INFORMATIONALISM: COMPUTER SYSTEMS AND

THE VALUES OF TRIPLE SURPLUS LABOR

By

Jordan Naod

Submitted to the

Faculty of the College of Arts and Sciences

of American University

in Partial Fulfillment of

the Requirements for the Degree

of Doctor of Philosophy

In

Sociology

Chair:

Russell Stone, Ph.D

Charles White, Ph.D

Jurg Siegenthaler, Ph.D

Dean of the College

October 18, 2011

Date

2011

American University

Washington, D.C. 20016

INFORMATIONALISM: COMPUTER SYSTEMS AND

THE VALUES OF TRIPLE SURPLUS LABOR

By

Jordan Naod

Submitted to the

Faculty of the College of Arts and Sciences

American University

in partial fulfillment of

the requirements for the Degree

of Doctor of Philosophy

in

Sociology

Chair:

____

Russell Stone, Ph.D

____

Charles White, Ph.D

____

Jurg Seigenthaler, Ph.D

____

Dean of the College

____

Date

2011

American University

Washington, D.C., 20016

INFORMATIONALISM: COMPUTER SYSTEMS AND

THE VALUES OF TRIPLE SURPLUS LABOR

BY

Jordan Naod

ABSTRACT

This dissertation analyzes how computer engineering organizations (firms that produce software and hardware) gained—and continue to gain—profit from labor wages in the production process and further demonstrates how these practices have caused a labor wage disadvantage for workers in this field. Computer engineering organizations devised unique operational mechanisms, functioning as the 3 components of their modes of production: original organization, management and production infrastructures. Computer engineering organizations formed a top-down pyramid organizational structure that synchronized with the hierarchy based micromanagement structure, and later developed an upside-down pyramid that worked with the macro-matrix management structure, requiring the development of their own production infrastructures to establish a complete modes of production. This study applies a historical analysis methodology by defining two periods: the *Transitional Phase* (1970-95) and the Age of Information (1995-2009), each of having its own modes of production with a distinct surplus value mechanism. During the *Transitional*

Phase computer engineering organizations utilized manual based single function computer systems, which necessitated the development of a pyramid based organizational structure, a hierarchy based micromanagement infrastructure and a manual based production infrastructure (the *Waterfall* production framework, an individual based production process, specialized based *Division of Labor* forces and production devices). The surplus value mechanism of this period was the increase in the number of specialized *Division of Labor* forces and the speeding up of the production process.

Nevertheless, as computer engineering organizations demanded more growth from the production process, they initiated a transformation of the production infrastructure by creating multitasking production devices, automation and internet communication. This production infrastructure was comprised by 4 new components: 1) *Waterfall* was changed to the *Iterative* production framework method, 2) single function base production devices were changed to multifunctional production devices, 3) singular specialization based *Division of Labor* forces were changed to multifunctional based *Division of Labor* forces, and finally, 4) the manual individual based production process became a multitasking based production process. This was followed by a transformation of the hierarchy management infrastructure to a macro-matrix management infrastructure, along with the replacement of the pyramid organizational structure with the upside-down and linear organizational structure.

By the mid-90's, the 4 components of production infrastructure were transformed, which gave birth to the Age of Information. These changes transformed the manual based single function production process and were meant to promote efficiency in production process. This was similar to the transformation of the previous classic surplus value mechanism (extracting surplus labor value from only one task in the manual production process) to a more effective mechanism of surplus value that would be compatible with the multitasking production process. Thus, applying the most advanced multitasking production devices and multitalented workforces, computer engineering organizations were able to extract surplus labor by assigning multitasking to a worker in production process, which became the operational mechanism for the Age of Information (1996-2009). With this transformation, computer engineering organizations developed a compatible surplus value mechanism (*Triple Surplus Value*, a concept and term coined by this present study), consisting of increasing the number of *Division of Labor* forces, speeding up the production process and multitasking multitalented professionals to perform the production process activities of multiple positions free of labor cost. The degree of *Triple Surplus Value* is measured using a formula introduced in Chapters 4 and 5, incorporating data from DOL's 41-year historical labor wage standard records and 13 years of historical job posting records. The results of this study indicate that computer engineering organizations gained large profits by controlling labor wages in production process and illustrate a labor wage disadvantage for workers.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ACRONYMS

BOE: The purpose of the BOE was to enable organizations to practice cost estimates against the work breakdown structure's labor efforts projections.

CMM: The capability maturity model was a framework that was created to be used by computer engineering organizations' management for implementing best practices in managing production process.

CMMI: The capability maturity model integrated introduced the concept of using best practices in forms of templates, devices, processes, procedures and sample product outputs, which were all essential relative to the production process phases to be shared across the management and production teams.

MS Project: Microsoft Project is an automated project management tool.

OB: The organizational behavior was a research aimed at creating new organizational structures.

OOAD: The object oriented analysis & design method was created to replace structure analysis & design methodology. The OOAD approach enables computer engineers to reuse a best practice development processes and procedure for all the modules until they are completed.

OOP: The object oriented programming (OOP) language that replaced the *Waterfall*'s structure based programming language

PTO: The second phase of management, which is the project tracking & oversight (PTO), begins after the computer systems engineering organizations' senior executives (known as steering committee's heads) approve the labor controlling mechanisms implemented in the plan.

WBS: The work breakdown schedule was created to specify production processes, allocate labor hours per tasks, specify deliverable date and assign workers.

WPPR: The weekly product progress report was a management methodology that was used to micromanage task performance at the production process level and to control production hours.

WPPR: The weekly project progress report was created for project managers and/or team leads to track the progress of ongoing production outputs.

UML: The unified modeling language was developed as a centralized and an integrated tool, commonly used by subsets of production process methods to complete tasks of the production phases.

CHAPTER 1

INTRODUCTION

The commercialization of computer systems (used here to mean the application of computer, communications and software to the management, processing and dissemination of information) in the last 15 years has transformed the organizational structure and function of computer engineering organizations. While it is generally recognized that computer systems have dramatically enhanced these firms' production processes and worker productivity, minimal attention has been paid to its impact on either organizational structure or the conditions of labor. Computer systems have played a pivotal role in transmuting the industrial modes of production of computer engineering organizations by replacing these organizations' former operational structure with advanced technology. This replacement simultaneously benefited stockholders on the one hand and disadvantaged workers on the other.

As conceptualized herein, this transformation occurred from 1970 to 2009; the period from 1970-1995 is denoted as the *Transitional Phase*, and the period from 1996-2009 is denoted as the *Informational Technology Era.* After studying the role that the Age of Science played in the emergence of the Industrial

Revolution and how science was used to redefine organization-operational structures, I began to see sociological inquiries as necessary for comprehending what led to the emergence of the Industrial Revolution. Who was impacted? Why? When? Where? How were the modes of production redefined? How did the relations of production become freed? How did these events move the peasants from their daily plantation labor to the industrialized, metropolitan factory environment? These questions seemingly having been answered by Enlightenment Thinkers and Industrial-Revolution historians, I began examining the emergence of the Age of Information.

It is essential to define a theoretical approach for this study to respond to the questions stated above. I will set the theoretical social themes in three timeframes. The first timeframe is the Industrial Revolution, coeval with the social assessment of the Enlightenment, Adam Smith's evaluation of relations of production being the central theme. This sets a foundation for Saint-Simon's doctrine of Progressive Industrialization, a theory that makes him arguably the last Enlightenment thinker due to his advocacy of Progressive Industrialization. The second theoretical social theme is Karl Marx's critical evaluation of Adam Smith's relations of production, and this will set the theoretical framework for an introduction of the classic concept of surplus value and Historical Materialism (relations and means of production) (Michael Burawoy, *Manufacturing Consent*, 1979). The third theoretical social theme is the evaluation of pre-1995 industrial organizational structures' development. Feeding off the work of Adam Smith, St.

\

Simone and Karl Marx set the foundation for the development of our theoretical framework from a production process perspective.

The focus of the third theoretical social theme will be placed on the objectivist social thinker, Max Weber, whose rationalized *Iron Cage* will shape this dissertation's theoretical framework (*Max Weber,* (1904-5a), *On Capitalism, Bureaucracy and Religion*. Translated by Stanislav, Andreski, 1983). Similarly, and within the same context—however, from a distinct perspective—the modernist Emile Durkheim's *Division of Labor in Society* will contribute to how this dissertation theorizes the modern industrial organization structure during the time period of 1970-1995, defined here as the *Transitional Phase* and thereafter as the emergence of the Age of Information during the time period of 1996-2009. When the theoretical framework developed in this dissertation is applied to organizations of the Age of Information generally (the software and hardware industries in particular), it becomes apparent that industrial organization structures have been transformed by advanced computer systems into informational organizational structures, with a negative impact on the work force.

This dissertation's theory regarding the nature of this transmutation may be summarized as follows: the growth of computer engineering organizations creates demand for development of advanced software and hardware systems. The application of the latter transforms how the organization's management structure and production processes operate. From a production perspective, this creates a new surplus value (exchange/surplus labor) mechanism that extracts profits from free labor. Concurrently, from a management perspective, management methods are developed to sustain this new surplus/exchange value, as well as the modes of production. The critical point is that the nature of the production process and the surplus/exchange value relationship depends on the type of advanced technological innovation occurring at a given time.

For example, from 1970 to 1995, a period that this dissertation calls the *Transitional Phase*, advanced technological innovation consisted of first generation technological infrastructure (e.g., client servers, local area networks with limited service capacity). The associated surplus/exchange labor mechanism applied by systems engineering organizations is known as the *Waterfall* production process (Edward Yourdon. (1997), *The Rise and Demise of the American Programmer*). Every worker worked in his own specialized role according to his expertise in the field. Production was performed in a controlled environment**,** in that workers reported to the owner's production facility to carry out their duties (Schermerhorn, Hunt, & Osborn, *Organizational Behavior*, 1998). The surplus/exchange labor value mechanism in effect was based on the industrial absolute and relative surplus values of the 19th century.

From a management perspective, to sustain the surplus/exchange labor value mechanism in practice, it was necessary to introduce a hierarchical, pyramid-shaped management structure. The structural similarity between the production and management mechanisms was essential to making the entire operation functional. With the *Waterfall* method in place and reinforcing surplus/exchange labor values, the pyramid management mechanism worked for the *Transitional Phase*. The resulting changes in the modes of production benefited owners of computer engineering organizations while having a negative impact on the computer production associates; there was a substantial decrease in the earnings of workers. In 1998, Rob Kling, a prominent techno-sociologist, stated that the computer engineering organizations create an unbalanced system. They set high standards of academic qualification and technical skills requirements, combined with high performance expectation, but grant relatively low compensation to hardware/software-development team associates. In addition, the elusive and complex nature of management techniques has created a lack of transparency for workers regarding their status, roles and responsibilities. This all occurs within the organizational structure, and it is the specific research problem of the proposed study.

As discussed in section 2 of this chapter, one primary and two secondary research questions are developed to examine this historical transformation of the modes of production using computer engineering organizations as a field, which are firms that produce software and hardware computer systems. In addition, these research questions cover the complete range of the problem as discussed in this section of the chapter. With the primary research question, the study will examine how the dynamics between computer engineering's organizational structure and software development have restructured management to benefit its stockholders. The same question further extends to inquiry how this transformation has created disadvantages for hardware/software-development associates in terms of labor wages. The details of the research questions are specified in section 2 of this introductory chapter. This dissertation identifies the most significant factors that have transformed the overall characteristics of the operations of computer engineering organizations, how the dynamics of this transformation create disadvantages in labor wages, and how computer engineering organizations both create and use advanced software to ensure their success.

1.1 Purpose, Goal and Research Question

The purpose of this study is to analyze the conflicting relationships (within a historical framework 1970-2009) among the organizational structure of computer engineering organizations, advanced computer devices and labor transformation in order to understand their effect on the modes of production in the Age of Information. Historically, those who owned an organization demanded growth and initiated the development of new means of production, which in this case are advanced software technologies (Schermerhorn, Hunt, & Osborn, 1998). Consequently, the emergence of these advanced computer devices and the organization's operational structures would then dynamically force a transformation of the relations of production. As a result, the classic industrial-pyramid-based organization's operational structure within the computer systems-development field has been replaced with the contemporary linear-based operational structure both in the production process and management structures.

A very similar transformation occurred during the emergence of the Industrial Revolution, in which the feudal modes of production were replaced by pyramid-based organizational structures (Burawoy, 1979). The Industrial Revolution transformed the relations of production and set the peasants free to sell their labor as a commodity at their will. However, as history revealed, the new bourgeois class then gained benefits from the modes of production. Even though the relations of production were freed, the industrial proletarians remained disadvantaged in terms of labor wages.

Another transformation has occurred during the Age of Information; just as there was a transition from feudal modes of production to industrial modes, so too was there a transition from industrial to informational modes of production. With the emergence of advanced computer devices, it was inevitable that the operational structure of the pyramid-based organization would change (Schermerhorn, Hunt, & Osborn, 1998). What makes the contemporary revolution more complex than the industrial revolution is its removing the means of production from the owner class, as exemplified by the case of the hardware/software production industry. Thus, the goals and research questions of this dissertation are primarily focused on analyzing the unique characteristics and scenarios of contemporary modes of production generally within computer engineering organizations. For instance, the *Division of Labor* in a pyramid-based organization's operational structure (characterized by single-focused specialization or expertise) will be analyzed. This pyramidal framework is no longer efficient in the linear structure of contemporary computer engineering organization. The contemporary modes of production are very complex because the production process has drastically transformed to sustain the exchange (surplus) labor cycle. The purpose of the current exchange (surplus) labor scenario is to ensure that the worker has the ability to perform multiple tasks simultaneously. Profit is gained from surplus labor because computer engineering organizations are now using multifunctional computer systems, the internet, wide area networks and advanced applications, which enables an individual worker to perform an increased number of production activities. For example, a software engineer can perform the following tasks alone: (1) analyzing business requirements, (2) developing analyses and designs and (3) programming, testing, configuring and installing (Schermerhorn, Hunt, & Osborn, 1998). The application of this complex production-line process causes a labor-wage disadvantage for the worker, because the worker is paid only one position's salary even though he or show performs the functions of multiple positions. However, this greatly benefits the owner-class, in terms of profits via exchange values, as they are saving on labor costs.

With such a complex scenario in mind, this study will focus on the following two issues: the role of advanced software technologies in transforming the modes of production of the computer engineering organizations (the organization's operational structures and technologies) to benefit those who own software-development organizations, and the negative impact on software-production associates in terms of labor wages, roles and status. To examine this interaction, I propose primary and secondary research questions, which should fully address the range of issues identified above.

Primary Research Question

Have the dynamics between the management structure of computer engineering organizations and advanced computer systems development transformed modes of production (i.e., management-and-production process)? If so, how has this benefited stockholders in terms of gaining profits, and how has this transformation created disadvantages in wages for the labor class?

Secondary Research Questions

How has the growth of computer systems engineering organizations created demand for the emergence of advanced software innovations?

How has the *Division of Labor* in production process of advanced computer system devices benefited stockholders and caused wage disadvantages for workers?

Within the organizational structure of computer high tech engineering institutions, what initiates the computer systems development process generally, or the software development production process in particular?

Have computer systems innovations been developed to help organizations manage increases in their size and complexity? What are they?

Have the industrial modes of production (management and production processes) been transformed by the emergence of advanced software technologies, and if so, how?

Has the advent of the Age of Information rendered details about modes of production opaque to the computer engineering workers, and if so, how?

1.2 Significance of the Study for the Field

The most significant contributions of this study to academia can be summarized by the following:

Historical Roots and Emergence of Sociological Theories: One of the important contributions of this dissertation to the field of sociological studies is its contemporary application of classical principles. It draws from the ideas of Enlightenment social thinkers and 19th century classical sociological theorists. Regarding the Enlightenment, the primary focus is on Adam Smith's transformation of Relations of Production, and for 19th century classical sociological thinkers, this dissertation will examine the works of Karl Marx and his Modes of Production, including his concept of Surplus Values, Emile Durkheim's *Division of Labor in Society* and most importantly, Max Weber's *Iron Cage* ("the shell is stronger than the steel"), including Bureaucracy & Rationalization. In sum, the dissertation forms its theoretical foundation by looking back to these theorists' work.

Trends of Continuity and Discontinuity of Sociological Theories: This dissertation makes an important contribution to contemporary social transformation studies because it addresses continuity and discontinuity of sociological theories about modes of production. For instance, this study employs Marx's theory of surplus values (absolute and relative). Using Marx's *Surplus Value* as a foundation, it builds a version of surplus value in the process of interpreting contemporary modes of production. The value of Marx's *Surplus Value* is apparent in this study's identification of the dependent variable (*Triple Surplus Value: Absolute, Relative & Complex Surplus Values*), discussed in Chapter 4. However, this dissertation parts ways from Marx's *Historical Materialism* because it finds the dialectical process to be of limited use with respect to understanding modes of production, which are means and relations of production (Friedrick Engels, (1868) *On Marx's Capital*, republished, 1972).

Although Marx was a dialectical thinker, he did not account for the dynamics of the modes of production and how those dynamics negatively affect labor force wages (*Ibid*); moreover, he overlooked the historical dynamics between organization's demand for innovation and the development of modern technology. By identifying and explaining its points of departure from Marx's version of dialectical analysis, this dissertation makes a significant contribution to the concept of social theoretical continuity and discontinuity in the field of social change.

Similarly, this dissertation relies on Max Weber's *Bureaucracy and Rationalization* framework as a building block to formulate its theory of organizations. Weber's theoretical framework is centrally focused on how organizations establish management mechanisms and implement them as controlling vehicles, which then became rationalized within organizations (Weber, M. (1904-5a), *On Capitalism, Bureaucracy and Religion*). Thus, his work is outstanding in terms of interpreting organizational structures and how they are rationalized. By applying Weber's theoretical framework, this dissertation sustains continuation of the Weberian theoretical approach in defining contemporary and classic organizational structures. However, this study also departs from traditional Weberian analysis and expands on it in order to demonstrate how the management controlling mechanism is utilized to gain surplus labor in the production process.

Techno-Sociology and the Structural Transformation of the Computer Engineering Organization: This dissertation constitutes a substantial addition to the fledgling field of techno-sociology because it incorporates into the traditional study of means of production concepts and theories unique to technological innovation: the complex dynamics of relations of production, in the form of the organizational structures of computer systems engineering organizations and means of production, in the form of advanced computer systems devices.

Development, Methods and Processes of Advanced Technology: One of the core premises of this study is that advanced computer systems are an independent variable in triggering changes in an organization's operational structures (e.g., management types/structures and production line processes). This premise implies the emergence of advanced computer systems devices and their role in contributing to the development of the organization's operational framework and productivity maintenance methods to ensure profit gains. The operational framework and productivity maintenance methods are the management mechanisms (e.g., types, structures and implementation processes) that are applied to maintain the structure in place by rationalizing them. This dissertation concludes that these elements are most critical in creating a lack of transparency among workers about their roles within the organization.

In closing, the dissertation suggests that both the academic and the professional community must examine the dynamics among (1) the structure of the computer systems engineering organization, (2) management mechanisms and (3) computer systems devices to understand their complex relationship. This can be accomplished only by applying a valid methodology to the exploration of the dynamics of these different dimensions. This dissertation contributes new knowledge to the field of sociology, information systems and computer science by clarifying how to analyze historical changes in reference to the organization's structure, characteristics, management mechanisms and product development. Most significantly, the study uncovers the causal factors, their dynamics and the process by which computer systems engineering organizations initiate computer system devices' development. In particular, this dissertation reveals how advanced computer systems are applied to restructure computer engineering organizations, which in turn yields profits—but by doing so creates disadvantages for software-development associates in terms of labor wages, status and organizational transparency.

1.3 Research Methodology

This study uses both interpretive and predictive historical research methods (published historical documentation and conclusions that can be reached from such documentation). The reason for combining these methodologies is that insights achieved through interpretive analyses can be enhanced by a predictive approach. The analysis also employs secondary statistical data to inform the historical comparative analysis of labor wages, the majority of the data about which was obtained from the U.S. Bureau of Labor Statistics (1970-2009). The data analysis is explained in greater detail in Section 4 (Data and Analysis) of this chapter.

For the period from 1970 to1995—the *Transitional Phase*—the historical analysis methodology is applied to analyze how the dynamics of advanced computer systems innovation and organizational demand for such innovation in parallel redefined modes of production. It also analyzes how owners of the means of production benefited from the historical occurrence at the same time that workers were subjected to labor wage disadvantages. For the period from 1996 to 2009—the Age of Information—the historical analysis methodology is applied to analyze the transition from industrial modes of production to the contemporary informational modes of production, in particular within the field of software development. Like the statistical data analysis, the application of the historical analysis methodology is discussed in greater detail below.

1.4 Conceptual Definitions

The Transitional Phase: The *Transitional Phase* encompasses the period from 1970 to 1995. During this period, the corporations that produced computer systems redefined skill set specialization and began to develop advanced software technologies, the commercialization of which ultimately led to the emergence of the Age of Information. The modes of production continued to reflect the traditional industrial organizational structure, where the means of production are controlled by those who own the organization and the individual worker is systematically deceived in order to increase efficiency, productivity and profits by implementing the classic surplus (exchange) labor value known as absolute and relative surplus values to extract free labor (Karl Marx. (1867), *Capital*: *The Process of Production of Capital*, Volume I, Trans. and Republished by Samuel Moore and Edward Aveling in (1954).

Age of Information: The period between 1996-2009, in which computer engineering organizations' modes of production were transformed via a switch to an upside-down organizational structure, the previous hierarchy micro based management to a macro-matrix management and from a top-down manual based production process to a linear, automated based production process.

Computer Engineering Organizations' Modes of Production: The computer engineering organizations' modes of production infrastructure has 3 components: The organizational structure, management infrastructure, and production infrastructure, which are defined below.

Pyramid Organizational Structure: Computer engineering organizations' top-down based organizational structure that existed during the *Transitional Phase* (1970-95).

Upside-down Organizational Structure: Computer engineering organizations developed this structure based on open communication with no chain of command, which was implemented in 1995 as the Age of Information was replacing the pyramid organizational structure.

Hierarchy Management Infrastructure: A top-down based management structure with the following infrastructures, which are also the mechanisms: Work Breakdown Structure (WBS), Gantt Chart and Bases of Estimates (BOE), manually practiced during the *Transitional Phase* (19970-95).

Upside-down Management Infrastructure: Macro-matrix management based on a linear and self-management structure with an automated WBS, Gantt Chart and BOE, using computer systems like Microsoft Projects as a managerial device.

Production Infrastructure: Computer engineering organizations' production infrastructure has 4 components: production framework, production process devices, *Division of Labor* forces and production process.

Waterfall Production Framework: A top-down based production process framework, which provided the methodology to computer engineering organizations of how to develop computer systems introduced. This framework was introduced by Winston Royce (1970) and was practiced during the *Transitional Phase*.

Iterative Production Framework: A linear and spiral based production methodology, which replaced the *Waterfall* production framework during the mid-90's. Grady Booch, Ivar Jacobson and James Rumbaugh collectively developed this framework in the early 90's.

Absolute, Relative and Triple Surplus Values: Marx and Engels dedicated 20 years to examining surplus labor values in production processes using sources such as factory labor reports, employment commission reports and other Parliamentary related reports. As summarized by Friedrick Engles in *On Marx's Capital*:

> The working-time in which the laborer reproduces the value of this labor-power, Mr. Marx calls "*necessary labor*"; the time worked beyond that, and during which surplus-value is produced, he calls "*surplus-labor*".

> Necessary labor and surplus labor combined form the "*working-day*". Suppose the working-day counts six hours of necessary and six hours of surplus-labor; then the laborer furnishes the capitalist with 36 hours of surplus-labor a week. He might as well have worked three days for himself and three days for the capitalist. But, this is not at once visible. Surplus-labor and necessary labor are more or less mixed together. I might express the same relation thus, that, in every minute, the laborer works 30 seconds for himself and 30 more for the capitalist. (Engels, 38-40, 1868 and republished, 1972)

It is this surplus labor value that possesses the attributes of the classical absolute and relative surplus labor values in the production process.

During the *Transitional Phase*, computer engineering organizations applied absolute and relative surplus labor values in the production process to gain profits from surplus (exchange) labor. The concept of the classic absolute surplus value is applied to measure how labor wage disadvantages for computer system engineers occurs in the production process and how this advantages owners in terms of gaining profits: organizations extracted free labor as the commodity production process came to require more time than the actual production process (Engels, 1868). This resulted in a surplus value for the owner even before the commodity was delivered to the market. This process by its very nature systematically deceives proletarians because they perceive that they have been paid for daily hours of work. Absolute surplus value manifests itself when working hours do not correlate with wages paid (Engels, 1868). The deception happens because wages appear to compensate for total hours worked; however, the pay for eight hours is actually the cost for five hours, giving the company three hours of free labor in producing additional commodities.

During the *Transitional Phase*, each individual computer systems engineer was producing computer systems within a *Waterfall* operational structure (Schermerhorn, Hunt, & Osborn, 1998). Marx and Engels defined the classical relative surplus value in this manner:

> There is another way of increasing surplus-value besides lengthening the working-day beyond the time required for the production of the necessary means of subsistence or their value. This can be achieved by a reduction in the working-time required to produce the necessary means of subsistence, in other words by cheapening the means of subsistence, and this in turn only by improving production. On this point Marx again gives a detailed illustration by investigating or describing the three main levers by which threes improvements are brought about: 1) *co-operation*, or multiplication of power, which results from the simultaneous and systematic joint work of a number of workers; 2) division of labor, as it took shape ion the period of manufacture [typos?] proper; finally 3) machinery by the help of which modern industry has since developed. (Engels, *On Marx's Capital*, 23-24, 1868).

Specialized computer system engineers worked together in the same production process or used advanced technologies to increase productivity without lengthening the number of working hours—for example, workers produced eight hours of goods in a matter of four hours but were led to believe that they were paid for eight hours, gaining the organization four hours of free labor. In short, the workers' labor wages did not reflect the increased profits that the company earned as a result of greater productivity. Put another way, workers were not compensated for their increased productivity. The management structures were micro based and hierarchies were visible to subordinate workers, and the Iron Cage mechanisms were indeed rationalized across the board to maintain the structure (Weber, 1904-5a).

The Age of Information: The Age of Information encompasses the period from 1996 to 2009. As a result of organizational growth during this period, demand has been created for advanced software technologies, the implementation of which transformed modes of production (i.e., organizations' operational structures). The modern industrial pyramid organizational structure gradually has transformed itself as the world has become more computerized; the contemporary working environment has been turned on its head, into an upside-down pyramid structure. The contemporary structure offers many advantages for corporations and relatively few for the individual worker.

This transformation has occurred in part due to the development of advanced computerization systems (Schermerhorn, Hunt and Osborn, 6, 1998). Unlike traditional directing-and-controlling managerial positions, today's upside-down pyramid charts show positions like "project director, coordinator and team leader" (Schermerhorn, Hunt and Osborn, 10, 1998). Subordinates' working positions are replaced by titles such as "team members and work associates" (Schermerhorn, Hunt and Osborn, 10, 1998). Through computerization, top managers can monitor subordinate workers' performance from different locations (Kling, 40, 1996). Furthermore, workers are motivated to be more productive by their coordinators instead of being directly supervised. Although this advanced computerized strategy gives some independence to workers, it also creates disadvantages.

Historical Transformation of Computer Engineering Organization's Operational Structures: This section refers to the transformation from modern industrial organization's operational structures to the Age of Informational Technology organizational structures, which redefined the modes of production to benefit owners with a negative impact on the workforce (the decrease in earnings of software engineers). The particular timeframe in which this transformation took place is 1996-2009.

Means of Production: The means of production are the software and hardware products used to produce commodities by the computer engineering organization staff. Historically, during, before and after the Industrial Revolution, the means of production were owned by the owners. However, as presented in this dissertation, the means of production continued to be owned by those who owned the organizations even during the *Transitional Phase* from 1970-1995 (Stephen Sanderson. (1995), *Social Transformation*). One of this study's core arguments states that in the Contemporary Age of the Information Technology Era, the means of production became freely available and affordable to the workers.

Computer Engineering's Organizational Structure: The organizational structure is the management and production layout of computer engineering organizations in terms of structural characteristics, management techniques, product (software) development and product life cycle.

Computer Systems Development Process: Production processes development methodologies encompass the defined tasks applied to develop a software and hardware products. I intend to demonstrate how computer engineering organizations are able to profit from their staff through a complex surplus value system.

False Consciousness: False consciousness, in the general sense, is a state of mind cultivated in workers by management pursuant to which workers define their roles based on incorrect information in production process (Burawoy, 1979). As used in this dissertation, the phrase false consciousness refers to workers misconceiving their actual roles in the production process and their status in the organizational structure. As the modes of production (.e.g., management structures and the methods of maintaining production process) were redefined during the Age of Information, workers developed a false consciousness not only regarding absolute and relative surplus value mechanisms (as in the *Transitional Phase*), but also with respect to the relations of production (management structure).

Triple Surplus Value: The modes of production in the Age of Information may subject workers to a *Triple Surplus Value* mechanism one that combines and exploits absolute, relative and complex surplus values. For example, from the production process perspective, during the *Transitional Phase*, absolute surplus manifests when a specialized worker devotes long hours to producing the same product throughout the course of his/her working hours. Similarly, relative surplus value is exercised when a specialized worker is expected to increase productivity by using technology. However, in the Age of Information, the modes of production (means and relations) are freed because the manifestation of surplus value in the production is complex.

During the Age of Information, the *Triple Surplus Value* mechanism is comprised by a combination of absolute, relative and complex surplus values. Using the field of computer-systems development, the following example provides a scenario of *Triple Surplus Value*: 1) A software engineer is required to be multi-talented to perform different aspects of the computer-software production process, 2) to increase productivity, in addition to using the organization's advanced computer equipment, the software programmer is required to utilize his own computer equipment to increase productivity wherever the programmer might be located to continue the long production process, 3) the programmer uses his/her own resources (e.g., overhead, technology and time), to produce the different production life cycle process (e.g., analysis, design, code, testing and deployment) to prepare the product for services in the market.

Each of these five specific production life cycle processes is a specialized field, which requires specific expertise with its own labor specific codes and salary ranges. As a result, the *Triple Surplus Value* benefits the owner of the software and hardware engineering organizations by enabling them to extract the surplus values from a worker, such as a software programmer. This enablement occurs when the programmer produces the production life cycle processes to save labor surplus values to gain profits from free labor.

1.5 Data Analysis

As stated above, the purpose of this study is to analyze the conflicting relationships (within a historical framework, 1970-2009) among computer systems engineering's organizational structures, technology (advance software technologies) and labor transformation to define the Age of Information's modes of production. My examination will focus on the use of computer systems to benefit those who control the means of software production and its negative impacts on software production associates.

In order to develop the explanations, my data consist of a purposive selection of written texts and supporting statistical data focusing on the research questions I defined in this study. The references that I use include various documents and publications, which include government and private sector project manuals. As the study requires, secondary statistical data generated by either public or privative organizations will be used to support the information obtained from the various documents and publications incorporated in this study. With reference to the labor wage historical comparative analysis, the majority of the data come from the U.S. Department of Labor's *Occupational Outlook Handbook and Bureau of Labor Statistics* reports published between 1974 and 2009. The historical comparative analysis is used as a tool for interpreting the data. In terms of the historical analysis of labor within the context of production during 1970-1994, Michael Burawoy's *Manufacturing Consent* (1979), Daniel Bell's *The Coming of the Postindustrial Society* (1974), and Edward Yourdon's *Classics in*

Structured Design (1979) are used as the primary sources to support the theoretical framework of this study.

My primary sources used to perform a historical comparative analysis on the two distinct software organizational management structures are the following: *Capability Maturity Module Integration*: *2003* by Carnegie Mellon Software Engineering Institute, and Stephen P. Robbins and Timothy A. Judge, *Organizational Behavior*: 2007, *The Iron Cage: A Historical Interpretation of Max Weber*, translated by Arthur Mitzman (1985), and *Closing the Iron Cage: The Scientific Management of Work & Leisure*, translated by Ed Andrew (1999), Manuel Castells, *End Of Millennium* (2000). Finally, Rob Kling, *Computerization and Controversy* (1996), Rob Kling, *Technology and the Transformation of White-Collar Work* (1987), Edward Yourdon, *Managing the Structured Development Techniques* (1986), Yoneji Mansuda, *The Information Society* (1981) and Peter Keen, *Tactical Management Models* (1985) are extensively referenced to support the study because of the valuable data they contain.

In sum, I selected the sources listed above based on their theoretical relevance to this dissertation. Using these sources, the study demonstrates the way in which the demand for the development of advanced computer devices was innovated, leading towards the emergence of the Age of Information. Also, this dissertation will show how the modes of production were transformed from the modern industrial capitalism to the contemporary informational capitalism modes of production structures. In the final analysis, the study evaluates the modes of productions in terms of management, characteristics, how profit was gained, how the dynamics consequently caused labor wage disadvantages for workers and how false consciousness manifests among the workers.

1.6 Measures

Historical Analysis: The interpretive historical analysis is applied to analyze the Modern Industrial and the Age of Information modes of production (an organization's operational structures) between the period of 1970 and 2009. Hence, a chart is used to provide graphic support for the interpretive analysis and quantitatively shows the following: 1) how computer systems' engineering organizational growth has created demands resulting in the emergence of advanced software technological innovations, and 2) how both the organization and advance software technologies helped lead to the creation of the Age of Information modes of production structures. Most significantly, the historical labor wage tracking chart shows how computer engineers were negatively impacted by the transformation. In addition, the historical analysis approach, using secondary case studies, examines what forms of computer devices have been developed to transform the organization from the modern industrial organizational structures to the informational organization's operational structure. Finally, the historical analysis will show how the reciprocal effect of the redefined management operation structures caused the false consciousness of the worker. Using this approach, the study analyzes the ambiguous current management and production operation mechanism that causes alienation of the computer engineering workers by using deceptive position titles—among other methods—to a point where workers have lost their sense of realistic roles within the organization.

The historical analysis includes dependent and independent variables to analyze the research questions defined above. The dependent variables are the growth (profit gain—profit gain for computer systems engineering organizations and the disadvantage to computer engineering proletarians in terms of wage and the creation of their false consciousness (unawareness of their true rank within the organization). For example, the first time period of 1970 to 1995, which this study defines as the *Transitional Phase,* had the modern industrial-based computer systems engineering's organizational structure. This time frame is dedicated to analyzing the practice of the classic surplus values in production process and to analyzing how the hierarchy-based management structures helped gain profits for owners and cause wage disadvantage for the proletarians. Two specific independent variables are applied for this time period.

The first independent variable is the computer engineering organizations' management structures; each organization's rationalized hierarchy-based characteristics used to sustain the production surplus value mechanism. For example, the employment titles are: Company Owner(s), Executive Manager, Senior Manager, Middle Manager, Supervisor and Subordinate Workers (Schermerhorn, Hunt, & Osborn, 1998). The second independent variable is the classical production operations' absolute and relative surplus values mechanism in the production process. In sum, the analysis shows how the independent variables contribute to the organization's growth and to the disadvantage to proletarian wages. The same dependent and independent variables are applied for the historical time period of 1996-2009.

The final time period is 1996 to 2009, which this study defines as the Age of Information. Although the specific independent and dependent variables are the same as for the *Transitional Phase*, they are distinctly defined. For instance, the first independent variable for the Age of Information is defined in terms of non-hierarchical management structures: 1) "linear," "matrix and upside-down pyramid management types" as a substitute for the "hierarchy", 2) "macro-matrix management" and "independent self-management" as a substitute for "micro supervisory management," (3) "steering committees" as a substitute for "ownership"; (4) "project sponsors" as a substitute for "ownership/partners"; (5) "coach" as a substitute for a "director"; (6) "program lead" as a substitute for a "senior manager"; (7) "project lead" as a substitute for "middle manager"; (8) "team lead" as a substitute for "supervisor"; and (9) "team members & associates" as substitutes for "subordinate workers" (Schermerhorn, Hunt, & Osborn, *Organizational Behavior*, 1998). Hence, these independent variables are used to analyze how management structure reinforces the production process to continue improving productivity for the organization.

1.7 Overview of Dissertation Chapters

Chapter 2 is titled "*Historical Technology and Controversy*." It covers the historical trends of the dynamics between organizations' demand to initiate advanced technological innovations and the impacts in redefining the modes of production. In addition, it highlights historical events, like how the Age of Science was used to trigger the emergence of mechanical tools to supplement the development of industrialization (Yoneji Masuda, *The Information Society*, 1981). At that point in time, that motion influenced the new class known as "bourgeois," which was primarily concentrated in the preliminary modern metropolitan areas, igniting the Industrial Revolution to transform the feudal modes of production (Karl Marx, *Capital*: *The Process of Production of Capital*, Volume I, 1867). As a result, the bourgeois' modes of production gave birth to a new conflict of interest between the factory owners and the workers in terms of profits and surplus value driven labor wages (Engels, *On Marx's Capital,* 1868). Subsequently, this condition evolved into what is known to us as a modern industrial capitalist organizational operational structure.

Thus, chapter 2 dedicates a section, entitled "Industrialization and the Enlightenment Perspectives on Modes of Production," to illustrating the historical trends as a background. In this section, this dissertation's theoretical frames are formulated by referring to social theorists of the Enlightenment period. In particular, it focuses on the modes of production subject matter expertise like those of the legendary Adam Smith's work, *Wealth of Nations* (1776), in which Smith examined the relations of production that allowed workers to sell their labor under the new structures developed around 1776 (Adam Smith, *The Wealth of Nation* (1776), republished by George J. Stigler (1954). This study refers to the classical social thinker Karl Marx's interpretation of the modern industrial modes of production structures, including his work on surplus values, which round out the shortcoming of Smith's relations of production theory (Turner, *Emergence of Sociological Theory*, 1998). This is how the historical background of this study will be structured. Marx's surplus value theory has a great influence on this study's *Triple Surplus Value* theory, which will be discussed in Chapter 4 in great detail. Furthermore, in an effort to establish a well-rounded theoretical framework and to demonstrate how this study justifies its theoretical foundation, this dissertation presents the social industrialist St. Simone's interpretation of industrialization.

In chronological order, the second section of chapter 2, entitled "Paving the Way: The Modern Capitalist Organizational Modes of Production," establishes a theoretical framework by examining the classical social thinkers' theories, including those of: (1) Emile Durkheim, *Division of Labor in Society*, 1893), in his efforts to define the modern industrial organizational structure, characterizing it as an "Organic Solidarity," (2) Max Weber, in his masterpiece, *On Capitalism, Bureaucracy and Religion* (1904), in an effort to interpret the modern industrial organizational structure's bureaucratic condition and its rationalization process. Weber arrived at the pessimistic conclusion known as the

"Iron Cage," which is a rather distinct projection of the modern industrial structures from Durkheim's optimistic conclusion (Emile Durkheim, *Division of Labor in Society* (1893), translated by W.D. Hall and Lewis Coser, in 1993**)**.

In the final closing sections of chapter 2, this study will proceed towards establishing its theoretical framework in the practical surplus values processes within the industrial organization's production process. It refers back to the work of one of the few leading industrial sociologists, Michael Burawoy (*Manufacturing Consent*, documented in (1979-1982). Most significantly in chapter 2, the dissertation's theoretical linage traces its roots in terms of defining labor processes during the industrial organization's operational structures back to Burawoy's hands-on participant observer findings. Burawoy's (1982) *Manufacturing Consent* is one of the few core sources used in this dissertation's chapter 2 to describe the production line processes within the context of surplus values and labor. Thus, Burawoy's *Industrial Consent* leads to the closing of chapter 2, which constitutes an examination of one of a few prominent techno-sociologists. Rob Kling's (1996) *Technological Utopianism vs. Anti-Technological Utopianism* will be analyzed. Using the sociological analysis of these sociologists' works in chapter 2, this dissertation sets its foundation for chapter 3's analysis, entitled "*Transitional Phase*," covering the historical time period of 1970-1995.

Using this study's theoretical linage and roots imbedded in chapter 2, chapter 3 progresses to theorize what it calls the *Transitional Phase*, arguing that the period of 1970-1995 was a timeframe that contributed to the emergence of the

Age of Information, which transformed the modern industrial modes of production. Chapter 3 presents the dissertation's theoretical explanations and answers its secondary questions regarding the *Transitional Phase*. This sets up the base to respond (and provides the answer) to its primary research question. The critical secondary research questions that chapter 3 responds to theoretically are the following: 1) During the *Transitional Phase*, how has the computer systems engineering organizational growth created demand for the use of advanced computer systems engineering development frameworks?, 2) What forms of technological innovations have been developed for solving organizational growth demands?, 3) What was the process of the production line within the context of labor and surplus values that caused wage disadvantages for workers, and 4) What were the management types and structures during this period that were used to maintain an organization's operational structures?

In chapter 4, the study responds to the secondary research questions outlined above. In responding to question 1, this study finds that the cost of specialized division of labor, overhead expenses and production devices, was increasing to produce a complete systems product becoming a challenge to those who owned the means of production (Rob Kling, (1987), *Technology and the Transformation of White-Collar Work*). Simply put, the maintenance cost of the means of production that were used to engineer the systems was unbearable (Kling, 1987). The study asserts that during the *Transitional Phase*, the development of desktop personal computers, client server based modern systems infrastructures, and total quality management methods developed by William

Golomski (1974), gave the green light to the computer systems engineering organizations to redefine their modes of production in an effort to increase their profits; however, the challenge of overhead cost remained persistent until 1995. In order to produce a finished product, the costs specialized *Division of Labor* included overhead expenses (Rob Kling, *Technology and the Transformation of White-Collar Work*, 1987). This disparity of cost raised the demand for computer systems engineering organizations to invent advance technological solutions. Based on this circumstance, those espousing the Technological Utopian School of Thought started encouraging the innovations of advance technologies (Schermerhorn, Hunt, & Osborn, *Organizational Behavior*, 1998).

Similarly, Grady Booch, Ivar Jacobson and James Rumbaugh's advanced Software Engineering Methodology—called unified modeling language—documented in (1983-1994) started developing advanced technological engineering methods during the *Transitional Phase*, which contributed to the emergence of the advanced software, world wide web and web-based applications (Schermerhorn, Hunt, & Osborn, 1998). As a result, the computer systems engineering organizations began to demand the use of those advanced technological innovations to transform their modes of production in an effort to improve their profits by transforming the production processes, which took place during the mid-90's (Keen, P. *Information Systems and Organizational Change*, 1981).

Finally, chapter 4 of this dissertation provides a theoretical framework that maps the evolving transformation of computer engineering organizations from hierarchy management structures to macro-matrix management ones and from industrial surplus values process to the *Triple Surplus Values*, leading to the present dynamics among them. Using chapter 3's theories and discussions as a *Transitional Phase*, chapter 4 is dedicated to establishing the theoretical framework of the 1996-2009 time period, defining it as the Age of Information. The central focus is analyzing how the dynamics between advanced computer devices and management operational structures have transformed the computer systems engineering organizations' modes of production.

As discussed in chapter 3, during the *Transitional Phase*, there was a major computer device innovation impacting the emergence of advance technologies (e.g., internet, software and wireless with advanced infrastructures) that transformed the modern industrial computer systems engineering organization's operational structures. Thus, this study uses chapter 4 to fully respond to the primary research question: How have the dynamics between computer systems engineering organizational structure and advanced computer system devices transformed the modes of production (management and production operational structures) to benefit stockholders, and how has this transformation created disadvantages for computer systems development associates? In sum, within its subsections, chapter 4 presents the processes during which the *Triple Surplus Value* manifests, causing disadvantage for the workers in terms of earnings, and how that translates into the advantage of profits for the computer systems engineering investors, including the linear based macro-matrix management structures created to maintain and to enhance the production processes.

In conclusion, chapter 4's covers the data and analysis aspects of this dissertation leads to supporting chapter 5's conclusion and provides recommendations. The dissertation concludes that both the studies and analysis of the great writers and the results of the data analysis fully answer the research questions presented by this study. The results of the data analysis and the findings are illustrated in chapter 3 and 4. The transformation that took place from 1970-95 was the result of the dynamics between the computer systems engineering's organizational demand to increase profits and the emergence of advanced computer system devices. Based on the results and this study's findings, chapter 5 concludes this dissertation by recommending future studies.

CHAPTER 2

HISTORICAL TECHNOLOGY and CONTROVERSY

2.1 Overview of Transformation of Modes of Production

As this chapter progresses, it will become evident that this dissertation is greatly influenced by the works of the Enlightenment and Classical period, in that some of the concepts developed by sociological scholars of this time will be used as valuable tools and starting points in evaluating the contemporary *Informational Era*. For example, in 1776 Adam Smith used his pin-production as a case study, the continuing trend 90 years later to Marx's shoes-production. 126 years later, within the same line of theoretical questioning, this is the case with computer systems production. The primary purpose of this chapter is to set the theoretical framework of this dissertation by tracing its historical roots and theoretical lineage back to some of the pioneer social thinkers in the field of social transformation with respect to modes of production. This chapter will examine the essential theories of the following scholars: (1) From the Enlightenments Era, Adam Smith and his examination of the *Division of Labor* in 1776, (2) Karl Marx and his assessment of Smith's relations of production model in 1867, with a focus on the concept of relative and absolute surplus values, in which he rounded out the limitation of Smith's theory, (3) St. Simone and his the pro-modern industrialization strategy of (1819), which further influenced the formation of

Emile Durkheim's sociology. Accordingly, section 2 of this chapter evaluates Durkheim's work on optimistic perspectives regarding modernization for social cohesion—within the context of the *Division of Labor in Society*—to promote the organic solidarity theorized in 1893. From this same period, the work of Max Weber (in terms of modern industrial organizational structures, the rationalization of modern bureaucracy process, which led to the pessimistic theoretical conclusion that he calls the "Iron Cage," or at times also known as "the Shell is stronger than the Steel") will be explored as a further foundation for the ideas in this dissertation. Critical to note here is Weber's examination of accounting practices and methods of management (relations of production processes, e.g. managerial types, structures and bureaucratically procedures). These bureaucratic methods maintained the production line mechanisms in practice. Following Weber's conclusion, the Iron Cage, the present study highlights the practical demonstration of these production line mechanisms, examined by Michael Burawoy within the modern industrial manufacturing environment documented during the 1970's (which is supported with 30 years of historical evaluations). The closing of this chapter will contain a conclusion just as pessimistic as Burawoy's finding; however, readers will encounter optimistic theoretical recommendations for future studies. Lastly, the final section of this chapter presents the theoretical debates of the Technological Utopian and Anti-Technological Utopian schools of thought to proceed towards chapter 3, which examines the trends of the modes of production's transformation with respect to computer-engineering organizations' operational structures that took place during the *Transitional Phase*.

2.2 *Industrialization and Enlightenment Perspectives*

The transformation from the feudal modes of production to the industrial modes of production occurred as a result of parallel revolutions, the emergence of industrial mechanical machines and the Enlightenment scholars' notions of reengineering social infrastructures (both to benefit the new bourgeoisie class and to put the industrial workers at a disadvantage in terms of labor wages). Thus, the analysis of how the Industrial Revolution transformed feudal modes of production raises three critical questions: (1) What led to the demand for more industrial-based mechanical machineries and the transformation of the means of production? (2) What caused the transformation of the relations of production, which set the peasants free to sell their labor in newly emerged metropolises and led to a disadvantage for workers in surplus labor wages? (3) Why did the Enlightenment scholars advocate the implementation of the *Division of Labor*, and what were their negative consequences for the work force in production process? In response to all three of the questions in this section, it is critical to evaluate the work of Adam Smith, Saint-Simon, Karl Marx, Max Weber and Michael Burawoy.

The study finds Adam Smith to be a pro-industrialist, whose work benefited the new bourgeois class and helped dismantle the old feudal modes production. However, his theoretical interpretation and projection of a labor wages model did not differentiate how the method of surplus value was applied by the bourgeoisie to gain profits from free labor in production process (Engles, Fredirick. (1868). *On Marx's CAPITAL*, republished (1940). In order to fully respond to the three questions in this section, it is relevant to reassess and analyze Adam Smith's (1776) *The Wealth of Nations*.

Smith views the demand of industrial mechanical machines as a factor that ultimately transformed the feudal modes of production (as he called it, "unproductive" to a "productive means of production") to improve production-line process (Smith, Adam. Chapter One of Book One, *The Wealth of Nations*, 1776b Trans. by Skinner, Andrew (1997). Earlier on in his social analysis, Smith noticed that as a result of the Age of Science's scientific discoveries, inventors began to develop mechanical machineries in ways that were never seen before (Smith, 1776b). As this progress was taking the course of its impact on urban development, Smith noticed how this began to set the direction for creating industrially-based social classes, perhaps later on known as the preliminary bourgeoisie class (Smith, 1776b). Once he learned that the industrial mechanical machineries were aggressively redefining communication, he started his evaluation of the feudal modes of production, a system by then challenged by the emergence of the new industrial class. Smith found that there was a high demand to replace the old means of production with the industrial machines to produce commodities gradually (Smith, 1776b). With the emergence of industrial machineries and new cities developing with new social classes forming, the weakening of the feudal modes of production became inevitable (Turner, 1998).

With this taking place, the bourgeoisie critical question became how to redefine relations of production and means of production (Turner, 1998). The means of production remained in the hands of a new industrial class (bourgeoisie) and this triggered a critical question about faith in the relations of production, which led to the inevitable human resource ownership collusion within the bourgeoisie, who were in search of laborers for factories and the landlord (Smith, 1776b). Smith, as an Enlightenment thinker, sought to take advantage of this historical event to force the liberation of the relations of production so that laborers could sell their labor freely, and there would not be a labor scarcity (Smith, 1776b). In this way, this dissertation finds answers for its historical questions raised within this section in terms of the impact of the industrial mechanical machineries on the demand for the redefining of the relations of production. Similarly, around this same time frame, Smith began forming the concept and the process of the *Division of Labor* to enhance the newly redefined relations of production to improve commodity production as the preliminary industrial-based organizations were developing, which would respond to the second question and argument of this section. Below, the figure demonstrates the causal factors that transformed the feudal modes of production to the industrial modes of production process.

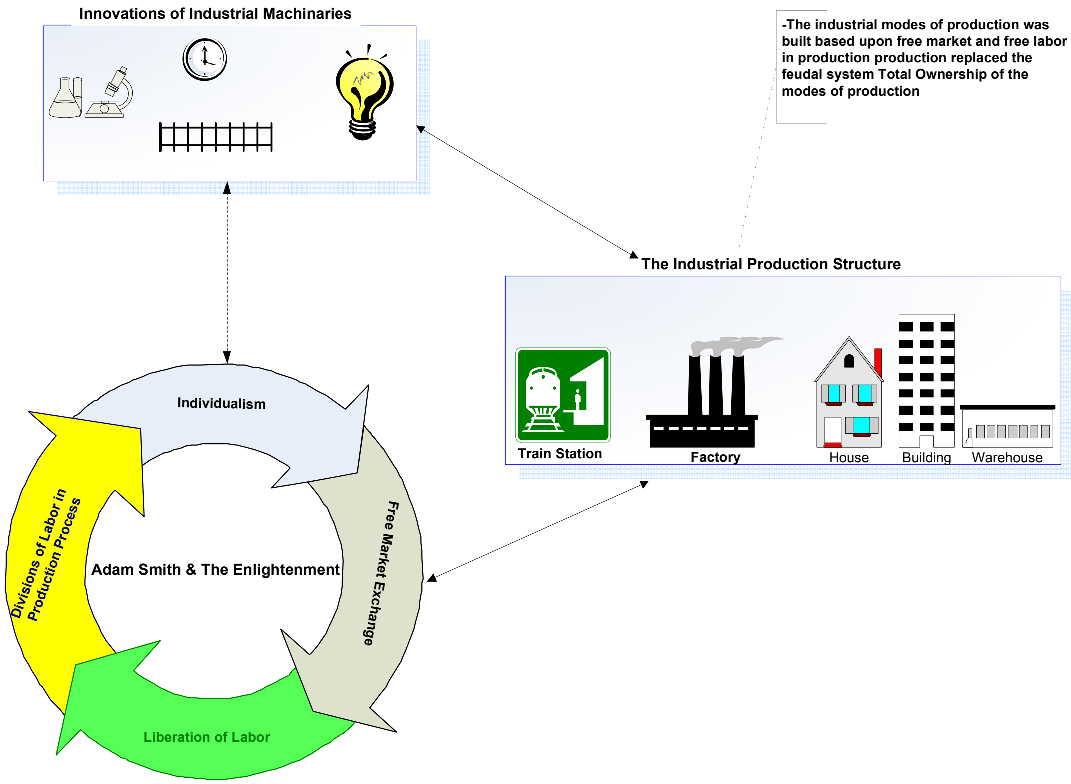


Figure 1: Transformation of Feudal's Modes of Production to Industrial

In responding to the prior raised question of Enlightenment scholars' advocacy regarding the implementation of the *Division of Labor* and the implications this brought upon the worker, it is useful to refer to Smith's *Division of Labor* theory within the production process (Smith, Adam. (1776a) Trans. (1957) by George J. Stigler. *Selections from The Wealth of Nations*). As the industrialization modes of production transformation was in progress, the demand by the factory owner for improving productivity in the production line became a central focus for Smith; he argued that the ownership of the modes of production must be redefined. By this, he meant that the relations of production had to be freed so that labor would be available to support the production line that was in demand; however, the ownership of the means of production was not a concern to him (Smith, 1776a). *Division of Labor* is a valuable commodity and to be competitive in the market, the success of the factory owner is measured by the number of productive labor forces he possesses (Smith, 1776a). Thus, he asserted that to support the type of industrial structure that was taking place in terms of labor process, individual-driven free labor was inevitable. Following this, he proposed concepts and processes to support his theoretical framework while formulating the *Division of Labor*. Using his famous pin-production line, he wrote:

> It is divided in to a number of branches, of which the greater parts are likewise peculiar trades. One man draws out the wire, another straightens it, a third cuts it, a fourth points it, a fifth grinds it at the top for receiving the head; to make the head requires two or three distinct operations; another; it is even a trade by itself to put them into the paper; and the important business of making a pin is, in this manner, divided into about eighteen distinct operations, which, in some manufactories, are all performed by distinct hands, thought in others the same man will sometimes perform two or three of them (Smith, 2, 1776a).

Thus, Smith's work responds to the second question of this section regarding the advocacy of the Enlightenment thinkers in terms of what led to the demand of the *Division of Labor*. At this point, the transformation of the relations of production has already taken place. Smith thought that the *Division of Labor* were the solution to the demand at the beginning of the industrial revolution regarding the needs of the newly emerging factories and manufacturers and their requests for labor power to increase productivity in the production process. Below, diagram 2 depicts Smith's *Division of Labor* in the industrial production process during his lifetime and thereafter, setting the preliminary foundation for the manual and sequence based production processes of the 19^{th} and 20^{th} century.

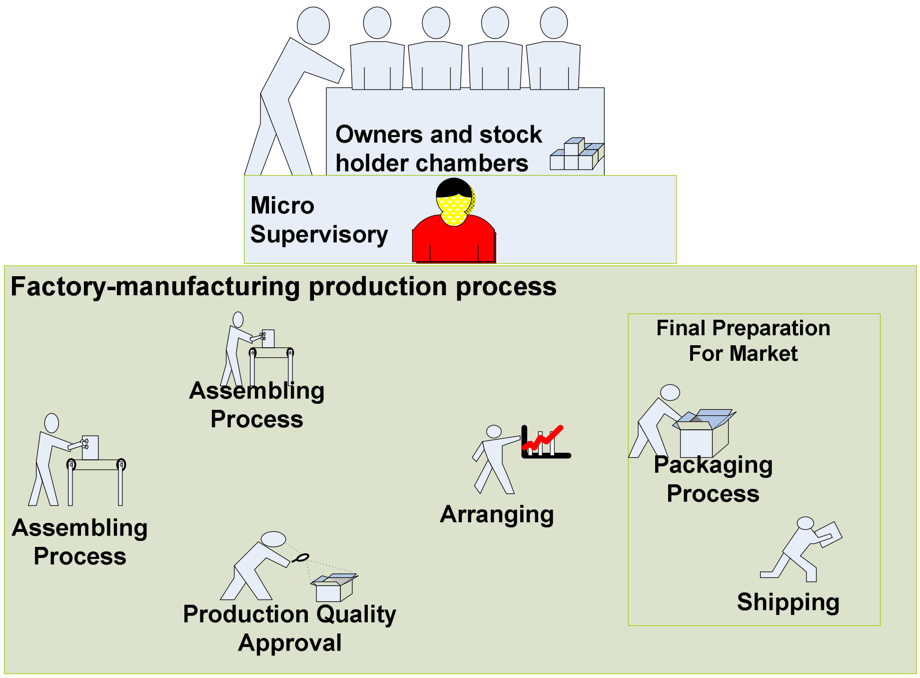


Figure 2: Adam Smith's Divisions of Labor Forces to Improve the Industrial Production Process

To measure the success of the *Division of Labor*, Smith performed a historical assessment on some of the commodities that were in demand during the 18^{th} century (Smith, 1776a). To reinforce the implementation of the productive mechanism known as the *Division of Labor* to improve productivity in production process, he proposed three critical directives. Although Smith considers these three directives supplementary for ultimately improving production processes, this dissertation perceives Smith's three supplementary directives of the *Division of Labor* as maintaining the *Division of Labor* mechanism applied by the industrial bourgeoisie to sustain their production line operations. Below, Smith defines the proposed three major directives:

> This great increase of the quantity of work, which, in consequence of the division of labor, the same number of people are capable of performing, is owning to three different circumstances: First, to increase of dexterity every particular workman; secondly, to the saving of time which is commonly lost in passing from one species of work to another; and lastly, to the invention of a great number of machines which facilitate and abridge labor, and enable one man to do the work of many. (Smith, 4-5, 1776a).

Accordingly, the dexterity of every workman is critical in making the *Division of Labor* work (Smith, 1776a). This success was measured by how well the worker became intimate with and proficient in his particular task. The more workers became intimate with their role, the more they became experts. This directive is the effort to establish specialization and expertise by embracing individualistic ideology to ensure that every workman has the confidence level to produce more in his specific task (Smith, 1776a). The new manufacturing factories brought in Smith’s dexterity as one of their directives to improve the expertise and specialization of every workman in the production line. Smith’s dexterity directive was defined and elaborated further in chapter one of the *Division of Labor*:

> The improvement of the dexterity of the workman necessarily increases the quantity of the work he can perform; and the division of labor, by reducing every man’s operation the sole employment of his life, necessarily increases very much the dexterity of the workman. (Smith 5, 1776a).

Thus, the overall aim was to develop a work force that was highly skilled in the particular task of producing only one part or component of the complete commodity. As previously noted, the market was the driving force for the demand of labor to support the new industrial production line in factories, and that led to the major task of analyzing the *Division of Labor*, undertaken by the Enlightenment scholars in general, but which became a particular quest for Smith after 1776. Following this progress, a second critical question was raised that Smith addressed in his second directive.

Time management in the production line processes became an issue in the new *Division of Labor* methodology. A second directive was introduced by Smith, whose advice was implemented as a policy for the new industrialized production process. Thus, the second directive was centrally focused on time management's reinforcing the newly implemented *Division of Labor* methodology for the industrial operation. In this light, Smith defined the second directive as pertaining to time management:

> Secondly, the advantage which is gained by saving the time commonly lost in passing from one sort of work to another is much greater than we should at first view be apt to imagine it. It is impossible to pass very quickly from one kind of work to another that is carried on in a different place, and with quite different tools. A country weaver who cultivates a small farm must lose a good deal of time in passing from his loom to the field and from the field to his loom. When the two trades can be carried on in the same workhouse, the loss of time is no doubt much less. (Smith, 5-6, 1776a).

With this directive in place, the industrial manufacturing and factory owners began to require the work force to perform the required tasks within facilities. This set the foundation for the development of the early phase of the industrial organization's operational structures, whereby the means of production and overheads were concentrated within the same location (Smith, 1776a). The primary goal was having the owners in full control of the means of production, with the production then improved by ways of a time management directive. This effort gave the owners a mechanism to increase productivity in the production line process, which became efficient throughout the course of the commodity's lifecycle (Smith, 1776a). Finally, the two directives discussed (specialization and production line process time management) were reinforced by a third directive that was expressed in the usage of means of production (industrial machineries) to make improve overall productivity in terms of production line performance (Smith, 1776a).

The third directive devised by Smith was the use of industrial mechanical machines in the production line to improve productivity. Here, the intention was to maximize production quantity to satisfy the market and the owners' demand (Smith, 1776a). This directive ensured that quantities of commodities were going to increase dramatically, because the specialized workman was already in an organized production facility using the required industrial machineries to perform the specific task assigned (Smith, 1776a). As was written by Smith, the machineries were important components in revolutionizing the way in which productions were processed to improve productivity:

> "I have seen a small manufactory where ten men only were employed, and where some of them consequently performed distinct operations using the necessary machinery, they could, when they exerted themselves, make among them about twelve pounds of pins in a day. There are in a pound upwards of four thousand pins of a middling size. Those ten persons, therefore, could make among them upwards of forty-eight thousand pins in a day. Each person, therefore, making a tenth part of forty-eight thousand pins, might be considered as making four thousand eight hundred pins a day." (Smith, 2, 1776a).

Thus, the application of industrial machineries promotes efficiency throughout the course of the production line process, satisfying the demand of the owners and the market. While these directives were in practice, the question of labor wage for the workman was placed on the back burner.

To a great extent, Smith's concern was the fact that relations of production were freed, in that the demand of the newly indoctrinated industrial market had received plenty of labor force to produce the commodities that the market demanded (Smith, 1776b). Thus, this scenario raises a question regarding wages in the labor force and profits gained by owners. Dealing with the sensitive relationship between labor wages and profit, Smith separates the two by defining labor as a productive force and profit as gained from market demand through the adjustment of the natural price of the commodities (Smith, 1776a). Smith defines labor as a valuable force for the production of commodities, and to be competitive in the market, the success of the factory owner is measured by the number of productive labor forces he possesses (Smith, 1776a). For Smith, surplus value is gained from the exchange value of the commodity at the market-retail level when a transaction is made. He strongly argued that surplus was not gained from the production process before the commodity goes out for market. Below, diagram 3 depicts Smith's industrial interrelationship among market, labor wages and profit gains:

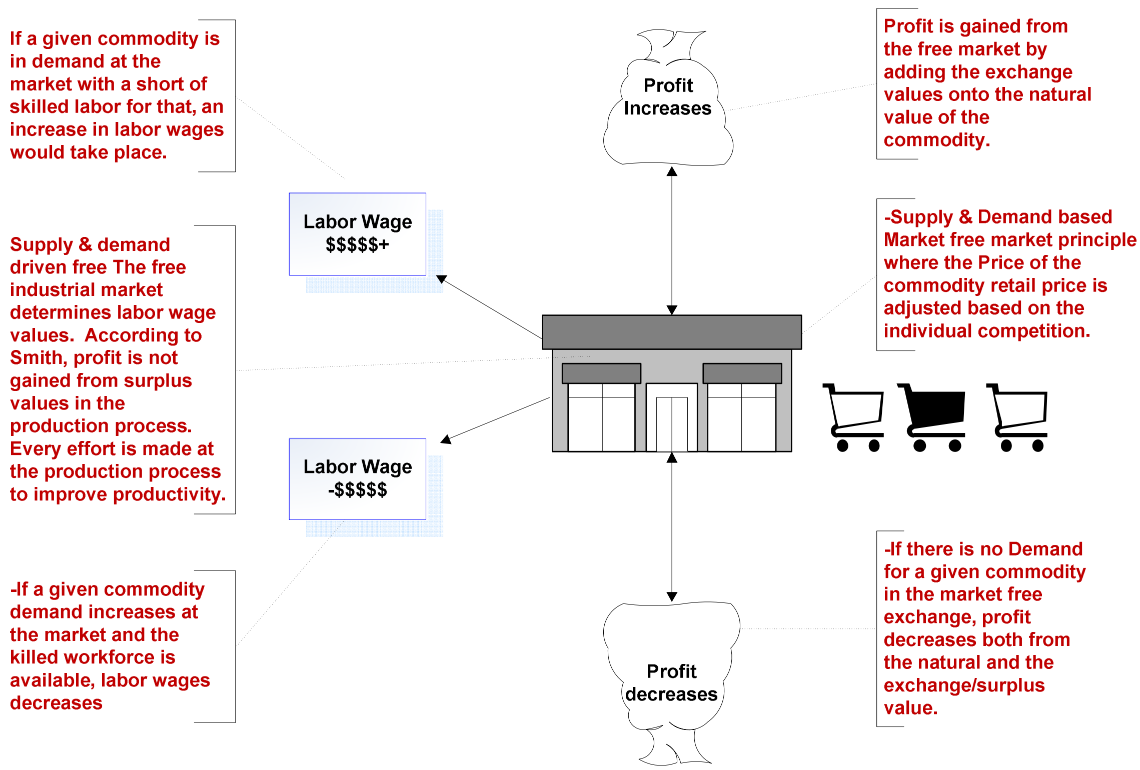


Figure 3: Smith's View of The Industrial Market Driven Profit and Labor Wage Structure

In 1776, Smith described how the market drove the newly emerged industrial organization-structures, and he established a correlation amongst industrial organization's blueprint, markets, and labor wages (including the profits):

> "The exclusive privileges of corporations' statutes of apprenticeship and all those laws which are applied, sort of enlarged monopolies, and may frequently, for ages together, and in whole classes of employments, keep up the market price of particular commodities above the natural price, and maintain both the wages of the labor and the profits of the stock employed about them somewhat above their natural rates. However, if profits are below the natural rate, the persons whose interest it affected would immediately feel the loss, and would immediately withdraw either so much land, or so much labor, or so stock, from being employed about it, that the quantity brought to market would soon be no more than sufficient to supply the effectual demand." (Smith, 45, 1776a).

Thus, for Smith, labor wage is driven by the market, since the wage is varied by the market demand for a given commodity, thereby making the profit increase right from the commodity price at the market. However, he asserted that this scenario can only increase the particular commodity's natural price; it does not increase labor wage for that commodity. What he meant is that a high demand in a given commodity does not guarantee an adjustment in labor wage. Labor wage can only be adjusted and/or increased if there is a scarcity of skilled laborers in the production line for a particular commodity that is in demand. Thus, both labor wage and the price adjustments of a commodity are regulated based upon competition amongst the factory owners.

By setting the relations of production free and using *Division of Labor* as a production improvement method, the industrial production structure would flourish for generations to come. Hence, the market was ultimately an engine of the new industrial production process, which was based upon supply and demand by adjusting the prices of commodities. Although Smith's realistic presentation has been found to be effective in terms of responding to this section's questions regarding the emergence of industrial machineries (the application of the *Division of Labor* and the transformation of the relations of production), Smith's market supply & demand-driven theory is limited in terms of labor wages and profits. This is supported in Turner's *The Emergence of Sociological Theory* (1998); even Smith himself ultimately dropped the issue of labor wages and became involved in the analysis of improving productivity.

The emergence of industrial machineries coupled with the application of the *Division of Labor* gave birth to the modern industrial period's modes of production. Thereafter, the newborn industrial modes of production brought with them a new conflict of interest between the owners of factories and their workers in terms of profits and surplus value-driven labor wages (Marx, Karl. (1867b), *Capital*: *The Process of Production of Capital*, Volume I, Trans. (1887), Edited by Engels (1890). This condition evolved into what is known to us today as modern organization's conflict in production process, which was reinforced by Saint-Simon, who was arguably the last Enlightenment scholar during the late 17th century and the early part of the 18th century (Zeitlin, Irvin. (2000), *Ideology and the Development of Sociological Theory*).

It must be asked: (1) What led to the demand for more industrial based mechanical machineries and the transformation of the means of production? (2) What caused the transformation of the relations of production, which set the peasants free to sell their labor in newly emerged metropolitans, and led to a disadvantage for workers in surplus labor wages? (3) Why did the Enlightenment scholars advocate the implementation of the *Division of Labor* and what were their negative consequences for the work force in production process? Saint-Simon, who was a protégé of Smith and his advocacy of advancement in the industrial modes of process, provides valuable historical responses in answering these questions.

Saint-Simon set out to take Smith's vision to the next level during the late 18th to early 19th century. Thus, in response to questions (1) and (2), for Saint-

Simon, the emergence of industrial machineries and the transformation of the modes of production were the results of social advancement, in which industrial and science forces developed within the feudal system (Saint-Simon, Henri, Comte De. (1817). The *New World of Henri Saint-Simon*. Trans. (1956) by Emanuel, Frank.). In his work, *Social Organizations* (1824), which focused on the historical development of society, Saint-Simon justified the idea that both the emergence of industrialization and the advancement advocated by Smith were indeed an inevitable historical occurrence built right within the feudal structures (Saint-Simon, 1824). However, Saint-Simon did not provide a causal factor for the emergence of the industrialized machineries or the transformation from feudal to the modern industrial modes of production, including the period's conflict between surplus values and profits in the production process (Saint-Simon, 1817). He was convinced that the industrial production process was not to be used as a mechanism for one class to maximize profits by exploiting subordinates; it was meant to be taken as having a positive impact on human development (Saint-Simon, 1814). With this dilemma in mind, it is critical to thoroughly examine Saint-Simon's work, *Social Organizations*, (1824) to seek historical responses to questions (1) and (2).

In an effort to systemize Adam Smith's lifetime work on the industrial relations of production and the *Division of Labor*, Saint-Simon outlined directives in his *Social Organization* (1824), which took Smith's work in this area to the next level in preparing it to be part of the apparatus of the newly emerged industrial structures. The first proposal was a plan for an industrial parliament, which included invention, examination and execution branches (*Saint-Simon*, 1817). For Saint-Simon, the invention body was a combination of engineers and artists that would be responsible for developing project plans to facilitate what the new industrial modes of production would require in production process (*Saint-Simon*, 1824). This was meant to further advance Adam Smith's *Division of Labor,* which was to improve production processes. For the examination branch, Saint-Simon recommended that scientists to take up monitoring and guidance responsibilities in directing a productive society. For the execution branches, he proposed that the industrialist ensure executions of programs and projects (*Saint-Simon*, 1824). Saint-Simon's proposal was made in an effort to further enhance Smith's industrial production process improvement by establishing a government branch to protect the industrial modes of production in general.

As Saint-Simon continued to immerse himself in the workings of Adam Smith's industrial modes of production improvement, he once again set out to enhance positions of Smith: concepts of individualism and competition driven laissez-faire advocacy. Saint-Simon began to ponder the question of individualism-driven free laissez-faire economic systems because he was not convinced Smith's version of individual-based success was going to promote advancement for the majority (*Saint-Simon*, 1817). Saint-Simon's primary concern was that this type of encouragement was going to be an unchecked and unbalanced anarchical industrial structure that would prevent the full potential advancement of the industrial system from coming to be in terms of inventions and modern creativity perspectives.

Thus, his second valuable work was focused on the new industrial society. In this treatise, Saint-Simon outlined the roles and responsibilities of artists to influence the poor class ethically to love the industrial system as a whole so that they would not be hindering factors for the development of the modern and science-driven industrial system (*Saint-Simon*, 1817). Thereafter, Saint-Simon's *Industrial Society* proposed that a hierarchal and organic society could measure productivity. It became the advanced version of Adam Smith's laissez-faire and free relations of production based upon the *Division of Labor* principle. Figure 4 demonstrates Saint-Simon's hierarchal and organic industrial system. Saint-Simon thought that the development of human knowledge, advancement in industrialization and the inevitable scientific forces within the feudal structure led to the emergence of the Industrial System.

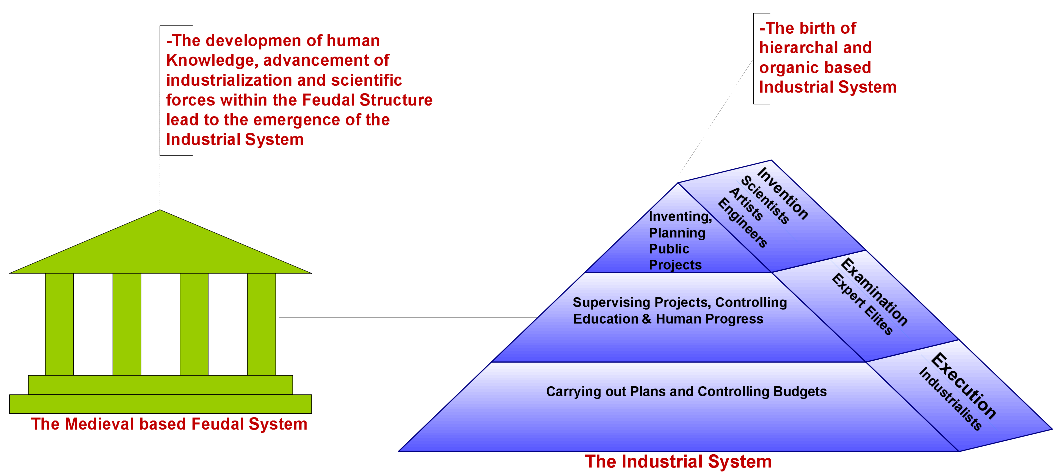


Figure 4: Saint-Simon's Progressive Industrial System

In response to the questions raised in this section, the dissertation finds adequate theoretical and practical responses by elaborating on distinct perspectives of the transformation of the modes of production. In terms of determining the causal factors for the emergence of industrial machineries, Adam Smith did not elaborate except to emphasize its usefulness in improving productivity in production process. However, the study finds Saint-Simon to be more concrete in expounding the emergence of the industrial machineries' development, as he concluded that the causal forces were the growth of knowledge and science (*Saint-Simon*, 1817). This comparison depicts both as industrialists lacking critical examinations of actual measurements of the implications within the new industrial modes of production (e.g., surplus labor wages and profit gains). While the relations of production were freed (peasants were freed) to sell their labor, the applications of absolute and relative surplus labor in the production process were soon implemented by the newly emerged bourgeoisie factory owners (Marx, Karl. (1867b). *Capital*: *The Process of Production of Capital*, Volume I.

In terms of the surplus labor wages that negatively affected workers in the production process and benefited the owners in squeezing profits during the industrial structure, both Smith and Saint-Simon arrived at relatively similar conclusions. While Smith was convinced that surplus value was gained from the market exchange value and not from production process, Saint-Simon did not even find surplus value implications in production process. Finally, Smith dropped the issue of surplus value in production process entirely, whereas Saint-

Simon concluded that it was part of the characteristics of the new industrial system that he foresaw. However, Saint-Simon's industrial system—designed to work as a hierarchal and organic structure—encouraged the elite artist to teach workers to accept the industrial production process and endorse it, or else to use alternative mechanisms to persuade them to accept the industrial production process as the only almighty religion (Saint-Simon, Henri, comte de. (1822). *Doctrine of Saint-Simon: An Exposition First Year*. Trans. (1958) by Georg G. Iggers). Hence, although classical thinkers like Marx and Engels found Smith's work to be a defense and justification for the capitalist to continue exploiting surplus labor wages in production, Saint-Simon's *Social Organization* (1824) is a systematic tool that goes beyond a mere defense to being a practice of surplus value in the production process. In reference to Saint-Simon's proposition to call for the elite artists to influence the workers to see the positive nature of the industrial system's production, Marx and Engels found him to be a naïve human being whose mind was still in the ancient world (Marx, 1867a). Nevertheless, this dissertation reverses Marx and Engles' motions in calling Saint-Simon naïve and Smith's work a defense for the surplus value practiced in the production process. As Smith and Saint-Simon's work has been compared and contrasted, it has been found that Saint-Simon was not naïve regarding these issues, and his industrial society was a scientific, industrial and systematic methodology that justified the modern industrial production process despite all its negative consequences to the labor force. It is important to note that his great concern was not with a perspective of inequality, an imbalance in operations between profit gains or a surplus of labor wages in production process.

This study furthermore argues that Saint-Simon's call for elite artists to influence workers was a manipulative attempt to systematically capture the minds of employees, convincing them to obey the new industrial production process so that his ideal industrial system would not be interrupted by social uprising. In fact, this dissertation holds that it was Saint-Simon who further systemized the industrial organizational structure in his hierarchal and organic-based industrial system that had been implemented in practice, which is relevant to a historical analysis of the *Transitional Phase*. Hence, in order to collectively and comprehensively analyze the shortcomings of Adam Smith and Saint-Simon's industrial systems' production processes, it is crucial to consider the industrial system historical evaluation performed by Karl Marx and Fredirick Engels in the 19^{th} century.

Contrary to Smith's relations of production theory and St. Simone's industrialization prophesy, this dissertation seeks a theoretical response to this chapter's questions from Karl Marx's critical interpretation of the industrial system and modes of production process. Before fully addressing Marx's work relevant to this issue, it is logical to highlight the similarities and differences among the perspectives of Adam Smith, Saint-Simon and Karl Marx: (1) What led to the demand for more industrial based mechanical machineries and the transformation of the means of production? (2) What caused the transformation of the relations of production, which set the peasants free to sell their labor in newly emerged metropolises and led to a disadvantage for workers in surplus labor wages?

Both Adam Smith and Saint-Simon attributed similar causal factors to the emergence of industrial machineries, stating that it was due to forces of social development and human progressiveness in production process. Smith sought to encourage the transformation of the feudal modes of production to the industrial modes of production to fully implement the free market system by setting the relations of production (labor free). For Saint-Simon, the transformation was due to the inevitable human knowledge-development and industrial advancement that that took place within the feudal system, giving birth to the industrial system. Marx, on the other hand, applied his dialectical framework to interpret the historical transformation of industrial modes of production and argued that the transformation occurred due to the internal conflict that took place within the feudal system between the newly emerging bourgeoisie class and the feudal land lords (Engles, Fredirick. (1868). *On Marx's CAPITAL*, republished (1940) Marx strongly disagreed with Saint-Simon's causal factors, which were the development of human advancement in knowledge and the forces of industrial development within the feudal structure that led to the transformation of the feudal modes of production (Engels, 1868).

Marx interprets the historical modes of production transformation as being based on class struggle, which was a rather distinct approach from those of Smith and Saint-Simon. To summarize, while Smith proposed production improvement and free market based upon free labor process, Saint-Simon attributed the transformation to forces of human advancement. Marx, in contrast, thought the transformation was due to inevitable class struggle. Even though he was a dialectician, Marx overlooked the interchangeability between the components of the modes of production (relations and means of production) and further neglected new modes of production birthed by the dialectical interchangeability of relations and means of production. He simply utilized relations and means of production to define the mechanics of surplus value in production process, to demonstrate class struggle between the have's and have not's and ultimately to interpret social systems. However, this dissertation takes a rather distinct approach to examine the dynamics between the relations and means of production to explain the birth of the new means of production, all the while with an emphasis on the transformation of labor process, organization and technology. With this approach, Marx's theory of transformation can be rounded out by incorporating the concept of interchangeability between the relations and means of production, which will be the primary purpose of chapter 3 and 4. In addressing Adam Smith's concept of *Division of Labor* being intended to improve production process, Marx put forth a criticism against Smith's *Division of Labor* and derives his conclusion by outlining its negative consequences in the production process for workers.

According to Marx, Smith focused on the improvement of production process, the implementation of the *Division of Labor* in the industries and the attempt to justify the concept of market-driven surplus value profit gains at the retail level (Marx, 1867a). In order to have a clear comprehension of their differences, it is important to thoroughly analyze their opposing views on the *Division of Labor* proposition. First, Marx critically assessed Smith's promotion of the *Division of Labor* in production process. They generated the concept of the *Division of Labor* in Society from the principle of the *Division of Labor* in production process. The *Division of Labor* in Society is a device meant to describe macro level social characteristics and institutional apparatuses' interactions (Engles, 1868). On the other hand, they defined the *Division of Labor* in production process as being specifically dedicated to the work forces, between which these is a direct theoretical relationship.

Marx found that Smith's indoctrination of the *Division of Labor* improved production process. He learned that the practice of specialization in tasks was hindering workers' potential self-development as human beings (Marx, 1867a). According to Marx, specialization drastically isolated workers and disconnected them from participating in the full development of the product in the production process. Marx took this preliminary finding further to examine the actual manifestation of this separation by introducing the concept of Alienation. Although Marx expanded the concept of Alienation over the course of 1848-1867 to examine many components of society (e.g., self-consciousness, nature, objectification and division of labor), the specific elements of Marx's Alienation that are relevant to questions about false consciousness are the components of production and the productive activity within the *Division of Labor* in production process (Marx, 1848).

Contrary to Smith's advocacy of specialization in the *Division of Labor*, an effort to improve production process, Marx concentrated on listing and defining all the negative consequences of the *Division of Labor* in production and argued that workers are alienated from their own products that they produce in the production process (Marx, 1867a). With this, Marx attempted to show that through constant production the worker becomes, in effect, the slave of his or her own product. As the object increases in value, workers ultimately decrease in worth as human beings. By saying this, he implies that individuals who work to satisfy others' desire for objects themselves become an object to the owner of the business (Marx, 1867a). What Marx purports is that if workers create a product for themselves instead of creating for the demand of others, they would not be estranged from the product they produce. He suggests that to reduce the level of impact of Alienation on workers, they would have to be capable of choosing what they would do to fulfill their daily needs without another's demands playing into that decision (Marx, 1848). In a nutshell, Marx's Alienation from product refers to workers producing products that do not belong to them, whereby they share no profits with the owner of the product who has no direct or indirect labor-participation in the production process. This form of Alienation challenges one of Smith's theories regarding having a high number of workers at a given time to increase and improve productivity in production process by applying the mechanism he called the *Division of Labor*. With this finding, Marx continued to advance to the second form of his Alienation, defined as a productive activity, which refers to the actual production process.

Marx's concept of workers' alienation from productive activity justified the hindering elements of the specialization principle of the *Division of Labor* in production process, a doctrine that Adam Smith promoted and encouraged for implementation. Marx's second form of Alienation that of a man in the act of producing is a core theoretical response to this dissertation's question regarding the specialization aspect of the *Division of Labor* role in production process. Marx sees a problem with the actual act of production—that is to say, he thinks the relationship between man and his actual activity of labor, if forced, is unnatural (Marx, 1848). According to Marx, in the modern industry production process, workers typically do not work under their own direction (Marx, 1848). Product owners, using the principle of specialization within the *Division of Labor* mechanism, require workers to be assembled in large factories or production facilities, and their activity is under the close supervision of a hierarchy of site supervisors (Marx, 1848). The site supervisor does the planning, and managers divide complex work processes into simple, repetitive tasks which workers can perform with machines (Marx, 1848). This means that Marx found that the application of specialization driven *Division of Labor* in production environment does not enable the individual worker to perform his or her required tasks, including planning and arranging the tasks to complete the product lifecycle. This is particularly relevant in supporting chapter 3, which analyzes the hierarchy-driven industrial management structure and the *Waterfall* specialization based production process methodology in computer engineering during the period of 1970-1995. Similarly to Marx's classic Alienation from productive activity, chapter 3 describes the practice of specialization-based *Division of Labor* in the computer engineering production process. A practical instance is the inability of a software programmer to perform all of the production process tasks, including the inability to plan the project, the product requirements, to analyze, to design, deliver for final testing and to release the end product to the user/customer. These tasks are performed by different individuals whereby each is performing a specific task due to the requirement of the specialization based *Division of Labor* practice in the production process. In this practical scenario, while productivity was increasing for the organization, the potential development and ability to perform the tasks that an individual software programmer executes was hindered because of the principle of the specialization-driven *Division of Labor* in production process during the period of 1970-1995.

This dissertation also seeks from Marx theoretical responses to its question pertaining to the disadvantage of workers' labor wages and the surplus value-driven profit gains for owners within the production process. In this regard, Marx primarily focused on the negative consequences of Smith's advocacy by differentiating necessity labor for use value from surplus labor for exchange value (Engles, 1868). Perhaps the only core similarity that this dissertation finds between Marx and Smith is that both of them dedicated 20 years of their scholarly work to modes of production study (from, of course, distinct perspectives) and arrived at distinct conclusions. Marx performed a criticism of Smith's political economy, and in the process, he redefined Smith's concepts of 'use value' and 'exchange value,' which later became the core dimensions of Marxist economic principles (Engles, 1868). Marx argued that Smith both separated the value of labor from the production process and claimed that the industrial society's production process used labor to increase productivity in production process (Engles, 1868). The surplus (exchange) value is achieved from the retail market exchange (Engles, 1868). Thus, according to Smith, the surplus process is achieved from market competition, whereby owners are able to adjust product prices by increasing and decreasing the commodity market price (Engles, 1868). According to Marx, Smith was simply justifying an industrial society that has no market law and regulation on the one hand; on the other, Smith was in denial about the real fact that surplus value is embedded within the actual production process while the workers are producing the commodity (Engles, 1868).

With this argument in mind, Marx discovered the concept of surplus values within the production process. He separated use value from surplus (exchange) value by arguing that use value is the necessary labor that the workers need to survive, and exchange value is the free surplus value labor furnished by the laborers during the production process. Smith, differing, introduced the concept of use value, applying it to the industrial production process by declaring that commodities not produced for market and exchange value are the surplus value that is added on top of the natural value of a commodity produced for market purposes (Smith, 1776b). In addition, Smith defined the basic concept of exchange value as a means of collecting surplus values at the retail level during the actual market-transaction by increasing the value of the commodity (Smith, 1776b).

Nevertheless, while this may be a valuable contribution from Smith's work, consider also an investigation of *Das Capital*'s (Marx, 1867) critical assessments of labor wage and profit-gains in a comparative analysis of Smith's aforementioned contributions. In *Das Capital* (1867), Marx altered Smith's exchange surplus value and introduced his classical surplus value components (absolute and relative surplus values). From a historical perspective, the table below specifies Marx's classical surplus values with their attributes:

Table 1: Marx's classical surplus values:

Social System	***Types of Surplus Values***	***Attributes***
Hunt and Gathering	None existed in the production process. Defined as communal based production environment.	Everyone worked independently.
Feudal	Absolute Surplus Value	Absolute surplus value was practiced in the production process. The land lord owned both the means and relations of production.
Industrial - Capital	Absolute and Relative Surplus Values	Absolute surplus is applied to more than half of the working days that is beyond the necessity labor to extract free labor long working days translated to profits.
Socialism	Wages distributed by the state to prevent and abolish surplus values.	Private properties and ownership are nationalized.
Communism	Surplus values do not exist.	A classless social system where everyone practices their daily productive tasks to their full potential capability to produce their daily commodities.

Source: Marx, *Das Capital*, 1867

Comparing Smith's theory to that of Marx effectively responds to this dissertation's fundamental research questions regarding labor wages and profit gains. In *Das Captial* (1867), Marx defines his classical discovery of surplus values in the following words:

> The surplus-value produced by prolongation of the working-day, I call absolute surplus-value. On the other hand, the surplus-value arising from

> the curtailment of the necessary labor-time, and from the corresponding alternation in the respective lengths of the two components of the working-day, I call relative surplus-value. (Marx, 299, 1867b)

In addition, Marx argued that the application of the *Division of Labor* in the production process coupled with the use of machineries is a typical attribute of the relative surplus value (Marx, 1867b).

Thus, in sum, this dissertation has set its theoretical formation by seeking responses to its research questions from the work of Adam Smith, Saint-Simon and Karl Marx in terms of the following: (1) What forced the modes of production to be transformed? (2) What led to the emergence of industrial machinery and technological development? (3) What caused wage disadvantages for workers in production process and how is profit gain accomplished in the production process? This dissertation's findings from the three aforementioned thinkers are valuable, and the research questions were effectively answered by an historical social-assessment of these theorists. For instance, in terms of modes of production-transformation, it was found that Smith's production process improvement was effective in providing a base theoretical understanding of the birth of the industrial modes of production and its new production process. Smith's work in encouraging the freeing of the relations of production from the feudal structure provided a causal explanation about rising demand of labor in the newly emerging industrial production process. Similarly, this dissertation found an advanced theoretical framework from Saint-Simon, which introduced a modern and systematic industrial system to fully implement Smith's vision of the free market-based industrial society. Saint-Simon further systemized the idea of industrial society and introduced a hierarchical and organic-driven industrial production process by engineering institutional apparatuses.

On the other hand, causal explanations were found regarding the above research questions on the modes of production transformation, wage and profit gains. In terms of analyzing the transformation of the modes of production, Karl Marx analyzed historical organizations' change in management structure and production (in relation to the labor force) by using the dimensions of the mode of production (relations of production and means of production) (Engles, 1868). From Marx, it was found that the ultimate modes of production takes place as a result of class struggle between the have's (owners of the means of production) and the have not's (proletarians) (Karl Marx, (1867b), *Capital*: *The Process of Production of Capital*, Volume I). In terms of wage and profit gains, Karl Marx's more than 20 years of industrial production process-investigation fully covered the complete range of this dissertation's questions regarding wage and profit gains.

Although Marx's surplus value theory had a profound impact on this dissertation's surplus value theory (coined *Triple Surplus Value*), Marx's dialectal methodology *did not* have an impact on the formation of this dissertation's dialectal approach regarding the transformation of the modes of production (a transformation that was due to the interchangeability of the modes of production's components: relations and means of production). This idea (regarding the dialectics of relations and means of production) is a novel contribution to the understanding of the history of the transformation of modes of production. Marx as a dialectician did not account for the dynamics of the means of production and relations of production in transforming the modes of production, nor did he explain how the dynamics negatively affect labor force wages. Instead, Marx argued that transformation was based on the end result of the inevitable class struggle between the have's and the have not's. Hence, this dissertation attempts to address the shortcomings of Marx's approach by arguing that the modes of production is transformed as a result of the dynamics of the means of production and the relations of production.

2.3 Paving the Way: Rationalization of Capitalist Organizational Modes of Production

The primary purpose of this section is to analyze both Emile Durkheim's characterization of the *Division of Labor* in production process and Max Weber's rationalization theory regarding industrial modes of production. This section will analyze the work of Emile Durkheim and Max Weber to acquire theoretical responses to the research questions defined in section 1 of this chapter. Emile Durkheim's general characterization of the *Division of Labor in Society*, an idea that expands beyond the traditional division in production process (that of Adam Smith and Saint-Simon), will be explored. Last but not least, this section analyzes Max Weber's Iron-Cage, as it has direct relevance to this dissertation in terms of relations of production.

The evolution of the *Division of Labor* in production process has been examined in an effort to answer critical questions in this study regarding the transformation of relations of production. As Saint-Simon systemized the original *Division of Labor* structure (belonging to Adam Smith), Saint-Simon had a major impact on influencing the thinking of Emile Durkheim, expanding work on the *Division of Labor* in society with specialization in production process. Durkheim’s classic *Division of Labor in Society* (1893) concerning modern organizational structure and specialization has its foundations in Saint-Simon’s theory of industrialization. With this in mind, the present study seeks theoretical responses from Durkheim to answer its questions about the implementation of the *Division of Labor* in production process and the implication of surplus labor for workers in production process.

According to Durkheim, the emergence of the transformation of the feudal relations of production was a result of the inevitable progress of the *Division of Labor* in society (Durkheim, 1893). For Durkheim, society has an organic characteristic to it because of the specialized tasks that workers perform in production process (Durkheim, 1893). So, for Durkheim, the *Division of Labor* is progressive because it is held together by the interdependence of specialized employees that complete the lifecycle of a product in the production process. The completion of one task’s output is the input and beginning of the next task. Thus, he looked at the causal factors that made the *Division of Labor* successful in production process and beyond.

As discussed above, Durkheim characterized the modern industrial organizations’ functions as analogous to a biological organism, wherein the structure and *Division of Labor* are interrelated, because all are playing roles according to their specialized forms (Durkheim, 1893). He also characterized the industrial organizational structures and their production processes by introducing mechanical and organic solidarity. Mechanical solidarity was put forth to address the feudal modes of production's structure, in which tasks were performed in a uniform manner; organic solidarity describes the *modern* industrialized modes of production's structure. As noted, the *Division of Labor*'s specialization in production process reinforced interdependence among workers throughout the course of the lifecycle of a product in the production process (Durkheim, 1893).

Durkheim argued that the disadvantage for workers in terms of wages in production process was due to a lack of common morality in the organic-based *Division of Labor* (Durkheim, 1893). Similar to his predecessors (Saint-Simon, who recommend that artists educate workers about the greatness of the industrial system, and Adam Smith, who came up short in logically proving the existence of surplus labor value), Durkheim dropped the issue of surplus labor value all together. Although his *Division of Labor in Society* (1893) attempts to examine organizational structures and their characteristics, the doctrine's limitations are an inability to discover and to specify the base causal factors for historical changes in organizational structures and operations. He suggests an alluring (but ultimately vague and unconvincing) explanation that states changes in historical organizational structure, which are the result of the development of the *Division of Labor*. Durkheim, with his optimistic view of specialization-interdependence as a sign of organizational advancement, became a target by the one of the 19th century's most prominent social thinkers, Max Weber.

An examination of Max Weber's theory of rationalization can establish historical trends of management structure, which can further be utilized in an analysis of pyramid-based management structures within computer engineering organizations' management structures. While Smith, Saint-Simon, and Durkheim focused on production process improvement, Weber looked at the rationalizations coming from the bureaucratic management structures of the modern industrial system, a consideration his predecessors (along with Durkheim) did not address in their analysis. Their assessment of the industrial system was partial in that it limited itself to the production process by focusing only on the implementation of the *Division of Labor* and the improvement of the production process. In contrast, Weber centrally focused on analyzing the overall complexity of bureaucratic management systems and how organizations were able to manage and control production process by applying methods of calculation (Weber, Max. (1909). *The Essential Weber*, Trans. (2004) by Sam Whimster). As *Division of Labor* and production process were the core for Smith, Saint-Simon, Durkheim and Marx, Weber's was rationality and the process of rationalization in the industrial system. To answer questions about the transformation of the relations of production from a feudal to an industrial social system, Weber applied and implemented the concept of rationalization. For him, the transformation of the feudal modes of production was the result of advanced development of the specialization skill sets and the emergence of advanced machineries, a position he shares with his preceding social thinkers. However, Weber's concept of rationality was the method he outlined as fundamental to examining the transformation from feudal relations of production (e.g., administration and management structures) to an industrial bureaucratic system (Weber, Max. (1904-5a). *On Capitalism, Bureaucracy and Religion*. Trans. (1983), by Stanislav Andreski). In order to analyze this change, Weber defined the feudal relations of production as a patrimonial administration, which he characterized as based on loyalty and kinship (Weber, 1909). However, according to Weber, the rationalization process became progressive, in that advances in technicality, specialization and modernization in formal management mechanisms eventually formed the management aspects of the relations of production, bureaucratizing (as he called it) the feudal administrative structures. (Weber, 1909). In this manner, Weber defined industrial organizations' management systems as bureaucracy-driven and complex in terms of calculation (e.g., accounting bases of estimates, mechanisms of computing profit gain and controlling production process). Weber's definition of bureaucracy is hierarchy-based, a system in which the individual worker rationalizes acquiescing to the organization's operational rules instructed to him/her to obey, which he calls the Iron Cage (Weber, 1904-5a). Weber's bureaucracy concept confirms this dissertation's defining the industrial management structure as a pyramid (hierarchy-based) management structure during the 1970-1995 time period (the *Transitional Phase*). In similar fashion to Weber's bureaucratic characterization of the industrial management structure, this dissertation defines the *Transitional Phase*'s management structure as a mechanism used to sustain the surplus/exchange labor value mechanism. To ensure this sustentation, it was necessary that the management structure be hierarchical and pyramid-shaped. The structural similarity between the production and management mechanisms was essential to making the entire operation functional. With this pyramid management mechanism in place reinforcing surplus/exchange labor values, the system worked for the *Transitional Phase*.

In terms of wage and profit gains, Weber focused on analyzing the methods of bookkeeping, accounting principles and how they were applied to increase profits. Although he agrees with Marx in terms of the process of surplus labor, his focus was how industrial organizations manage to utilize the calculation devices in production process. Weber thought production line process and bureau/office operational processes were quite similar given how specific tasks were conducted (Weber, 1909). The workers accepted their roles and responsibilities at the office level by executing the process and procedures implemented by high-level bureaucrats; these mass-subordinate workers blindly gave their allegiance to the organization that hired them (Weber, 1909). Wage payment was accordingly calculated, controlled and distributed by specialized accounting officers backed by written consent of the mass-workers in the production process (Weber, 1909). This scenario was described by Weber as the Iron Cage (1909) and was translated by Sam Whimster in the *Essential Weber* (2004):

> The management of the modern office is based upon written documents (the 'file'), which are preserved in their original or draft from, and upon a staff of subaltern officials and scribes of all sorts. The body of officials working in an agency along with the respective apparatus of materials implements and the files makes up a bureau (in private enterprises often called the 'counting house', Kontor)." 246

With this type of bureaucratic administrative system, management was used to reinforce the production process by their determining wages and guaranteeing profit gains.

In sum, Weber's rationality, rationalization and Iron Cage concepts effectively respond to this dissertation's question regarding the role of management in production process. While rationality as a theory defined the management mechanism (the hierarchy-driven bureaucratic management structure), rationalization ensured the implementation of bureaucratic mechanisms' procedures that reinforced how production process was executed. Most significantly, Iron Cage surfaced the reality that both the management worker at the bureau/office level and the specialized workers from the production process perspective are equally hindered by the Iron Cage, which was used by Weber as an analogue to describe the industrial systemic mechanisms that organizations utilized to sustain surplus labor in production.

2.4 Workers' Consent in Production Process

Michael Burawoy assessed the manifestation of workers' consent in production process, surplus labor and the management mechanisms that reinforce and guarantee sustainable profit gains (Burawoy, 1979). Although Burawoy concludes that workers are unaware of surplus labor in production process, they do consent to capitalist managerial agent/representatives to enter into production process. However, his attempt to theorize causal factors leading to the transformation of the industrial production process did not contribute core theoretical dimensions to justifying how the production process transformation was limited, because the emergence of electronic-based technological devices that revolutionized the production process was not part of his theoretical equation.

Nevertheless, Burawoy's work is critical to the present analysis. His study's timeframe happens to be right around 1970, the beginning of the *Transitional Phase*, and he conducts an historical analysis of this time period, drawing from classical theorists. Most critical to analyze is his work regarding both surplus labor in production process and organizational perspectives. Burawoy began his research of the labor process in hopes of determining the cause(s) of both the production process transformation and workers' consent in an organization. His core research question was, "How is unpaid labor mystified to the worker under capitalism?" (Burawoy, 28, 1979). During his observation of modern industrial organization's management principles that began in the early to mid-1970's, Burawoy found and wrote:

> Just as serfs have to fend for themselves in the time remaining for their own production, so capitalists or their managerial agents have to organize the labor process so as to ensure the extraction of unpaid labor. (Burawoy, 26, 1979).

Here, one can find a viable theoretical and practical response to the aforementioned research questions in terms of how the role of an organization's management forms mechanisms to maintain surplus labor in production process. Burawoy's organizational management analysis is linked to the 20th century Weberian school, because a strong correlation has been found between Weber's rationalization of organizations' systemic-accounting methods—used to extract surplus labor to gain profits—and Burawoy's observation of the management functions in developing complex management processes. Also, as discussed earlier in this chapter and in chapter 1, this connection confirms the validity of this dissertation's approach in terms of its argument that computer engineering organizations' management creates the methods that enable them to control the surplus labor in production process during the *Transitional Phase* and the Age of Information (a period after 1995 coined by this dissertation). As presented in chapter 1 and 2, the computer engineering management methods *Waterfall* and pyramid are applied to control surplus labor in production process. In these structures, during the *Transitional Phase*, management calculated the wage for each specialized field in advance (e.g., requirements analyst, designer, programmer, tester and deployment analyst). According to the *Occupational Out-Look of Department of Labor Statistics*, 1970-2009, the average wage-rate in each computer engineering field is specified. However, depending on the demand of the specific field, wage rates are adjusted by the accounting control offices of the computer engineering entities to match such demands.

In 1974-75, Burawoy observed this accounting practice in the Allied Corporation while working there for 10 months, during which he said, "When restrictions on working hours were legislated, capitalists resorted to intensification of the labor process through speedups, introduction of piece wages, and mechanization" (Burawoy, 1979). Furthermore, in this same assessment, he found that the sophistication of the organization's operation principle motivates workers' consent by their succumbing to management's methods of incentives, bonuses and benefits practices (Burawoy, 1979). Burawoy concludes that workers who are unaware of the concept of surplus labor in production process generally consent to an organization's operational principles in advance before entering the production process (Burawoy, 1979). Thus, Burawoy's theory is that the production process-transformation was caused by the complexity of advanced organizational management-mechanisms, entities that reinforced more specialized Division of Labor in the production process.

Burawoy further advanced a concern that workers in the production process realize surplus value in the form of profit only through the sale of commodities in the market. He theorized that the implementation of complex management schemas (e.g., concepts of bonuses, incentives, and benefits) coupled with the rationalization of these mechanisms made workers unable to realize that the real value of profit-gaining is embedded in the labor production process (Burawoy, 1979). Further intriguing in Burawoy's study is that the implementation of management schemas did not only prohibit workers from realizing where the real value of profit emerges but also hindered them to a point where they were not able to separate surplus labor from necessity labor in the production process. This is quite similar to the dissertation's findings in terms of the contemporary computer engineering organization's management schema and its impact on causing false consciousness in workers. Buraway wrote of his observation:

> How, then is unpaid labor mystified to the worker under capitalism? We have already seen that necessary and surplus labor time is not distinguished within the labor process and also how the wage conceals

> such a distinction. Workers have no sense of producing the wage equivalent or the means of subsistence, since what they produce is but a fraction of a useful object that they may never even see. Nevertheless, the process of production appears to workers as a labor process, that is, as the production of things-use value rather than the production of exchange value. (Burawoy, 28, 1979)

For the same reasons, workers are not able to realize that without their labor forces there would not be any successful profit-gaining for stakeholders (owners of organizations). As Burawoy observed while at Allied Corporation's production process, workers were committed to the success of their stakeholders' profit increase, believing that such gains would guarantee their continued employment with the company. According to Burawoy, management mechanisms made the entire surplus value operation so invisible that the worker realized neither his or her wanting to quit nor his or her hope that eventually there would be some type of promotion as a reward for motivation and dedication in productivity (Burawoy, 1979).

In sum, this dissertation finds common denominators with Burawoy's *Manufacturing Consent* in terms of a focus on production process and organizations' management methods and an inquiry regarding the cause(s) of the transformation of the production process. Pertaining to the latter, it is apparent that both studies found distinct variables and arrived at very distinct conclusions. While Burawoy proposed that the (relations of production) management's mechanisms were causal factors in ultimately transforming the production process, this dissertation searches for a dialectical relationship between the relations of production and the means of production. As has been stated, this dynamic causally factors into the transformation of the following: the production process, the management structure and, ultimately, the transformation of the modes of production from the industrial modes of production to those of the Age of Information.

Nevertheless, Burawoy's *Manufacturing Consent* provides valuable answers to questions about discussions of workers' consciousness. In fact, Burawoy's *Manufacturing Consent* can be seen as a theory that both measures the ability of workers to understand surplus labor in production process and finds the causal factor(s) for why workers believe that profit is gained from market retail transactions. Hence, it is apparent that this dissertation challenges *Manufacturing Consent* by arguing that it is the dynamics between the relations of production (management) and the means of production (advanced technology) that transform production process. The present study further disputes with Burawoy what exactly transforms the *Division of Labor* in production processes, surplus labor wages and profit gains. Within this context, the next section of the dissertation will analyze the role of the means of production by exploring technological utopianism and anti-technological utopianism.

2.5 Technological Utopianism vs. Anti-Technological Utopianism

Here, the theories of Technological Utopianism and Anti-Technological Utopianism are analyzed to respond to three questions: (1) What led to the demand for more industrial-based mechanical machineries and the transformation of the means of production? (2) What caused the transformation of the relations of production, which set the peasants free to sell their labor in newly emerged metropolitans and led to a disadvantage for workers in surplus labor wages? (3) Why did the Enlightenment scholars advocate the implementation of the Division of Labor, and what were its negative consequences for the work force in production process?

As discussed by Rob Kling's *Computerization and Controversies* (1997), some techno-social theories have formulated opposing sides (Technological Utopianism and Anti-Technological Utopianism) regarding the role of technology within the contexts of *Division of Labor* in production process, implications of labor wages and organizational growth in profit gains (Kling, 1997). There are similarities in the thinking of Technological Utopianism and that of St. Simone and Adam Smith, the former building on a foundation from the latter. While the pro-technological utopians endorse only the advantage of technological advancement in improving the ways in which production was processed for organizations, they do not consider the negative consequences posed towards the work force in terms of surplus labor wages in production process. In contrast, the Anti-Technological school of thought blames advancements in computerized-based innovations as responsible for all the negative consequences that occur to the workforce, such as surplus labor in production process, jobs becoming obsolete and the cumbersome complexities in *Division of Labor*'s implementation in the production line (Kling, 1997). In order to have a clear understanding of these opposing schools of thought, it is important to assess their positions regarding the research question mentioned above.

What the contemporary advocates of Technological Utopianism are misconceiving is what the eighteenth century advocates of industrialization production process, like Saint-Simon and Adam Smith, also did not conceive: that the advanced innovations implemented to improve production process consequently causes severe conditions in the work force. As illustrated earlier, both Adam Smith and Saint-Simon contributed greatly to the shaping of the industrial system's production process. As noted above, their contribution set the foundation for the development of hierarchy-driven organizational structures with their own *Division of Labor*-based production process. This is the notion influencing the formation of the Technological Utopian school of thought's prophecies on computerization and applications of advance technologies to improve production process (Kling, 1997). Within this context, the dissertation argues that the theoretical perspective of the Technological Utopian school of thought advocates computerization as a means to improving production process. The Technological Utopian school of thought further advances its argument stating that the great transformation of the industrial production system was solely due to the development of computerization over the most recent four decades (Kling, 1997). According to the Technological Utopian school of thought's argument, technology has added value in the production process. However, this view is limited, lacking consideration of certain negative consequences: namely, that the application of advanced computerization causes disadvantages in surplus labor wages and alienation in production processes. This limitation of the Technological Utopian school of thought is challenged by the Anti-Technological Utopian school of thought.

Although Anti-Technological Utopian school of thought has taken a stand against the technological prophecies of the Technological Utopian school of thought, the Technological Anti-Utopian school of thought is presenting only partial analytical approaches in terms of the following concepts: 1) They hold that innovations in advanced technology, computerization and scientists in the field were responsible for the drastic transformation of the industrial production process, a change to more complex computerized production processes (which resulted in negative consequences of surplus labor wages and in the production process). 2) The implementation of contemporary computerization and its continual innovation by technologists are transforming the *Division of Labor* in production process and causing jobs to become obsolete by condensing varieties of expertise, which brings new social stress to the job market (Kling, 1997). 3) Organizations are applying advanced production devices and implementing computerization to automate tasks in production processes to improve productivity, which leads to promising increases in profit gains. The Anti-Technological Utopian school of thought has erroneously conceptualized the roles of the technologist and computerization in production process. Both schools' analysis focuses on how computerization affects society, but their analyses should examine how organizations utilize technologies as a means to an end in accumulating wealth, gaining surplus value and deluding workers. While the

Technological Utopian school of thought emphasizes social progressiveness through computerization, the Anti-Technological Utopian school of thought blames scientists and technologists for the possibly negative social consequences of computerization. As has been argued, leading techno-society schools of thought tend to negate each other's perspectives while the real essence of the issue remains untouched in virtue of their partial analyses.

Advanced computerization has been used by organizations as a means to an end, whereby organizational goals are achieved in improving productivity in production process. Indeed, technologies have been used by certain groups to transform the relations of production (Kling, 1997). For instance, the Age of Science led to the emergence of the Industrial Revolution, which in turn partially transformed production process, paving the way for the birth of the traditional hierarchy-driven production process (Saint-Simon, 1824). This only logically brings us to an historical analysis of the *Transitional Phase*. The *Transitional Phase* (1970-1995) is a period dedicated to the analysis of the industrial modes of production (management structure, specialization-based *Division of Labor* in production process) and further examines surplus labor wages and profit gains within the context of the computer engineering industry. Under this system, according to the findings in this dissertation, computer engineering entities increase efficiency, productivity and profits, and they systemically control the production process by implementing surplus labor wages. These two components of surplus labor method work to guarantee organizations' growth from gaining profits and increasing productivity (Burawoy, 1979). In this surplus method, workers in production process are not able to differentiate necessity labor from surplus labor, in that for them wages appear to compensate for the totality of their working hours (Burawoy, 1979). As discussed, computer engineering entities increase their production by utilizing advanced software and hardware devices in the production process, which translates into growth (Kling, 1997) Thus, computer engineering organizations utilized computerization (coupled with surplus-sustaining managerial methods) during the mid-70's and 80's to maximize growth, and consequently, workers' labor wages in the production process did not parallel the increased profits that the company earned as a result of greater productivity. As chapter 3 portrays in its data-analysis (wage data) of the Department of Labor's 25 years' (1970-1995) record of wage compensation in the field of computer-engineering, workers were not compensated for their increased productivity. Hence, what this proves is that computer-engineering entities have benefited in terms of maximizing their growth by applying computer devices to (and implementing surplus labor methods in) production process, causing disadvantages to workers.

In sum, neither the Technological Utopian school of thought nor the Anti-Technological Utopian school of thought was able to explore this effect on the labor force as a core dimension in their theoretical framework; they lack a critical approach. Blaming computerization implies that technologically related organization does not constitute a contribution to worker's disadvantages in terms of surplus labor wages and their being deluded about the reality of surplus in the production process. The analysis should focus on how organizations utilize computerization to reengineer their operational systems by applying systematic deception to workers in the production process (Triple Surplus Value: absolute, relative, and complex surplus value).

2.6 Summary

In conclusion, this chapter sought an explanation regarding its research inquires discussed above. This section highlighted social thinkers (Smith, Saint-Simon, Marx, Durkheim, Weber, Burawoy and Kling) that focused on distinct subject matters and arrived at distinct conclusions. Core explanations were extracted from Adam Smith's theory of free labor advocacy, which was ultimately responsible for the transformation of the relations of production from the feudal production process to the industrial. Although Smith's concept of free labor liberating the relations of production was effective in facilitating labor demands for the newly emerging industrial production process, it has been argued that it set peasants up for complex surplus labor disadvantages. In regards to the present study's inquiry regarding the role of the *Division of Labor,* this study finds an historical explanation from Adam Smith's *Division of Labor* in production process, as well as from his successor, Saint-Simon's *Industrial System.* While Adam Smith set the foundation for the formation of the *Division of Labor* (supplementing it with concepts of specialization, use of machineries and time saving methods), Saint-Simon justified and introduced measurable roles and responsibilities in production process. However, their theoretical framework was limited, because they did not concern themselves with surplus labor and profit-gaining from production process. As discussed above in this section, this scenario was targeted by Karl Marx and Fredrick Engels in the 1900^{th} century.

Adam Smith's use value and exchange value were fundamental concepts for the formation of Karl Marx's surplus value theory. Using Smith's concepts of use value and exchange value, Marx developed absolute and relative surplus labor in production process. Further, this helped Marx in developing the advanced version of surplus value, defined as necessity labor (paid/use labor) vs. surplus labor (free/exchange labor) in production process. In addition to finding explanations to this dissertation's inquiry regarding wage and profit gains, Marx's surplus value theory is further explored and applied as a foundation in the formation of this dissertation's *Triple Surplus Value* Theory presented in chapter 4. Surplus labor, wage and profit gains in production process are further explored by Max Weber and later by Michael Burawoy. Although on a different timeline in history, Weber's rationalization and bureaucracy theory argued that the industrial management structure was used as a method to control workers and sustain the practice of surplus value in organizations. Similarly, about 55 years later, in his *Manufacturing Consent*, Burawoy argued that the production process was transformed by the complexity of the management technicalities, causing transformation in production process, as well as hindering workers' realization of surplus labor in production process (profits gained from market transaction). With these arguments in mind, chapter 3 of this dissertation will apply an historical comparative method and use the Department of Labor's statistical data to explore/demonstrate a *Transitional Phase* tension between surplus labor and profit gains in the production process, the conflicting roles of *Division of Labor* and the computerization and management practices of computer engineering organizations.

CHAPTRE 3

TRTNASITIONAL PHASE (1970-1995)

3.1 Introduction

The overall theme of this chapter is to analyze the *Transitional Phase*'s industrial modes of production infrastructures, which are the pyramid organizational structure with its relations of production (management), and means of production (production devices), which are inseparable from the production process. These dimensions will be analyzed in four sections. While 3.1 provides the highlights of each section, 3.2 analyzes the pyramid organizational structure, which is the base for the development of the industrial based modes of production infrastructures. Section 3.3 focuses on the analysis of the computer engineering organization's management infrastructure (the industrial hierarchy based management) mechanisms. This section's analysis responds to the dissertation research question of how computer engineering organizations utilized the industrial hierarchy based management mechanism to control production process. Section 3.4 focuses on the development of computer engineering organizations' production infrastructure, which consists of three elements: production process framework, computer systems as production devices and specialized *Division of Labor* forces. Within this section, the dissertation responds to its core research questions in a logical order: First, how did computer engineering organizations create a production process framework? Second, how did the growth of computer engineering organizations create a demand for the emergence of advanced computer systems as production devices? Third, how did computer engineering organizations use production devices and a production process framework to create specialized *Division of Labor* forces?

Section 3.5 analyzes how computer engineering organizations used combinations of production infrastructure (production process framework, computer devices as productions devices and specialized *Division of Labor* forces) and management infrastructure (the industrial based hierarchy management mechanisms) to improve productivity (speed of production) and increase efficiency (low labor wages/mass production), all of which were equally mechanisms of extracting free labor hours from the production process during the *Transitional Phase* (1970-1995).

3.1 Overview of Organizations' Relations of Productions Infrastructure

This section analyzes two computer engineering modes of production: the Pyramid organizational structure and the hierarchy management infrastructure, which were practiced during the *Transitional Phase* (1970-1995). During this period, computer engineering organizations utilized the Pyramid organizational structure, which was similarly used by other industries at this time. The Pyramid organizational structure set the foundation for the development of the hierarchy management framework that was used both for the management infrastructure and to control the production development process (Thomas Allen, *Information Technology and the Corporation of the 1990s*, 1994). The structure was based on a top-down framework. From a management infrastructure perspective, the hierarchy framework enabled organizations to implement a top-down and micro-controlling management structure during the *Transitional Phase* (1970-1995). As part of the industrial modes of production infrastructure, the Pyramid organizational structure worked hand-in-hand with the hierarchy management infrastructure effectively. The Pyramid organizational structure was top-down driven, and the fact that the hierarchy management practice was a top-down based framework made for the cohesiveness between the two that made the organizational operations functional (Allen, 1994). The Pyramid organizational structure was established as an industrial organizational system (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). Its top-down structure is recognizable when put into a diagrammatic form. Below, diagram 5 depicts the pyramid organizational structure:

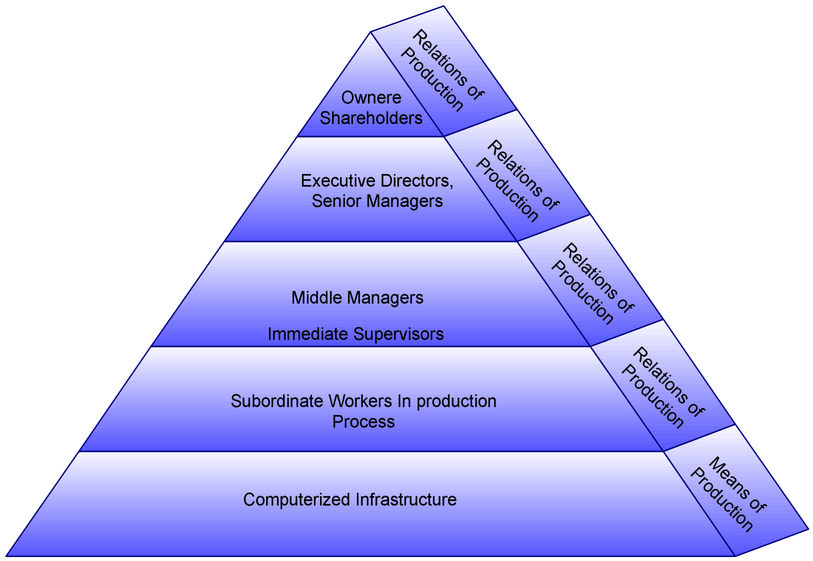


Figure 5: The Pyramid Organizational Structure

The diagram above depicts that each department sits below the top in descending order of subordination. Within this structure, management ensures that different groups and individual workers are structured as subordinate groups at the bottom of the pyramid (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). On the other hand, the partners, executive managers and/or departments in charge sit at the top of the pyramid (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). Thus, in this fashion, the pyramid structure facilitates the command and control framework for the management infrastructure to handle communication channels and to exercise commanding and micro-controlling power over the rest of the workers being organized at the bottom of the pyramid structure. (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). During the industrial period in which the *Transitional Phase* took place, the pyramid organizational structure was the standard infrastructure that was implemented in most of the industrial organizations (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). The industrial modes of productions infrastructure was confined within the same facility, whereby means of productions (e.g., computerized machines and networks) were stand-alone, which meant they only supported production processes limited to one facility, because computerized networks until the mid-90's were not capable of facilitating production services across multiple facilities (Schermerhorn, Hunt and Osborn, 2008b). This type of organizational structure set the foundation of micro management principles and the production process controlled by on-site supervisors because the entire operation was confined within the same facility. Hence, the Pyramid organizational structure collaboratively worked in conjunction with the available mainframe computerized based means of production during the *Transitional Phase*. This structure enabled industrial organizations, including computer engineering organizations, to have complete control over the operations, which included administration (management) activities, production process, and means of production, further controlling the communications path across the different hierarchies (Allen, 1994). The organization initiatives, directives and standards involving the organization as a whole were provided by the executive directors from top downwards to the middle managers to establish the operational processes and procedures for each department (Allen, 1994). Thereafter, in the case of computer engineering organizations, the operational processes and procedures were established by the senior and middle managers, who were sent specifically to the superior of each management department and to the technical supervisors of the actual production line process (Schermerhorn, Hunt and Osborn, 1997a). In this manner, organizations' owners and shareholders used the executive directors, managers and immediate supervisors to carry out the goals of the organizations' owners (Schermerhorn, Hunt and Osborn, 1997a). From a production line process perspective, the immediate supervisors were held responsible for implementing the operational processes and procedures that were formulated using the top-down project management methodology to manage the technical workers in production process (Allen, 1994).

As introduced above, the hierarchy management infrastructure was top-down based with an emphasis of micro-managing the subordinate workforce in production process (Schermerhorn, Hunt and Osborn, 1997a). In order to have a comprehensive understanding of the industrial top-down project management practice and method that was used during the *Transitional Phase* (1970-1995), an in-depth analysis is critical. Historically, both the Pyramid organizational and the hierarchy management structures were already in practice since the industrial revolution until the end of the 20th century. The hierarchy management structure became advanced due to the incorporations of scientific management methods and advanced managerial accounting principles (Jaince M. Roehl-Anderson, *Controllership*, 2004). The scientific management methodology introduced the techniques of applying measureable project schedules using work breakdown schedules (WBS) to improve efficiency and manage human resources to best utilize workers' hours during production process (Roehl-Anderson, 2004). These scientific management concepts were formulated by Henri Gantt in the early twentieth century and were later implemented as a scientific management methodology in the 50's to enhance management practices in the fields of engineering and manufacturing during the industrial era (Roehl-Anderson, 2004). In Henri Gantt's scientific management framework, known as the Gantt-Chart, project management methodology goes hand-in-hand with the traditional pyramid organizational structure and the hierarchy model, which is used both for the management activities and for the production life cycle process. The purpose of the Gantt-Chart approach was to specify the tasks that were required to complete each deliverable product in chronological order according to the life cycle of a product's development; the production process followed an exact sequential process from start to end in a way similar to the managerial activities (Roehl-Anderson, 2004). While figure 7 illustrates the Gantt-Chart inputs and outputs in sequential order, table 8 shows the WBS methodology as a master for controlling the resources allocated by the schedule of each production process phase.

Table 2: WBS managerial methodology

Production Process Phases	Description	Start Date	End Date	Actual End Date	Personnel
Phase #1	Staff Coordination	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx
Phase #2	Business Requirements Development	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx
Phase #2.1	Product Analysis & Design	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx
Phase #2.3	Product Programming & Testing	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx
Phase #3	Product Deployment	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx
Phase #4	Product Maintenance	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx	xx/xx/xxxx

Source: Roehl-Anderson, *Controllership*, 2004

The Gantt-Chart model, in conjunction with the work breakdown structure (WBS), was used by the *Waterfall* model (applied both for the management and the production process, including the managerial accounting principle) for management practice and for production process to improve efficiency (Roehl-Anderson, 2004). As noted above, the Gantt-Chart and the WBS were used to specify the tasks involved in the hierarchy management activities, and in the production process deliverables, to measure production time and control cost—and to use them as historical data to project future estimates (Roehl-Anderson, 2004). From a management perspective, the hierarchy model for management activities in the engineering and manufacturing fields had 5 components forming the modern project management infrastructure as following in hierarchical order: 1) project initiation and planning stages of the project management tasks, 2) product development tracking, monitoring and controlling stages of the production process, 3) measurement and analysis, 4) taking corrective actions, and 5) project completion stages of the project management process (Roehl-Anderson, 2004). As can be seen in the hierarchy model for the management above, managerial accounting fits right in stage 1 and 4 of the project management model. Thus, whereas the scientific management model (Gantt-Charting & WBS) modernized the ways in which projects' operations were managed and controlled and tasks were specified for each phase of the product life cycle, the managerial accounting principle facilitated methods of how the tasks assigned to the specialized workers were measured in the production process (Roehl-Anderson, 2004). Once the WBS with Gantt-Chart model was fully developed by middle managers and approved by the senior executive directors, it contained the following information that was assigned to the production process's immediate supervisor to be implemented in the production process: the WBS contains the specific tasks, human resources assigned to the tasks, labor cost and schedule time against the list of tasks defined (Roehl-Anderson, 2004). In this complex methodology, workers' productivity was measured depending on how fast they completed their assigned tasks, so that cost is saved, and an allocated budget was rolled over to another project to fund labor wages for the same type of task and/or distributed to other tasks as needed (Roehl-Anderson, 2004). As discussed in chapter 2, this very concept of increasing productivity by completing tasks as quickly as possible to save labor cost was initially put forth by Adam Smith in 1776 (later negated by Marx in 1867) and was reexamined by Burawoy in 1978; the complexity of its methodology and practice in production process was not comprehensible to workers. Burawoy confirmed that this practice reinforced the practice of exchange (surplus) labor in production process and made its manifestation inseparable from that of use labor (necessity labor) (Burawoy, *Manufacturing Consent*, 1979). According to Burawoy, the average worker did not differentiate the hours spent for exchange values from those for use value (Burawoy, 1979). Workers simply perceived that it was just a process of improving productivity (Burawoy, 1979). Below, figure 8 depicts the hierarchy industrial project management, which was built to work hand-in-hand with the Pyramid organizational structure across engineering and manufacturing organizations.

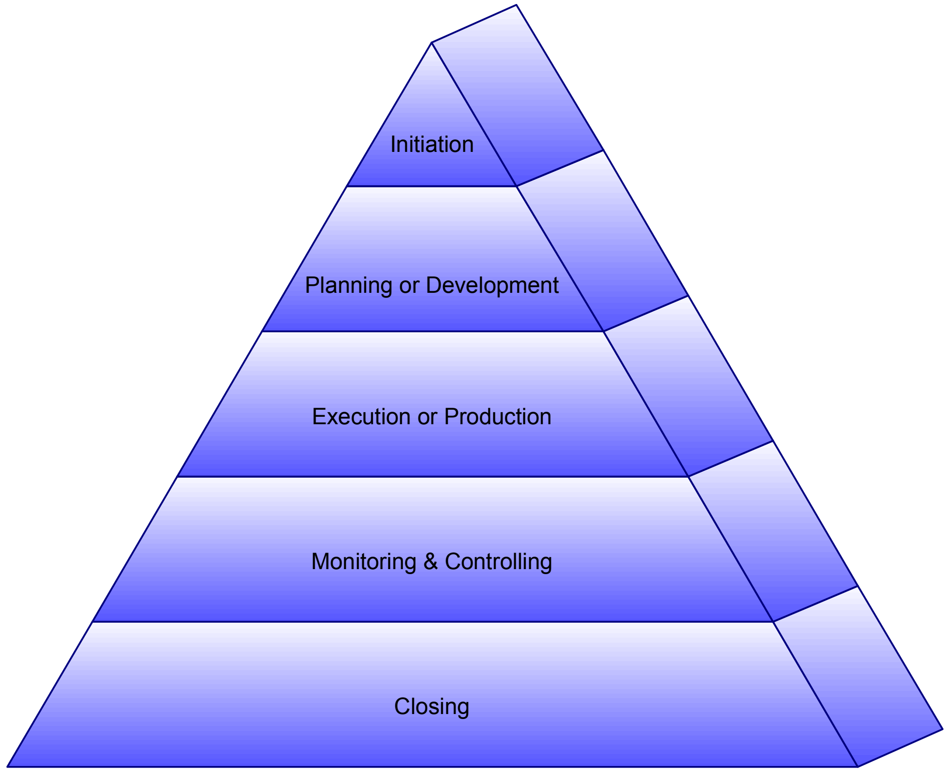


Figure 6: The *Hierarchy Management* Framework
Source: *History of Project Management*, Mark Kozak-Holland, 2010

This project management model used managerial accounting principles (process and procedures) to control the cost of labor and overhead for projects (Roehl-Anderson, 2004).

The hierarchy project management model adapted managerial accounting techniques of cost controlling by calculating hours spent on production against the WBS (Roehl-Anderson, 2004). As discussed above, scientific management combined the WBS and the Gantt-Chart, which then enhanced the process of defining tasks for each phase of the production life cycle, and the incorporation of managerial accounting techniques set the basis for the development of bases of estimates (BOE) methodology. The purpose of the BOE was to enable organizations to practice cost estimates against the WBS's labor efforts projections (Roehl-Anderson, 2004). With this advancement, organizations particularly in the engineering and manufacturing fields applied managerial accounting principles to control production cost defined by WBS in accordance with the total budget of the project. Figure 9 illustrates the WBS planning specification, which includes cost controlling methodology: 1) compare budget to actual hours worked, 2) compare budget to actual project start dates, 3) compare budget to the actual project completion dates, and 4) compare budget to actual staffing requirements (Roehl-Anderson, 2004).

Table 3: WBS with Managerial Accounting's cost controlling methodology for all projects in progress:

Project	Hours	Start Date	End Date	Personnel
1	240	xx/xx/xxxx	xx/xx/xxxx	xxxxxxx
2	280	xx/xx/xxxx	xx/xx/xxxx	xxxxxxx
3	240	xx/xx/xxxx	xx/xx/xxxx	xxxxxxx
4	240	xx/xx/xxxx	xx/xx/xxxx	xxxxxxx
5	280	xx/xx/xxxx	xx/xx/xxxx	xxxxxxx

Source: Roehl-Anderson, *Controllership*, 2004

Similar to the hierarchy project management's project tracking methodology that tracks the progress of the production process, the managerial accounting principle has its own labor planning and controlling mechanism that works hand-in-hand with the hierarchy project management methodology. As discussed by Roehl-Anderson (2004) in *Controllership*, every engineering and/or manufacturing organization's financial controller requires middles managers and project managers to develop WBS based upon direct and indirect labor categories. This method is further applied to specify the tasks involved in every project phase that had to be pre-defined, and workers have to carry out the tasks. According to

Roehl-Anderson, direct labor is defined as tasks in the WBS, which are conducted as work inputs in the production process, whereas tasks like administrative supports are categorized as indirect labor (Roehl-Anderson, 2004). Roehl-Anderson elaborates:

> Direct labor is only that labor that adds value to the product or service. However, there are many activities in the manufacturing or service areas, not all of which add value to the final product, as one must be careful to segregate costs into the direct labor and indirect labor categories. Direct labor is typically incurred during the fabrication, processing, assembly, or packaging of product or service. Alternatively, any labor incurred to maintain or supervise the production or service facility is categorized as indirect labor. Roehl-Anderson, 416-417, 2004.

This managerial accounting methodology (e.g., direct and indirect labor categories) was also used to compute bases of estimates for the engineering and manufacturing organizations. Using the direct labor category, the middle managers together with the most experienced technical manager (including the financial controller) develop the bases of estimates (BOE) for the entire production that needs to be produced (Roehl-Anderson, 2004). This activity was initiated at the beginning of each project initiation phase while finalizing the WBS and the Gantt-Charts. Roehl-Anderson further elaborates this process of applying direct labor category to perform BOE:

> This involves having a qualified industrial engineer team with a controller to conduct exact measurements of how costs relate to specific measurements. For example, this approach may use time-and-motion studies to determine the exact amount of direct labor that is required to produce one unit of finished goods. Roehl-Anderson, 436, 2004

Similarly, the indirect labor category was used to quantify the BOE, calling for a budget for the entire project (Roehl-Anderson, 2004). As introduced above, the indirect labor category is used to define and control production support related tasks (e.g., administration, purchasing and vendor processes). The indirect labor types at times blend with the direct labor tasks production in production process (Roehl-Anderson, 2004). Thus, during the BOE development, expertise of engineering and manufacturing fields' management life cycle is required to identify indirect labor types. This aspect of the methodology was described by Roehl-Anderson:

> Sometimes the correlation may not be as close as desired and a more analytical technique may be necessary, which involves the aid of industrial or process engineers. The method, which closely resembles the calculation of the required direct labor for any given manufacturing operation, essentially is: The engineers study the specific function to be performed by the departmental indirect labor crew, including the exact labor hours required at differing activity levels. An activity base is selected, such as standard machine hours that would be a fair and easily determinable measure of just what labor hours are needed for each function of the indirect labor crew. Roehl-Anderson, 451, 2004

This approach was also used to train workers to become specialized in a given field and to enhance labor force recruiting requirements. A worker was trained to become an expert in a specific field that was defined according to the direct or indirect labor categories. For example, in computer engineering and computer hardware manufacturing organizations, an engineer can be trained to become a blue print designer, developer or assembler in the production life cycle process (Thomas Hempell, *Computers and productivity*: *How Firms Make a General Purpose Technology Work,* 2006). As originally theorized by Adam Smith in 1776, and later by Michael Burawoy in 1979, the motive behind making workers specialized in a specific field was to make them proficient so that the production process improves and workers can speed up production time by increasing efforts—in the end, production increases with low labor cost. To figure out the level of efficiency poured into the production process, the organization middle managers and the financial controllers track progress in terms of output by reviewing the actual cost associated with the specific tasks performed (Roehl-Anderson, 2004). The standard output of a given task by a worker is computed as paid labor effort, and the productivity of that laborer performing the same tasks without labor costs is categorized as part of the productivity efforts for the organization with recognition given to the worker (Roehl-Anderson, 2004). Furthermore, because of this task specification, workers specialization and motivation methodology were used to develop labor recruiting directives for organizations.

In the process of recruiting workers, organizations required long term employment history that became part of the criteria of recruiting (Roehl-Anderson, 2004). Organizations required that every candidate have long term records with a previous company, because production process requires long term experience and cost to retain workers (Roehl-Anderson, 2004). This practice was implemented to reduce levels of turnover in terms of losing specialized laborers in the production process because organizations invest in specialization trainings to increase efficiency in production process (Roehl-Anderson, 2004). Roehl-Anderson (2004) points out the disadvantages of continuous turnover, particularly for organizations from an increased cost perspective:

> There are the added and clearly defined costs of training new employees and correcting the inevitable mistakes that they will make as they learn their jobs. In addition, the controller must pay for replacement staff to fill in during the period when the old employee has already left the company and the new one has not yet started. (Roehl-Anderson, 164, 2004)

Accordingly, these concerns were taken into major consideration when recruiting workers setout to search for specialized workers in support of production process efforts (Roehl-Anderson, 2004). From an increased inefficiency perspective, an organization's financial controllers' concern was the loss of productivity due to new workers' lack of expertise in production process (Roehl-Anderson, 2004). According to Roehl-Anderson (2004), unless vastly experienced workers are hired, organizations would not reach the expected efficiency levels from a new worker. This is further elaborated by Roehl-Anderson, "Inefficiency will appear in several ways: the extra time required to complete tasks, the extra temporary help needed to support the person, and the time of other staff people needed for training," (Roehl-Anderson, 129, 2004). Thus, avoiding a high turnover is crucial in keeping specialized workers in production process and maintaining steady productivity level. Similarly, checking employment history in terms of longevity became one of the major criteria of hiring (Roehl-Anderson, 2004). Once the qualified worker was hired, motivation methods were applied to encourage the worker to increase productivity, which enables him/her to produce more. This result was measured based on the amount of production accomplished with no labor cost to the organization (Roehl-Anderson, 2004). In addition to the longevity criteria, organizations amended long, midrange and short range motivational standards with procedures to enable them to prolong longevity of workers in production process (Roehl-Anderson, 2004).

The purpose of amending the long, midrange and short range motivational standards with procedures was to motivate workers for the following reasons: 1) to maintain specialized workers for a long time in the same organization, 2) to increase productivity by improving the pace of producing products because the longer a worker stays in the same production process, the more he/she becomes proficient in the production process, 3) to enhance the practice of productivity, so that workers could double production levels and produce more than what they were paid to produce, which was an invisible practice of surplus labor. Once the workers started working, the project management team that was consulted by the financial controllers assessed workers' performance relative to the amount of revenue that needed to be earned for a given business quarter (Roehl-Anderson, 2004). Based on the result of the workers performance in production process, the immediate supervisor classified workers among the long, midrange and short range motivational categories (Roehl-Anderson, 2004). This motivation methodology was further assessed by Roehl-Anderson's *Controllership*:

> Motivating employees clearly improves morale as well as department performance, but motivation should be based on a multilayered motivation scheme that covers the long, medium, and short term. By using this multilayered approach, employees are presented with a richer environment in which to work, which is more satisfying for them, and gives them many good reasons for staying with the company. Roehl-Anderson, 171, 2004

Accordingly, production process supervisors instructed by financial controllers from the executive management level applied the long-term, mid-range and short-range motivational standards to promote specialization in specific tasks for workers to become experts (Roehl-Anderson, 2004). The multilayered motivational method was based on going to on-job trainings and organizational wide recognitions of productive workers in production process (Roehl-Anderson, 2004). Equally, this long term motivation was used as a mechanism to promote workers longevity in the organization (Roehl-Anderson, 2004). However, although organizations take a great advantage of motivation to improve productivity and worker longevity in the organization, the major drawback for the workers was that they became dependent on the organization and would not leave their post (Burawoy, 1979). Workers tended to convince themselves that organization was doing them a favor by providing them with the long term motivational practices to be productive (Burawoy, 1979). Workers never thought that it was their surplus labor in production process that kept them within the organization and guaranteed the success of the organization (Burawoy, 1979). As a result, this dissertation argues that during the hierarchy management era the motivation method was devised by the financial controllers to improve productivity by encouraging workers in production process to double the levels of productivity in a given hour—and to promote workers longevity in the production line is profitable to an organization. However, during the hierarchy management, both the motivation method and productivity were part of the mechanism to delude workers into not realizing the distinction between necessity labor (paid labor) and exchange (surplus/unpaid) labor.

In sum, this section primarily focused on the analysis of the development of the industrial based relations of production infrastructure (pyramid organizational structure and the hierarchy management structure). As detailed in this section, both the pyramid organizational structure and the hierarchy project management structure were top-down based, in which they complimented each other during the industrial era, before the emergence of the informational era towards the end of the 20's century. Organizations applied the hierarchy project management methodology, adapting scientific management methods (Gantt-Chart, WBS and BOE) to implement specialization based division of labor specification in production process to improve productivity. Organizations further reinforced their hierarchy management practice by amending financial controlling principles, such as direct & indirect labor categories, human resource's employee longevity recruiting criteria, and the multilayered motivational practices to reinforce the surplus labor practices in production process. In this way, the pyramid organizational structure with its modes of production infrastructure (hierarchy management, stand-alone production devices and facilities with sequential driven production process) set the foundation for the development of the computer engineering field's specific modes of production infrastructure, which started formulating its management infrastructures beginning in the early 1970s (Thomas Hempell, *Computers and productivity*: *How Firms Make a General Purpose Technology Work*, 2006). The next section analyzes the development of computer engineering organizations' means of production infrastructure (production framework, *Division of Labor* force & production devices) during the *Transitional Phase* (1970-1995).

3.2 Production Process Framework 1970-1995

As analyzed in section 3.2 in this chapter, the Pyramid organizational structure set the foundation for the development of the industrial modes of production infrastructure. The pyramid organizational structure was a standard organizational structure for industrial organizations until about the end of the 20^{th} century. During that period, industrial organizations adapted the industrial modes of production infrastructure (hierarchy management and the stand-alone production devices as means of production), which the Pyramid organizational structure helped to develop. In the case of the computer engineering field, since the field itself was new to the industrial era, experts in the field were searching to define specific modes of production infrastructure. Another unknown challenge was defining the tasks involved in the management and production process life cycles. Furthermore, since the specific tasks in the computer engineering project management and the production process were not defined, these issues became obstacles to implementing task specialization based *Division of Labor* practice in the production process. The newly formed computer engineering organizations were not even near bothering with improving productivity and calculation of return on investment; they were still in search for definitions of tasks from which to develop specialization based on *Division of Labor* schemas to determine if the computer engineering venture was a long lasting and profitable field (Allen, *Information Technology and the Corporation of the 1990s,* 1994). Therefore, this section, first analyzes the historical development of the management structure of computer engineering organizations, followed by the analysis of the computer devices as a means of production to demonstrate their role in *Division of Labor* and surplus labor values in production process.

At around 1970, which this dissertation claims to be the start of the *Transitional Phase* period, the computer engineering field was in search for its own specific framework of industrial modes of production infrastructures (Theo Eicher, *Information technology and productivity growth*, 2009). According to historical development of computer engineering field development, up until about the early 1970's, the field did not have its own relations of production infrastructure (management) with standards and procedures to command production processes (Eicher, 2009). As noted in section 1, for the computer engineering field, production process and management tasks were not specified to create a specialized *Division of Labor* forces and measure productivity outputs, or provide training to improve production processes (Eicher, 2009). For example, according to Department of Labor (DOL) data on computer engineering positions during the late 1960's and early 1970's, there were about 3 known position titles. Workers were performing their tasks within the same facility confined with the management personnel including the stand-alone mainframe-computers, which were used as means of production to produce software applications (Eicher, 2009). With this type of environment setting with no specified tasks, computer engineering organizations were not able to form an effective *Division of Labor* force to improve productivity. As businesses increased the demand for computer usage grew, and computer engineering organizations realized the inevitable emergence of desktop computers. Consequently, computer engineering organizations set out to devise their own framework for project management. In addition, they created production development processes to specify tasks to implement a specialization based *Division of Labor* and surplus values to succeed to guarantee their profit gains. Hence, by 1970, developing a framework enabled computer engineering organizations to accomplish their goals, which became inevitable (Eicher, 2009). At this junction, computer engineering organizations relied on Winston W. Royce, a scholar and a practitioner computer scientist to devise the framework.

Winston W. Royce's answer to the problem was to devise an integrated framework that synchronizes project management and production process for the computer engineering field (Eicher, 2009). Royce's initial task was assessing the organizational environment, which encompasses organizational structure, management, production process and hardware/software systems (Eicher, 2009). During the process of the assessment, Royce learned that the foundation of the industrial system was top-down and thus could only accommodate a top-down based framework (Eicher, 2009). The reason is that from an organizational structure perspective, the industrial system had a pyramid organizational structure. As described in section 1 of this chapter, the pyramid organizational structure operates in a form of superiority whereby the top layer controls the subordinates beneath in a controlling fashion. Similarly, the management structure was also hierarchy based, built within the pyramid organizational structure—managing a sequential based production process that was hierarchy driven (Eicher, 2009). Further, the production process facility and the stand-alone computer hardware/software devices used as the means of production were reasoning factors that set the foundation for the formation of Royce integrated framework called *Waterfall* in 1970 (Eicher, 2009).

Royce's *Waterfall* framework used the industrial system's modes of production infrastructure (e.g., the Pyramid organizational structure, the hierarchy management, the facility and the stand-alone production devises as means of production) as its foundation. While computer engineering organizational structure continued to be Pyramid based throughout the course of the 70's, 80's and the mid-90's, Royce's *Waterfall* framework introduced its own integrated framework for production development process that enabled computer engineering organizations to measure success, specify tasks in developing specialized *Division of Labor* and apply its own cost controllership in handling surplus values (Eicher, 2009). As noted in section 1 of this chapter, the need for the development of a new framework for the computer engineering field was due to the field's newness in the late 60's and the early 70's with no task specifications in which owners did not have any standardized mechanism that enabled them to maximize their profits from the production process and from the market (Eicher, 2009). This meant that there was a need to devise modes of production infrastructure (management, production development process and production devices as a means of production) for the computer engineering field. In 1970, Royce attempted to introduce an integrated mechanism called *Waterfall* framework based on a top-down foundation.

The integrated *Waterfall* framework had industrial characteristics in which the management, production process, and the production devices as means of production complemented one another. The relations of production (management) structures and the production process had the same infrastructural foundation. Royce's contribution was that he was able to define the tasks involved in the production process in the computer engineering field (Eicher, 2009). With this incorporation of the industrial hierarchy management infrastructure, which was devised by Henry Gantt (1916) in his *Work, Wages, and Profits,* the integrated *Waterfall* became the standard framework for the computer engineering field. For example, both of these frameworks were built with a top-down foundation, which was based upon an industrial system (Eicher, 2009). Also, they commonly shared the pyramid organizational structure as a foundation and operated within in it (Eicher, 2009). For the relations of production (management), the *Waterfall* framework was structured as a top-down micro based practice, which had life cycle phases starting from initiation, planning, executing, tracking & controlling and closing with hierarchy management devised by Henry Gantt in 1916. Similarly, for the production process, the *Waterfall* structure had a top-down based practice, which synchronized with Gantt's hierarchy management structure. As the table below shows, the integrated *Waterfall* framework of the production process phases and the hierarchy management phases are near one-to-one identical in terms their task execution. These very intertwined and complemented modes of production infrastructures were reinforced by industry standard best practices in terms of the application of processes and procedures to reassure the success of the organization production process (Eicher, 2009). Hence, below, table 4 depicts the synchronization of the hierarchy management and the production process phases.

Table 4: Synchronization of hierarchy management and production process phases:

Management Infrastructure Phases	Management Infrastructure Phases	Production Process Phases
Phase #1	Project Initiation & Planning	Staff Coordination
Phase #2	Project Tracking & controlled	Business Requirements Development
Phase #2.1	Project Tracking & controlled	Product Analysis & Design
Phase #2.2	Project Tracking & controlled	Product Programming & Testing
Phase #3	Project Transitioning	Product Deployment
Phase #4	Project Closing	Product Maintenance

Source: Roehl-Anderson *Controllership*, 2004

In this manner, both modes of production infrastructures were the vital mechanisms as a pair to ensure the success of computer engineering organization in general during the *Transitional Phase* (1970-1995). Computer engineering organizations adapted the industrial modes of production (pyramid organizational structure and the hierarchy management structured) as a base foundation. As analyzed in this section, the integrated *Waterfall* framework borrowed WBS, Gantt-Chart and BOE management dimensions from the industrial hierarchy management structure to develop the computer engineering organization management infrastructure to manage production process (Roehl-Anderson

Controllership, 2004). The next section analyzes how the industrial hierarchy management components (WBS, Gantt-Chart and BOE) were integrated with the *Waterfall* framework, and also analyzes how the tasks involved within each production process phase were defined. Further analysis in the next section includes how the integrated *Waterfall* framework specified production process phases and tasks to established specialized based *Division of Labor* and the manifestation of surplus labor in production process in computer engineering organizations.

3.3 Production Infrastructure: Production Process & Division of Labor

This section analyzes the relationship among the industrial hierarchy management, the development of specialized *Division of Labor* (roles & tasks) and the production process phases for computer engineering organizations' production process. First, the management aspect of Royce's integrated *Waterfall* framework took the WBS to specify computer engineering management phases with tasks and to define the required specialization driven *Division of Labor* in the production process (Eicher, 2009). Although the WBS has its own templates and logical procedures to follow, Royce's major challenge was in defining the specific phases involved within the production process for the computer engineering field (Eicher, 2009). With this in mind, Royce defined 5 production process phases (Eicher, 2009). The diagram below, illustrates the top-down, *Waterfall* production process phases:

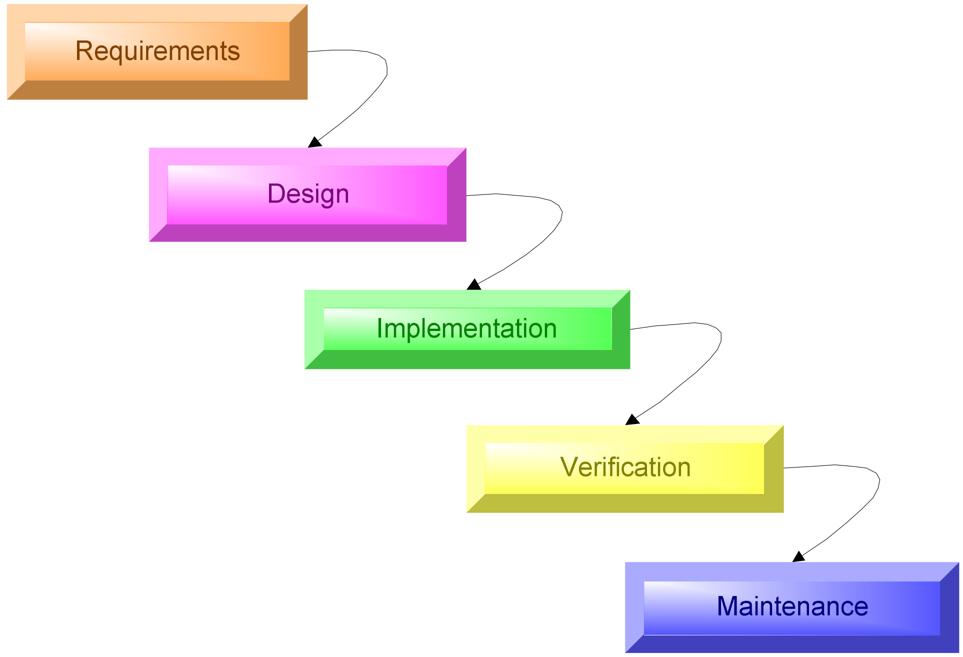


Figure 7: The *Waterfall* Framework. Source: Winston W. Joyce, 1970

Establishing a mirrored synchronization between the industrial management structure and the production process phases was part of Royce's efforts.

The next challenge for Royce was to define the detailed production process tasks involved within each phase of the production process, so that Gantt's industrial WBS could be fully utilized. The primary reason why the detailed tasks' description for each phase was needed was to enable computer engineering organizations to maximize their profits through specialization of the *Division of Labor*, improving productivity to double the speed of production efforts, which led to the equivalent of surplus labor gain. This means workers use value (necessity) payment remains constant per hour even though they are doubling the production speed per hour. Workers would not even comprehend the distinction between their surplus value and use value because the manifestation of this practice is invisible to workers as both values are blended in the production process (Burawoy, 1979). Thus, according to the phase, Royce defined specific tasks, which were involved within each phase. The following table illustrates the types of specified tasks for each production process phase:

Table 5: Production Process Phases:

Production Process Phases	Tasks Specification
Systems Requirement	• Requirements elicitation • Requirements analysis and negotiation • Requirements specification • System modeling • Requirements validation
Systems Analysis & Design	• Logical Design • Physical Design • Structure Charts • Software Specification
Programming	• Programming Specification • Coding
Testing & Verification	• Manual Testing • Automated Testing
Deployment	• Hardware Infrastructure Development & Implementation • End Users Training • Software Release
Maintenance	• Tech-Support • System Change Request

Source: Winston Royce, *Waterfall*, 1970

DOL develops job category descriptions based on what is practiced in the computer systems engineering organizations. As illustrated in sections 3.4 of and 4.4, job category definitions and position titles have been redefined as a result of the development of the modes of production in general and more specifically the advancement of the production infrastructure (production framework, production process, types of *Division of Labor* forces and production devices). For example, during the individual based specialized production process in the 70's, 80's and mid-90's, specialized roles (*Programmers, Systems Programmers* and *Programmer Analysts*) were used; however, as a result of the emergence of automation, advanced network systems and internet communications of the multifunctional base production process during the late 90's and 2000's, the job categories' descriptions and position titles were changed to *Applications Programmers, Systems Software Engineers and Applications Software Engineers* (DOL, 2004-05). The changes in categories and names of positions occur as a result of the development of the modes of production, but connecting position categories across time can be done by comparing production process tasks/activities. In essence, the continuation of a position is determinable by its main function. For example, while *Programmers* have taken on new tasks over time because of positions consolidation (as a result of the emergence of automated production devices), their core responsibility has always remained programming. Below, DOL describes how job titles and categories change:

> Many technical innovations in programming-advanced computing technologies and sophisticated new languages and programming tools-have redefined the role of programmer and elevated much of the programming work done today. Job titles and descriptions may vary, depending on the organization. In this occupational statement, computer programmer refers to individuals whose main job function is programming; this group has a wide range of responsibilities and educational backgrounds. (Occupational Outlook Handbook, 96, 2004-05)

Hence, as the components of the production infrastructure change, the job categories and positions' titles are transformed.

As shown in table 5, the integrated *Waterfall* framework defined the tasks involved in the production process; however, there was scarcity in the specialized labor force for the computer engineering field between the late 60's and the early 70's. The table below illustrates Department of Labor's (DOL) data on the specialized positions with their roles and responsibilities in the computer engineering field:

Table 6: Specialized *Division of Labor* force 1968-69:

Types of Specialty	Roles & Responsibilities in Production Process
Computer Systems Analyst	Analyze business processes and design workflows with coding specification.
Computer Programmers	Validate workflows process with coding specification and perform programming codes and develop user manual instructions.
Computer Data Processors	Produce retrievable data by converting codes to magnetic tapes and deliver to librarian to do daily backups of data.

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1968-69a

The data indicates that during the late 60's and early 70's, there were indeed limitations in specification of tasks for computer engineering production process as the integrated *Waterfall* framework was being implemented. Furthermore, there was specialized labor force scarcity in the field of computer engineering to fulfill the specialized tasks indoctrinated by the integrated *Waterfall* framework to carry out production process. According to DOL records published in *Occupational Outlook Handbook* of 1970 through 1974, computer engineering organizations began to implement in-house training methods to produce a specialized based *Division of Labor* force for each phase of the production process (Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1970-71a). Furthermore, in addition to the tasks' process and procedures that workers were trained to perform, workers were given specialized training on the computer devices that they used to perform the actual production process (*Occupational Outlook Handbook*, 1970-71a). This concept of training workers in specialized tasks and production devices was in an effort to develop a specialized based *Division of Labor* in production process. The specific tasks defined for the production process for each phase was then used to train workers to become experts in their specialized task assigned to them. This approach has its foundation in Adam Smith's 1776 *Division of Labor*, where he initially proposed his directives on job training in terms of production processes and procedures and production device training to reinforce the application of the *Division of Labor*. This was done by increasing workers' expertise to improve productivity. As this on-job training dilemma was in progress, academic institutions began to provide curriculums based on Royce production phase specification introduced in the integrated *Waterfall* framework. According to DOL assessment and data from the early to mid-70s, computer engineering organizations had reported positive increases in the availability of a specialized labor force as a result of the on-job training. Equally, by that time, the academic sector had formed accredited two-year associate degrees and four-year bachelor degrees for students to produce a future labor force (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1970-71a). Although the specialized labor force increase was not for every production process phase defined in the *Waterfall* framework, by interpreting DOL historical data trends, an informative analysis can be drawn about the computer engineering production process's tasks development and the labor force during the 70's, 80's and the mid-90's. The table below presents DOL's historical data on the assessment of the computer engineering field, which can be interpreted to analyze the development of production process phases' tasks specification throughout the course of the early to mid-70's through the early to mid-90's.

Table 7: Specialized *Division of Labor* force 1970-71 & 1972-73:

#	Specialization 1970-71 & 1972-73	Description 1970-71 & 1972-73
1	**100,000 Computer Systems Analyst**	
1.1	**Computer Systems Analyst**	Responsible to analyze complex "As-is" business processes using workflows, design "To-be" business process and develop coding instructions for computer programmer.
2	**200,000 Computer Programmers**	
2.1	**Computer Programmers**	Responsible to translate coding instruction to programming languages to instruct the computer machine to process data.
Total	**2 specializations were implemented**	**300,000 were employed**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1970-71 & 1972-73a

The interpretation of the data in table 7 confirms that the computer engineering organizations did not have a production process framework during the late 60's throughout the early 70's. Further confirming is the need for a computer engineering framework as demonstrated by Royce's *Waterfall*. Although Royce introduced the integrated Waterfall framework, it was in the process of being implemented in computer engineering organizations' production process. The DOL data trends assert that there were only three production process phases' tasks known to the computer engineering field. This scenario suggests that there was indeed a specialized labor force shortage in the computer engineering field as the computer engineering field was newly indoctrinated into the industrial modes of production. According to a DOL assessment performed during 1970-71, computer engineering organizations computer analysts and programmers were employed to develop computer systems to support a variety of industries. This suggests that the 2 tasks were used as production process to develop computer systems, which reconfirms that the computer engineering field was premature during the late 60's and early 70's. As this was in progress, according to DOL's assessment of the computer engineering organizations' *Division of Labor* development in 1970-71, the demand of acquiring specialized computer engineering labor forces increased as computer systems became more in demand to run daily business operations (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1970-71a). Although DOL's data in 1970-71 and 1972-73 indicated 100,000 computer analysts and 200,000 programmers hired by computer engineering organizations, the increase was only by 50,000 from what was reported by DOL data in 1968-69, which indicated only 50,000 computer analysts and 100,000 programmers were hired by computer engineering organizations in the United States (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1970-71a). This suggests a need for promotions of specialized *Division of Labor* in the computer engineering field. It is worthwhile noting that the efforts for the promotion of the specialized *Division of Labor* can only be accomplished by aggressive on-job trainings and the involvement of academia; the massive training depends on the innovation of an effective computer engineering production process framework like the integrated *Waterfall* framework introduced by Royce, which was just in its state of being introduced.

This confirms that the integrated *Waterfall* framework production process phases and specialization of tasks involved within the production process phases were crucial dimensions. Thus, at that point in time, the development of specialized *Division of Labor* depended on the successful indoctrination of the computer engineering field's production process phases with their specific tasks into the industrial modes of production infrastructure. Furthermore, the Department of Labor reported, "There is no universally acceptable way of preparing for work in systems analysis. Some employers prefer that candidates have a bachelor's degree and experience in mathematics, science and engineering, (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 258, 1970-71a). In addition to the demand of computer engineering specialized labor forces for production process, implementing the integrated *Waterfall* framework that enables computer engineering organizations to maximize their profits while in production process became one of the major challenges during the early to mid-70s (Eicher, 2009). What this boiled down to was the shortage of the specialized *Division of Labor* to fully implement the integrated *Waterfall* framework to carry out the complete tasks defined for each production process phase. Obviously, it was unknown and immeasurable to plan out how long it would have taken to complete a computer system from analysis & design through coding and testing production phases (Thomas Hempell, *Computers and productivity*: *How Firms Make a General Purpose Technology Work*, 2006). As a result, computer engineering organizations were not able to take a great advantage of the principle of the specialized based industrial *Division of Labor* and the available industrial organizations accounting principles to extract surplus labor from the production process before the computer product is set for market. At this junction, the focus on the increase of the specialized *Division of Labor* and the implementation of the integrated *Waterfall* framework started to surface; tables 8 & 9 illustrate DOL's data from 1974-75 and 1976-77 regarding the implementation of some of the tasks introduced by the integrated *Waterfall* framework.

Table 8: Specialized *Division of Labor* force 1974-75:

#	Specialization 1974-75	Description 1974-75
1	**100,000 Systems Analysis**	
1.1	***Systems Analysts***	Produced business workflows, code specifications and designs for programmers
2	**186,000 Programming**	
2.1	***Programmers***	Produced computer systems
2.2	***Lead Programmers***	Produced final codes, provided on-job training and supervised mass programmers
3	**480,000 Electronic Computer Operating Personnel**	
3.1	***Console Operator***	Produced final computer user's and administrators manual, computer punch cards, magnates and tapes.
3.2	***Keypuncher Operator***	Produced computer readable symbols, words and numbers by using a computer devise to punch holes and cards that were input data inserted into the converter devise.
Total	**5 specialization were implemented**	**766,000 were employed**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1974-75a

Table 9: Specialized *Division of Labor* force 1976-77:

#	Specialization 1976-77	Description 1976-77
1	**115,000 *Systems Analysis***	
1.1	***Systems Analysts***	Role's description remained unchanged.
1.2	***Lead Computer Systems Analyst***	Produced final designs and provided training to systems analysts and supervised Analysis & Design production process
2	**200,000 *Programming***	
	Programmers	Role's description remained unchanged.
2.1	***Lead Programmers***	Produced final computer systems, validated codes, provided on-job training and supervised mass programmers in the production process.
3	**500,000 *Computer Operating Personnel***	
3.1	***Console Operator***	Role's description remained unchanged.
3.2	***Lead Console Operators***	Produced final users' and administrators manual, provided on-job training and supervised mass console operators in production process.
3.2	***Keypuncher Operator***	Role's description remained unchanged.

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1976-77a

Some degree of progress was made on the development of the specialized *Division of Labor* during the 1974-75 and 1976-77 per DOL's data on the assessment of the computer engineering field. In table 8, one can see that the 1974-75 DOL data showed more progress for the computer engineering organization specialized *Division of Labor* than it did during 1970-71 and 1972-73 in terms of amending a new production phase called the *Electronic Computer Operating Personnel* role within its roles and tasks. DOL had reported that this specialized production phase had about 480 specialized computer technicians. Although this *Electronic Operating Personnel* role title was replaced by *Computer Operating Personnel* in 1976-77, DOL data on table 9 indicates that the specialized technicians' size in this role had increased to 500,000, which was an increase of about 20,000. Similarly, DOL's data in 1976-77 in terms of numbers of specialized workers in *Systems Analysis* indicated an increase to 115,000, up from 100,000 in 1974-75. Also, the same data showed that the size of specialized workers in *Programming* production phases increased to 200,000 from 186,000 in 1974-75. Further progress was made by implementing 2 of the tasks defined by the integrated *Waterfall* framework (Lead Computer Systems Analyst and Lead Programmer) and created roles with tasks called "*Console Operator*" and "*Keypuncher Operator*" under a category of computer related occupations (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1974-75a). These specialized roles were incorporated into the production process to micro-supervise analysts and programmers daily production output and ensure that tasks assigned to the individual analyst and programmer were completed in a timely manner. As a result, the left over time from the completed task is saved to produce other similar tasks from different projects to increase efficiency, in which the individual programmer would not earn use value in this scenario. The direct interpretation of such a scenario is that while it translates to profit gains for the computer organization as an exchange value in production process, it is a form of exchange value for the individual programmer. Hence, DOL's data of 1974-75 can be interpreted as an initial step towards the implementation of defined production process tasks to systematically gain surplus labor value. This scenario confirms that during this period, computer engineering organizations had begun amending production process tasks. Furthermore, the incorporation of console operators and keypuncher operators in 1974-75 and the lead console operator in 1976-77 shows that computer engineering organizations were indeed developing their own specialized *Division of Labor* to facilitate production process. As DOL reported during these periods, there was no accredited/formal university degree for computer engineering for computer analysis, programming or other computer related fields; computer engineering organizations managed to provide on-job training to generate specialized roles of console operators, lead console operators and keypunchers. The data can also be interpreted as a justification of Adam Smith's 18th century industrial *Division of Labor*, which was developed by the means of creating a specialized labor force through on-job training to double the speed of productivity as workers develop expertise. The belief was that the more a worker repeats the same process to produce the same task, the more proficient the worker becomes. This is determined by computing the amount of output achieved through surplus labor while the good is in production process. Applying the same principle in this case, the aim of the computer engineering organizations was to generate specialized console operators to handle the day-to-day transactional data processed by the legacy systems (pre-mainframe computer systems) to migrate to the new mainframe computer system that was introduced during the early 70's. Within the same context, computer engineering organizations used a mechanism of career transformation of existing employees. According to DOL's record, this career transformation took place because there was a shortage of specialized labor force in the computer engineering field as the field was new and there was no accredited/formal education during that period (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1974-75a and 1976-77a). For example, "Through on-job experience and additional study, some console operators qualify as programmers," (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 108, 1974-75a). Similar on-job career training was applied for typists, "some organizations will train good typists in the operation of keypunching equipment," (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 108, 1974-75a). This was one of the efforts made by the computer engineering organizations during the early phase of the computer engineering field. Sure enough, towards the end of the decade, DOL's data of 1978-79 can be interpreted as suggesting that there was more progress made in the creation of specialized *Division of Labor* force to support the computer engineering organizations' production process.

Table 10: Specialized *Division of Labor* force 1978-79:

#	Specialization 1978-79	Description 1978-79
1	**160,000*Systems Analysts***	
1.1	***Systems Analysts***	Role's description remained unchanged for the entire 70's.
1.2	***Lead Systems Analysts***	Role's description remained unchanged for the entire 70's.
2	**230,000 *Programming***	
2.1	***Programmers***	Role's description remained unchanged for the entire 70's.
2.2	***Application Programmers***	Specialized in industry specific computer systems development. Produced computer systems for specific industry.
2.3	***Systems Programmers***	Produced standard programming codes and drafts instructions for manuals.
2.4	***Supervisor Programmers***	Produced final computer systems, validated codes and supervised mass programmers in the production process.
	565,000 *Computer Operating Personnel*	
3	***Computer Key Punch Operators & Data Typists***	Role's description remained unchanged since 1974-75.
3.1	***Console Operators***	Role's description remained unchanged since 1974-75.
	Lead Console Operators	Role's description remained unchanged since 1976-77.
3.2	***High-Speed Printer Operators***	Produced outputs of transactional reports using centralized printing computers. Specialized in database backup devise operator.
3.3	***Converter Operators***	Produced computer readable data in computer cards and magnets. Specialized in transforming input data to computer readable cards and magnets to save transactional data.
3.4	***Tape Librarians***	Produced database backup reports using computer backup devices.
Total	**12 specializations were implemented**	**955,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a

As table 10 shows above, DOL reported 12 specialized computer engineering fields with roles and tasks within 3 production process phases (*Systems Analysis*, *Programming and Computer Operating Personnel*) in the production process. As table 9 shows, there were 4 more specialized fields with roles and tasks than the 1976-77 statistical record for computer engineering organization during 1978-79. The 1978-79 data indicates that there was a persistent increase in sizes of the specialized *Division of Labor* forces in computer engineering organizations in production process throughout the course of the 70s. For example, table 10 shows that the number of *Systems Analysts* increased to 160,000 from 115,000 in 1976-77, that *Programmers* increased to 230,000 from 200,000 in 1976-77 and *Computer Operators Personnel* increased to 565,000 from 500,000 in 1976-77. Thus, according to DOL data, the *Division of Labor* in production process has grown in specialization and expertise in the life cycle of the computer engineering production process. For example, the *Computer Operating Personnel* had five specializations. The data indicates that computer engineering organizations were making progress in developing the industrial specialized *Division of Labor*. Furthermore, it confirms the implementation of the integrated *Waterfall* framework production process phase's tasks specification, which helped to create on-job training programs to generate more specialized workers. As the demand increased for computer systems to support daily businesses operations, it came to the attention of the computer engineering organizations to establish a production process role called "*Application Programming*" to develop industry specific applications (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). As described in table 10, the *Application Programming* specific roles and tasks were distinct from the standard programmer's roles and tasks specification introduced by the integrated *Waterfall* framework. The *Application Programming* roles and tasks were to specialize in a specific industry in developing a software application for a specific industry like Financial, Human Resource, Chemical and Pharmaceutical and so on (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). There was a scarcity of specialized workers in the work force that remained persistence in the computer engineering field throughout the course of the 70's. Computer engineering organizations continued to use the on-job career training as an alternative to producing such industry specific *Division of Labor* to facilitate the need in the production process. On-job training was provided to standard programmers who were already producing computer systems, which made for an easy transition given their prior skill (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). At the time, a similar effort was made by the computer engineering organizations to generate a specialized labor force as trained high speed printing operators.

The high speed printing operator role became in demand by businesses that started utilizing centralized (auxiliary computer equipment) printing computers that were connected to local area mainframe computer systems (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). According to the DOL data captured in reference to this role and described in table 10, this specialized labor force was generated by on-job training that was provided to the *Keypunch Operators*. Thus, computer engineering organizations were responsible for creating new task definitions for high speed printer operators to support production process. For very similar reasons, converter operators and tape librarian roles were created to support production process. Workers that were trained to specialize as converters worked directly with output transformer computer systems to develop systems that provided reports to businesses (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). At this point, as computer systems had already become the main means of production devices in processing day-to-day business transactions, there became a requirement to secure and backup databases to maintain records of transaction outputs. As reported by DOL:

> Computer systems have become an increasingly important part of everyday life. Today these machines bill customers, pay employees, record airline and hotel reservations, and monitor factory production processes. Scientific and engineering research relies on computer systems to solve complex equations as well as to collect, store, and sort vast amounts of data. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 111, 1978-79a

With this in demand, the more the usage of computer systems increased, the more computer engineering organizations became challenged with a shortage of experienced workers available to perform day-to-day transactional database backups to maintain transaction records. For this very reason, the role of tape librarians was introduced using the on-job training method to facilitate this demand. By the end of the 70's, it became apparent that computer engineering organizations were beyond being acquainted with the integrated *Waterfall* framework and began practicing a semi-standardized framework in terms of managing production process. In parallel, the production development process sequence had shown some degrees of standardization along with the specification of the specialized *Division of Labor*. For example, the following specialized roles and tasks descriptions of the *Division of Labor* with the sequential production process were interpreted from DOL's 1978-79 data: First, the specialist *Systems Analyst* analyzes the business process in a form of a workflow to develop the business requirements with the specifications of the involved process scenarios to complete a given task. For instance, in a context of financial and accounting computer systems, the systems analyst draws a flow chart diagram sketching out the business scenarios (e.g., obtaining old balances, new charges, calculating financial charges and deducting payments) to specify what the computer system must perform to complete the transactions (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). This step-by-step process specification became an input to the coding phase. Second, the *Programmer* who specializes in the financial and accounting industry then develops the actual coding instructions that the computer systems must follow. The specialist *Programmer* then validates the instruction using test data to ensure that the code instructions work accordingly. If any errors were found, then the programmer performs a task called “debugging” to resolve the error until the computer systems produce what was intended (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). Third, thereafter, these results were delivered to the independent testers to test the overall functionality of the computer system to measure if the business requirements were met. If any bugs were found by the specialized *Testers*, then there was a reciprocal dialog between the *Programmers* and the *Tester* until final resolutions were made (Eicher, 2009). Finally, an end user manual was developed by the *Systems Analysts* who designed the system and conceptualized the business scenarios. In addition, a user manual for the *Console Operators* was developed by the *Programmers* who coded the computer system. The *Console Operator* manual was part of the production phase defined by Royce’s integrated *Waterfall* framework in table 5, as part of the maintenance production phase tasks. The role of the *Console Operator* was described by DOL’s 1978-79 report:

> Once the input is coded, prepared in a form the computer can read, it is ready to be processed. Console operators examine the programmer's instructions for the processing the inputs, make sure the computer has been loaded with the correct cards, magnetic tapes, or disks and then start the computer. While it is running, they watch the machine, paying special attention to the error lights that could signal a malfunction. If the computer stops or one of the lights goes on, operators must locate the problem and remove the faulty input materials. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 112, 1978-79a

Further descriptions of the final production phase (maintenance and outputs) regarding the roles and tasks of the specialized workers like *Converters*, *Tape Librarians* and *High-Speed Printer Operators* were provided by DOL's 1978-79:

> In some systems, machines directly connected to the computer translate output into the form desired by the programmer. In others, high-speed printers or converters run by auxiliary equipment operators-high-speed printer operators and converter operators perform this function. Frequently, data on punched cards, magnetic tape on punched kept for future use. Tape librarians classify and catalog this material and maintain files of current and previous versions of programs, listings and test data. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 112, 1978-79a

Hence, to some extent, during the late 70s, computer engineering organizations had reached some level of maturity in terms of utilizing a standardized way of conducting production process. Furthermore, this confirms the implementation of the integrated *Waterfall* framework to standardize the specialized *Division of Labor* operation in production process. This degree of maturity increased levels of demands in specialized *Division of Labor* forces to produce computer systems for the market and to improve productivity, so that production doubles by ways of efficiency (low resources cost & high productivity). Overall, based on the interpretation of DOL's data of the 70's, it is valid to objectively conclude that throughout the course of the 70's, computer engineering organizations had spent the entire decade searching for a framework that enabled them to develop a standardized production process for the newly emerged field and to indoctrinate this framework into the industrial production modes of production infrastructure. As the market demand was increasing for computer systems and businesses became more and more dependent on computer systems to conduct their daily transactions, further challenges for computer engineering organizations developed, such as how to produce computer systems to satisfy the market demand while profiting from it, as the computer systems were going to be the devices to be used as the means of production. With the use of Royce's integrated *Waterfall* framework, computer engineering organizations were able to define production process phases and the tasks involved within each production phases. They further were able to develop specialized *Division of Labor* for the production process. It was because of all of the challenges discussed earlier in this chapter that the computer engineering organizations spent the entire decade of the 70's searching for a framework. Once the 70's was used to standardize the framework, it paved the way for further advancement of specialized *Division of Labor* and the implementation of production process phases with roles and tasks involved within each, which is reported by the DOL data throughout the course of the 80's.

Table 11: Specialized *Division of Labor* force 1980-81:

#	Specialization 1980-81	Roles & Tasks Description 1980-81
1	**182,000 *Systems Analysis***	
1.1	***Systems Analysts***	Roles & tasks description remained unchanged.
1.2	***Lead Systems Analysts***	Roles & tasks description remained unchanged.
2	**228,000 *Programming***	
2.1	***Programmers***	Roles & tasks description remained unchanged.
2.2	***Application Programmers***	Roles & tasks description remained unchanged.
2.3	***Lead Application Programmers***	Produced final computer systems, validated application programmers' codes, developed production schedule and training materials for programmers.
2.4	***Systems Programmers***	Produced standard computer systems using programming languages.
2.5	***Lead Systems Programmers***	Produced final computer systems, validated programmers' code and provided training for programmers.
3	**666,000 *Computer Operating Personnel***	
3.1	***Computer Key Punch Operators & Data Typists***	Roles & tasks remained unchanged.
3.2	***Console Operators***	Roles & tasks remained unchanged.
3.3	***Lead Console Operators***	Roles & tasks remained unchanged.
3.4	***High-Speed Printer & Card-Tape-Convertor Operators***	Roles & tasks remained unchanged.
3.5	***Converter Operators***	Roles & tasks remained unchanged.
3.6	***Tape Librarians***	Roles & tasks remained unchanged.
Total	**13 specializations were implemented**	**1,076,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1980-81a

As interpreted in DOL's 1980-81 data in table 11, there were additional specializations of *Application Programmers Supervisor* and *Systems Programmers Supervisor*. This is an indication of using these roles to provide more on-job training, so that more specialized *Division of Labor* are produced to satisfy the market demand. The most important advantages of creating these lead roles and tasks were to ensure that the specialized *Division of Labor* in *Computer Systems Analysts* and *Computer Application Programmers* size were increased for the following crucial advantages for the computer engineering organizations: 1) to micro manage the workforces using the industrial hierarchy management mechanisms to double outputs by speeding the daily production process, 2) to double the daily production outputs by controlling labor cost, which guarantees surplus labor, and 3) to customize production processes and procedures in that workers follow the production requirements of computer engineering organizations' production process (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79a). Although the production process phases continued to be limited to 3 phases as it had been during the mid-to-late 70's, DOL had reported that there was an increase in the specialized *Division of Labor* size. For example, according to the DOL data during the late 70's, the number of *Computer Systems Analysts* was about 160,000 *Programmers* were about 230,000 and *Computer Operator Personnel* were about 565,000 (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1980-81a). However, as shown in table 11, the size of the specialized *Division of Labor* was increasing during the period of 1980-81: *Systems Analysis* increased to 182,000, *Programming* increased to 247,000 and *Computer Operating Personnel* increased to 666,000. The demand for the development of the specialized *Division of Labor* remained steady throughout the course of the 70's and also became evident in the early 80's. As discussed earlier in this section, the demand for independent desktop computers in the market continued to increase as businesses became more dependent on their day-to-day transaction processes on computer systems. Although there was an attempt to implement the integrated *Waterfall* framework during the 70's, computer engineering organizations were only able to adopt 3 production process phases (*System Analysis*, *Programming* and *Maintenance*). The inability of the computer engineering organizations to fully implement the integrated *Waterfall* framework for production process was primarily due to a lack of specialized *Division of Labor*. The shortage of specialized *Division of Labor* was due to the computer engineering field's newness in the industrial system with its unknown production process phase, including unknown roles and tasks. Thus, computer engineering organizations were not fully able to meet the market demand, which became one of the major challenges for computer engineering organizations because the computer field was newly emerged with no specific production process framework. Nevertheless, the efforts to increase the size of specialized *Division of Labor* for the computer engineering organization had shown an improvement. In addition to the gains of the specialized *Division of Labor* below in table 12, DOL's 1982-83 data shows a progressive step in terms of implementing the integrated *Waterfall* framework in the production process.

Table 12: Specialized *Division of Labor* force 1982-83:

#	Specialization 1982-83	Roles & Tasks Description 1982-83
1	**205, 000 *Systems Analysis***	
1.1	***Systems Analysts***	Roles & tasks description remained unchanged.
1.2	***Lead Systems Analysts***	Roles & tasks description remained unchanged.
2	**247,000 *Programming***	
2.1	***Application Programmers***	Roles & tasks description remained unchanged.
2.2	***Lead Application Programmers***	Roles & tasks description remained unchanged.
2.3	***Systems Programmers***	Roles & tasks description remained unchanged.
2.4	***Lead Systems Programmers***	Roles & tasks description remained unchanged.
3	**666,000 *Computer Operating Personnel***	
3.1	***Data Entry***	Produced input put data compilation and served as database processer. This roles and tasks were replacing the *Computer Key Punch Operators & Data Typist.*
3.2	***Console Operators***	Roles & tasks description remained unchanged.
3.3	***Lead Console Operators***	Roles & tasks description remained unchanged.
3.4	***High-Speed Printer & Card-Tape-Convertor Operators***	Roles & tasks description remained unchanged.
3.5	***Converter Operators***	Roles & tasks description remained unchanged.
3.6	***Tape Librarians***	Roles & tasks description remained unchanged.
4	**83,000 *Computer Tech-Services***	
4.1	***Computer Technicians***	Served as a computer machine repairer and produced computer parts by customizing to meet the environment.
4.2	***Senior Computer Service Technicians***	Produced final computer systems, validated *Computer Service Technicians*' outputs and provided training for *Computer Service Technicians*.
Total	**15 specializations were implemented**	**1,201,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1982-83a

As shown in table 12, in 1982-83, computer engineering organizations were able to implement one more production process phase called "*Computer Tech-Services*" including the roles and tasks involved (*Computer Technicians* and *Senior Computer Technicians*). During the same year, the roles and tasks of *Computer Keypunch Operators & Data Typists* were replaced by *Data Entry* for entering input data into the databases during 1982-83. Although the primary reasons for defining these specialized roles and tasks as specialized *Division of Labor* were the same as they were during the previous years, there was an additional causal factor here: the increasing development of terminal based desktop computers (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1982-83). The increase in terminal based desktop computers required more computer technicians to provide computer repairing services. This led to the implementation of *Computer Tech-Services* into the production process phase. Additionally, according to DOL data, this was also a product of the emergence of minicomputers and the growing demand for computer technical supports to provide maintenance services. As shown in Appendix A, the terminal based desktop computers and in Appendix B, the network systems were both being gradually accepted and they required computer technicians to maintain them. These computer systems and network connection systems were considered major advancements for the 80's. This meant another challenge for computer engineering organizations, as these computers and network developments were just coming to the surface as new innovations. Thus, computer engineering organizations were set out to define roles and tasks in the production process phase as they were not part of Royce's original integrated *Waterfall* framework for the production process phase. As discussed in earlier in this section, the 70's was the decade for development of the production process phase framework and the development of specialized *Division of Labor* for the computer engineering field. Steadily, as shown in DOL's data of the early 80's, computer engineering organizations were persistent in increasing the size of specialized *Division of Labor* and in implementing required production process phases with their roles and tasks to meet the market demand and secure production level surplus labor advantages. Although the implementation of Royce's integrated *Waterfall* framework was the base factor for sustaining and continuing the development of the specialized *Division of Labor,* the early 80's suggests that computer engineering organizations were amending new production process phases and developing the specialized *Division of Labor* (roles & tasks) in accordance with Royce's integrated *Waterfall* framework. DOL's data of the early 80's suggests that there was a uniquely diverse way of developing the specialized *Division of Labor* in the early 80's, distinct from the 70's. This trend is due to the emergence of computer machines from a large mainframe system that formed minicomputers (small desk-top computers) and network systems were developing faster than they did in the 70's. This trend of advancement in creation of specialized *Division of Labor* is shown in table 13, which contains DOL's 1984-85 data.

Table 13: Specialized *Division of Labor* force 1984-85:

#	Specialization 1984-85	Roles & Tasks Description 1984-85
1	**254,000 *Systems Analysis***	
1.1	***Business Systems Analyst***	Roles & tasks description remained unchanged.
1.2	***Lead Systems Analysts***	Roles & tasks description remained unchanged.
2	**266,000 *Programming***	
2.1	***Programmers***	Roles & tasks description remained unchanged.
2.2	***Lead Programmers***	Roles & tasks description remained unchanged.
2.3	***Application Programmers***	Roles & tasks description remained unchanged.
2.4	***Programmer Analysts***	Roles & tasks description remained unchanged.
3	**580,000 *Computer Operating Personnel***	
3.1	***Data Entry***	Roles & tasks description remained unchanged.
3.2	***Console Operators***	Roles & tasks description remained unchanged.
3.3	***Lead Console Operators***	Roles & tasks description remained unchanged.
3.4	***High-Speed Printer & Card-Tape-Convertor Operators***	Roles & tasks description remained unchanged.
3.5	***Converter Operators***	Roles & tasks description remained unchanged.
3.6	***Tape Librarians***	Roles & tasks description remained unchanged.
4	**124,000 *Computer Tech-Services***	
4.1	***Computer Service Technicians***	Roles & tasks description remained unchanged.
4.2	***Senior Computer Service Technicians***	Roles & tasks description remained unchanged.
Total	**14 specializations were implemented**	**1,224,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1984-85a

Table 14: Specialized *Division of Labor* force 1986-87:

#	Specialization 1986-87	Roles & Tasks Description 1986-87
1	**308,000 *Systems Analysis***	
1.1	***Business Systems Analyst***	Roles & tasks description remained unchanged.
1.2	***Lead Systems Analysts***	Roles & tasks description remained unchanged.
2	**341,000 *Programming***	
2.1	***Programmers***	Roles & tasks description remained unchanged.
2.2	***Lead Applications Programmers***	Roles & tasks description remained unchanged.
2.3	***Applications Programmers***	Roles & tasks description remained unchanged.
3	**565,000 *Computer and Peripheral Equipment Operators***	
3.1	***Data Entry Keypuncher***	Roles & tasks description remained unchanged.
3.2	***Console Operators***	Roles & tasks description remained unchanged.
3.3	***Lead Console Operators***	Roles & tasks description remained unchanged.
3.4	***High-Speed Printer & Card-Tape-Convertor Operators***	Roles & tasks description remained unchanged.
3.5	***Converter Operators***	Roles & tasks description remained unchanged.
3.6	***Tape Librarians***	Roles & tasks description remained unchanged.
4	**164,000 *Computer Services Technicians***	
4.1	***Computer Service Technicians***	Roles & tasks description remained unchanged.
4.2	***Senior Computer Service Technicians***	Roles & tasks description remained unchanged.
Total	**13 specializations were implemented**	**1,378,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1986-87a

As shown in table 13 and 14 of DOL's data, the development in specialized *Division of Labor* was continuing. Although by the mid 80's sizes of specialized workers in computer engineering for the 4 production process phases were showing some degrees of increase, the degrees of progress in generating a specialized workforce was slow. When comparing DOL's 1984-85 and 1986-87 data with the early 80's data, one can find slight progress in the generation of sufficient specialized workers, which remained to be one of the computer engineering organization's challenges. As for the production process, there was no change from a production process phase perspective. They remained the same, having only 4 production phases since 1982-83. The number of *Systems Analysts* grew slightly from 205,000 reported in 1982-83 to 225,000 in 1984-85, increasing by 20,000 and in 1986-87 it grew to 308,000 showing an increase of 54,000 specialized *Systems Analysts* (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1984-85a). Similarly, the number of *Programmers* changed from 228,000 in 1982-83 to 266,000 reported in 1984-85, an increase of 32,000 and a further leap to 341,000 was reported in 1986-87, which made for 75,000 more specialized programmers (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1984-85a). However, there was a decrease in *Computer Operator Personnel* in 1986-87 to 565,000 from 580,000 reported in 1984-85, which was a reduction of 15,000; it was reported to be 666,000 in 1982-83, the highest number ever achieved for the *Computer Operators Personnel*. The *Computer Operator Personnel* specializations were originally created during the 70's for the mainframe (large computer systems) that operated within a data center environment, which required *Computer Operating Personnel* to run the system. Thus, the decrease from 666,000 to 580,000 was caused by the development of the terminal based minicomputers that required more *Computer Technicians* for minicomputer systems' repairs, configurations and maintenance. As analyzed in the table 12 of 1982-83 data, the change in production devices took place, moving from the large mainframe computer systems to terminal based minicomputer systems, and businesses became dependent on them for their daily business transactions. As a result, the market demand for terminal based minicomputer systems and network systems increased. In 1984-85, this market demand of minicomputer systems was assessed by DOL:

> As the Nation's economy expands, more computer equipment will be used, and more technicians will be needed to install and maintain it. Business, government, and other organizations will buy, lease, or rent additional equipment to manage vast amounts of information, control manufacturing processes, and aid in scientific research. The development of new uses for computers in fields such as education and medicine also will spur demand. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 267, 1984-85

This scenario demanded its own specialized workers to provide services. Hence, by 1984-85, computer engineering organizations were pressed with the creation of roles and tasks for a new specialized workforce called "*Peripheral Computer Equipment Technicians*" within the Computer Tech-Services production process phase (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1984-85). According to DOL, computer engineering organizations managed to provide on-job training for the *Computer Operating Personnel* to become *Computer Technicians* as there was still a specialized workers shortage to accommodate for in this newly emerging minicomputer systems service. Although in 1984-85, computer engineering organizations were able to train about 124,000 specialized *Computer Technicians* for production process, the shortage in specialized computer technicians persisted throughout the 80's and the early 90's. For the most part, this particular shortage was caused by the emergence of the minicomputer systems, requiring new technical expertise that was hard to come by (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1984-85). In sum, the birth of the minicomputer systems placed more requirements on the specialized *Division of Labor*, which reinforced the creation of specialized *Division of Labor* starting in the 70's that continued throughout the course of the 80's and the early to mid-90's. As noted earlier in this section, the creation of specialized *Division of Labor* continued to accommodate the requirements of the newly emerging computer systems to provide technical services and to fully implement the integrated *Waterfall* framework's production process phases' roles and tasks, which set the standard for the computer engineering production process. During the later 80's and early 90's, computer engineering organizations continued to be challenged by the shortages of specialized workers, which hindered them from gaining surplus profits right from the production process, because during the time, the more specialized workers became available, the more the production was doubled and profit was gained from the production process. Table 15 shows the continuation of this trend in DOL's 1988-89 data.

Table 15: Specialized *Division of Labor* force 1988-89:

#	Specialization 1988-89	Roles & Tasks Description 1988-89
1	**331,000 *Systems Analysis***	
	Business Systems Analyst	This position title was changed. This is not reported. Perhaps, this is one of the changes during this period in the division of labor process.
	Lead Systems Analysts	
2	**479,000 *Programming***	
2.1	***Programmers***	Roles & tasks description remained unchanged.
2.2	***Lead Applications Programmers***	Roles & tasks description remained unchanged.
2.3	***Applications Programmers***	Roles & tasks description remained unchanged.
3	**565,000 *Computer and Peripheral Equipment Operators***	
3.1	***Data Entry***	Roles & tasks description remained unchanged.
3.2	***Console Operators***	Roles & tasks description remained unchanged.
3.3	***Lead Console Operators***	Roles & tasks description remained unchanged.
3.4	***High-Speed Printer & Card-Tape-Convertor Operators***	Roles & tasks description remained unchanged.
3.5	***Converter Operators***	Roles & tasks description remained unchanged.
3.6	***Tape Librarians***	Roles & tasks description remained unchanged.
4	**109,000 *Computer Services Technicians***	
4.1	***Computer Service Technicians***	Roles & tasks description remained unchanged.
4.2	***Senior Computer Service Technicians***	Roles & tasks description remained unchanged.
Total	**13 specializations were implemented**	**1,484,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1988-89a

Table 15 shows the continuation of creating specialized *Division of Labor* to facilitate the market demand for minicomputer systems and network communication systems. While the 70's was spent on the standardizations of a computer engineering framework, which became the reason for the development of the integrated *Waterfall* framework, computer engineering organizations spent the entire 80's dealing with the development of specialized *Division of Labor* and in the process of modifying terminal based minicomputer systems with network communication systems to replace the mainframe based computer systems. DOL's data has shown positive indications of increase in specialized *Division of Labor* for *Systems Analysis*, *Programming* specializations and *Computer Operating Personnel*. According to the historical trends analysis of DOI's data, *Computer Operating Personnel* specialized workers were increasing throughout the course of the 80's as their profession was needed by the computer engineering organizations to continue the operation of the large mainframe computer systems (Bureau of Labor Statistics, *Occupation Outlook Handbook*, 1980-89a). Similarly, computer engineering organizations made an effort to increase the number of *Systems Analysts* and *Programmers*. *Systems Analysts* were needed to design programmable workflows for different industries, so that *Programmers* could develop industry specific computer systems to be used in minicomputers for daily business transactions (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1988-89a). Due to this need, efforts were made by the computer engineering organizations to provide career development training to increase the number of *Systems Analysts* and *Programmers*. Similarly, *Computer Operating Personnel* were needed and subsequently trained to continue producing codes to sustain functionalities. The increase in these specialized *Division of Labor* is shown by DOL's data throughout the course of the 80's in table 16. *Computer Services Technicians* specialization did not exist in 1980-81, represented by "N/A" in the table below.

Table 16: Specialized *Division of Labor* force 1980-89:

#	Specialization	1980-81	1982-83	1984-85	1986-87	1988-89
1	***Systems Analysis***	**182,000**	**205, 000**	**254,000**	**308,000**	**331,000**
2	***Programming Analysis***	**228,000**	**247,000**	**266,000**	**341,000**	**479,000**
3	***Computer and Peripheral Equipment operators***	**666,000**	**666,000**	**580,000**	**565,000**	**565,000**
4	***Computer Services Technicians***	**N/A**	**83,000**	**124,000**	**164,000**	**109,000**

Source: Department of Labor: *Occupational Outlook Handbook*, 1980-1989a

Further motivating the development of minicomputer systems was the fact that large mainframe computer systems could only serve a few computer workers at a time, as they had to be shared by the workers. Thus, this large mainframe computer system could not be used by every specialized worker simultaneously. This became another challenge to computer engineering organizations in their efforts to increase the size of the specialized *Division of Labor*, because the large mainframe computer system could only be used by one type of worker at a time. Specialized *Programmers* would have to code on the mainframe computer *before* the *Converter Technician* could transform data cards to magnates on the same machine, and similarly was used at a different schedule by the *Computer-Tape-Librarian* to store data to storage archival (Department of Labor: *Occupational Outlook Handbook*, 1980-1989a). As a result, the large mainframe computer systems became a major factor in slowing the efforts of increasing the size of the specialized *Division of Labor.* Realistically, computer usage became a scarcity for specialized workers as they had to wait for their schedule to perform their tasks, let alone receiving on-job training. By the mid to late 80's, computer engineering organizations were convinced that the mass production of minicomputer as terminals became inevitable. They aimed to transform the production process environment from a single data center, where there were only a few large mainframe computer systems, into minicomputer based terminals, which were available for each worker to carry out production processes and to improve the on-job training process.

The emergence of the transformation of production environment from large mainframe to minicomputer based terminals, which began in the mid 80's and through the later 80's led to a new challenge for computer engineering organizations in the early 90's. The *Computer and Peripheral Equipment Operators* as a specialized role with its tasks that was developed to produce and maintain the large mainframe computer systems lasted only about until the late 80's. According to the DOL data trends, this happened because the terminal based minicomputer systems were replacing the large mainframe computer systems and the *Computer and Office Machines Repairers* specialized role introduced in the mid-80s was gradually replacing the *Computer and Peripheral Equipment Operators* (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1988-89a). Although DOL's data trends shows that the number of the *Computer and Peripheral Equipment Operators* was increasing throughout the course of the 80's, it started to decline as the lager mainframe based computer systems were being replaced by the terminal based minicomputer systems, along with the innovation of network communication to connect all the terminals. This gradual transformation event was described by DOL in the 1988-89 report:

> Advances in technology have reduced both the size and the cost of computer equipment while at the same time increasing their capacity for data storage and processing. These improvements in technology have fueled an expansion in the use of computers in such areas as factory and office automation, telecommunications, medicine, and education. Computer and peripheral equipment operators work mainly with large computer systems-the part of the overall computer market that has slowed down-employment of operators is not expected to rise as rapidly as in previous years. As the trend towards networking-making connections between computers-accelerates, a growing number of these workers can be

expected to operate minicomputers. *Occupational Outlook Handbook*, 234, 1988-89

With this trend in progress, computer engineering organizations began to setup on-job training to alter the career of *Computer and Peripheral Equipment Operators* to reinforce the creation of the specialized role known as *Computer and Office Machine Repairers*, which started in later 80's. This trend is shown in table 17 of DOL's 1990-91 data.

Table 17: Specialized *Division of Labor* force 1990-91:

#	Specialization 1990-91	Roles & Tasks Description 1990-91
1	**403,000 *Computer Systems Analysis***	
1.1	***Business Systems Analyst***	Roles & tasks description remained unchanged.
1.2	***Systems Analysts***	Roles & tasks description remained unchanged.
1.3	***Lead Systems Analysts***	Roles & tasks description remained unchanged.
2	**519,000 *Programming Analysis***	
2.1	***Programmers***	Roles & tasks description remained unchanged.
2.2	***Lead Programmers***	Roles & tasks description remained unchanged.
3	**275,000 *Computer and Peripheral Equipment Operators***	
3.1	***Data Entry***	Roles & tasks description remained unchanged.
3.2	***Computer Operator***	This position title was changed. This is not reported. Perhaps, this is one of the changes during this period in the division of labor process.
3.3	***Lead Computer Operator***	This position title was changed. This is not reported. Perhaps, this is one of the changes during this period in the division of labor process.
3.4	***Peripheral Computer Equipment Operator***	The salary decreased. Perhaps, more operators were available.
4	**128,000 *Computer and Office Machine Repairers***	
4.2	***Computer Service Technicians***	This position title was changed. This is not reported. Perhaps, this is one of the changes during this period in the division of labor process.
4.3	***Senior Computer Service Technicians***	This position title was changed. This is not reported. Perhaps, this is one of the changes during this period in the division of labor process.
4.4	***Computer and office machine repairers***	The roles and tasks for this specialization was to maintain computers and other terminal devices.
Total	**12 specializations were implemented**	**1,325,000 were employed.**

Source: Department of Labor: *Occupational Outlook Handbook*, 1990-91a

As computer engineering organizations spent the 70's standardizing their process framework and the 80's creating more specialized *Division of Labor*, the 90's brought them a new dilemma. During the early 90's, the workstation based multitasked minicomputer system was replacing terminal based single function minicomputer systems, which had replaced the large mainframe computer systems that were thought to be an impediment for speeding up the creation of the specialized *Division of Labor*. This brought a challenge with it, in that during the early 90's, computer engineering organizations were forced to mass produce minicomputer systems to be used as terminals and to create a new specialized *Division of Labor* known as *Computer and Office Machine Repairers* (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1990-91a). In the light of this dilemma, DOL's 1990-91 data in table 17 shows the sharp decrease of *Computer and Peripheral Equipment Operators* to 275,000 from 663,000 in 1988-89. To a great extent, this took place due to the replacement of this role by *Computer and Office Equipment Repairers* as discussed earlier considering that the *Computer and Office Equipment Repairers* specialization indicated 128,000 as it was implemented to the computer engineering production process. This shows that the speed of providing on-job training to produce a specialized role of *Computer and Office Equipment Repairers* was improving due to the availability of the minicomputer terminals. This particular specialized role with its computer technician based tasks was required for the new production device, in that the emerging office computerized devices were automating manual processes, replacing large printers and mainframe computer systems. These office peripheral systems like portable printer software, industry specific applications and drives were produced by the *Computer Programmers* and *Computer Systems Analysts*. Hence, the DOL data in table 17 shows a steady increase in *Computer Systems Analysts* and *Computer Programming* specialization, as these two roles were in demand to develop more minicomputer systems. Specialization in *Computer Systems Analysts* increased from in 1990-91 to 403,000 from 331,000 in 1988-89 and specialization in *Computer Programming* increase to 519,000 in 1990-91 from 479,000 in 1988-89. As the micro-computerized systems for office peripherals, network communications and minicomputer systems as terminals were developing during the early to mid-90s, computer engineering organizations continued to promote the creation of specialized *Division of Labor* in *Computer Systems Analysis*, *Computer Programming* and *Computer and Office Machine Repairers*, which are shown in table 18 below, respectively.

Table 18: Specialized *Division of Labor* force 1992-93:

#	Specialization 1992-93	Roles & Tasks Description 1992-93
1		**666,000 *Computer Systems Analysis***
1.1	***Business Systems Analyst***	Roles & tasks description remained unchanged.
1.2	***Systems Analysts***	Roles & tasks description remained unchanged.
1.3	***Lead Systems Analysts***	Roles & tasks description remained unchanged.
2		**555,000 *Computer Programming***
2.1	***Programmers***	Roles & tasks description remained unchanged.
2.2	***Lead Programmers***	Roles & tasks description remained unchanged.
3		**266,000 *Computer and Peripheral Equipment Operators***
3.1	***Data Entry***	Roles & tasks description remained unchanged.
3.2	***Computer Operator***	Roles & tasks description remained unchanged.
3.3	***Lead Computer Operator***	Roles & tasks description remained unchanged.
3.4	***Peripheral Computer Equipment Operator***	Roles & tasks description remained unchanged.
4		**143,000 *Computer and Office Machine Repairers***
4.2	***Computer Service Technicians***	Roles & tasks description remained unchanged.
4.3	***Senior Computer Service Technicians***	Roles & tasks description remained unchanged.
4.4	***Computer and office machine repairers***	Roles & tasks description remained unchanged.
Total	**12 specializations were implemented**	**2,955,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1992-93a

Table 19: Specialized *Division of Labor* force 1994-95:

#	Specialization 1994-95	Roles & Tasks Description 1994-95
1		***Computer Systems Analysis* (About 666,000 Employed)**
1.2	***Business Systems Analyst***	Roles & tasks description remained unchanged.
1.3	***Systems Analysts***	Roles & tasks description remained unchanged.
	Lead Systems Analysts	Roles & tasks description remained unchanged.
2		***Computer Programming* (About 555,000 Employed)**
2.1	***Programmers***	Roles & tasks description remained unchanged.
2.2	***Lead Programmers***	Roles & tasks description remained unchanged.
3		***Computer and Peripheral Equipment Operators* (About 266,000 Employed)**
3.1	***Data Entry***	Roles & tasks description remained unchanged.
3.2	***Computer Operator***	Roles & tasks description remained unchanged.
3.3	***Lead Computer Operator***	Roles & tasks description remained unchanged.
3.4	***Peripheral Computer Equipment Operator***	Roles & tasks description remained unchanged.
4		***Computer and Office Machine Repairers* (About 143,000 Employed)**
4.2	***Computer Service Technicians***	Roles & tasks description remained unchanged.
4.3	***Senior Computer Service Technicians***	Roles & tasks description remained unchanged.
4.4	***Computer and office machine repairers***	Roles & tasks description remained unchanged.
Total	***12 specializations were implemented***	**1,630,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1994-95a

According to DOL's early to mid-90's data, computer engineering organizations increased the number of specialized *Computer Systems Analysts* to 666,000 in 1992-93 and 1994-95, up from the 403,000 that was reported in 1990-91. Similar increase was shown by the *Computer Programming* role, where 555,000 in both 1992-93 and 1994-95 exceeded the 519,000 of 1990-91. On the contrary, during the same time frame, there is a sharp decrease in the *Computer and Peripheral Equipment Operators* to 266,000 from the 519,000. This sharp decrease was due to the transformation of the large mainframe computer systems to minicomputer systems. The *Computer and Office Machine Repairer*s specialized roles showed an increase to 143,000 from 128,000 reported in 1990-91, which showed only an increase of 15,000. This increase was very low and according to DOL's reports, this low figure was due to the emergence of the aforementioned computer systems as well as advance network communication systems called the world wide web (www) shown in Appendix B. For example, the advance computer systems of the 90's had built-in software automated to perform their own malfunctioning, which was the primary reason the decrease in the demand for specialized *Computer and Office Machine Repairer*s. This transformation automated most of the manual tasks that were performed by the specialized *Computer and Office Machine Repairer*s who were trained to perform command line troubleshooting from the terminal manually. The trends of automation during the mid-90s became progressive as more automating software applications were encouraged to be developed by the *Computer System Analysts* and *Computer Applications Programmers* (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1994-95a). This computerization advancement that automated manual procedures was described by DOL in 1994-95:

> The expanding use of software that automates computer operations gives companies the option of making systems user-friendly, which greatly reduces the need for operators. Even if firms continue to use operators, which for many are extremely likely in the near future, these new technologies free these new technologies free the operator to concentrate on unique problems and monitor a greater number of operations at the same time. The result is that as a few as 3 operators can accomplish the work previously done by 10. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 262, 1994-95a

Thus, during the mid-90s, this event revolutionized the way in which the computer engineering organizations created specialized *Division of Labor*. As the use of software for automations became the trend, the roles of *Computer Systems Operators and Office Peripheral Equipment Technicians* were redefined and with this in progress, more network connection specialists and computer workstation specialists became in demand. According to DOL's report in the mid-90's, some of these specialized *Computer Systems Operators and Office Peripheral Equipment Technicians* were transformed to become trainers of the future workstation computer systems and overseers of the overall computer systems. As reported by DOL in 1994-95:

> Computer operators or peripheral equipment operators who are displaced by automation may be reassigned to support staffs assisting other members of the organization. Others may be retrained to perform different job duties such as supervising an entire operations center, maintaining automation packages, and analyzing computer operations to recommend ways to increase productivity. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 262, 1994-95

The specialized *Division of Labor* force in the computer engineering field was still low in size except for *Computer Systems Analysts* and *Computer Programmers*. Most of the senior computer operators or peripheral equipment operators were tasked to create new specialized *Division of Labor*, namely computer systems specialists to configure and install microcomputer systems as workstations and specialized network communication specialists to configure and install network systems to interconnect workstation computers. As market demand increased for software applications and network connectivity, computer engineering organizations prepared specialized labor force to facilitate the demand. In this way, during the *Transitional Phase* (1970-95), computer engineering organizations were able to create specialized *Division of Labor* (roles & tasks) for the production process.

In sum, this section analyzes the development of the hierarchy management infrastructure and the standardizations of the *Waterfall* framework for production process with a particular emphasis on the creation of the specialized *Division of Labor* force during the *Transitional Phase* (1970-95). The lesson from this section was that, once the integrated *Waterfall* framework (production process phase, roles and tasks) was standardized for production process, computer engineering organizations continued to create the specialized *Division of Labor* at different events for different scenarios. For example, as analyzed earlier in this section, specialized *Division of Labor* were modified according to the production process occurring at a given time: 1) During the large mainframe computer systems era, 2) during the terminal desktop that was based on minicomputer systems, 3) and finally, during the compact workstations era that was microcomputer systems. Most significantly, as businesses were becoming more dependent on using computer systems for their daily transactions, computer engineering organizations were responsible to create specialized *Division of Labor* to provide the required services to the market during the mainframe, terminal and workstation eras. As this concluded the development of specialized *Division of Labor* force for the industrial production process, the analysis of DOL's historical data reveals that this period set the foundation for advanced specialized creation of the *Division of Labor* force for computer engineering organizations, which emerged during the informational era. This became evident later during the era as the emergence of the www brought on a new wave of distinct types of specialized *Division of Labor*, which will be analyzed in chapter

4 in great detail. For instance, www improved communications across the different production facilities, easing the production process and producing more computer systems. Computer engineering organizations started streamlining their ways of creating specialized *Division of Labor* forces, which meant that computer engineering organizations were preparing for improving productivity, which translates to surplus profit gains by introducing multitasking types of specialized *Division of Labor* forces in the production process. The trends that took place during the *Transitional Phase* (1970-95) had its foundation in the original Adam Smith's (1776) *Division of Labor* principles that were implemented in the industrial factories. For example, the concepts of specializing each worker using on-job training for specific tasks and computer systems was the exact principle that was used by computer engineering organizations' industrialism. After the successful creation of specialized *Division of Labor* for the production process, the challenge for the computer engineering organization became how to use each specialized worker to improve production process, so that the surplus labor is gained. This also pertains to Adam Smith's exchange vs. use value principles. Hence, as the dynamics amongst the industrial based hierarchy management infrastructure, production process, surplus gains and wage are the core dimensions in this study, the next section is dedicated to analyzing these dynamics' manifestation in computer engineering organizations during the *Transitional Phase* (1970-95).

3.4 Development of Surplus Values

This section analyzes how the production infrastructure (production process framework, computer systems as production devices and the specialized *Division of Labor* forces) and the management infrastructure (the hierarchy based management mechanisms) together were used by the computer engineering organizations to improve productivity by ensuring efficiency, which is the equivalent of taking advantage of free labor hours from the production process and generating profits. Responding to the aforementioned research question pertaining to this use of production and management infrastructure, this section analyzes the application of the industrial hierarchy management infrastructure (WBS, Gantt-Chart and BOE), including the accounting controllership methods (direct & indirect labor categories) to command production process operations. There will be a central focus in analyzing these methods' application to maximize surplus labor while computer products were being produced in production process. This section further analyzes the dependence of computer engineering organizations on the industrial production process environment (devices used as a means of production and facility) that was available to create a 3-shift based production process and surplus labor practice. This section analyzes the manifestation of surplus value during the *Transitional Phase* (1970-95) in computer engineering organizations. In the process of this analysis, this section analyzes the dynamics among industrial hierarchy management infrastructure, production process, surplus gains and wage disadvantages for worker manifest within the computer engineering organization. As analyzed in section 3.3 and 3.4, during the 70’s computer engineering organizations set the foundation of the production process framework, inherited by the industrial Pyramid based organizational and hierarchy management structure, including the managerial-accounting controlling mechanism. While the development of the production process framework contributed to the development of the specialized *Division of Labor* for computer engineering organizations, the hierarchy management structure provided the mechanism of scientific project management—with the insight of a labor wage controlling mechanism. This was done in an effort to gain surplus labor profits in the production process before the product was out for the market. This trend had its roots in Adams Smith’s 1776 work in specialized *Division of Labor* relative to production process and exchange vs. use labor values. As elaborated in chapter 2, Smith devised this mechanism in such ways that the specialized *Division of Labor* improved production process, and the surplus was to be acquired from the market price. During the era, computer engineering organizations did have a well-defined mechanism of gaining surplus labor from the production process. As analyzed in the previous sections of this chapter, computer engineering organizations were just beginning to implement production process frameworks like the integrated *Waterfall* framework and the industrial hierarchy management methods. Hence, during this time frame, computer engineering organization did not have a standardized framework to be able to extract surplus labor while the product is still in production process. Thus, following Smith's 1776 industrial *Division of Labor* framework's logical sequence, they started to develop their surplus labor in production process. As analyzed earlier in this section and in previous chapters, one of Smith's principles in 1776 was to develop the specialized *Division of Labor* forces, in which computer engineering organizations based their initial step to the development of surplus labor by forming first the specialized *Division of Labor*. A second principle of Smith's was to train workers according to their specialization roles and the tasks associated with these roles, so that they become proficient, which was also adopted by computer engineering organizations during the early 1970's. A third principle of Smith's logic was one of time saving, whereby all the production devices are within the same vicinity, which was also adopted by computer engineering organizations. This meant that a specialized worker could produce more in a short time frame, so that computer engineering organizations can maximize their surplus labor while the product is still in production process. Following these principles, once the specialized *Division of Labor* (roles & tasks) involved within each production process phase were standardized, computer engineering organizations implemented the industrial hierarchy management methodology discussed earlier in this chapter (e.g., WBS, Gantt-Chart and BOE), including the financial controllership methods (direct labor category) to enforce surplus value mechanisms. As described above in the list of tables, the integrated *Waterfall* framework contributed to the development of production process phases and the tasks specification within each production phase. Furthermore, as noted earlier, computer engineering organizations adapted the industrial hierarchy management infrastructure's WBS and Gantt-Chart to manage production process during all throughout the course of the 70's, 80's and through the mid-90's. In a very similar fashion, computer engineering organizations also adapted managerial accounting controllership principles (e.g., direct and indirect labor categories) to manage and control costs of the labor for tasks defined by the integrated *Waterfall* framework. Within the context of computer engineering organizations, the concept of direct labor category of accounting principle fits the tasks defined by Royce's integrated *Waterfall* framework (Eicher, 2009). However, within the context of computer engineering organizations, the indirect labor category is comprised of tasks pertaining to administrative support (e.g., administration, human resource recruiting, purchasing, and vendor processes). The purpose of adapting managerial accounting techniques of cost controlling (calculating hours spent on production against the WBS) was to enable production managers to perform direct category based labor cost bases of estimates (BOE) against the WBS's labor cost projections for each specialized task in production phases (Roehl-Anderson *Controllership,* 2004). As discussed in sections 3.2 and 3.3, every industrial organization's financial controller's standard practice was to require production managers to develop WBS based upon direct and indirect labor categories. In this way, the effectiveness of a production manager was measured by how much he/she saves from the total labor cost in production process (Thomas Hempell, *Computers and productivity*: *How Firms Make a General Purpose Technology Work,* 2006). This activity was practiced by ensuring that the daily output is measured by the production supervisor (e.g., supervisor programmer). As the production management mechanism was micromanagement, the supervisor programmer micromanages each worker

(Hempell, 2006). At the end of each work day, a site production manager (supervisor programmer) validates the output of each technical worker (Hempell, 2006). He then, puts the progress report in the form of a reporting template. Furthermore, this managerial controlling mechanism was applied to control the labor cost, as was determined in previous sections of this chapter. The accounting aspects of managerial controlling methodologies like direct and indirect labors were used to fully exercise surplus labor in production process while seemingly implying an improvement in productivity (Burawoy, *Manufacturing Consent*, 1979). This study analyzes the historical data on computer engineering labor forces and how the hierarchy management methods were applied to extract the maximum surplus labor in production process. Table 20 illustrates DOL's early 70s labor wages for Computer Systems Analysis and Computer Programming specialized *Division of Labor* in production process phases.

Tab1e 20: Illustrates the salary ranges of computer engineering workers during 1970-71

Specialization 1970-71	Annual Salary for Experienced	Annual Salary for Beginners
Computer Systems Analyst	**$12, 000 - $22, 000**	**$8, 950 - $12, 700**
Computer Programmers	**$12, 170 - $17, 000**	**$6, 550 - $8, 530**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1970-71b

Tab1e 21: Illustrates the salary ranges of computer engineering workers during 1972-73

Specializations 1972-73	Annual Salary for Experienced	Annual for Beginners
Computer Systems Analyst	**$14, 400 - $25, 000**	**$8, 462 - $14, 000**
Computer Programmers	**$14, 400 - $16, 700**	**$7, 850 - $10, 600**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1972-73b

As shown in table 20 and 21 above, DOL's record in 1970-71 showed the salary of computer engineering organizations' production process workers (*Computer Systems Analysts* and *Computer Programmers*). As can be seen in table 20 and 21 during the 1970-71, DOL has reported the workers' salary ranges in two categories: annual salary range for experienced and annual salary range for beginners. Following these categories, for example, for *Computer Systems Analysts*, DOL reported a range between $12,000 - $22,000 a year for experienced *Computer Systems Analysts* and $8,950 - $12,700 for beginners *Computer Systems Analysts* (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1970-71b). Similarly, during the same year, DOL reported a range of $12,170 - $17,000 a year for experienced *Computer Programmers*' salary and between the ranges of $6,550 - $8,530. Workers in the production process were paid for producing products according to the specialized roles and tasks they were performing. In DOL's 1970-71 report, the *Computer Systems Analysts* were earning more than the *Computer Programmers*. According to DOL, *Computer Systems Analysts'* roles and tasks were more involved than the *Computer Programmers*. During this era, computer engineering organizations did not have a well-defined mechanism of gaining surplus labor from the production process. As analyzed in the previous sections of this chapter, computer engineering organizations were just beginning to implement a production process framework like the integrated *Waterfall* framework and the industrial hierarchy management methods. Hence, during this time frame, computer engineering organization did not have a standardized framework to be able to extract surplus labor while the product was still in production process. As shown below in table 22 computer engineering organizations integrated a managerial labor cost controlling template, which includes the tasks names with labor cost associated with each production phase, compares the budget to actual hours worked, compares the budget to actual project start dates, and compares the budget to the actual project completion dates (Roehl-Anderson, *Controllership*, 2004). For example, putting the managerial accounting's cost controlling methodology in the context of computer engineering management's controlling their production process during the late 60's and early 70's, the template is captured by table 22 below.

This table shows a scenario whereby the hierarchy managerial controlling mechanism was applied to only one project's production process during the stand-alone mainframe computer systems era, because sharing the few available mainframe computer systems was the trend at that period. This computer system can only serve one worker at a time because it processes one task at a time due to the way the computer systems infrastructures were capable of performing during the early 70's. Thus, based on this circumstance, the computer engineering organizations were able to apply the industrial hierarchy WBS methodology as shown in table 22 below to control the surplus labor in production process.

Table 22: WBS methodology template:

Production Process Phase	Description	Start Date	End Date	Personnel	Labor Hours
Computer Analysis	Analyze, Design, develop programming specification and the test scripts	xx/xx/xxxx	xx/xx/xxxx	xxxxxxx	xxxxxxx
Programming	Coding, unit testing and providing user manuals/instructions	xx/xx/xxxx	xx/xx/xxxx	xxxxxxx	xxxxxxx

Source: Roehl-Anderson, *Controllership,* 2004

As analyzed above, the production devices used to produce computer systems and production facility infrastructures that were available determined the type of surplus labor mechanism that computer engineering organizations were able to apply while the products (computer systems) were still in the production process. For example, table 22 shows how the manager controlling mechanism was applied in the production process. But, the time set by the organizations to complete a project was based on the 3 shifts of a workday (e.g. 7a.m. – 3p.m., 3p.m. – 11p.m., 11p.m. – 7a.m.) around the clock in which projects are completed in sequential order, one after another. According to DOL's earning and working condition report during the late 60's and early 70's, computer engineering organizations were only able to apply shift based surplus mechanisms, which meant that one group consisted of both *Computer Systems Analysts* and *Computer Programmers*. They would produce in the 1st shift until the product was completed, and a second group would come in for the 2nd shift to do the same, and the 3rd shift followed after, all because the mainframe computer systems could only be used by so many at a time. Thus, production was around the clock due to the limited available infrastructure during the early 70's (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1970-71b). Based on this scenario, production supervisors were assigned for each shift and monitored the production process of each worker using the production progress tracking template. The use of the hierarchy management WBS methodology seen in table 22 above was created to be used in managing the labor cost of a single project's production process in production. For example, every *Computer Systems Analyst* was assigned to produce based on his/her specialized role (system flow analysis, proposed systems analysis or systems architect). Similarly, each *Computer Programmer* does produce according to his/her specialized role (code specification, entering codes or test codes). Hence, this set of workforce worked in one of the 3 shifts, while one analyzed, another designed and the last drew the architect and continued to work individually throughout the course of the production process. Thus, this scenario was applied to the different shifts in the same way. Based on production devices available as a means of production, hierarchy management WBS and production facility for production process, computer engineering organizations could only use the method of increasing production speed to complete the project as quickly as possible, so that another project could start with the same budget or remaining hours from the project just completed (Hempell, 2006). This means the computer engineering organizations were saving money by getting surplus labor from making the workers produce faster using mechanisms of training to make them proficient in their specialized roles and tasks involved in the particular production process phase. Nevertheless, similar to other engineering and manufacturing organizations' workers, the computer engineering organizations' workers remained earning the same amount of wages even though they were proficient enough in their specialty to increase the speed of production completion and doubled the level of productivity (Burawoy, *Manufacturing Consent*, 1979). As was analyzed by Burawoy (1979), this type of exchange labor value was not perceivable to the workforce as it seemed to them a part of improving productivity, and the workers believed that the existence of its success meant a lot to their existence in the organizations. They did not think that it was their exchange labor in the production process that continued to benefit the organizations. Nevertheless, for computer engineering organizations, the surplus labor in production process was not satisfactory, so they decided to improve production devices by switching from mainframe computer systems to terminal computer systems for the production process to maximize exchange labor by the mid-70's. This change enabled them to increase the number of workers in the production process and to introduce a different exchange labor mechanism. At this point, computer engineering organizations were still using the mainframe computer systems while the minicomputer systems were being tested for use. One of the primary purposes of implementing the minicomputer systems was to enable the worker to speed up the production process while producing the same amount of output for two projects. However, the earning wage remained the same for the worker even though the computer engineering organizations were getting exchange labor from two angles (speed of production and getting one project completed for no additional cost). Nevertheless, this took time, as there was a slow paced development and implementation of the minicomputer computer systems infrastructure. As shown in table 22, the trend of production process within the mainframe computer systems infrastructure continued to be the same throughout the course of the mid-to-late 70's forcing computer engineering organizations to continue to practice the shift based surplus labor process. This confirms that the industrial hierarchy management infrastructure (WBS, Gantt-Chart and BOE) with the available production process device (mainframe computer systems) allowed computer engineering organizations to practice shift based surplus labor process. Table 23 shows the continuation of the shift based surplus labor scenario.

Table 23: Illustrates the salary ranges of computer engineering workers during 1974-75

Specializations 1974-75	Annual Salary for Experienced	Annual for Beginners
Computer Systems Analyst	**$14, 400 - $22, 000**	**$8, 462 - $14, 000**
Computer Programmers	**$14, 600 - $17, 000**	**$8, 500 - $11, 000**
Lead Programmers	**$14, 000 - $18, 900**	**$10, 600 - $14, 000**
Console Operator	**$8, 256 - $8, 496**	**$6, 096 - $8, 256**
Keypuncher Operator	**$6, 336 - $6, 720**	**$5, 760 - $6, 000**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1974-75b

As table 23 illustrates, by the mid-70's, computer engineering organizations had two more additional specialized roles and tasks added into the production process phase. This increase was due to market demand for the more mainframe computer systems and the fact that the computer engineering organizations needed to increase their labor force for their production process. As analyzed above, by the mid-70's, the industrial computer engineering organizations management continued to adapt the industrial hierarchy management mechanisms to control labor cost and revamp production process. As shown in table 23 the 1974-75 salaries of workers, there was a decrease made in the yearly salary range for the experienced *Computer Systems Analysis* to $14, 400 - $22, 000 from $14, 400 - $25, 000, which was shown in table 21 of 1972-73. According to DOL report of 1974-75 shown in table 23, the yearly salary of the experienced *Computer Systems Analysis* was reduced by about $2,000. This might have happened as computer engineering organizations attempted to apply one of the major principles of supply and demand, which was analyzed in chapter two stating that if there was a market demand for a particular product, the producer would have to produce to supply more of that product; however, that would not have guaranteed the increase in the labor wages of the workers because there might be plenty of trained workers to hire. So, the owner/s would maintain the same salary range. Wage increase could only take place if there was a trained workers scarcity, with no one to produce that product in demand. Hence, applying the same concept in the context of computer engineering organizations, one possible interpretation of DOL's salary of the *Computer Systems Analysts* is that computer engineering organizations had sufficient *Computer Systems Analysts* to hire so that there was not a specialized workers scarcity. To make the matters even more intriguing, the annual salary for the beginners remained the same, from $8, 462 - $14, 000 to $8, 462 - $14, 000. Similarly, during the same year, DOL reported there was a similar adjustment made for the experienced *Computer Programmers* salary range from $14,400 - $16,700 to $11, 600 - $17, 000, showing a $300 difference from the year 1972-73. The salary range of the beginners in *Computer Programmers* was $7, 850 - $10, 600 and $8, 500 - $11, 000, which showed a slight increase of about $650 as reported in 1974-75. Hence, in the case of the *Computer Programmers* salary adjustment in 1974-75, the opposite of the *Computer Systems Analysts* supply and demand theory theorized above holds true. Interpreting the DOL salary report of 1974-75 on the *Computer Programmers*, one possible explanation could be the fact that there was a shortage of trained *Computer Programmers* to fulfill the market demand for mainframe computer systems. Computer engineering organizations had also amended two more specialized *Division of Labor* forces (*Console Operator* and *Keypuncher Operator*) and a *Lead Programmers* role for the production process phase. According to DOL's record, these specialized roles were in demand between the mid-70's to mid-80's as market demand continued to grow and computer engineering organizations scrambled for surplus labor and the creation of specialized *Division of Labor*. As described in section 3.5 of this chapter, the specialized role of the *Console Operators* was to translate the programming codes of the mainframe computer systems to administrate the system. The specialized role of the *Keypuncher Operators* was to insert the program codes into the magnetic devices, so that there was a communication between the hardware systems and the coded computer programs. For most part, these two specialized roles were operating during the 2nd and 3rd shifts of production process. Hence, the shift based management approach was applied as these roles were crucial for that period. Computer engineering organizations were attempting to utilize the available hierarchy management methods to control labor cost, so that they could get surplus labor right out of the production process. Table 24, which illustrates DOL's 1976-77 workers salary data, shows the trends and continuation of shift based surplus labor.

Table 24: Illustrates the salary ranges of computer engineering workers during 1976-77

Specializations 1976-77	Annual Salary for Experienced	Annual for Beginners
Computer Systems Analyst	**$16,080 - $18,240**	**$11, 040 - $14,400**
Lead Computer Systems Analyst	**$18,480 - $22,000**	**$18,240 - $18,480**
Computer Programmers	**$10,600 - $17,000**	**$7, 850 - $10, 600**
Lead Programmers	**$14,000 - $17,700**	**$10, 600 - $14, 000**
Console Operator	**$8, 256 - $8, 496**	**$6, 096 - $8, 256**
Keypuncher Operator	**$6, 336 - $6, 720**	**$5, 760 - $6, 000**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1976-77b

As the DOL's 1976-77 data indicated, computer engineering organization had continued the application of the industrial hierarchy management methods

(WBS, Gantt-Chart and BOE) to practice the shift based surplus labor. According to DOL's record, in 1976-77, the annual salary of the experienced *Computer Systems Analyst* remained the same as it was during the 1974-75 DOL's data. While the maximum cap remained at $22,000 since 1974-76, computer engineering organizations started using a complex method for adjusting the lower end cap from $14,400 to $16,080 for experienced *Computer Systems Analyst*s. Similarly, computer engineering organizations applied the same methodology to gain surplus labor when hiring the rookie *Computer Systems Analysts*, in that the starting annual salary range in 1976-77 was $11,040 - $14,400 vs. $8,462 - $14,000 in 1974-75. For example, the use of this complex methodology was undetectable by the average workers because the adjustment of the lower cap was increased from $8,462 to $11,040; however, the higher end cap remained at $14,000, which might have made workers feel as if salary range increase was made. This principle was also applied to the *Computer Programmers*' salary range; table 24 showed that in 1976-77 the higher end salary cap remained the same, as it was $17,000 since 1974-75 and the lower end salary cap was even further reduced to $10,600 in 1976-77 from 1974-75's $14,600. A more intriguing trend of reduction was that in 1976-77, the annual salary range for beginner *Computer Programmers* was reduced to $7,850 - $10,600 from 1974-75's $8,500 - $11,000. A very similar trend of a shift based surplus labor extraction method was applied for *Console Operators* and *Keypuncher Operators* in 1976-77. Their annual salary range remained the same; the salary range for the experienced *Console Operator* was $8,256 - $8,496, and the beginner *Console*

Operators remained exactly the same at $6,096 - $8,256. Finally, the *Keypuncher Operators*' annual salary range showed no change at all, as it remained the same since 1974-75, showing $6,336 - $6,720 for the experienced *Keypuncher Operators* salary range and $5,760 - $6,000 for the beginners. In a nutshell, 1974-75 and 1976-77's DOL's data provides two findings: First, what this data shows is that computer engineering organizations were beginning to utilize complex accounting principles, in which one was to adjust the lower end of the salary cap by leaving the higher end constant in a way unperceivable to workers as it was directly applied in the shift based surplus labor mechanism to extract exchange labor value to get production completed efficiently. Nevertheless, workers could not perceive the indirect meaning of efficiency as it implies the concept of improving productivity. According to Burawoy (1979), the application of complex accounting principles was unperceivable to workers, as they could not differentiate the use value from exchange value in production process. Second, the trend of the unchanged salary value shows the persistence of the market supply and demand for computer products. This is related to the reduction of labor wages for the purpose of extracting surplus labor while the computer products are in the production process. This means that if there is an increase in the size of a given specialized laborer group, the wage indicates a decrease and/or it remaining constant. For instance, as table 24 shows in 1976-77, there were a sufficient number of *Console Operators* and *Keypuncher Operators*, thus computer engineering organizations applied supply and demand driven labor wage controlling practices. Recall from section 3.4 of this chapter that in 1976-77 the number of *Computer Operating Personnel* was 500, and this production phase included: *Console Operators*, *Lead Console Operators* and *Computer Keypuncher Operators*. As described above, this meant that computer engineering organizations had sufficient numbers of *Console Operators* and *Keypuncher Operators*. So, when the number of these specialized workers continued increasing, computer engineering organizations consequently kept reducing the labor wages. In some instances, however, they kept the salary range constant. For example, as seen in table 24 illustrating the 1976-77 DOL data, labor wages were unchanged in the case of *Console Operators* and *Keypuncher Operators*, and a further reduction of labor wages was found in the case of *Computer Programmers*, which are practical indications of applying complex accounting methods and supply and demand based labor wage controlling methods utilized to maximize surplus labor in production process. Table 25 & 26 below illustrate the continuation of the shift based surplus labor trend throughout the course of 1978-79 and 1980-81.

Table 25: Illustrates the salary ranges of computer engineering workers during 1978-79

Specializations 1978-79	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Computer Systems Analysts	**$16,320 - $18,240**	**$11,040 - $12,000**
Lead Systems Analysts	**$18,480 - $22,000**	**$16,320 - $18,480**
Application Programmers	**$9,600 - $14,880**	**$9,120 - $9,600**
Systems Programmers	**$17,280 - $17,280**	**$9,120 - $9,600**
Lead Programmers	**$18,480 - $18,480**	**$9,600 - $14,000**
Computer Keypuncher Operators & Data Typists	**$6,336 - $6,720**	**$5,760 - $6, 000**
Console Operators	**$8,256 - $8,496**	**$6,720 - $7,200**
High-Speed Printer Operators	**$7,200 - $8,640**	**$5,760 - $6,720**
Converter Operators	**$9,840 - $10,320**	**$7,200 - $7,200**
Tape Librarians	**$7,680 - $7,680**	**$7,680 - $7,680**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1978-79b

Table 26: Illustrates the salary ranges of computer engineering workers during 1980-81

Specializations 1980-81	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Systems Analysts	**$16,320 - $18,240**	**$12,000 - $16,320**
Lead Systems Analysts	**$18,480 - $22,000**	**$16,320 - $18,480**
Application Programmers	**$9,600 - $17,280**	**$9,120 - $9,600**
Lead Application Programmers	**$11,040 - $19,920**	**$9,600 - $11,040**
Systems Programmers	**$9,600- $20,640**	**$9,120 - $9,600**
Lead Systems Programmers	**$11,040- $22,320**	**$9,600 - $11,040**
Computer Keypuncher Operators & Data Typists	**$6, 336 - $6,720**	**$5,760 - $6,000**
Console Operators	**$8,256 - $8,496**	**$6,720 - $7,200**
Lead Console Operators	**$12,480 - $14,400**	**$9,840 - $12,480**
High-Speed Printer & Card-Tape-Convertor Operators	**$7,200 - $8,640**	**$5,760 - $6,720**
Converter Operators	**$9,840 - $10,320**	**$7,200 - $7,200**
Tape Librarians	**$7,680 - $9,120**	**$7,680 - $7,680**

Source: Department of Labor: *Occupational Outlook Handbook*, 1980-81b

Above, table 25 illustrates the DOL's 1978-79 and 1980-81 data, whereby computer engineering organizations had incorporated 5 production process phases with roles and tasks, which were an indication of maturation in terms of standardization of the production process and the development of specialized *Division of Labor*. With this progress, computer engineering organizations were able to fully utilize the industrial hierarchy management methodology to practice the shift based surplus labor in production process. As illustrated in table 25 of 1978-79 and 1980-81, the experienced *Computer Systems Analysts*' wage remained the same at $16,080 - $18,240 and the *Lead Computer Systems Analysts*' wage remained unchanged at $18,480 - $22,000 as it was in 1976-77. Hence, in 1978-79 and 1980-81, the labor wage remained constant for the experienced *Computer Systems Analysts* and the experienced *Lead Computer Systems Analysts*, no different from 1974-75 and 1976-77. Furthermore, during the same year, a reduction to $11, 040 - $12,000 from $11, 040 - $14,400 was reported in the wages of the beginner *Computer Systems Analysts* to confirm that computer engineering organizations continued to apply the supply and demand based surplus labor extraction mechanism in production process, which meant that as the number of specialized workers increased, the labor wage remain constant or decreased. According to the DOL's specialized *Division of Labor* report of 1978-79, shown in table 25, the number of *Computer Systems Analysts* was 160,000 during this time, and in 1980-81, the number of *Computer Systems Analysts* increased to 182,000. Hence, it was this specialized *Division of Labor* increase that enabled computer engineering organizations to apply the industrial supply and demand based exchange labor value while computer systems were still being produced. DOL's wage report has been interpreted as showing that data trends during 1978-79 and 1980-81 indicate that labor wages have remained constant or reduced; this dissertation holds that computer engineering organizations had used this approach as a method of accumulating free labor while computer systems analysis was in production process. A similar trend of exchange labor value practice was deduced from DOL's wage report on *Computer Operating Personnel* during 1978-79 and 1980-81.

From the *Computer Operating Personnel* perspective, computer engineering organizations had created new specialized *Division of Labor* in 1978-79 and 1980-81. Table 25 contains DOL's 1978-79 and 1980-81 report, showing that computer engineering organizations incorporated *Computer Keypuncher Operators & Data Typists*, *Lead Console Operators, High-Speed Printer & Card-*

Tape-Convertor Operators, *Converter Operators* and *Tape Librarians'* roles and tasks as part of the production process phases. Hence, trends of surplus labor practices remained the same. For example, with these new incorporations, according to table 25 and table 26, there was a combination of supply and demand type of exchange labor value (constant or reduction) in wages and account controllership based exchange labor value (wage adjustment), which indicates an adjustment of the lower end of the wages in the average parameter. As noted in this section, the application of the exchange labor methodology was dependent upon the size of labor force. This means that during 1978-79, the number of the *Computer Operating Personnel* was 565,000 and further increased to 666,000 in 1980-81. Thus, during 1978-79, the salary of beginner *Console Operators* was reduced to $6,720 - $7,200 from $6,096 - $8,256 and remained constant at $6,720 - $7,200 in 1980-81. Similarly, the experienced *Console Operators*' salary range remained constant at $8,256 - $8,496 from 1976-77 through 1980-81 as illustrated in table 24 and table 26. Both the experienced and the beginner *Computer Keypuncher Operators & Data Typists* salary range remained constant from 1976-77 (table 24) throughout the course of 1978-79 (table 25) and 1980-81 (table 26). It also remained constant at $6,336 - $6,720 for the experienced *Computer Keypuncher Operators & Data Typists* and it was $5,760 - $6,000 for the beginners. During 1980-81, computer engineering organizations added specialized *Division of Labor* (*High-Speed Printer & Card-Tape-Convertor Operators, Converter Operators* and *Tape Librarians*) to reinforce their shift based production process, so that the supply and demand based industrial exchange labor cost would become cheaper as the size of the specialized *Division of Labor* increased. For instance, for the experienced *High-Speed Printer & Card-Tape-Convertor Operators*, it was $7,200 - $8,640 while the beginners it was $5,760 - $6,720. The experienced *Converter Operators* started at $9, 840 - $10,320, whereby the starting wage for beginners was $7,200 - $7,200. This was an example of the industrial supply and demand based exchange labor method that was applied in production process, whereby computer organizations continued to extract exchange labor value from workers by increasing the size of the specialized *Division of Labor*.

As described in section 3.5, computer engineering organizations began to focus on producing terminal based minicomputer systems to provide to the market and to implement into their production process during the early 80's (Theo Eicher, *Information technology and productivity growth,* 2009). Computer engineering organizations wanted to use a terminal based minicomputer dedicated to each specialized worker's desk to produce the specific task as defined according to the integrated *Waterfall* framework. In this way, the mainframe computer system that was shared by workers in different shifts was going through a disposition phase, because both the market and the computer engineering organizations were determined to improve the production process that suites their goal. To make this goal a success, computer engineering organizations revamped the idea of having *Computer Programmers* develop terminal based applications for the minicomputer systems. With this at hand, DOL's data in table 11, referencing specialized *Division of Labor* of 1980-81, confirms that there was a still shortage of *Computer Programmers*, while simultaneously the demand for *Computer Programmers* was increasing in the early 80's. As discussed in chapter 2 of this dissertation, this scenario demonstrates that the industrial supply and demand based surplus methodology operated when there was a labor scarcity (shortage), which showed a relative increase in labor wage. While this trend was in progress, from the Computer Programming perspective, computer engineering organizations had created new specialized *Division of Labor* in 1978-79 and 1980-81. Table 10 and table 11 contain DOL's 1978-79 and 1980-81 report that computer engineering organizations incorporated *Application Programmers*, *Systems Programmers* and *Lead Programmers*' roles and tasks as part of the production process phases. As can be seen in table 26 illustrating the trends of 1980-81, *Application Programmers* and *Systems Programmers*' wage increased somewhat for the experienced *Computer Programmers*. This suggests that computer engineering organizations were taking advantage of the expertise of the experienced *Computer Programmers* to speed up production process, so that more terminal minicomputer systems could be produced. This guaranteed the extraction of exchange labor value, which was blended with the use labor value in the production process, and at the same time, was imperceptible to the workers (Burawoy, *Manufacturing Consent*, 1979). For example, in 1980-81, DOL reported an increased at \$9,600 - \$17,280 from \$9,600 – 14,000 in 1976-77 for the experienced *Application Programmers*; however, in contrast, the beginner *Application Programmers*' wage remained the same, which is an indication that computer engineering organizations knew that the beginners had not yet acquired the expertise to perform in the same way as the experienced programmers that were used to speed up the production of the terminal minicomputer systems. Thus, to compensate for the exchange labor value, computer engineering organizations kept the wage of the beginner *Application Programmers* constant at $9,120 - $9,600 in 1976-77 and 1980-81. The demand for the *Computer Programmers* continued throughout the mid-80's as computer engineering organizations were transforming the production process devices from mainframe computer systems to terminal minicomputer systems, which were introduced with their own production process. This new trend of the mid-80's is analyzed in table 27.

Table 27: Illustrates the salary ranges of computer engineering workers during 1982-83

Specializations 1982-83	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Systems Analysts	**$18,720 - $22,080**	**$15,840 - $18,720**
Lead Systems Analysts	**$18,720- $23,520**	**$16,320 - $18,480**
Application Programmers	**$9,600 - $19,200**	**$9,120 - $9,600**
Lead Application Programmers	**$11,040- $20,640**	**$9,600 - $11,040**
Systems Programmers	**$9,600- $22,560**	**$9,120 - $9,600**
Lead Systems Programmers	**$11,040- $24,240**	**$9,600 - $11,040**
Data Entry	**$12,000 - $14,400**	**$10,560 - $12,000**
Console Operators	**$8,256 - $8,496**	**$8,400 - $8,400**
Lead Console Operators	**$12,480 - $14,400**	**$8,400 - $12,480**
High-Speed Printer & Card Tape Convertor Operators	**$7, 200 - $8,640**	**$5,760 - $6,720**
Converter Operators	**$9,840 - $10,320**	**$7,200 - $7,200**
Tape Librarians	**$7,680 - $9,120**	**$7,680 - $7,680**
Computer Service Technicians	**$18,480 - $20,640**	**$12,960 - $18,480**
Senior Computer Service Technicians	**$20,640 - $27,600**	**$18,480 - $20,640**

Source: Department of Labor: *Occupational Outlook Handbook*, 1982-83b

The early 80's was a major event that transformed the production process and the production devices (means of production). Simultaneously, as this transformation was taking place, computer engineering organizations had incorporated an additional specialized *Division of Labor* called *Computer Services Technicians* to create a labor force for this new type of transformation, which is shown in table 12. As analyzed above, during the early 80's, computer engineering organizations had transformed their production process from shift based to the individual based, and from a production devices perspective, from mainframe based to the terminal minicomputer systems. This wage parameter indicates that there was a *Computer Services Technicians* scarcity because this new production process and its new production devices required new specialized roles and tasks to be created to facilitate the computer engineering organizations' production process. This transformation from mainframe to terminal based minicomputer systems as a production process device enabled computer engineering organizations to utilize available labor forces without sharing devices across the different production processes. Hence, the emergence of the terminal based minicomputer systems dramatically increased the speed of production process, as it was available at each worker's desk. This enabled them to produce independently with no production time wasted, which had been a problem during the shared based mainframe computer systems era. In this new trend, a programmer could use his/her terminal to conduct all tasks involved (e.g., developing pseudo-codes specification, running coding, validating codes and developing systems user manuals) without waiting for another shift worker to complete the tasks, which wasted production hours. Nevertheless, although the surplus labor value methodology continued to be the same, wages were reduced or held constant relative to the size of the labor force. In another instance, the lower range of the wages was adjusted without actually increasing the upper end of the wage range. The only change from the emergence of the terminal based minicomputer system was that the production process became individual based, whereby workers completed tasks without sharing production devices sped up the completing of tasks. This was a great improvement in that workers did not have to wait for another shift crew to complete required tasks to in the final stages of production of the computer systems. In addition, computer engineering organizations management was challenged by this new era of production process that took place in the early 80's.

From management perspective, the transformation of the shift based production process and the mainframe computer systems to the terminal based minicomputer with individualized production process required its own management methodology within the industrial hierarchy management (WBS, Gantt-Chart and BOE) methodologies. Hence, computer engineering organizations had to develop a methodology that could be used to control the daily production progress of the individual worker. As Appendix C shows, computer engineering organizations started applying an engineering industry standard methodology called *weekly product progress report* (WPPR), which was used to micromanage task performance at the production process level and to control production hours. The designated project manager, first, outlined the task using the *Waterfall* specification of the production process path (programming: validating the architect of the system integration, validating the pseudo-code specification that was written by the *Systems Analysts*, running programming codes, performing unit testing and releasing final build to the independent testers). Second, the production supervisor allocates hours according to the specialized task to track how many production hours could be saved from this project, which could then be allocated to another project and translated to an exchange labor value for the computer engineering organizations. Third, the same production supervisor tracks issues that might prevent production process from being completed in a timely fashion to ensure that during production process, time was not wasted, which was adopted from the original Adam Smith's 1776 *Division of Labor* directives for industrialization to reinforce production time saving. Using this as a base, the computer engineering organizations' production supervisor then prevents logical issues, which are technical and training related issues, capacity issues, all related to production environments and infrastructure. Finally, human and financial resources related issues were prevented, so that time is saved to speed up the production process. This methodology was applied to control the production process and hours to gain exchange labor values during the new transformation era. Thus, in the era that brought about a new production process with its own production device (terminal based minicomputer systems), the size of the specialized *Division of Labor* started off at 83,000 *Computer Services Technicians* and their wages ranged started off at $18,480 - $20,640 for experienced workers and $12,960 - $18,480 for the beginners in 1982-83. Similarly, during this new transformation, the *Computer Systems Analysts* were in demand as they were designing terminal based systems in the early 80's. Hence, since the number of *Computer Systems Analysts* was still no more than 183,000, there was scarcity and as analyzed above, when there is scarcity in specialized *Division of Labor*, there is an increase in wages. In the case of the *Computer Systems Analysts*, as illustrated in table 27 of DOL's 1982-83, the experienced *Computer Systems Analysts* wage range had increased to $18,720 - $23,520 from $16,320 - $18,480 of the 1980-81 wage report. This increase was on both the lower and the higher end of the wage range, which can be interpreted as saying that there was indeed a shortage of *Computer Systems Analysts* to design terminal based minicomputers for the market and as production device for computer engineering organizations. However, during the same year in 1982-83, the wage remained constant at $16,320 - $18,480 for the beginner *Computer Systems Analysts*. This trend indicates that the beginners were considered less productive in terms of producing as compared to the more experienced *Computer Systems Analysts* that could produce at the expected level of speed to double the production, so that ample time is saved to extract unpaid labor in production process. As indicated in section 3.5 of the chapter in table 12, in 1982-83, there was still scarcity in *Computer Systems Analysts*, which forced computer engineering organizations to increase the wage parameter and the trend was also the same for *Computer Programmers*. *Computer Programmers* were as in demand as the *Computer Systems Analysts* during the same time frame for producing the terminal based computer systems. Hence, as the scarcity of labor became a preventing factor for the computer engineering organizations, they started increasing the number of *Computer Programmers* and *Computer Systems*

Analysts to control for wage range increases. For example, DOL's data in table 27 of the 1982-83 indicates that the lower side of the wage range remained constant and the higher end of the wage range slightly increased to \$11,040-\$20,640 for the *Lead Application Programmers*, and for the beginner *Applications Programmers* it remained constant at \$9,600 - \$11,040 as had been the case in DOL's 1980-81 reports in table 26. As this trend of labor scarcity continued throughout the course of the 80's, it became a challenge for computer engineering organizations because as they were going through the transformation of the production process and the production devices, they needed the *Computer Programmers* and the *Computer Systems Analysts* to produce the terminal based minicomputer systems. By analyzing the DOL data and reports of the computer engineering organizations on wages and sizes of specialized *Division of Labor*, the following can demonstrate what moves computer engineering organization had taken: 1) increasing the size of the specialized *Division of Labor* in *Computer Programming* and *Computer Systems Analysts* to control the wage range, 2) eliminating the *Computer Operating Personnel* specialized *Division of Labor*, as their roles and tasks were no longer needed due to the transformation of the shift production process and the replacement of the production devices (mainframe based computer system replacement by terminal based minicomputer systems) and 3) by replacing the roles of the *Computer Operating Personnel* with *Computer Service Technicians* to control labor cost in production process (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1982-83b). Thus, during 1982-83, DOL reported that the specialized *Computer Operating Personnel* wage range remained constant both for the experienced and beginners. This is one of the typical wages controlling mechanism practiced within the industrial production process. For example, the experienced *Console Operators* wage remained constant at $8,256 – $8,496 in 1982-83 and 1980-82, and the beginners' wage remained constant at $6,720 - $7,200. In addition, computer organizations began to replace specialized roles and tasks as they were no longer needed due to the transformation of the production device, which replaced the mainframe computer systems with terminal based minicomputer systems. In this instance, *Computer Keypunch Operators & Data Typists* were replaced by *Data Entry* as data transactions on terminal minicomputer systems became inevitable. Hence, the DOL wage records in 1982-83 showed a starting wage range between $12,000 - $14,400 for experienced *Data Entry* personnel, and $10,560 - $12,000 for the beginner *Data Entry* workers.

In sum, as this major transformation of production process and production device occurred, the specialized roles within the *Computer Console Operating* production process slowed in the increase of their sizes. In fact, DOL's data trends (e.g., wage range and size of the *Computer Console Operators* remained constant at 666,000 during the 1980-81 and 1982-83) suggest that computer engineering organizations were redefining their production process from shift to individual based. There was also an indication of a change in the exchange labor extraction methodology, which was based on speed of production process to save time so that it is used to process another production without additional labor cost. This new model became possible as a result of the replacement of the production device, which was at the birth of the terminal based minicomputer systems available for each worker. It was convenient to work without sharing mainframe systems with other shift members to complete the life cycle of a product completion. This trend continues on to 1984-85 & 1986-87 and the rest of the late 80's until it brought its own thesis and antithesis that sparked a new transformation in the production process and its exchange labor value methodology.

Table 28: Illustrates the salary ranges of computer engineering workers during 1984-85

Specializations 1984-85	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Business Systems Analyst	**$23,040 - $26,400**	**$21,408 - $23,040**
Lead Systems Analysts	**$26,400 - $28,608**	**$15,840 - $18,720**
Programmers	**$20,640 - $22,800**	**$14,480 - $20,640**
Lead Programmers	**$22,640 - $27,600**	**$20,640 - $22,640**
Application Programmers	**$21,840 - $25,600**	**$16,000 - $21,840**
Programmer Analysts	**$20,640 - $23,800**	**$14,480 - $20,640**
Data Entry	**$12,000 - $14,400**	**$10,560 - $12,000**
Console Operators	**$12,000 - $16,560**	**$10,560 - $12,000**
Lead Console Operators	**$16,560 - $18,400**	**$12,480 - $12,480**
High-Speed Printer & Card-Tape-Convertor Operators	**$7, 200 - $8, 640**	**$5, 760 - $6,720**
Converter Operators	**$9,840 - $10,320**	**$7,200 - $7,200**
Tape Librarians	**$7,680 - $9,120**	**$7,680 - $7,680**
Computer Service Technicians	**$20,640 - $23,040**	**$12,960 - $20,640**
Senior Computer Service Technicians	**$24,640 - $30,000**	**$23,040 - $24,640**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1984-85b

Table 29: Illustrates the salary ranges of computer engineering workers during 1986-87

Specializations 1986-87	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Business Systems Analyst	**$23,040 - $28,080**	**$21,408 - $23,040**
Lead Systems Analysts	**$28,608 - $33,120**	**$18,720 - $26,400**
Programmers	**$20,640 - $22,800**	**$14,480 - $20,640**
Application Programmers	**$21,840 - $25,600**	**$16,000 - $21,840**
Lead Application Programmers	**$22,640 - $27,600**	**$20,640 - $22,640**
Data Entry	**$12,000 - $14,400**	**$10,560 - $12,000**
Console Operators	**$12,000 - $16,560**	**$10,560 - $12,000**
Lead Console Operators	**$16,560 - $18,400**	**$12,480 - $16,560**
High-Speed Printer & Card-Tape-Convertor Operators	**$7, 200 - $8, 640**	**$5,760 - $6,720**
Converter Operators	**$9, 840 - $10,320**	**$7,200 - $7,200**
Tape Librarians	**$7, 680 - $9,120**	**$7,680 - $7,680**
Computer Service Technicians	**$21,960 - $23,040**	**$12,000 - $21,960**
Senior Computer Service Technicians	**$24,640 - $30,000**	**$23,040 - $24,640**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1986-87b

By 1986-87, it is apparent that computer engineering organizations had continued to apply the weekly production progress report (WPPR) mechanism to manage the new individual based production process. This hierarchy management mechanism was used by the project managers to control the individual worker's production hours according to the tasks assigned to ensure that remaining hours are transferred to another production task to be performed by the same worker free of additional labor costs, which translates to an exchange labor value (*Manufacturing Consent*, Burawoy, 1979). As analyzed above, the transformation of the shift based production process to the individual based production process was caused by the need for speeding up the production process to save production hours to be used for other tasks without incurring additional labor cost. Hence, using the WPPR mechanism, a project manager could simply control production hours of a worker as he/she was using his/her issued terminal based minicomputer system as a production device.

Analyzing DOL's data on computer engineering organizations' labor wage, during the periods of 1984-85 and 1986-87, one can see that computer engineering organizations continued to operate with the same trends of individual based production process with terminal based minicomputer system production devices. As discussed above, this new transformation of production process and production device brought about its specialized *Division of Labor* by replacing *Computer Operating Personnel* and by introducing a new specialized *Division of Labor* called Computer Tech-Services in 1984-85 as was required for the new terminal based minicomputer systems as a production device. Furthermore, by 1986-87, computer engineering organizations altered the roles and tasks of the *Computer Operating Personnel* by replacing them in accordance with the roles and tasks of the *Computer and Peripheral Equipment Operators*. Similarly, the roles and tasks of the *Computer and Peripheral Equipment Operators* were replaced by *Computer Services Technicians.* Analyzing DOL's report in 1984-85 and 1986-87, this event led to the demise of *Computer and Peripheral Equipment Operators*, as their size decreased to 565,000 from 580,000 of the 1982-83 size. Also, their labor wage remained constant at $16,560 - $18,400 as it was for the previous two years. By the contrary, the number of the *Computer Services Technicians* was increased to 124,000 from 83,000 of 1982-83, which indicated this specialized role and that its tasks were in demand, and this demand led to the scarcity of *Computer Services Technicians* workers in 1986-87. Due to this scarcity, the labor wage of those in existence increased to $24,640 - $30,000 from $20,640 - $27,600. As can be seen, both sides of the wage ranges had increased for the *Computer Services Technicians* and this was uncontrollable for computer engineering organizations, as they were still scrambling to fully implement the new terminal based minicomputer system as a production device to do mass production of additional terminal minicomputer systems for the market. Thus, according to DOL's assessment, in an attempt to have a better control of this rising labor wage for *Computer Services Technicians*, computer engineering organizations began to task *Computer Systems Analysts* and *Computer Programmers* to automate some of the manual technical services tasks that were performed by the *Computer Service Technicians* (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1988-89b). By 1986-87, this scenario caused an increase in the number of *Computer Systems Analysts* (*Business Systems Analysts* and *Lead Systems Analysts*) and *Computer Programmers* (*Programmers*, *Application Programmers* and *Lead Application Programmers*). Particularly, the size of the *Computer Systems Analysts* increased in 1986-87 to 308,000 from 254,000 that was reported in 1984-85. In a similar fashion, during the same year, DOL had reported an increase for the *Computer Programmers* to 341,000 from 266,000. Examining labor wage records of the *Computer Systems Analyst* and *Computer Programmers*, there was a trend of labor wage increase indicating a scarcity for these two specialized *Division of Labor*. For example, table 29 of 1986-87 shows that the labor wage for *Lead Systems Analysts* increased to $28,608 - $33,120 from $26,400 - $28,608 in 1984-85. Similarly, during the same year, the labor wage for the *Lead Application Programmers* showed an increase to $22,640 - $27,600 from $21,840 - $25,600 reported in 1984-85. At this point in time, computer engineering organizations were determined to introduce a major transformation to help them control the labor wage increase that was taking place as described for specialized *Division of Labor* of *Computer Services Technicians*, *Computer Systems Analysts* and *Computer Programmers*. Thus, in an effort to resolve the issue of controlling labor wage, so that efficiency in production process is accomplished (which then translates to the extractions of exchange labor wage hours in production process), computer engineering organizations were set to introduce two major solutions throughout the year 1988-89 and the early 90's: 1) Computer engineering organizations started increasing the size of these specialized *Division of Labor* to eliminate scarcity, so that labor wage increase was reduced, remained constant and/or the lower wage range was adjusted. 2) From a production device perspective, they began automating the daily tasks of computer systems maintenance duties. 3) Finally, from the perspective of putting it all together, computer engineering organizations brainstormed to invent a production device that would transform the terminal based minicomputer systems into something called workstation based multitasked computer systems, which were integrated with a local area network (LAN). Putting this historical transformation in chronological order, as similarly done earlier in this dissertation: first, the terminal based minicomputer systems transformed the mainframe based computer systems, which were shared among workers, and then, the workstation based multitask computer systems transformed the terminal based minicomputer. Similarly, from the production process perspective, during the early 80's, a historical transformation had taken place: first, the individual production process, which was implemented with the terminal based minicomputer systems as a production device replaced the shift based production process, which belonged to the mainframe based computer systems. Second, the individual production process that was implemented with the terminal based minicomputer systems as a production device was transformed to the multitasked based production process during 1988-89 and the early 90's. Framing it all in an historical perspective, computer engineering organizations spent the 70's standardizing the framework of the hierarchy management and the integrated *Waterfall* framework for production process. In the 80's, this was followed by a development of the specialized *Division of Labor*, the invention of production devices and refining of the surplus labor wage. Finally, the early 90's entailed the analysis of the early phase of the multitasked production processes and their multitasked based computer system as a production device, including the labor wage value, which are analyzed in upcoming parts of this section of the dissertation.

As can be seen in the DOL's labor wage reports, the transformation from the terminal based minicomputer systems as a production device to the multitasked based computer systems (coupled with the transformation of the individual based production process to the multitasks production process) ultimately controlled the labor wage. Table 30 of the 1988-89 and table 31 of the 1990-91 DOL's labor wage data indicates that their labor wages remained constant or reduced. This practice of controlling labor wages in production process was implemented by computer engineering organizations to promote efficiency (high productivity and low labor cost), which is equivalent to surplus value, the extraction of free labor in the production process.

Table 30: Illustrates the salary ranges of computer engineering workers during 1988-89

Specializations 1988-89	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Business Systems Analyst	**$23,040 - $28,080**	**$21,408 - $23,040**
Lead Systems Analysts	**$28,608 - $33,120**	**$18,720 - $26,400**
Programmers	**$20,640 - $27,600**	**$14,480 - $20,640**
Application Programmers	**$21,840 - $25,600**	**$16,000 - $21,840**
Lead Application Programmers	**$27,600 - $30,600**	**$22,640 - $27,600**
Data Entry	**$12,000 - $14,400**	**$10,560 - $12,000**
Console Operators	**$12,000 - $16,560**	**$10,560 - $12,000**
Lead Console Operators	**$16,560 - $18,400**	**$12,480 - $16,560**
High-Speed Printer & Card-Tape-Convertor Operators	**$7,200 - $8,640**	**$5,760 - $6,720**
Converter Operators	**$9,840 - $10,320**	**$7,200 - $7,200**
Tape Librarians	**$7,680 - $9,120**	**$7,680 - $7,680**
Computer Service Technicians	**$21,800 - $23,040**	**$12,960 - $21,800**
Senior Computer Service Technicians	**$23,040 - $30,000**	**$20,640 - $22,640**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1988-89b

Table 31: Illustrates the salary ranges of computer engineering workers during 1990-91

Specializations 1990-91	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Business Systems Analyst	**$23,040 - $28,080**	**$21,408 - $23,040**
Systems Analysts	**$26,400 - $33,120**	**$18,720 - $22,640**
Lead Systems Analysts	**$28,608 - $35,800**	**$18,720 - $26,400**
Programmers	**$20,640 - $27,600**	**$14,480 - $20,640**
Lead Programmers	**$27,600 - $30,600**	**$22,640 - $27,600**
Data Entry	**$12,000 - $14,400**	**$10,560 - $12,000**
Computer Operators	**$12,000 - $16,560**	**$10,560 - $12,000**
Lead Computer Operators	**$16,560 - $18,400**	**$12,480 - $16,560**
Peripheral Computer Equipment Operators	**$17,800 - $21,100**	**$10,320 - $17,800**
Computer Service Technicians	**$21,800 - $23,040**	**$12,960 - $21,800**
Senior Computer Service Technicians	**$23,040 - $30,000**	**$20,640 - $22,640**
Computer and office machine repairers	**$25,300 - $30,000**	**$22,640 - $25,300**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1990-91b

During the years of 1988-89 and 1990-91, computer engineering organizations had a better grip on controlling the labor wage of *Computer Services Technicians*; *Computer Systems Analysts* were *Computer Programmers*. What made this a success is that there were three major contributing factors: 1) As analyzed above, this became possible due to the fact that computer engineering organizations were able to increase the size of the these specialized *Division of Labor*, so that the labor wage cost could be controlled during the production process, 2) The transformation of the production devices from the terminal based minicomputer systems to the workstation based multitasked minicomputer systems became an additional factor, which automated daily manual tasks, 3) Finally, a contributing factor was that the transformation of the production process from individual based to multitasked based, which ultimately enabled computer organizations to have a worker perform multiple tasks at the same time using workstation based multitasked minicomputer systems, whereby each individual task was formerly performed by a specialized worker during the individual based production process era. Thus, although the pace of increasing the size of specialized *Division of Labor* was slow, as the field of computer engineering was new and was going through a continuous transformation every 5 years or so, computer engineering organizations managed to take advantage of the size of the labor force that might be available to them during the early 90's. Hence, from the perspective of increasing the size of specialized *Division of Labor*, by 1988-89, DOL reported that the size of *Computer Systems Analysts* increased to 331,000 from 308,000 in 1986-87. Similarly, in 1990-91, DOL records showed that the number of *Computer Systems Analysts* increased to 403,000 from 331,000 in 1988-89, showing a positive increase of 72,000. With this increase, during the same year, the *Lead Computer Systems Analysts* labor wage range remained constant at $28,608 - $33,120 in 1886-87, 1988-89 and 1990-91 indicating that computer engineering organizations controlled labor wages successfully. The same methodology of increasing the number of *Computer Programmers* and *Computer Services Technicians* was applied, so that the labor wage remains constant to guarantee the extraction of free labor.

During the period of 1988-89, computer engineering organizations had combined the roles and tasks of the *Application Programmers* and the *Lead Application Programmers* into a specialized role called *Lead Programmers*. By 1990-91, DOL reported that there were only two specialized roles called *Programmers* and *Lead Programmers*. With this change, the number of

Computer Programmers increased to 479,000 from 341,000 of 1986-87 and showed even more of an increase in 1990-91 to 519,000 from 479,000 of 1988-89. With these increases, computer engineering organizations managed to keep the labor wage of the *Lead Computer Programmers* constant at $27,600 - $30,600 during 1988-89 and 1990-91. For example, as shown in the tables 30 & 31 of 1988-89 and 1990-91, the labor wage for the beginner *Computer Programmers* also remained constant at $22,640 - $27,600. By using the method of increasing the size of the workforce, computer engineering organizations gained a great deal of advantage in terms of controlling the labor wage by following Adam Smith's 1776 *Division of Labor* theory within the context of supply and demand based production process, which worked by keeping the wage constant, reducing it or adjusting the lower side of the wage range. In this instance, as computer engineering organizations knew that *Computer Systems Analysts* and *Computer Programmers* were going to be in demand to develop workstation based multitasking minicomputer systems, the organizations had to increase the number of these specialized roles to continuously control the labor wage to make the specialized roles labor as cheap as possible as one of the means of controlling production costs. This concept of increasing the size of the specialized *Division of Labor* to control the labor wage before even the workers engaged themselves in production process was fully implemented in most engineering based organizations, and most of them were already beneficiaries of this implementation (Burawoy, 1979). Based on the interpretation of DOL's data on labor wage and creations of specialized *Division of Labor* of computer engineering organizations, it is apparent that the increase in size of the computer engineering workforce throughout the 70's, 80's and 90's was one crucial methodology that computer engineering organizations implemented to control labor wage before the specialized workers engaged themselves in production process. Further intriguing was that once workers were engaged in the production process, project managers were applying the hierarchy management methods, such as the Weekly Product Progress Report (WPPR) to monitor each worker's production hours against the tasks performed to extract left over hours that could be utilized to produce another task without additional labor cost known as exchange labor value for profit.

The WPPR, coupled with the method of increasing the size of specialized workers, reinforced the efforts of extracting free labor value, a method created to gain profit for when the product was still in production process before it went out for market transaction. In sum, computer engineering organizations had the following framework implemented to succeed in gaining profits from the production process: 1) increasing the size of specialized labor, 2) applying WPPR management methodology to control production hours to rollover the left over production hours to another project and 3) applying production process methods depending on the historical timeframe (e.g., shift based, individual based or multitasked based) using the production devices as a means of production (e.g., mainframe based, terminal based and workstation based) to speed up the production process. With this in mind, during the early 90's, computer engineering organizations transformed the production process and the production process devices to a framework of multitasking to promote efficiency in production process, which translates to gaining exchange labor value to control for production cost. For instance, according to DOL's data of 1988-89 and 1990-91, specialized roles such as *Computer and Peripheral Equipment Operators* and *Computer Services Technicians* were shrinking in numbers as most of their tasks were being automated by the workstation based multitasked minicomputer systems production devices. Furthermore, the transformation of the production process from one that was individual based to the multitasking production process occurred, which meant one worker is capable of performing multiple tasks at the same time using the multitasking based minicomputer system as a production device.

According to DOL data, during early 1988-89, the number of *Computer Services Technicians* was reduced and the number of *Computer and Peripheral Equipment Operators* remained the same. For example, during 1988-89, the size of *Computer Services Technicians* was reduced to 109,000 from 164,000 reported in 1986-87 and their roles were to provide services for the terminal based single function minicomputer systems. In 1990-91, as the workstation based multitasking minicomputer systems began to integrate the single function terminal computer system into their workstation and automate most of the manual tasks, a labor demand was brought on, which required computer engineering organizations to replace the *Computer Services Technicians* with a specialized *Division of Labor* called *Computer and Office Machine Repairers*. According to DOL data in 1990-91, the number of *Computer and Office Machine Repairers* was 128,000, and as the workstation based multitasking minicomputer system was becoming the production device, its role became maintaining the production devices by automating manual processes and office peripheral systems installations and configuration (e.g., replacing large printers with portable printers, installations and configurations of hardware/software systems and industry specific applications as well as developing networks and server infrastructures). Furthermore, computer engineering organizations were able to use the emergence of the new workstation based production device in the early 90's to permanently redefine the role of the *Computer and Peripheral Equipment Operators* in production process.

During the same year 1988-89, the size of *Computer and Peripheral Equipment Operators* remained the same at 565,000 as it was in 1986-87 and a dramatic decrease to 275,000 in 1990-91. Similarly, the labor wage remained constant for both the *Computer Services Technicians* and *Computer and Peripheral Equipment Operators*. During 1990-91, DOL's labor wage report stated that for the *Lead Computer and Peripheral Equipment Operators* the labor wage remained constant at $16,560 - $18,400, the same as it was in 1988-89 and 1986-87. During the same, the beginner *Computer and Peripheral Equipment Operators* were subjected to the same in which their labor wage remained constant at $12,480 - $16,560 and that remained the same in 1988-89 and 1986-87. Similarly, the *Senior Computer Services Technicians'* labor wage was held to remain constant at $23,040 - $30,000 and the beginner *Computer Services Technicians*' was constant at $20,640 - $22,640 during 1988-89 and 1990-91.

Hence, in the case of the *Computer Services Technicians* and the *Computer and Peripheral Equipment Operators*, computer engineering organizations successfully managed to reduce the number of *Computer Services Technicians* and *Computer and Peripheral Equipment Operators* by implementing the newly transformed production process (multitasking) that replaced the individual based production process and the workstation based multitasking minicomputer system as a production device that replaced the terminal based minicomputer systems. This transformation benefited computer engineering organizations in that their ultimate goal was to promote a multitasking based production process to be performed by each worker. The new multitasked workstation minicomputer was automating the manual tasks in single/individual based production process that was supported by terminal based single function minicomputer systems. DOL performed an assessment of the automation that was taking place during 1990-91, “As the benefits of automating computer operations become more recognized, the number of new operator positions in growing data centers will decline,” *Occupational Outlook Handbook*, 1990-91. This trend continued throughout the course of the mid-90’s. As can be seen in table 32 of 1992-93 and table 33 of 1994-95, the number of *Computer and Peripheral Equipment Operators* continued to decrease while the fall in labor wage remained the same throughout the course of the mid-90’s.

Table 32: Illustrates the salary ranges of computer engineering workers during 1992-93

Specializations 1992-93	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Business Systems Analyst	**$25,040 - $28,080**	**$21,408 - $23,040**
Systems Analysts	**$26,400 - $33,120**	**$18,720 - $22,640**
Lead Systems Analysts	**$28,608 - $35,800**	**$23,040 - $26,400**
Programmers	**$27,600 - $35,600**	**$14,480 - $27,600**
Lead Programmers	**$35,600 - $38,600**	**$27,600 - $35,600**
Data Entry	**$12,000 - $14,400**	**$10,560 - $12,000**
Computer Operators	**$12,000 - $16,000**	**$10,560 - $12,000**
Lead Computer Operators	**$16,000 - $16,400**	**$12,480 - $12,480**
Peripheral Computer Equipment Operators	**$16,800 - $21,100**	**$10,320 - $16,800**
Computer Service Technicians	**$15,900 - $23,040**	**$12,960 - $15,900**
Senior Computer Service Technicians	**$20,000 - $26,700**	**$15,900 - $20,000**
Computer and office machine repairers	**$22,400 - $25,300**	**$20,000 - $22,400**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1992-93b

Table 33: Illustrates the salary ranges of computer engineering workers during 1994-95

Specializations 1994-95	Annual Salary Range for Experienced	Annual Salary Range for Beginners
Business Systems Analyst	**$25,040 - $28,080**	**$21,408 - $23,040**
Systems Analysts	**$28,400 - $33,120**	**$18,720 - $26,400**
Lead Systems Analysts	**$30,608 - $35,800**	**$23,040 - $26,400**
Programmers	**$29,600 - $35,600**	**$14,480 - $27,600**
Lead Programmers	**$35,600 - $38,600**	**$27,600 - $35,600**
Data Entry	**$12,000 - $14,400**	**$10,560 - $12,000**
Computer Operators	**$12,000 - $16,000**	**$10,560 - $12,000**
Lead Computer Operators	**$16,000 - $16,400**	**$12,480 - $12,480**
Peripheral Computer Equipment Operators	**$17,800 - $21,100**	**$10,320 - $7,200**
Computer Service Technicians	**$15,900 - $23,040**	**$12,960 - $15,900**
Senior Computer Service Technicians	**$20,000 - $26,700**	**$15,900 - $20,000**
Computer and office machine repairers	**$22,400 - $25,300**	**$20,000 - $22,400**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1994-95

As shown in table 32 and 33, DOL reported that throughout the course of the early and mid-90's, computer engineering organizations continued to maintain the labor wage of the *Computer Systems Analysts*, keeping it constant. The *Lead*

Computer Systems Analysts labor wage remained \$28,608 - \$35,800 in 1992-93 and 1990-91. By 1994-95, computer engineering organizations applied the industrial accounting method of adjusting the lower end of the labor wage range for the hiring standard. In this case, the lower end of the labor wage for the *Lead Computer Systems Analysts* was increased to \$30,608 in 1994-95 from \$28,608 in 1992-93 and 1990-91. During the mid-90's, *Computer Systems Analysts* were highly in demand to develop the workstation based multitasking minicomputer system to automate manual tasks and enhance the multitasking production process. The data indicates, when comparing the adjusted labor wage of the *Lead Computer Systems Analysts* with the number of the *Lead Computer Systems Analysts*, that computer engineering organizations were taking advantage of the emergence of the workstation based multitasking computer systems and the transformation of the individual based production process to the multitasking production process. They gained two advantages: 1) By implementing the multitasking production process in practice, computer engineering organizations would no longer have to increase the number of *Computer Systems Analysts* like they did during the individual production process era that occurred throughout the course of the mid to late 80's. Instead, the focus became providing the training for *Computer Systems Analysts* to become multitasked, so that each individual could perform multiple tasks at the same time in an efficient manner. 2) By implementing the workstation based multitasking minicomputer systems, computer engineering organizations were able to replace the single/manual based function that occurred during the terminal computer systems by automation and then integrating functions into logical groups, which ultimately increased the speed of the worker in production process. Hence, the advantages that the computer engineering organizations had by implementing both the multitasking production process and the workstation based multitasking minicomputer systems was that efficiency in production process was guaranteed. This translated to extracting free labor hours from the *Computer Systems Analysts* while computer systems were still being produced in the production process. Computer engineering organizations applied this very same ideology that prevailed in the specialized *Computer Programmers* during the same time frame.

According to DOL's data illustrated in table 32 of 1992-93 and 33 of 1994-95, similar to the *Computer Systems Analysts*, labor wage adjustments were made; the lower end of the labor wage of *Computer Programmers* was adjusted to $29,600 from $27,600 of 1992-93 while the higher end of the labor wage remained constant at $35,000 in 1999-91, 1992-93 and 1994-95. This is an indication that the multitasking minicomputer system was automating the tasks of the programmers, which used to be performed manually. This confirms that the replacement of the terminal based single function computer systems by the workstation based multitasking minicomputer systems, coupled with the multitasking production process, and had altered the way in which a *Computer Programmer* performs the daily tasks. The automation and integration of the single function computer into multifunctional computer systems made the multitasking production process for the *Computer Programmers* efficient in the way the computer engineering organizations had demanded. It was a change from performing a single task at a time to performing multiple tasks at a time. For example, a programmer could perform the following tasks using his/her multitasking capable minicomputer system right from the workstation environment: validate the coding specification, view data structure and architects, execute codes, tests codes and write applications for an administrator's user manual. However, during the terminal based single function computer systems era, it was impossible to perform these tasks, as the operating system of the terminal computer system was not able to perform multitasking in an integrated structure. DOL reports its assessment of the emergence of the multitasking based computer systems in automating the manual programming tasks during the mid-90's:

> Employment, however, is not expected to grow as rapidly as in the past as improved software and programming techniques, including CASE and 4GL, simplify or eliminate some programming tasks. Someone who can apply CASE tool programming along with design and systems analysis is able to produce applications quickly and more cheaply. Employers are increasingly interested in workers who can combine both of these skills.
> Bureau of Labor Statistics, *Occupational Outlook Handbook*, 226, 1994-95

With this transformation, computer engineering organizations were able to control the hiring standard labor wage range and the ability to produce the daily programming tasks in production in an efficient manner by using one programmer to multitask and save production hours that can later be used for the tasks of other projects' production process to decrease labor cost. However, at the same time this scenario puts the programmer in the state of labor wage disadvantages as he/she produces without pay for the additional project's' tasks in production process. Furthermore, the number of *Computer Programmers* remained the same at 555,000 in 1994-95 as it was during 1992-93 DOL report. This trend of automating the manual tasks that were performed by the *Computer and Peripheral Equipment Operators* and *Computer and Office Machine Repairers* had changed during the mid-90's.

For the *Computer and Peripheral Equipment Operators* and *Computer and Office Machine Repairers*, the daily production process tasks were related to installation, configuration and troubleshooting of hardware and software computer systems. Thus, production was manually performed by *Computer and Peripheral Equipment Operators* and *Computer and Office Machine Repairers* during the terminal based single function computer systems and during the individual based production process. With the emergence of the workstation based multitasking computer systems, most of these tasks became automated. DOL reports its assessment of this transformation to minicomputer systems integration and automation on the specialized *Division of Labor force* in 1994-95:

> As automated equipment is developed further, smaller versions and lower prices will induce smaller organizations to invest in these technologies as well, further dampening demand for peripheral equipment operators. Employment of computer and peripheral equipment operators is expected to decline sharply through the year 2005. Advances in technology have reduced both the size and the cost of computer equipment while at the same time increasing its capacity for data storage and processing. Computer and peripheral equipment operators; however, will not benefit because they work mainly with large computer systems-the part of the overall computer market that has slowed down. Furthermore, the expanding use of software that automates computer operations gives companies the option of making systems user-friendly, which greatly reduces the need for operators. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 261-262, 1994-95

Thus, during mid-90's, computer engineering organizations started reducing the number of *Computer and Peripheral Equipment Operators* and *Computer and Office Machine Repairers* as they sought advantages of the new multitasking based minicomputer systems as a production device and at the same time the multitasking production process. By 1994-95, the number of *Computer and Peripheral Equipment Operators* remained constant at 266,000 as it was in 1992-93 and also the labor wage for the *Lead Computer Operating Personnel* remained constant at $16,000 - $16,400 as it was in 1992-93. Similarly, the *Computer and Office Machine Repairers* remained constant at 143,000 as it was during the 1992-93 and the labor wage for the *Computer and Office Machine Repairers* remained constant at $22,400 - $25,300 as it was during the 1992-93. This was the result of the automation of the manual tasks by the emergence of the new production device, which was built with multifunctional operating systems capabilities to run multiple tasks (e.g., an integrated software that has capabilities to prevent malfunctions, run system diagnostics and monitor systems functions) at the same time (cite). In addition, from production process perspective, computer engineering organizations benefited from the new multitasking based production process by downsizing their massive specialized *Division of Labor* because a worker was able to perform multiple tasks using one integrated workstation based computer system, which eliminated labor costs and saved labor hours during the production process. As demonstrated by this dissertation, the historical development of the specialized *Division of Labor* size and the labor wage were decreasing or remaining constant.

Hence, based on the analysis of the data trends during the *Transitional Phase* (1970-1995), the conclusion is that computer engineering organizations created the production infrastructure in logical order, whereby the first creation was the computer systems as production devices, followed by the development of the production process framework, which was the *Waterfall*, to define tasks and roles to create specialized *Division of Labor* force. Finally, computer engineering organizations adapted industrial hierarchy management methods (WBS, Gantt-Chart, WPPR & BOE) to reinforce efficiency, which could translate to profit gains by ways of free labor from the production process.

3.5 Conclusion

As analyzed earlier in this section, computer engineering organizations had reinforced the development of the production devices and the production process methods, which combined make up the production infrastructure and the management infrastructure to promote efficiency in production process; this constitutes an advanced mechanism of extracting free labor in the production process. It is critical to note that these production infrastructures and management infrastructure were developed by computer engineering organizations in such ways to complement one another. Below, table 34 illustrates this historical development phase.

Table 34: Historical Transformation of Production Infrastructures (Production Process and Production Devices):

Historical Timeframe	Production Device	Production Process	Specialized Division of Labor
1970 - 79	**Mainframe based shared large computer systems**	**Shift based production process**	**Required the size of labor forces be increased to control labor wages**
1980 - 92	**Terminal based single function minicomputer computer systems. *Eliminated sharing production devices**	**Individual based production process. *Increased the speed of completing a task in production process**	**Required the size of labor forces increased to control labor wages**
1993 - 95	**Workstation based multitasked minicomputer systems. *Integrated single functions, *Enabled Multi-system processors to produce multitasks**	**Multitasked based production process. *Enabled a worker to produce multitasks**	**Increasing the size of labor force was not required to control labor wage**

For example, these inseparable production and management infrastructures were complemented through historical development phases in the following ways: 1) In the late 60's and the 70's, the production infrastructures had mainframe based computer systems as production devices and the shift based production process as a pair made up the production infrastructure. Micro-supervising the outputs of each shift at the end its production process using the industrial hierarchy management mechanism (WBS, Gantt-Charts & BOE) made up the management infrastructure. 2) The second historical development phase of the production infrastructures and the management infrastructure took place in the early 80's and matured out towards the early 90's, as computer engineering organizations strove for efficiency (low labor cost & high production). As a result, the mainframe production device was replaced by the terminal based single function computer system and the shift production process was replaced by individual based production process. While the industrial hierarchy management mechanism (WBS, Gantt-Charts & BOE) as a foundation continued to be the same, micro-supervising the outputs of each shift at the end of its production process was replaced by the weekly product progress report (WPPR) method to best supervise the output of each individual. Accordingly, important to note is that both the production device, which was terminal based single function and the individual based production process were synchronized, one-to-one, perfectly. The single function production device can only produce one task at a time and a worker can use this device individually without waiting for another shift to engage in the production to complete the production process. This advancement benefited computer engineering organizations by doubling productivity by saving production time that was wasted by waiting for shifts to continue the production process. Most significantly, computer engineering organizations benefited from the fact that the production process hours were saved from each individual worker as he/she was able to use an independent production device without sharing with other fellow workers to complete tasks. 3) The third historical development phase of the production infrastructures and the management infrastructure started surfacing during the mid-90's. As analyzed above, computer engineering organizations had strived to implement the most effective methods that benefit them to improve production process, so that production time is saved that can translate to labor capital. Although the terminal based single function production infrastructure and its individual based production process had improved the speed of production and enabled individual workers to complete tasks without having to waste production hours in waiting for other workers to complete the production process, the infrastructure was not effective in controlling labor wage increase and to control the size of labor forces needed to complete tasks. Hence, again, for the third time in twenty-five years, the primary goal of computer engineering organizations became the transformation of the production infrastructure (production devices and the production process) and the management infrastructure. Thus, by the mid-90's, computer engineering organizations transformed the terminal based production infrastructure, which could only provide a single function at a time and its individual based production process method. At this point, computer engineering organizations were able to transform the production infrastructure to enable them to accomplish their ultimate goal, which was to control labor wage and size of labor forces. Hence, this made the production infrastructures the foundation factors that determined how labor wage was controlled and the size of specialized *Division of Labor* force. For example, as illustrated in table 34, the new transformation, which was the workstation based multitasking computer systems integrated the single function based terminal systems into modules using its capability of multi-functional operating systems to process multiple tasks at the same time. Furthermore, its production process eliminated the individual based production process by enabling a worker to complete all the tasks involved: analyzing the As-is workflows, developing the To-be workflows, developing data flow diagrams and finally developing the required structure charts with their coding specification to be submitted as a deliverable to the programmer. Thus, all of these tasks were not produced using the terminal production infrastructure era because of the fact that neither the terminal based single function computer system nor the mainframe computer system process were capable of doing so. Hence, computer engineering organizations built a multifunctional production infrastructure to accomplish their everlasting goal, the ability to save labor cost and to double production "efficiency" that was equally translated as a means to extract free labor hours from the production process. While this new transformation showed a promising future, computer engineering organizations knew that the *Waterfall* production infrastructure that was used to build the mainframe and the terminal production infrastructures would not be a feasible framework, as it was developed to serve only single function based top-down production processes, enabling a worker to complete only one task at a given time. Hence, computer engineering began to deal with the challenge of searching for a framework that was used as a method to create the multitasking based production infrastructure, which consisted of the production devices, production process framework and specialized *Division of Labor* forces. Thus, the framework that they intended to create for the new multitasking production infrastructure and the type of management infrastructure would be the one that could be utilized to effectively save production hours, which ultimately leads to the extraction of labor wage within the production process. In the quest of creating the appropriate framework for the multitasking based production process, computer engineering organizations began to reassess the Transitional Phase's *Waterfall* production process framework. This aspect had to be created first as a foundation, so that it could define the production process's tasks and roles, which ultimately creates the specialized *Division of Labor* force for the multitasking production process. As the *Waterfall* production framework was going through assessment, it was discovered that the *Waterfall* framework that it could only work as a top-down methodology throughout the course of the analysis, design, coding and test of the entire computer system development and integrate all the modules and finally perform testing right before releasing it to the market. Hence, if defects are found during the integration test phase, it could be costly to reverse engineer the product because each module has to be dismantled and reprogrammed. This worked for the shared based production process, which used the large mainframe computer systems, and for the individual based production process, which used the terminal based single function computer system capable to execute only a single function at a time. However, since the terminal based production infrastructure was replaced by the multitasking based production infrastructure, computer engineering organizations learned that the *Waterfall* framework could not work for multitasking based production process and its production devices, which was the workstation based multitasking computer systems, because it did not have the iterative capability to develop prototypes for every design without waiting to complete the coding of all the designs. Hence, for this reason, computer engineering organizations decided that the *Waterfall* production process framework was not cost efficient, as it could not complement the multitasking based production process and production devices. Further motives for the demand of a creation of a new production process framework was that the continued emergence of advance communication network systems (e.g., www, wide area network and local area network), which were implemented as production devices requiring a flexible model that defined the new roles and tasks to create multitasked based specialized *Division of Labor* forces. With this unprecedented advancement in progress, by the mid-90's, computer engineering organizations also learned that the hierarchy based management infrastructure had to be transformed, as micro supervision of the production process was no longer needed because the communication network was transforming as the internet was starting to be implemented as the production device to reinforce the production process (Theo Eicher, *Information technology and productivity growth*, 2009. Using the wide area network and world wide web internet capabilities, computer engineering organizations figured that they could cut production supervision cost, in that their managers could monitor production from a different facility (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). By the mid-90's, both the production infrastructure and the management infrastructure were going through a transformation phase, and this resulted in putting the computer engineering organizations in the situation of having to transform the pyramid industrial organizations structure, as it could complement only the hierarchy management structure. In sum, in the quest to analyze and demonstrate how computer engineering organizations gained free labor hours, which translate to profits from the production process, this chapter has shown that during the *Transitional Phase* (1970-1995), computer engineering organizations created a sophisticated production process framework like the *Waterfall* that was used to specify the production devices, process and specialized

Division of Labor forces. This chapter further showed that once the framework was successfully implemented, to micro supervise the production process for production outputs and closely control the production hours to save labor costs, computer engineering organizations adapted the industrial hierarchy management mechanisms (WBS, Gantt-Chart, BOE & WPPR). This chapter also shows how the creations of the production devices (e.g., mainframe, terminal & workstation) to improve productivity (speed of production output) and achieve sufficiency, which has also an equivalent meaning as gaining surplus labor hours from the production process. Despite the computer engineering organizations achieving their ultimate goals in saving production hours to gain free labor by having workers produce free of production costs, by the mid-90's, computer engineering organizations were set out to transform the modes of production (production devices, production process and creating new specialized *Division of Labor* force). As a result, the transformation was not limited to the modes of production, as it led to the inevitable transformation from the industrial organizational structure to the informational organizational structure, changing the way in which computer engineering organizations gain free labor hours from the production process. It also transformed the production devices, the production processes and the management infrastructure, which had been taking place over the course of 1996-2009.

CHAPTER 4

AGE OF INOFORMATION (1996-2009):

FROM MANUAL TO AUTOMATION

The overall theme of this chapter is an analysis the *Informational Era*'s modes of production infrastructures, which are the upside-down pyramid organizational structure and its management infrastructure (the macro and the matrix management infrastructures as mechanisms of managing the production process) and the means of production infrastructure (the production process framework, the multifunctional minicomputer systems as production devices and the specialized *Division of Labor* force, all three of which must simultaneously be in place for the computer engineering organizations' operations). This chapter has 5 sections: Section 4.1 introduces sections of chapter 5. Section 4.2 analyzes the upside-down and linear-horizontal organizational structure and the computer engineering organization's management infrastructure (macro and matrix based management) mechanism. Section 4.3 analyzes the development of computer engineering organizations' production infrastructure, which consists of three elements: production process framework, computer systems as production devices and specialized *Division of Labor* forces. Section 4.4 analyzes how computer engineering organizations used combinations of production infrastructure (production process framework, computer devices as productions devices and specialized *Division of Labor* forces) and management infrastructure (the industrial based hierarchy management mechanisms) to improve productivity (speed of production) and increase efficiency (low labor wages), which were equally translated as mechanisms of extracting free labor hours from the production process during the *Transitional Phase* (1996-2009).

4.1 Informational Organization and Management Infrastructure

This section focuses on the analysis of the informational modes of production, which are the organizational, the management and production infrastructures within the context of the contemporary era of the information age. As chapter 3 concluded, during the *Transitional Phase* (1970-95), computer engineering organizations created mechanisms that continually modified the management and the production infrastructures to reinforce efficiency and high productivity to achieve their goals in maximizing growth. Hence, during 1970-95, computer engineering organizations created and modified management and production infrastructures. Despite the fact that computer engineering organizations simply adopted the industrial pyramid organizational structure and the hierarchy management infrastructure during 1970-95, they created their own production infrastructure.

Highlighting the historical development of production infrastructures that were created and implemented during 1970-95: 1) as for the production process framework, computer engineering organizations created *Waterfall*, which worked in harmony with the hierarchy management infrastructure and the pyramid organizational structure. More significantly, the *Waterfall* production process was used to define process phases, which then were used to specify roles and tasks for creating the specialized *Division of Labor* force. 2) Using *Waterfall*, computer engineering organizations were able to create the step-by-step phases required by the tasks of each production process phase to develop computer systems in the early 70s. With this setup in place, computer engineering organizations created the large mainframe based computer system during the late 60s and the early 70s with its shared based production process, which was followed by the terminal based single function computer system and its individual based production process. The latter set the foundation for the development of the workstation based multitasking computer systems used as production devices. 3) The *Waterfall* production framework, with its top-down structure, effectively synchronized with the hierarchy management infrastructure because the management mechanism during that time was micromanagement, whereby a production supervisor would watch over the shoulder of each worker to ensure production time was invested efficiently. Furthermore, since the production framework and the management mechanisms were top-bottom based, computer engineering organizations used the *Waterfall* production framework to develop the production devices. As a result, computer engineering organizations used the mechanisms they created to maximize their potential in efficiency, saving labor hours in production process, which translated to free labor to save labor wages. However, as the during the early 90s, computer engineering organizations began to demand advanced innovations for production infrastructure, and with this motivation, computer systems analysts and programmers were assigned to create the multitasking based minicomputer systems as production devices to fulfill the vision of computer engineering organizations. Once this multitasking minicomputer was invented, the goal was to put it in production process to enable a worker to perform multiple tasks at the same time. Although this minicomputer with its capacity of handling numbers of tasks was revolutionizing production processes, it became impossible to successfully implement in practice because multitasking capabilities of specialized *Division of Labor* force were not available. Further, there was no scientific way of controlling the production process because the hierarchy management infrastructure mechanisms could not handle the increased micro-managerial supervision required by multitasking based production process and there was no mechanism to control labor hours and ensure efficiency within this new multitasking production process. A further intriguing aspect of the multitasking revolution of the production device was that it enabled macro-management and matrix management mechanisms, which required creating an organizational structure more flexible than the rigid pyramid structure of the industrial age.

In order to bring about a complete transformation of the organizational structure and the management and production infrastructures, computer engineering organizations began tasking technologists and management researchers to innovate mechanisms. As analyzed in chapter 3, the multifunctional based minicomputer system including the internet as a production device was developed during the early to mid-90s as a result of computer engineering organizations' demands to improve efficiency. The first quest for computer engineering organizations was to create a compatible organizational structure, which could be used as a structural foundation to create a management mechanism to enable computer engineering organizations to control production process. They simultaneously aspired to create a compatible production process framework to generate the appropriate multitasked based *Division of Labor* force for the informational era. With this in progress, the initial challenge for computer engineering organizations became creating the organizational structure.

As the multitasking based minicomputer systems with internet were revolutionizing the production process, computer engineering organizations started to fully benefit from these new mechanisms of efficiency because the production process changed from a single function to multitasking, which enabled a worker to perform a multitude of tasks with no micro-supervision and with great speed in completing the process. Hence, during the early to mid-90s, computer engineering organizations started initiating research to be done by the organizational behavior school, also known as "OB" (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). The research aimed at creating new organizational structures that would be compatible with the production process, production devices and management mechanisms horizontally encompassing open boundary communication (Schermerhorn, Hunt and Osborn, 2008b). In addition, the new management infrastructure had essential mechanisms that enabled open communication across the organizations, making it apparent that the pyramid based industrial organizational structure was no longer needed because internet emailing broke the top-down power that prior was too rigid in controlling communication between workers, managers and different department became open (Schermerhorn, Hunt and Osborn, 2008b). This task was carried out by members of the OB scholars like John Schermerhorn, James Hunt and Richards Osborn, all of whom produced scholarly work for the most part focused on organizational transformation in general and with particular attention given to computer engineering organizational structures. As they started their research in the mid-90s, the scholars stated:

> By the dawn of the twentieth century, consultants and scholars were giving increased attention to the systematic study of management. This gave impetus to research dealing with individual attitudes, group dynamics, and the working relationships between managers and their subordinates. Eventually, the discipline of organizational behavior emerged as a broader and encompassing approach. (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 7, 1997a)

The OB scholars began to assess the innovations that were emerging during the mid-90s to help them understand the type of organizational structure that needed to be created. They knew the 90s was revolutionizing traditional organizational operation, as computer systems were becoming far more advanced and day-to-day business processes were becoming more and more dependent on them. It was obvious to them that a base for a major transformation was being set. The OB scholars characterized the 90s based on their assessment in this manner:

> The 1990s may well be remembered as the decade that fundamentally changed the way people work. We have experienced the stresses of downsizing and restructuring; we have gained sensitivity to the peaks and valleys of changing economic time; and we have witnessed the advent of the internet with its impact on both people and organization. Truly progressive organizations, however, are doing much more than simply cutting employees and adding technology to reduce the scale of operations in the quest for productivity. They are changing the very essence of the way things are done. Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 5, 1997a

Hence, by mid-90s, their initial proposal of organizational structure for the informational age was known as an “upside-down pyramid”. The upside-down pyramid organizational structure was created to compliment the multitasking based production infrastructure and the macro-management based matrix management infrastructure (Schermerhorn, Hunt and Osborn, 1997a). The two major features of the upside-down pyramid structure were open communication and process engineering (Paulk, Weber and Curtis, *The Capability Maturity Model: Guidelines for Improving the Software Process*, 1994a). Open communication was possible for management entities and the workers, as they communicated across the entire organization via the new means of electronic email messaging and other advances in the local and wide area network systems that were available for accessing data (Schermerhorn, Hunt and Osborn, 2008b). With this new communication capability, computer engineering organizations were able to break the barriers (rigid chains of command, vertical management and controlled communications) of the industrial pyramid based organizational structure. Furthermore, the availability of internet emailing systems and access to production data within the organization unshackled the workers that had previously been subjected to scrutinizing micro-managerial supervision, which had promoted hierarchical superior/subordinate relationships practices in production process.

The emergence of this era's production devices radically changed computer engineering organizational structure. The internet, coupled with the development of workstation based minicomputer systems with multitasking capabilities, was one of the major components that the OB scholars applied to sketch the upside-down pyramid organizational structure (Schermerhorn, Hunt and Osborn, 1997a).   The diagram below demonstrates the contents of the upside-down pyramid organizational structure.

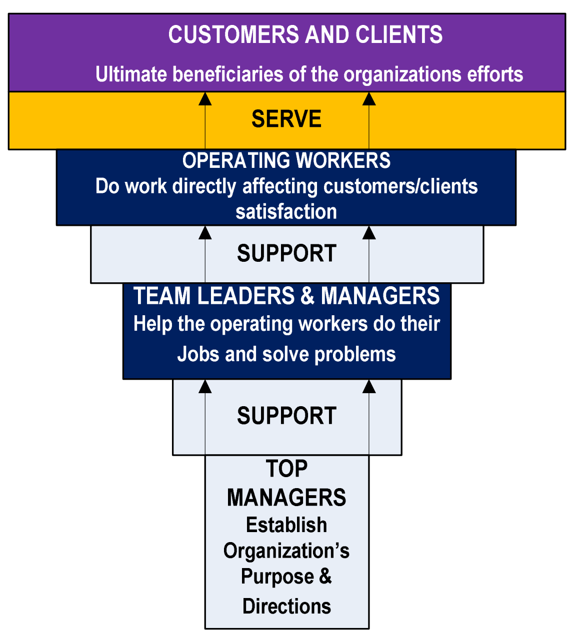


Figure 8: The Upside-down Pyramid Organizational Structure

The OB scholars created the upside-down pyramid demonstrated in figure 8. According to the structure, customers and clients are to benefit from all the services that are provided by the organization as a whole. The top executives are at the bottom and their roles and tasks are to develop ideas that support the operating workers (Schermerhorn, Hunt and Osborn, 1997a). This structure was centrally focused on breaking the traditional top-down based pyramid structure's controlled type of rigid communication culture (Schermerhorn, Hunt and Osborn, 1997a). The OB scholars define the upside down pyramid organizational structure as the following:

> The upside-down pyramid view of organizations focuses attention on total quality service to customers and clients by placing them at the top of the upside-down pyramid. It requires that workers operate in ways that directly affect customers and clients; it requires that supervisors and middle managers do things that directly support the workers; it requires that top managers clarify the organizational mission and objectives, set strategies, and make adequate resources available. Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 5, 1997a

Hence, this structure became compatible with the open work environment, using internet communication capabilities that were not available in the confined production facility operating during the single function production infrastructure that occurred prior to the mid-90's. This completely transformed the production environment and the ways in which workers engaged in their daily production process. As observed by the OB scholars:

> The workplace is changing and it will continue to do so. Call it whatever you like-reengineering, restructuring, transformation, flattening, downsizing, rightsizing, a quest for global competitiveness-it's real, it's

> radical, and it's arriving every day at a company near you. Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 3, 1997a

Today's computer engineering organizations have implemented the production environments that were part of the upside-down pyramid model.

With this in progress, by the late 90's to early 2000's, the upside-down pyramid organizational structure, coupled with the advanced production infrastructures, brought about the emergence of two new production environments on two different timelines. The first new production environment known as "telecommuting" was implemented during the late 90's and the advanced version of that was known as virtualizations, which was implemented between the early to mid-2000's. According to the OB scholars, "Telecommuting describes work done at home or in a remote location via use of a computer and/or facsimile machine linked to a central office or some other employment location. Sometimes this arrangement is called *flexiplace*" (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 168, 1997a). In addition to the telecommuting capabilities that the computer engineering organizations benefited from in their decreasing overhead costs via reducing production facility expenses (which was part of the issue aside from labor cost before the mid-90's), the upside-down pyramid organizational structure's flexible communication principle included virtual corporation capabilities whereby independent organizations collaborated to share resources for common production goals (Schermerhorn, Hunt and Osborn, 1997a). The virtual corporation aspect of the new organizational structure became possible because the upside-down pyramid organizational structure freed up communication flow across management chains and intended to use the full capacities of the advance internet systems as a means of production devices. This created a whole new organizational operation where computer engineering organizations would not run into labor scarcity as they were preparing for a new type of multitasking *Division of Labor* forces, since they were now able to hire workers regardless of their physical locations (Schermerhorn, Hunt and Osborn, 1997a). Hence, a group of computer engineering organizations form a virtual membership and share their resources through the means of internet access. This is an aspect of multitasking and automation at the stage of flexible communication through virtualization. The content of virtual corporation was defined by the OB scholars in this way:

> New technology has made possible the virtual corporation, one that exists only as a temporary network or alliance of otherwise independent companies jointly pursuing a particular business interest. Members of a typical virtual corporation consist of independent suppliers, customers and even competitors, who link up with the latest electronic information technologies and share such things as skills, costs, and access.
> Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 168, 1997a

To implement all of these aspects of the upside-down pyramid organizational structure, the OB scholars outlined an organizational change based process engineering framework.

As this transformation was based on revolutionizing the entire industrial pyramid organizational structure and modes of production infrastructure, which are the production and the management infrastructure, the OB scholars recommended that there be a methodology to make the transition as smooth as possible to the upside-down pyramid based organizational structure. Most significantly, an organizational change methodology was critical, as the transformation involved replacing the hierarchy micro based management to the macro matrix based management infrastructure. Concerning the production infrastructure, the transformation involved replacing the three components: 1) The *Waterfall* production process framework was to be replaced by the *Iterative* based production process framework, 2) the single functioned production devices were replaced by multi-functional production devices, and finally, 3) the single functioned based specialized *Division of Labor* force was replaced by the multitasking talent based *Division of Labor* force. Thus, an organizational change methodology was to be created to accommodate the newly opened communication based upside-down organizational structure, since the transformation was meant to revolutionize the entire modes of production infrastructure (Schermerhorn, Hunt and Osborn, 1997a). Hence, the organizational change management methodology that the OB scholars developed was shifting/transitioning the old, pre mid-90s type of workforce, management and organizations structure to a desired future state. Thus, the organizational change methodology introduced its three core components, which were the operational, production process changes, computer systems changes and changes in work ethics. Hence, the first step was to orient the organizational body about the open communication based organizations' operations with its process and procedures. The second step was to orient the organizational body about the emergence of the computer systems capabilities in improving productivity as the computer systems were becoming multitask based and automations of manual tasks were taking place. The third step was to change the work ethics from the single function based specialized *Division of Labor* to accommodate the abilities of a multitasking worker and to embrace the concept of self-motivation. For example, as the internet was becoming part of the production device, it was evident that the production process environment was going to be flexible in that work could be done remotely. Hence, these changes in the production environment could then lead workers to conduct their tasks with no supervision, which meant the organizational change methodology must then aim at empowering employees to promote self-motivation to produce, to accept their role and to embrace changes in their production environment.

From the management infrastructure perspective, computer engineering organizations were seeking a management model that could co-exist with the newly opened communication based upside-down organizational structure. Hence, the proposed management infrastructure, at this junction in the mid-90's, was the matrix management model that was based upon macro- management principles to compliment the new organizational structure and to replace the hierarchy framework based on micro-management structures of the *Transition Phase* (1970-1995). As part of the informational modes of production infrastructure, the upside-down organizational structure worked hand-in-hand with the macro based matrix management infrastructure; because both had the foundation of open communication and self-managing oriented structures. The two most important components of the upside-down pyramid organization structure were open communication and process engineering among with the multitasking internet based computer systems used to formulate the matrix-macro-management structure of the contemporary informational era of the post mid-90's (Paulk, Thayer and Mark, *Software Process Improvement*, 2001b).

As introduced above, the matrix-macro management infrastructure was linear based with an emphasis on macro-managing the subordinate workforces in production process. In order to have a comprehensive understanding of the contemporary informational era's management practice and method that was used from the beginning of the mid-90s, an in-depth analysis is critical. First, computer engineering organizations automated the hierarchy management's scientific management principles (WBS, Gantt Chart & BOE) using one of the multifunctional computer systems software known Microsoft Project (MS Project) and adopted it as part of the matrix-macro management methodology (Paulk, Thayer and Mark, 2001b). This was in an effort to continue the technique of applying measureable project schedules using a work breakdown schedule (WBS) and to improve efficiency and manage human resources to best utilize workers' hours in production process (Paulk, Thayer and Mark, 2001b). Using the MS Project multifunctional capabilities, a project's team leader is able to specify the tasks required to complete a module instead of specifying the entire computer system's full modules, as was the case during the hierarchy management of the industrial era (Paulk, Thayer and Mark, 2001b). Right about the mid-to-late 90's, computer systems engineering organizations standardized the MS Project tool to manage the multitasking production process, which directed a worker to perform required tasks using the multifunctional computer systems. Simultaneously, during the same time frame, computer systems engineering organizations began implementing an automated production framework known as the *Iterative* framework, which was developed by a team of methodologists like James Rumbaugh, Grady Booch and Ivar Jacobson (1998). The *Iterative* production has introduced its own automated production process method known as the Unified Modeling Language (UML), which is a centralized and an integrated tool, commonly used by subsets of production process methods to complete tasks of the production phases (Rumbaugh, Booch and Jacobson, *The Unified Modeling Language Reference Manual*, 1998). Hence, both the MS Project and the UML tools worked hand-in-hand in an integrated way, as they were both multifunctional computer systems. While the *Iterative* production framework was used to create the multitasking based production process phases, including the involved tasks and the new multitasked based *Division of Labor* forces' roles and responsibilities, its automated UML tool served crucial purposes that presented three great advantages for computer systems engineering organizations: 1) The automated aspect of the UML tool helped to promote efficiency, as it was multitasking based, combining all the required production process devices—this aspect enabled a multitasking worker to produce the expected output from the production process. 2) The centralization aspects of the tool enabled workers to access the production process devices remotely. 3) Finally, the UML tool integrated with the MS Project management tools, so that project leads and management administration could easily monitor the product in progress and final outputs in real-time, regardless of their location. Thus, these features enabled computer systems engineering organizations to promote efficiency by using multitasking devices to increase productivity and reduce labor costs, as it was now possible to assign multiple tasks to a worker.

Matrix-macro management is built on the open communication dimension of the upside-down pyramid organizational structure. This aspect created another common ground of co-existence between the upside-down organizational structure and matrix-macro management. This hand-in-hand relationship served as open communication, enabling project managers to macro-manage workers' production output and daily activities wherever the worker is located. As described in the previous section of this chapter, the open communication aspect of the upside-down organizational structure has enabled computer engineering organizations to run production processes in an open production environment (remotely accessing and telecommuting). Hence, based on these aspects of open communications, computer engineering organizations were able to practice matrix-macro management to manage workers remotely using the internet network access; a single project team lead or manager was now able to instantly apply macro-management to control the production process as the worker was logging into or out of the network (Paulk, Thayer and Mark, 2001b). The implementation of the multifunctional MS Project computer system, the internet network access and the open communication to the matrix-macro management were all reinforced by process engineering method to ensure that the mechanism worked effectively in controlling for the ultimate goal, which was the extraction of free labor hours in production process.

Computer systems engineering organizations sought to implement the methods of process engineering developed by scholars of the Capability Maturity Model (CMM) during the mid to-late 90's. The purpose of the CMM was to provide appropriate templates, accompanied with their process and procedure, as a framework for matrix-macro management practices (Paulk, Weber and Curtis, *Capability Maturity Model*, 1994a). The templates for project planning and project tracking & controlling (progress report, action items & issues tracking, risk & resolution tracking) were to be utilized when controlling production process, which were carried out in an open production environment, remotely. From a project planning perspective, the matrix-macro management infrastructure utilizes an automated template to plan the project (Paulk, Weber and Curtis, 1994a). As the matrix-macro management infrastructure automated the project planning template using the multitasking computer systems, computer engineering organizations began to implement the template to plan their product process activities to control production hours (Thomas Hempell, *Computers and productivity*: *How Firms Make a General Purpose Technology Work*, 2006). During the late 90's, the MS Project management device was beginning to redefine the activities of the project leaders and managers that were planning the multitalented workers' activities in production process (Paulk, Thayer and Mark, 2001b). Once the activities of the assignment were specified, schedules, costs and references with support resources were also specified (Paulk, Thayer and Mark, 2001c). As the production process environment such as the facilities were open, depending on the advancement of the wide area network and the internet, the activities of the assignments for the production process could simply be delivered to the worker through emailing (Thomas Hempell, *Computers and productivity*: *How Firms Make a General Purpose Technology Work*, 2006). Although the project planning template was automated into the MS Project management system to increase productivity, it made the activities of the project leaders and managers broad based by combining all the tasks into one integrated MS Project management system, which then assigned it to be performed by one project leader or manager (Hempell, 2006). By this time, computer engineering organizations had a project leader or manager utilizing the methods of project planning, tracking & oversight, deployments, maintenance, measurements and taking corrective actions (Paulk, Thayer and Mark, 2001c).

The weekly project progress report (WPPR) as part of the management planning template is shown in Appendix D. The template is automated into the MS Project management system and used by a project lead, team lead or manager to specify the production process in conjunction with the WBS, Gantt Chart and basis of estimates (BOE) (Paulk, Weber and Curtis *Capability Maturity Model*, 1994b). As discussed in chapter 3, it is during this planning phase that management implements the accounting labor controlling mechanisms (specification of direct labor categories and wage range adjustments) to estimate the amount of surplus labor that would be extracted while products are still in production process. The second phase of management, which is the project tracking & oversight (PTO), begins after the computer systems engineering organizations' senior executives (known as steering committee's heads) approve the labor controlling mechanisms implemented in the plan. This ensures efficiency and successful profit gains while the product is still in the production process. The designated multitasking project lead or team lead starts applying PTO methods of weekly product progress reports (WPPR), risk controlling, and issues & action items to monitor the production process in a real-time manner (Paulk, Thayer and Mark, 2001b). As shown in Appendix C, the automated WPPR template contains the required production phases with their required work activities that the worker is assigned to perform and tracks hours completed, issues & action items, and resolutions associated with each work activity. The term activity has been introduced as a substitute of the term task, which was used during the industrial management infrastructure. According to the CMM best practice standard, the multitalented worker must receive his/her specific activities at the beginning of the assignment (Paulk, Weber and Curtis, 2001b). The assignment form contains the hours and associated references that are required for the assignments listed items, and using the automated MS Project management tool, the project lead can remotely monitor the production process. In this scenario, during the mid to-late 90's, each worker was responsible for delivering his/her production progress update according to the given production progress update schedule for the given projects' production (using the WPPR) to the project lead or manager (Paulk, Weber and Curtis, *Capability Maturity Model*, 2000b). Thereafter, as required by the CMM best practices, the team lead or the project lead makes the necessary production progress updates through the automated WBS template for the steering committee members and stakeholders. As the team lead or project lead has access to the UML production tool, he/she verifies the WPPR report submitted by the worker to ensure that the production progress status is valid (Paulk, Thayer and Mark, 2001b). Although the automated MS Project management tool, in conjunction with the UML production device revolutionized, the way in which production and management operated to promote efficiency, computer engineering organization desired more integrated production devices, processes and procedures. As a result, by the early 2000's, the CMM methodologists incorporated a best practice known as Capability Maturity Model Integrated (Paulk, Weber and Curtis, 2001b). The CMMI introduced sharing artifacts, which were knowledge & information about product development templates, devices, processes, procedures and sample product outputs, which were all essential relative to the production process phases to be shared across the management and production teams (Ralf Kneuper, *CMMI*: *Improving Software and Systems Development Process*, 2009). In order to successfully implement this concept in practice, computer engineering organizations devised the following approach: 1) they realized that automation solution was effective in promoting efficiency in production process; however, an integrated and centralized production environment supported by production devices, processes and procedures was needed to share knowledge, information and monitor production process. 2) As the production process was based on multitasking, computer engineering organizations believed that this integrated production environment would provide devices, processes and procedures that a given worker needs to complete multiple production activities for in a given project (Kneuper, 2009). For example, using the integrated and centralized production environment, a systems analyst is able to utilize the knowledge and information sharing function to perform a search for artifacts such as sample analysis outputs, design methods and workarounds to perfect production outputs (Kneuper, 2009). 3) Most significantly, computer engineering organizations wanted to use the integrated and centralized production environment's knowledge and information sharing function for training purposes to handle multiple trainings for workers regardless of their location. In addition, the product output artifacts in the repository could be used by the team lead and project lead to extract data that could be used to develop bases of estimates (BOE) for the WBS before projects. In essence, they want to have every piece of information that they need to promote efficiency and control production cost available in the knowledge and information sharing function of the integrated production environment. Hence, by 2003, an integrated and centralized production environment known as Microsoft SharePoint Workspace was engineered by Microsoft (Paulk, Thayer and Mark, 2001b). This integrated and centralized production environment enabled a team lead or a project lead to monitor production process, outputs and verify WPPR all at once (Paulk, Thayer and Mark, 2001b). Furthermore, he/she was able to control a worker's production hours right from the system when logging in and out of the system (Kneuper, 2009). Thus, computer engineering organizations were able to revamp their ability to control production process, control labor cost and revolutionize the manual management templates into automation and then store them to the integrated & centralized management, which enabled team leads and project leads to maintain a twenty-four-hour capability to control production process in real-time sets (Hempell, 2006).

In summary, the primary focus of this section was to analyze the development of the informational based structure known as the upside-down organizational structure and the macro and matrix management infrastructure. As analyzed in this section, both the upside-down pyramid organizational structure and the macro-matrix project management structure were a combination of linear and upside-down based, whereby they were also based on open communication and enhanced by best practice process engineering methods.

While the upside-down organizational structure used organizational change as a best practice guideline, the macro-matrix management structure similarly used the CMM and the CMMI as best practice guidelines, which made both of them synchronize, effectively. This management infrastructure used the multitasking computer devices to automate its manual management templates (Gantt-Chart, WBS and BOE), processes and procedures into a MS Project management device. With this progress, computer engineering organizations reinforced their management infrastructure to become able to assign only one team lead or project lead to perform multiple management activities, which in turn saves resource cost management by reducing the number of management staff (Hempell, 2006). The open communication capabilities became available as a result of the internet, emailing and wide area network communication to enable team leads and project leads to monitor production process remotely. This was even further revamped by early to-mid 2000's when the Microsoft SharePoint Workspace system was implemented to have an effective way of monitoring the multifunctional based production process as a production environment open to remote access and telecommuting. The next section analyzes the development of computer engineering organizations' multifunctional based production process infrastructure (production framework, *Division of Labor* force and production devices) of the *Informational Era* between the periods of 1996-2009.

4.2 Production Process Framework 1996-2009

The analysis of section 4.3 is focused on the development of the *Iterative* production framework and its synchronization with the macro-matrix management infrastructure to reinforce computer engineering organizations' success in promoting efficiency through increasing productivity and controlling labor wages during the period of 1996-2009. As analyzed in section 4.2, computer systems organizations transformed the pyramid based industrial organizational structure to the upside-down pyramid based informational organizational structure. As the upside-down pyramid based informational organizations' structure was built upon an open communication in its foundation, it required its own set of modes of production that would work collaboratively. This scenario led computer engineering organizations to replace the industrial hierarchy management infrastructure with the macro-matrix management infrastructure to coincide with the upside-down organizational structure. Similarly, during that period, computer engineering organizations were set to replace the *Waterfall* production framework by creating a new framework that would synchronize with the macro-matrix management infrastructure and the upside-down organizational structure. As discussed in chapter 3, the reasoning was that computer engineering organizations' goal was to improve production process by transforming industrial production infrastructure by replacing the production framework and the single functional production process, changing to a multifunctional based production infrastructure. At this point, one of the first major challenges for computer engineering organizations was creating a production framework that aligned the production phases with their required work product activities to enter them into the automated WBS of the MS Project management system. This controlled production process better than the manual management mechanism of the *Transitional Phase* that occurred during (1970-1995). Hence, with the quest to accomplish their ultimate goal, computer engineering organizations' fate of success depended on a group of computer science methodologists. During the mid-90's, a group of methodologists named James Rumbaugh, Grady Booch and Ivar Jacobson introduced the *Iterative* production framework that worked with the automated macro-matrix management mechanisms (automated MS Project management system) (1995). In order to analyze the role of the *Iterative* production framework to become so instrumental as to meet the computer engineering organizations' goal, it is critical to highlight the causal factors that led to the emergence that transformed the *Waterfall* production framework (single functional production process) to the multifunctional production process.

As analyzed in chapter 3, during the 70's, computer engineering organizations searched for a production framework for the then fairly new computer engineering field, which never had a formalized production framework that defined the production phases and tasks or created specialized *Division of Labor* force's roles & responsibilities, all of which had to be quantified in the WBS of the management mechanism to setup a measurable production process. This led to the creation of *Waterfall*, which worked effectively with the manual hierarchy management mechanism and with the manual based single function production devices. The *Waterfall* production framework was built from scratch, as there were no preexisting production process frameworks for the field of computer engineering and was not available for the computer engineering organizations to implement in their operation. In 1970, Winston Royce did not have any formal computer engineering production process methodology to refer to when he developed *Waterfall*. However, for the computer engineering methodologists after 1995, James Rumbaugh, Grady Booch and Ivar Jacobson, there was already a production process framework to refer to (*Waterfall*). The production phases with tasks specification, single functional production process, the single functional based terminal production devices and specialized *Division of Labor* force were already in place. In addition, as analyzed in chapter 3's section 3.4, it is important to note that by the early to-mid 90's, the emergence of multifunctional based minicomputer system as a production device was replacing the single functional production devices, which caused incompatibility with the *Waterfall* production process framework, as it was created to work with the single functional production process, the single functional based manual production devices, specialized *Division of Labor* force and the manual based hierarchy management mechanisms. The emergence of the automation and multifunctional based minicomputer system with internet, including the wide area network communication systems as production process devices, was to promote efficiency in production process by using a worker to produce multiple tasks to squeeze labor cost as much as possible. However, this method was incompatible with the entire production and management infrastructure during the mid-90's, calling for a creation of a new production infrastructure (production framework, production process and multifunctional *Division of Labor* forces) as had been the case in transforming the hierarchy management infrastructure to the automated based macro-matrix management infrastructure analyzed in section 4.2 of this chapter. Thus, to analyze this transformation, it is important to analyze the content of *Iterative* production framework and its relationship with the macro-matrix management mechanisms discussed in section 4.2 of this chapter.

The integrated *Iterative* production framework had linear and spiral characteristics that enabled computer engineering organizations to build different computer modules independently and release them for market, using the framework's 4 major production process phases: inceptions, evaluation, construction and transition (Rumbaugh, Booch and Jacobson, *The Unified Modeling Language Reference Manual*, 1998). The table below shows the linear based structure coinciding with the macro-matrix management infrastructure, because it enables computer engineering organizations' team leads and project leads to use the multitasking production process across multiple production phases using a multitasking capable worker (Rumbaugh, Booch and Jacobson, 1998).

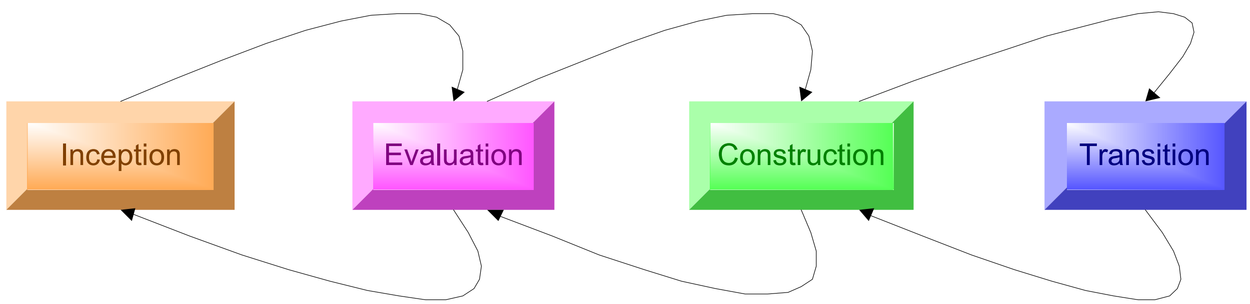


Figure 9: The *Iterative* Production Proces Framework
Source: James Rumbaugh, Grady Booch and Ivar Jacobson

Thus, figure 9 shows the linear aspect of the *Iterative* production process framework. Table 35 describes the meanings of the Iterative production process phases:

Table 35: The Iterative production process phases

Phase #	Production Process Phases	Production Process Phases' Descriptions
Phase #1	Inception	Inception identifies project scope, risks, and requirements (functional and non-functional) at a high level but in enough detail that work can be estimated.
Phase #2	Evaluation	Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
Phase #3	Construction	Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
Phase #4	Transition	Transition delivers the system into the production operating environment.

Resource: James Rumbaugh, Grady Booch and Ivar Jacobson (1997)

Another component of the *Iterative* framework, the spiral aspect pertains specifically to the production phases and methods created to define and specify the actual activities that are used to create the multifunctional *Division of Labor* roles and responsibilities (Rumbaugh, Booch and Jacobson, 1998). The spiral content enabled computer engineering organizations to apply the integrated industry's best practice processes and procedures as recommended by CMMI to complete computer systems' modules independently and to deliver them for market. The same approach could be repeated with remaining modules that would finally be integrated to make up the complete computer system (Rumbaugh, Booch and Jacobson, 1998). This meant the spiral aspect of the *Iterative* framework enabled computer engineering organizations to develop a module piecemeal, e.g. analyze systems, design prototypes, code programs, test the module in the computer systems and deliver it to market without waiting for the all of the entire computer systems' modules to be completed (Rumbaugh, Booch and Jacobson, 1998). Thus, according to the *Iterative* production process methodologists, James Rumbaugh, Grady Booch and Ivar Jacobson, the *Iterative* production process's spiral aspect to create multiple iterations recurs to create a fully integrated computer system as an end result. The *Iterative* production process uses an automated Unified Modeling Language production device to work in an integrated production environment. Thus, the common denominator among the upside-down, macro-matrix management mechanisms and the *Iterative* production process framework is the fact that they are built based upon open communication, automation and multifunctional supported by the CMMI best practices as foundations (Rumbaugh, Booch and Jacobson, 1998). This common denominator led to the promotion of efficiency by increasing production and controlling labor wages using the automated MS Project management device. Computer engineering organizations needed such a production process framework as the *Iterative* framework because they synchronize effectively. The table below shows the synchronization between the macro-matrix management mechanism and the *Iterative* production process framework. Hence, below, table 36 depicts the synchronization of the hierarchy management and the production process phases.

Table 36: Synchronization of macro-matrix management and *Iterative* production process framework phases:

Management Infrastructure Phases	Management Infrastructure Phases	Production Process Phases
Phase #1	Project Initiation & Planning	Inception
Phase #2	Project Tracking & controlled	Evaluation
Phase #3	Project Measurement, Quality Assurance, Change Control Management, Configuration Management & Corrective Actions	Construction
Phase #4	Project Closing	Transition

Source: James Rumbaugh, Grady Booch and Ivar Jacobson (1998)

In this manner, both the automated based macro-matrix management infrastructures and the *Iterative* production process framework in their synchronized fashion were used to control a successful management and production operation to promote a measureable operation that enabled computer engineering organizations to achieve their ultimate goal during the post 1996 - 2009.

As stated above in this section, the spiral aspect of the *Iterative* production process framework was created to specify the activities within the production process phases. This production process phases' activities specification is required by the macro-matrix management mechanism for the following purposes: 1) to specify the activities within the WBS, 2) to allocate labor hours, 3) to create multitasking based *Division of Labor* forces, 4) to assign multitasking capable workers according to their skills sets, 5) to be able to perform bases of estimates (BOE) during the project planning phase. All of these items listed are required for every module, as they are repeated the same way for every module until the complete computer system finalized and delivered to the market (Rumbaugh, Booch and Jacobson, 1998). Thus, the next section analyzes the development of the spiral components within the *Iterative* production process phases and how activities specified are used to fulfill the 5 stated above that are needed by the macro-matrix management.

4.4 Production Process, Division of Labor & Management 1996-2009

Although this section analyzes the development of the spiral based production process phases' activities, which is part of the *Iterative* production process framework, the section's primary focus will be analyzing the role of the spiral based production process as it relates to the following dimensions: 1) its relationship to the macro-matrix management mechanisms, 2) its contribution to the development of multitasking based *Division of Labor* (roles & tasks) and the development of multitasking production process. During the mid-90s, as the macro-matrix management structure was automated by MS Project management software device to provide multitasking capabilities to team leads and project leads, and as the production devices became multitasking based and accessible remotely through the internet communication network, computer engineering organizations needed production process phases' activities that were required by the macro-matrix management mechanisms for planning, tracking and measuring and ultimately controlling labor wage (Hempell, 2006). Hence, the first objective of the computer systems methodologist was to create the spiral based production process phases' activities that could be used for every module in a repeatable manner, whereby the specific activities are used in the WBS to plan out the required number of worker(s) amount of labor hours and labor wage (Rumbaugh, Booch and Jacobson, 1998). Simultaneously, the specification of the spiral production process activities were used to create the required multitasking based *Division of Labor* forces to overcome the limitation of *Waterfall* (replacing it due to its incompatibility) and ultimately provide the means to create a multifunctional production environment that allows computer engineering organizations to accomplish their goal in promoting efficiency. Hence, computer systems methodologists, James Rumbaugh, Grady Booch and Ivar Jacobson introduced 8 production process phases activities, which increased the numbers of activities by 3 when compared to the 5 found in the *Waterfall* production process framework between 1970-1995 (Rumbaugh, Booch and Jacobson, 1998). The diagram below illustrates the *Iterative* production process phases' activities:

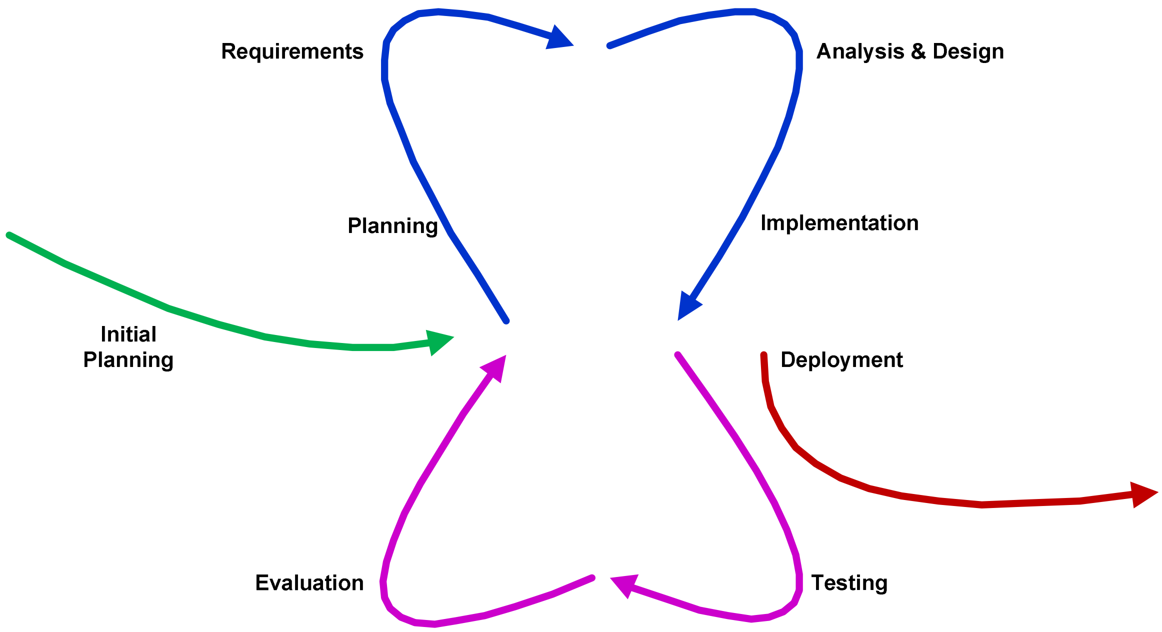


Figure 10: Production Process Phases' Activities
Resource: James Rumbaugh, Grady Booch and Ivar Jacobson

Hence, as can be seen in figure 8's diagram, the production phases' specific activities were established as one-to-one synchronization between the macro-matrix management structure and the production process phase. Every module could go through each step in the production process phases' activities as many times as needed before being released to market. Similarly, the remaining of the modules for the system would also go through the same process, and as each is completed, it would go through the system integration process and the same process would be repeated until all the entire system completed (Rumbaugh, Booch and Jacobson, 1998). In addition to increasing productivity in the production process, this incremental production approach's product was used as input in automated based macro-matrix management mechanisms (WBS and BOE) to perform planning, tracking and measuring to control labor wages in production process.

Once the production process phases were defined, the next challenge for James Rumbaugh, Grady Booch and Ivar Jacobson was to create a development methodology for each of the production process activities (Rumbaugh, Booch and Jacobson, 1998). Accordingly, the following production development methodologies were created within the *Iterative* production process framework: 1) To do this, they first developed a methodology for the requirements production process phase as, it is the base which is used as an input to the analysis & design, programming and constructs the testing scripts to measure whether the final computer system is accurate (Rumbaugh, Booch and Jacobson, 1998). They formulated a workflow methodology that could be used to define the exiting business process with identifications of limitations of existing systems that needed to be modified, and at the same time, could be used to conceptualize a new computer system that never existed (Rumbaugh, Booch and Jacobson, 1998). Once the workflow is completed, the output is a static prototype that illustrates production process and simultaneously, from management perspective, it is an input to develop the BOE to develop the WBS for the project planning. Once these activities are completed, the workflow is used as input to develop a case that converts the workflow business requirements to detailed functional requirements process specification to be used as inputs to the analysis & design production phase (Rumbaugh, Booch and Jacobson, 1998). At this point, the final requirement specification is also used as inputs to refine the BOE and update the WBS by management to further control the allocated labor cost by assigning more multitasking activities to a given worker. 2) The second production process phase development that the methodologist team, James Rumbaugh, Grady Booch and Ivar Jacobson created was the object oriented analysis & design (OOAD) methodology within the *Iterative* framework to replace the *Waterfall*'s structural based analysis & design development methodology (Rumbaugh, Booch and Jacobson, 1998). The purpose of this OOAD was to comply with the multitasking production process and incremental approach that enables computer engineering organizations' production process to reuse a best practice development processes and procedure for all the modules until they are completed (Rumbaugh, Booch and Jacobson, 1998). This reusability aspect helps in speeding up the process, as sample artifacts are used to repeat the sample methodology to develop the remaining modules in a timely manner by preventing delays. As discussed earlier in section 4.3 of this chapter, all of these development methodologies are available in the Microsoft SharePoint system, which is used as shared knowledge to make resources communal to promote efficiency. Thus, as multiple activities are produced by a single worker, all the required product development methods are available as samples from where a worker can extract them (Rumbaugh, Booch and Jacobson, 1998). Furthermore, the final outputs of the analysis & design that is developed using the OOAD methodology is used as inputs to develop programming specification in ways that are in compliance with the programming methodology (Rumbaugh, Booch and Jacobson, 1998). 3)

Similarly, for the programming production phase activities, Object Oriented Programming (OOP) language was created to replace the *Waterfall*'s structure based programming language (Rumbaugh, Booch and Jacobson, 1998). The OOP is created to coincide with the OOAD. OOP methodology has its own best practices' processes and procedures that coincide with the OOAD (Rumbaugh, Booch and Jacobson, 1998). As the outputs of OOAD are used to formulate the programming specification, the OOP then can simply use the OOAD programming specification to code without going through any conversion process to prevent delays in production (Rumbaugh, Booch and Jacobson, 1998). If there is an incompatibility in production development, then there is a delay while the output goes through a conversion process, which can result in high labor cost (Rumbaugh, Booch and Jacobson, 1998). That is why *Waterfall*'s structural programming methodology was replaced by the OOP to establish production development methodologies compatible across the entire production process phases spectrum to promote an overall efficiency. 4) Finally, for product testing, a test case methodology was introduced (Rumbaugh, Booch and Jacobson, 1998). As noted above, the use case based requirements development methodology was created, which was used to expand the workflows to functional specification, and was equally used as an input to develop the test scripts for validating each module to ensure that it fulfills the initial requirements (Rumbaugh, Booch and Jacobson, 1998). Hence, the use case requirements methodology is compatible with the test case in that it follows the same exact methodological structure; while the use case produces the functional requirements process specification of a module as outputs, the test case uses these outputs to develop the testing scripts specification without going through inputs and outputs conversions to save production time (Rumbaugh, Booch and Jacobson, 1998). Once the production development methodologies were developed and implemented based upon the CMMI recommend best practice processes and procedures, the challenge became to develop the detailed production process tasks involved within each phase of the production process, so that they could be used as inputs to the macro-matrix management's automated MS Project management system (Rumbaugh, Booch and Jacobson, 1998).

As the production phases' activities specification is required as an input to formulate the BOE and WBS for production process planning, it is equally required to enable computer engineering organizations to maximize their profits through creating multitasking based *Division of Labor* forces to improve productivity by using a multitasking worker and multitasking production devices, which led to the equivalent of surplus labor gain. Thus, to accomplish these goals, the methodologist team, James Rumbaugh, Grady Booch and Ivar Jacobson specified the activities that were involved within each production process phase. Table 37 illustrates the types of activities descriptions for each production process phase:

Table 37: Production Process Phases' Activities during the 1996-2009:

Production Process Phases	Production Process Phases' Activities	Production Process Phases' Activities Specification Description
Inception	***Initial Planning***	• Bases of Estimates
	Planning	• Preliminary Planning
	Requirements	• Requirements elicitation • Requirements analysis and negotiation • Business requirements development • Functional requirements specification • specification • Requirements validation • Systems and technological environments requirements
Evaluation	***Analysis & Design***	• Logical Design • Physical Design • Structure Charts • Software Specification
Construction	***Implementation***	• Hardware Infrastructure Development & Implementation • End Users Training • Software Release
	Testing	• Manual Testing • Automated Testing
Transition	***Deployment***	• Tech-Support • System Change Request
	Evaluation	• Lesson Learn

Source: James Rumbaugh, Grady Booch and Ivar Jacobson (1997)

Thus, the specification of the production phase activities was completed by the mid-90's. After a successful completion of the specifications of the activities for each phase in the production process, computer engineering organizations were able to use this specification as an input to the automated WBS to develop the production planning by using the input to establish the following: assigning production process phases' activities to a given worker, determining production time with its labor rate, and finally to associate them with weekly product progress reports (WPPR). By this time, the *Iterative* framework with its linear and spiral contents was standardized and automated through a production device known as unified modeling language (UML) to be accessible to management and workers remotely (Rumbaugh, Booch and Jacobson, 1998). This advancement led to an interface based production environment for computer engineering organizations, whereby the UML production device was interfacing the MS Project management system, so that team leads and project leads could simply monitor work in progress while a product is still in the production process (Ralf Kneuper, *CMMI*: *Improving Software and Systems Development Process*, 2009). For example, a team lead can simply view the work in progress by logging into the UML production device (without the awareness of the worker) to get a real-time status of the progress (Kneuper, 2009). With this advancement, using the production process phases' activities specification listed above, computer engineering organizations started providing the object oriented project management training to team leads and project leads, which prepared them to utilize the UML system within the context of MS Project management system to promote multitasking (Rumbaugh, Booch and Jacobson, 1998). For example, using the WBS template, a given team leader or a project leader was then able to plan out the WBS for a module. Furthermore, using the WPPR template that can be seen in APPENDIX C, the same team lead or project lead can assign multitasks based on the production process phases' activities specified in table 37 above. Further, multitasking capabilities benefits are reaped when a team lead or a project lead is able to track product issues, supervise action items and mitigate risks using automated templates. All of these improvements have become possible for the macro-matrix management because the *Iterative* framework and the macro-matrix management systems are interfacing and use automated templates. Both are built with open communication to be accessible, remotely, and they included the CMMI best practice process and procedures as foundations. In a similar fashion, simultaneously, the production process phases' activities specification was applied to create the multitasking *Division of Labor* force as part of the production infrastructure.

As the production process phases' activities were used to create multitasking based *Division of Labor* forces, broad based skill sets of the production development methodologies were cultivated (workflow & use case for business requirements development, OOAD for analysis & design, OOP for programming). This meant that it was not only that workers were trained to master requirements production phases' specific activities advanced techniques (workflows & uses cases, as development methodologies that were applied to produce the actual requirements as outputs to be delivered to the analysis & design production phase), but they were also trained to master other production phases' development methods (Rumbaugh, Booch and Jacobson, 1998). For instance, using this approach, computer engineering organizations can create a multitalented business analyst to produce all the 6 requirements of the production process activities listed using the workflow and use case requirements development methodologies (Rumbaugh, Booch and Jacobson, 1998). This approach combines all the activities involved within the requirements' production phase in such ways to be performed by a multitalented requirements analysis, which was not possible before the mid-90's. This was because the *Waterfall* framework was only able to create a *Division of Labor* force that specialized in a single task. The production process was single function based and the production device was also single function based (minicomputer system). However, there was even further advancement in creating multitasking *Division of Labor* forces during the early to mid-2000's, as computer engineering organizations began to implement the Microsoft (MS) SharePoint production environment, which was the main knowledge sharing mechanism for all the production process phases' activities (Kneuper, 2009). This knowledge sharing system was created as a centralized production environment that could be accessed remotely both by the production team and management (Kneuper, 2009). The advantage with regard to the creation of multitalented *Division of Labor* force was that all the production process phases' activities and their specific development methodologies, including best practice processes and procedures, were available as shared knowledge for workers. Hence, trainings were provided with real-time production artifact examples for workers to use when producing (Kneuper, 2009). For instance, a multitalented requirements analyst who was trained to produce the entire requirements production process phases' activities can be trained to become a cross functional worker, who has the capability of producing in multiple production process phases. In this scenario of cross functions, a business analyst can then produce analysis & design related production process phases' activities, including the OOAD production development methodology. As the multitalented worker is capable of producing two production process phases with their activities within the same centralized production environment, the team lead or the project lead is able to monitor the production progress instantly without having to wait for WPPR to be delivered by the worker (Kneuper, 2009). Furthermore, the real-time artifacts being available within the MS SharePoint, the same team lead or project lead is able to develop BOE for another module to formulate the WBS with close to an actual estimate based upon the size and type of the product that needs to be produced (Kneuper, 2009). In a similar scenario, the same multitalented worker can be trained to perform programming within the same multitasking centralized production environments. Hence, with all these automated, multifunctional production processes and devices including a centralized production environment, the demand for creating multitasking *Division of Labor* became more and more crucial because it became advantageous for a worker to multitask, which controlled for labor wage while increasing productivity. According to the Department of Labor's (DOL) 1996-97 record published in *Occupational Outlook Handbook* in 1996-97, computer engineering organizations were increasing the development of automation and focusing on investing in new inventions of multitasking production devices to expedite the process of creating multitasking based *Division of Labor* forces. Table 38 shows DOL's data on computer engineering field's multitasking *Division of Labor* and their roles according to the *Iterative* production process phases' activities implemented during the 1996-97 years.

Table 38: Specialized *Division of Labor* force 1996-97:

Production Process Phases	Positions 1996-97	Roles & Activities' Description 1996-97
1: *Inception*	**828, 000 *Computer Systems Analysis***	
1.1	***Computer Scientists***	Produce new computer concepts as solution given to them by Systems Analysts and Computer Engineers.
1.2	***Computer Engineers***	Produce hardware designs, architectural specification, chips and test scripts for hardware and software automation.
1.3	***Systems Analysts***	Produce designs, structure charts and code specifications and test scripts for software.
1.4	***Database Administrators***	Produce data software architects, design, test data storage software system.
1.5	***Computer Support Analysts***	Produce user's manuals, analyze problems and provide tech-support hardware, software and systems.
2: *Construction*	**537,000 *Computer Programming***	
2.1	***Application Programmers***	Produce application designs, architects, codes & configuration specifications, test scripts for custom software including for package software systems.
2.2	***Systems Programmers***	Produce hardware components integration codes and configurations test scripts to ensure communication among infrastructures and software applications are functioning, accordingly.
2.3	***Programmer Analysts***	Produce codes, test custom software including configuring package software systems.
3: *Deployment*	**259,000 *Computer & Peripheral Equipment Operators***	
3.1	***Computer Operators***	Multitasking roles & tasks description remained unchanged.
4: *Maintenance*	**134,000 *Computer and Office Machine Repairers***	
4.1	***Computer Service Technicians***	Multitasking roles & tasks description remained unchanged.
4.2	***Computer and Office Machine Repairers***	Multitasking roles & tasks description remained unchanged.
Total	**11 specializations were implemented**	**1,758,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1996-97c

With the goal of investing in new production devices for the purpose of expediting the creation of multitasking based of *Division of Labor* force for the computer engineering organizations' production process, by 1996-97, computer engineering organizations were beginning to incorporate some of the basic multifunctional *Division of Labor* roles to pave the way for the implementation of the *Iterative* production process phases' activities. As table 38 shows above, DOL reported 14 computer engineering fields with roles and activities within the 4 the production process phases. They adopted inception and construction production phases from the *Iterative* production process framework and kept deployment and maintenance production phases from the *Waterfall* production process framework. The reason was that computer engineering organizations during the mid-90's had an initiative to invent new computer systems for production processes, so they needed the *Inception* production phase to create a multitasking position called *Computer Scientists* (that could be comprised of effective visionaries who create new concepts) and to create position called *Computer Engineers* (that could develop and implement new concepts created by the *Computer Scientist*). Thus, within the first production process phase, computer engineering organizations created 3 new positions called *Computer Scientists*, *Computer Engineers*, *Database Administrators* and *Computer Support Analysts* and kept *Systems Analysts* from the year 1994-95. This increased the size of the *Division of Labor* in 1996-97 to 828,000 from 666,000 of the 1994-95 by 162 (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1996-97c). The increase in the size of this *Division of Labor* was critical as computer engineering organizations were striving to find a better way to promote efficiency in production process. As computer engineering organizations were increasing the number of workers to decrease and/or keep the wage constant from the 70's through the mid-90's to gain surplus labor, they began to seek an alternative solution that could promote efficiency without increasing the size of the *Division of Labor* forces. This could be done by taking advantage of the emergence of internet communication, automation and multitasking computer systems. Hence, the search for a change became inevitable, and computer engineering organizations sought to take advantage of newly invented multitasking based minicomputer systems including the internet as production devices to change their operational strategies in production process. Hence, by 1996-97, they created a position called *Computer Scientists* to invent new computer systems that could become a production device. As recorded in DOL's 1996-97 report:

> Computer Scientists generally design computers and conduct research to improve their design or use, and develop and adapt principles for applying computers to new uses. Computer scientists perform many of the same duties as other computer professionals throughout a normal workday, but their jobs are distinguished by the higher level of theoretical expertise and innovation they apply to complex problems and the creation or application of new technology. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1996-97c

With this demand, *Computer Scientists* introduced a concept called ".com" (dot-com) using the automated and multifunctional computer systems including the internet and the UML production environment (Rumbaugh, Booch and Jacobson, 1998). ".com" was invented to develop computer systems that integrated software applications with the internet that could be accessed by users regardless of their locations (Rumbaugh, Booch and Jacobson, 1998). With this advancement, computer engineering organizations successfully implemented a mobile production environment in which the workers simply login regardless of their location to produce their daily production activities, enabling computer engineering organizations to decrease their overhead cost as the production environment became mobile. Hence, to implement this newly invented computer systems developed by the *Computer Scientists*, the standard best practice in the production process became viable as the *Computer Scientists* invented the ideas and completed the proofs of concepts, and the *Computer Engineers* took over the development and deployment aspects of the newly invented computer systems. DOL's record describes the *Computer Engineers* position, stating, "*Computer Engineers* work with the hardware and software aspects of systems design and development. *Computer Engineers* may often work as part of a team that designs new computing devices or computer-related equipment, (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 93, 1996-97c)". Thus, both the *Computer Scientists* and *Computer Engineers* which were primarily to invent new production devices that could be used to enhance the efforts of creating more multitasking based *Division of Labor* forces, so that the ultimate goal of controlling labor wages by assigning multiple production activities to a worker could be achieved. Based on the DOL's 1996-97 data, the indication is that the demand for the *Computer Scientists* and *Computer Engineers* would increase throughout the course of the late 90's and 2000's as computer engineering organizations would need them to produce new computer systems that could improve productivity within their own production processes environment and create new industry specific computer systems. DOL's labor report on computer engineering organizations states:

> The demand for Computer Scientists and Computer Engineers is expected to rise as organizations attempt to maximize efficiency of their computer systems. There will continue to be a need for increasingly sophisticated technological innovation. Competition will place organizations under growing pressure to use technological advances in areas such as office and factory automation, telecommunications technology, and scientific research. As the complexity of these applications grows, more computer scientists and systems analysts will be needed to design, develop, and implement the new technology. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 95, 1996-97c

In addition, during the same time-frame in 1996-97, computer engineering organizations redefined the position and roles of *Systems Analyst* by combining it with the *Business Systems Analyst*. To make this merging of positions a success, computer engineering organizations trained *Systems Analysts* to master the activities and methods of business requirements development, like workflows and use cases, to analyze and evaluate business needs and problems from a business analysis perspective as applied to different industries (Rumbaugh, Booch and Jacobson, 1998). This aspect of *Systems Analysts'* ability to perform production process activities was reported to DOL's labor description in 1996-97 as following:

> Far more numerous, systems analysts use their knowledge and skills in a problem solving capacity, implementing the means for computer technology to meet the individual needs of an organization. They study business, scientific, or engineering data processing problems and design new solutions using computers. This process may include planning and developing new computer systems or devising ways to apply existing systems to operations still completed manually or by some less efficient method. Systems analysts may design entirely new systems, including both hardware and software, or add a single new software application to harness more of the computer's power. They work to help an organization realize the maximum benefit from its investment in equipment, personnel, and business processes. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 93, 1996-97c

This became advantageous for computer engineering organizations as they could now assign multiple tasks to a given *Systems Analyst* to perform broad based computer systems development analysis and evaluation for different industries to present the findings to the *Computer Scientists*, enabling them to create new, marketable solutions. With this initiative, computer engineering organizations were successful in utilizing the multitasking production process, which was an indication of its success. This promoted an ability to produce more, given the *Systems Analysts'* technical and analytical skill sets and their use in production process. Thus, computer engineering organizations effectively used the multitasking roles of the *Systems Analysts* to analyze and evaluate the business problems of different industries for two important advantages: 1) to get the final outputs of the *Systems Analysts* and deliver them to the *Computer Scientists* as inputs for devising new computer systems concepts, 2) to use the *Systems Analysts'* outputs as coding specification inputs for *Programmers* to develop industry specific computer systems. Thus, in this scenario, once a module is completed, its database structure has to be constructed. To do this activity, computer engineering organizations created a position called *Database Administrator*. The *Database Administrator* is responsible for building the database structure for every module until all the modules are developed and integrated to make a complete, sellable computer system. The roles of the *Database Administrator* were reported to DOL's in 1996-97:

> Database Administrators works with database management systems software. They reorganize and restructure data to better suit the needs of users. They also may be responsible for maintaining the efficiency of the database, system security and may aid in design implementation. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 93, 1996-97c

Hence, a given that *Database Administrator* can produce the entire database structure by her/his self-using UML compatible database development tools, some of which are commonly used as production devices (e.g., Erwin (data modeling tool), oracle database developer, object-oriented database developer and power database builder), using these production devices, computer engineering organizations can assign production process activities to multitask a *Database Administrator*. Thus, the increase in the size of the multitasking based *Division of Labor* forces of the *Computer Scientists, Computer Engineers, Systems Analysts, Database Administrators* and *Computer Support Analysts* within the *Inceptions* production phase grew from 666,000 reported in 1994-95 to 828,000 in 1996-97, which was an increase of 162,000, proving that computer engineering organizations were striving to invent new computer systems, such as the ".com," using internet communication as a foundation, along with the automated multifunctional minicomputer systems to promote efficiency in their production process. As this was the case for the inception production phase, according to DOL's 1996-97 computer engineering organizations' labor report, the *Programmers* number indicated a decreased to 537,000 in 1996-97 from 555,000 of the 1994-95 report, which shows a difference of 18,000, suggesting that the there was no high demand. That the decrease was not high suggests that there was uncertainty as what roles the *Application Programmers, Systems Programmers* or *Programmer Analysts* were to play in the production process.

The demand for the mass production of the newly invented .com web-based application for the different industries in the market did not take place until about the late 90's. Although there was a slight decrease in numbers, DOL's 1996-97 labor report indicated that *Programmers* were still a significant part of the multitasking *Division of Labor* force using the simplified OOP development methodology to improve production. Thus, during 1996-97, computer engineering organizations adopted the *Iterative* production framework to create multitasking *Division of Labor* forces as *Application Programmers*, *Systems Programmers and Programmer Analysts* to develop industry specific web-based applications for the market (Rumbaugh, Booch and Jacobson, 1998). As analyzed above, the spiral aspect of the *Iterative* production framework created development methodologies for each production phase. Thus, a spiral based object-oriented programming (OOP) was created that works directly with object oriented analysis and design (OOAD) methodology. Hence, using this common denominator (computer systems development methodology) the outputs from the systems analysis phase activities are used directly in the coding environment without any programming specification conversion, as both programming and systems analysis have the same spiral based object oriented as a foundation. Furthermore, these object-oriented development methodologies were integrated into the automated UML production device, in which the work product artifacts were accessible for both the *Systems Analysts* and the *Programmers* (Rumbaugh, Booch and Jacobson, 1998). According to DOL's 1996-97 report, object-oriented programming development methodologies were going to be highly used by computer engineering organizations:

> Object-oriented languages will increasingly be used in the years ahead, further enhancing the productivity of programmers. Programmers will be creating and maintaining expert systems and embedding these technologies in more and more products. Bureau of Labor Statistics *Occupational Outlook Handbook*, 223, 1996-97c

With this availability, computer engineering organizations began to multitask an *Application Programmer, Systems Programmer* and *Programmer Analyst* to develop the web-based applications in an efficient manner without wasting production time. According to DOL's report on the labor description of the *Programmers*, their demand would increase towards the later 90's to produce the industry specific web-based applications as the market demand increased.

The computer engineering organizations kept the 3rd production phase (deployment) with its *Computer and Peripheral Equipment Operators* specialized *Division of Labor* force and the 4th production process phase (maintenance) of the *Waterfall*'s production framework during 1996-97. With the 4th production process phase (maintenance), computer engineering organizations eliminated *Console Operators, High-Speed Printer & Card-Tape-Convertor Operators Converter Operators* and *Tape Librarians* positions and created a position called *Computer Operators* with its production process activity. The maintenance production phase's *Computer and Office Machine Repairers* as a specialized *Division of Labor* force was also retained during the same time frame. The three fundamental reasons for keeping these phases comprised of the exact same specialized *Division of Labor* forces were: 1) to continue to support existing single functional minicomputer systems and appliances (DOL reports the future fate of these specialized *Division of Labor* forces as following, "In the computer centers that lack this level of automation, some computer operators still may be responsible for the tasks traditionally done by peripheral equipment operators," (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 262, 1996-97c). 2) As the newly invented .com web-based computer systems in the form of software applications was at the stages of being formulated, computer engineering organizations did not need to create new multitasking based *Division of Labor* for its production environment. 3) The newly invented industry specific web-based was not yet developed enough for the market, so computer engineering organizations were not required to create a new multitasking based *Division of Labor* to support the 3rd and the 4th production process phases. Thus, according to DOL's 1996-97 labor report, the number of *Computer Operators* was slightly reduced to 259,000 from the 266,000 of the 1994-95 record. Similarly, during the same year, *Computer and Office Machine Repairers* showed a slight decrease to 141,000 from 143,000 from DOL's 1994-95 labor report. As the single functional computer systems were being replaced by multifunctional computer systems and the emergence of internet based web computer systems, these specialized roles were not needed by the computer engineering organizations in production process. Furthermore, as the automation and multitasking computer systems were implemented as production devices, the need for these specialized based *Division of Labor* forces declined. As the implementation of the *Iterative* production process framework replaced by the specialized based *Division of Labor* force, it became a practical decision for computer engineering organizations to reduce the number of specialized *Division of Labor*, such as the *Computer and Peripheral Equipment Operators* and the *Computer and Office Machine Repairers*.

DOL's prediction is that these specialized *Division of Labor* forces would have to change their skill sets with advance training and education on newly emerging computer systems to be part of the multitasking based *Division of Labor* force. DOL's 1996-97 report projected:

> As technology advances, many computer operators will essentially monitor an automated system. As the role of operators changes due to new technology, their responsibilities many shift to system security, troubleshooting, desk help, network problems, and maintaining large databases. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 223, 1996-97c

Hence, although the number of specialized *Division of Labor*, such as the *Computer and Peripheral Equipment Operators* and the *Computer and Office Machine Repairers*, did not have a significant increase throughout the late 90's due to automation and systems changes that were taking during that period, computer engineering organizations continued to keep these specialized *Division of Labor* forces by providing on-job training, and some of them were forced to change careers, as their tasks in production process were replaced by automated computer systems (Rumbaugh, Booch and Jacobson, 1998). However, based on the DOL 1998-99 labor report shown below on table 39, computer engineering organizations increased the number of *Computer Scientists* and *Computer Engineers* for the purpose of maximizing the number of new computer system inventions, and the *Systems Analysts*, *Database Administrators* and *Computer Supports Analysts* were also increased to implement industry specific web-based computer systems. This trend continued throughout the course of the late 90's and the early 2000's.

Table 39: Specialized *Division of Labor* force 1998-99:

Production Process Phases	Multitasking Positions 1998-99	Roles & Tasks Description 1998-99
1: Inception	**933, 000 *Computer Systems Analysis***	
1.1	***Computer Scientists***	Multitasking roles & tasks description remained unchanged.
1.2	***Computer Engineers***	Multitasking roles & tasks description remained unchanged.
1.3	***Systems Analysts***	Multitasking roles & tasks description remained unchanged.
1.4	***Database Administrators***	Multitasking roles & tasks description remained unchanged.
1.5	***Computer Support Analysts***	Multitasking roles & tasks description remained unchanged.
2: Construction	**568,000 *Computer Programming***	
2.1	***Application Programmers***	Multitasking roles & tasks description remained unchanged.
2.2	***Systems Programmers***	Multitasking roles & tasks description remained unchanged.
2.3	***Programmer Analysts***	Multitasking roles & tasks description remained unchanged.
3: Deployment	**259,000 *Computer and Peripheral Equipment Operators***	
3.1	***Computer Operators***	Multitasking roles & tasks description remained unchanged.
4: Maintenance	**141,000 *Computer and Office Machine Repairers***	
4.1	***Computer Service Technicians***	Multitasking roles & tasks description remained unchanged.
4.2	***Computer and office machine repairers***	Multitasking roles & tasks description remained unchanged.
Total	**11 specializations were implemented**	**1,901,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1998-99c

As the demand for the new ".com" web-based computer system increased for the computer engineering organizations' production devices both to revamp their production environment and for the industry specific web-based applications for the market, computer engineering organizations increased the number of *Computer Scientists and Computer Engineers* to continue generating new computer systems inventions. The number of *Systems Analysts*, *Database Administrators* and *Computer Support Analysts* was also increased to develop the industry specific web-based computer systems. During 1998-99, as the production process was already established in a form of multifunctional based *Division of Labor* production process, computer engineering organizations started searching for highly educated professionals who held backgrounds in diversified curriculums in the field of computer science to reinforce the creation of a multitalented workforce. With this initiative, the basic academic bachelor degree in the computer science field became a standard prerequisite for computer engineering organizations, and they searched for applicant who met this criterion. Regarding the academic standard for the computation field, DOL's 1998-99 report stated:

> A bachelor's degree is virtually a prerequisite for most employers relevant work experience also is very important. For some of the more complex jobs, persons with graduate degrees are preferred. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 109, 1998-99c

With a bachelor degree as the minimum standard criteria, in 1998-99, the number of *Computer Scientists, Computer Engineers, Systems Analysts*, *Database Administrators* and *Computer Support Analysts* increased to 933,000 from 828,000 of 1996-97, which was an increase of 105,000 (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1998-99c). While a bachelor's degree was the minimum prerequisite criteria for *Computer Engineering, Systems Analysts, Database Administrators* and *Computer Support Analysts*, a Ph.D., or at least a master's degree was a minimum prerequisite for the *Computer Scientists* (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1998-99c). According to DOL's 1998-99 labor report in this regard:

> *Computer hardware engineers* generally require a bachelor's degree in computer engineering or electrical engineering, whereas software engineers are more likely to need a degree in computer science. For *Systems Analyst* or even *Database Administrator* positions, many employers seek applicants who have a bachelor's degree in computer science, information science, computer information systems, or data processing. *Computer support specialists* many also need a bachelor's degree in a computer-related field, as well as significant experience working with computers, including programming skills. Generally, a Ph.D., or at least a master's degree in computer science or engineering, is required for *Computer Scientist* jobs in research laboratories or academic

> institutions. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 111, 1998-99c

With academic standards, the *Computer Scientists* began to participate in academic institutions to create curricula in addition to working within the computer engineering organizations in inventing new computer systems as devices for production environments. The advantage of involving *Computer Scientists* in the academic institutions was to invent new programming techniques and computer systems, which were used to establish curricula. These curricula were implemented to improve educational standards and to create diversified core courses that exposed students to materials necessary to become experts in multiple fields of computer science in which they were prepared to become part of the multitasking *Division of Labor* forces that the computer engineering organizations could use in their production process. DOL described the *Computer Scientists* in 1998-99:

> *Computer Scientists* can work as theorists, researchers, or inventors. Those employed by academic institutions work in areas ranging from complexity theory, to hardware, to programming Language design. Some work on multi-discipline projects, such as developing and advancing uses of virtual reality in robotics. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 109, 1998-99c

Thus, as the *Computer Scientists* continued to invent new computer systems concepts both for the academic institution and for computer engineering organizations' production environment to reinforce the creation of a multitalented workforce, computer engineering organizations sought to redefine a new employment process. This new employment process was created for the new highly educated professionals to be hired as temp-contractors in order to eliminate full-time employment. This change in employment process was already influenced by the growing demand of implementing automated based multifunctional computer systems coupled with the newly invented ".com" web-based computer systems that replaced the old manual production devices and single functional production process. Hence, as most of the manual operations were automated and systems were web enabled and multifunctional, computer engineering organizations no longer kept the permanent full-time employees since the new computer systems were implemented, because these devices did not require ongoing manual maintenance. DOL's 1998-99 employment report states:

> A growing number of computer professionals are employed on a temporary or contract basis-many of whom are self-employed consultants. For example, a company installing a new computer system may need the services of several Systems Analysts just to get the system running. Because not all of them would be needed once the system is functioning, the company might contract directly with the systems analysts themselves or with a temporary help agency or consulting firm. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 111, 1998-99c

This in turn was also used for knowledge transfer purposes to acclimate key permanently employed *Systems Analysts, Database Administrators* and *Systems Support Analysts*, so that in-house skills would be supplemented. In this manner, new computer systems concepts invented by *Computer Scientists* were applied to establish a high standard educational prerequisite that was in existence during1998-99, and this was effective in creating educated *Systems Analysts, Database Administrators* or *Systems Support Analysts* professionals that computer engineering organizations used in their production process to promote efficiency. Similarly, from a computer programming perspective, a bachelor's degree was the minimum standard as a prerequisite for *Programmers* (Department of Labor: *Occupational Outlook Handbook*, 1998-99). As shown in table 39, during 1998-99, the number of *Programmers* increased to 568,000 from 537,000, showing an increase of 31,000. This increase was due to the growing demand for the implementation of industry based ".com" web-based computer systems that required programmers to program the system. However, the number of *Computer and Peripheral Equipment Operators* remained the same at 259,000 in 1998-99 as it was during 1996-97. This is an indication of this profession not being in demand by the computer engineering organizations' production process, as most of the manual production devices were being automated. This is an indication that this specialized *Division of Labor* that was needed to maintain the remaining single function based minicomputer systems up until 1998-99 was no longer part of the production process as the ".com" web-based. Automation and multifunctional computer systems did not need ongoing maintenance activities on a daily basis as had previously been done by specialized *Console Operators*. DOL reported this change during 1998-99:

> As organizations continue to look for opportunities to increase productivity, automation is expanding into more areas of computer operations. Sophisticated software coupled with robotics; enable the computer to perform many routine tasks formerly done by computer operators. Scheduling, loading and downloading programs, mounting without the intervention of an operator. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 276, 1998-99c

This change extirpated the labor cost incurred by computer engineering organizations for the deployment production phase. As a result, the 3rd production phase, which was kept from the *Waterfall* production process phase, was replaced by the *Iterative* production framework's transition production process phase. In sum, from a production process perspective, the late 90's was spent implementing the *Iterative* production process framework and its linear and spiral development methods, which came with the automated unified modeling language (UML) as a production device that became an interface with the automated MS Project management system. As a result of these establishments, the multitasking based *Division of Labor* forces were created, and the spiral development methodology made module based development possible. With these advancements, *Computer Scientists* invented ".com" web-based computer systems for computer engineering organizations' production environments and for industry specific products as the market demanded. These inventions were used to establish academic curricula to standardize prerequisites for employment beginning in the period of 1998-99. This trend of invention has continued throughout the 21st century. 2000-01 was a period for invention and refinement of the computer hardware platform architectures known as the 3 tier architecture (Web-browser, application server and database server). The analysis of DOL's data recorded in table 40 below demonstrates the continuous creation of the multitasking *Division of Labor* forces at the beginning of the 21st century.

Table 40: Specialized *Division of Labor* force 2000-01:

Production Process Phases	Multitasking Positions 2000-01	Roles & Tasks Description 2000-01
1: Inception	**1.5 Million *Systems Analysts, Computer Scientists & Database Administrators***	
1.1	***Systems Analysts***	Produce systems' problem specification, designs, structure charts and code specifications and test scripts for software, hardware, and networking and communication systems.
1.2	***Computer Engineers***	Produce hardware designs, architectural specification, chips and test scripts for hardware and software automation.
1.3	***Computer Scientists***	Produce new computer concepts as hardware and software problems given to them by Systems Analysts and Computer Engineers.
1.4	***Database Administrators***	Produce data software architects, design, test data storage software systems including database infrastructures configuration.
1.5	***Computer Support Analysts***	Multitasking Roles & tasks description remained unchanged
2: Evaluation	**648,000 *Computer Programmers***	
2.1	***Application Programmers***	Produce application designs, architects, codes & configuration specifications, test scripts for custom software including for package software systems.
2.2	***Systems Programmers***	Produce hardware components integration codes and configurations test scripts to ensure communication among infrastructures and software applications are functioning, accordingly.
2.3	***Programmer Analysts***	Produce codes, test custom software including configuring package software systems.
3: Construction	**141,000 *Computer and Office Machine Repairers***	
3.1	***Computer Technicians***	Multitasking Roles & tasks description remained unchanged.
3.2	***Computer and Office Machine Repairers***	Multitasking Roles & tasks description remained unchanged.
Total	**10 specializations were implemented**	**2,289,000 were employed.**

Source: Department of Labor: *Occupational Outlook Handbook*, 2000-01c

As analyzed above, computer engineering organizations took advantage of the *Computer Scientists* as inventors of the new ".com" web-based computer systems to improve production devices to be used for production environments, for industry specific products in the market and for academic purposes to create multitasking *Division of Labor* forces. As the *Computer Scientists* were able to invent the concepts of web-based computer application systems using *Iterative*'s spiral development methodology created for modules, inventing an architectural platform methodology to host the web-based computer application systems became urgent. The goal was to replace the 2 tier architect of *Waterfall* production devices, which was limited in hosting, performance, maintenance and security for the newly implemented web-based applications both for the production devices for the production environments and for the industry specific computer systems in the market. The *Waterfall*'s 2 tier computer systems architectural platform was created to serve the single functional computer systems where the graphic user interface (GUI) and the programming codes were combined within one single server environment (known as a client), and the database log was at the backend environment (known as a server) (Lars Mathiassen, *Improving Software Organizations*, 2002). Figure 11 below illustrates the 2 tier architectural platform:

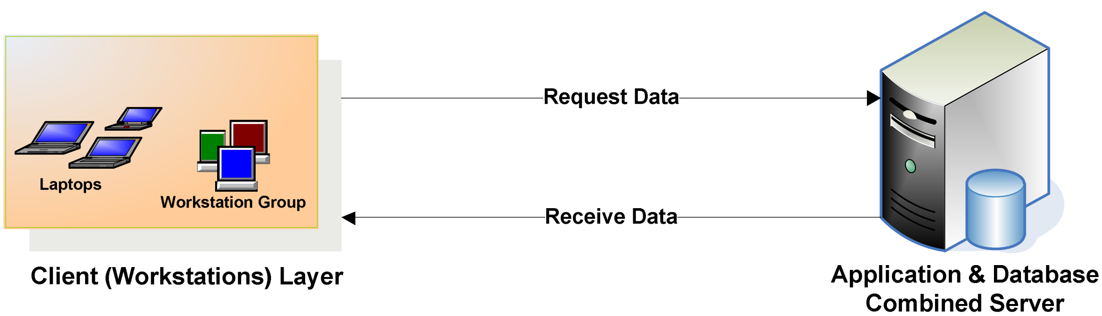


Figure 11: 2 Tier Computer Systems Architectural Platform

Hence, during the late 90's as the web-enabled computer application systems were implemented, computer engineering organizations began to learn that the 2 tier systems architectural platform could not serve the ".com" web-based computer application systems. The new web-enabled production devices had transformed the single production variety to multitasking production devices and further revolutionized the single facility based production environment into open production environments with remote accessibility (Mathiassen, 2002). With this transformation, as the new multitasking *Division of Labor* forces began to remotely access the production environment to perform their activities, the 2 tier architectural platform was not able to provide sufficient services to meet the high volume of access requested. This performance issue became a major bottle-neck, making the newly implemented web-based computer application systems insufficient by reducing the speed of production process (Mathiassen, 2002). Hence, it became obvious that there was an incompatibility between the web-based computer application systems and the 2 tier infrastructural platform forcing computer engineering organizations to replace the latter to improve efficiency (Mathiassen, 2002). Thus, the web-based computer application systems first were innovated by the *Computer Scientists* and tested by *Computer Engineers* using the spiral module based development methodology that provided the capability of developing each module and continuing until all the modules that would make up the complete system had been finally integrated. Similarly, using this spiral module based development methodology, by the early 2000's, *Computer Scientists* introduce a 3 tier architectural platform to replace its 2 tier counterpart (Mathiassen, 2002). With this invention as a need, by 2000-01, the demand rose for *Computer Scientists* to invent, for the *Computer Engineers* to test new concepts, for *Systems Analysts* to evaluate computer systems, for the *Database Administrator* to maintain data and for the *Systems Support Analysts* to serve as subject matter experts (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2000-01c). As DOL's 2000-01 data above in table 40 indicates, the number of *Computer Scientists, Computer Engineers, Systems Analysts, Database*

Administrators and *Systems Support Analysts* had increased to 1.5 million from the 933,000 reported in 1998-99, which was a staggering jump of 567,000. The *Computer Scientists* were highly in demand during this critical period to create the concept of a 3 tier architectural platform. The basic reasoning for the need was that the computer hardware systems' architectural platform must be in sync and compatible with the computer application systems to work effectively in accomplishing the goal of computer engineering organizations' effort to promote efficiency in production process (Mathiassen, 2002). Hence, using the same spiral development methodology that was a module based development approach, each production process phase's activities maintained their own methods that could be applied independently to each production phase, such as object oriented analysis (OOA) for analysis, object oriented design (OOD) for design phase, and object oriented programming (OOP) for programming phase. Hence, this approach was used to build the ".com" web-based computer application systems as it enabled workers to build module based systems that could be released for use as a production device for the production environments and for the industry specific market. In a similar fashion, *Computer Scientists* used the spiral development methodology as a foundation to invent the 3 tier computer hardware systems' architectural platform, so that the web-based computer application systems could be synchronized to increase the speed of production as the 3 tier computer hardware systems' architectural platform increased the speed of the overall systems' performance (Mathiassen, 2002). Unlike the *Waterfall*'s 2 tier infrastructural platform illustrated in Figure 9, the *Iterative* production framework's 3 tier computer hardware systems' infrastructural platform had three independent computer machines in which each had its own development methodology defined in the following respective ways: The 1^{st} layer of the 3 tier architectural platform is a web server known as a presentation (front end) server, which is created to be responsible for communication with the users and the systems. In the 2 tier architectural platform, this functionality was combined with the programming log of the computer application systems in the same environment, causing a major delay in the speed of the overall systems at a time where there was a high volume of users attempting to access the web-based application system (Mathiassen, 2002). Hence, to overcome this issue of performance, the *Computer Scientists* created the 1^{st} layer of the 3 tier infrastructural platform as a web server layer platform for the purpose of improving the speed of systems by enabling the users to communicate with the system through the login screen. Figure 12 below illustrates the 3 tier computer hardware architectural system:

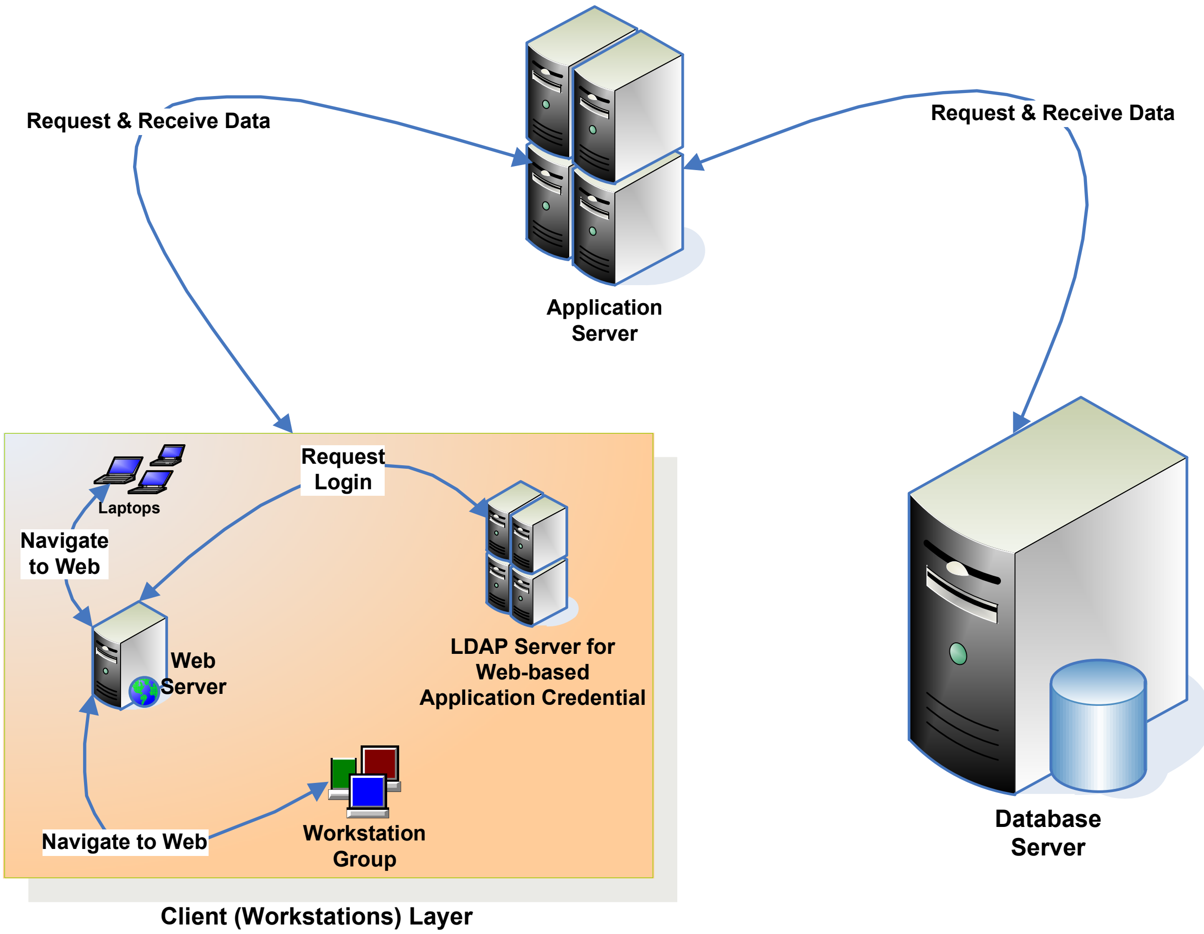


Figure 12: 3 Tier Computer Systems Architectural Platform

This approach separated the 1st layer web server from the 2nd layer where the programming codes are kept (Martin Fowler, *Pattern of Enterprise Architecture*, 2003b). For this reason, the *Computer Scientists* created the 2nd layer known as the application server within the 3rd tier architecture platform where the programming codes reside. The purpose of creating the 2nd layer within the 3 tier architectural platform was to fully implement the spiral development methodology of a module based approach, which enables computer engineering organizations to continue to produce modules and release them while the system is still running to prevent the loss of production hours, which included the following issues: 1) Losing production hours was the major limitation of the 2 tier architectural platform, as it was not able to allow modules to run while new upgrades to the system were performed (Fowler, 2003b). 2) Module based testing could be performed while other modules were running because it was not capable of running modules and testing others simultaneously (Fowler, 2003b). This limitation was costly as the test results for the entire system came up at the end of the system and the labor hours for debugging was a major loss for computer engineering organizations. 3) As the programming codes environment and the web server of the front end login server were combined in the same environment, the environment caused a major performance issue in delaying the production process (Steve McConnell, *Code Complete*, 2004b). Hence, as a solution to these issues with the 2 tier computer hardware architectural platform, *Computer Scientists* invented the 2^{nd} layer called the application tier with the 3 tier architectural platform in which computer engineering organizations would not lose production hours, and the new architectural platform was improving the speed of production (Steve McConnell, *Software Estimation: Defining the Black Art*, 2006c). The 3^{rd} layer they created was known as the database server within the 3 tier architectural platform. The purpose of this layer was to enhance the data processing capability of the web-based computer application systems (Fowler, 2003b). In this way, the 1^{st}, 2^{nd} and 3^{rd} layers have their own module based hardware machines, which enable them to be developed independently, until all the modules are developed, and then are finally the 3 layers are integrated to make up the complete 3 tier computer hardware platform to run the web-based computer application systems and improve the speed of production hours (Fowler, 2003b). As the *Computer Scientists* and *Computer Engineers* completed the proof of concept on the 3 tier computer hardware architectural platform during the early 2000's, the *Systems Analysts* and *Database Administrator* professionals became in demand for analyzing, developing and implementing the newly created 3 tier computer hardware architectural platform for industry specific market products throughout the course of this time period, which increased the number of *Systems Analysts* to 1.5 million in 2000-01 from the 933,000 of the 1998-99 report to make for a dramatic increase of 567,000. With this progress, computer engineering organizations increased the number of *Computer Programmers* to 648,000 in 2000-01 from 568,000 of the 1998-99 year report, which was an increase of 80,000. However, during 2000-01, they eliminated the 3^{rd} production process phase, which was the deployment phase with the specialized *Computer and Peripheral Equipment Operators*. This elimination resulted from the replacement of the 2 tier computer hardware platform by the new 3 tier computer hardware platform. Web-based computer application systems that automated the manual based single functional computer systems, coupling with the multifunctional workstation, also contributed the deployment phase's disappearance. With this, computer engineering organizations completely removed the 3^{rd} production process phase, originating from *Waterfall*'s production framework, and transitioned to the *Iterative* framework production process. As a result, specialized *Computer and Peripheral Equipment Operators*' job was obsolete and was never recovered. For the same reasons, although not eliminated during 2000-01, the number of *Computer and Office Machine Repairers* did not show any increase. The number of specialized *Computer and Office Machine Repairers* remained the same at 141,000 in 2000-01 as it was in 1998-99. In sum, the use of *Computer Scientists* to invent new computer systems concepts that started in the late 90's continued onto the early 2000's with the invention of the 3 tier computer hardware architectural platform, which revamped the interests of the computer engineering organizations to promote efficiency. This enabled them to effectively improve the speed of production by improving performance to save production's labor hours, in that workers are able to remotely access production devices in the open production environments with no delays. According to the DOL labor report in table 41 below, this trend continued throughout the course of 2002-03.

Table 41: Specialized *Division of Labor* force 2002-03:

Production Process Phases	**Multitasking Positions 2002-03**	**Roles & Tasks Description 2002-03**
1: Inception	**887,000 *Computer Systems Analysts, Engineers & Scientists***	
1.1	***Systems Analysts***	Multitasking Roles & tasks description remained unchanged.
1.2	***Network Systems & Data Communications Analysts***	Produce design architects of local area networks (LAN), wide area networks (WAN), internet, and intranets test scripts to ensure data is transmitted, accordingly.
1.3	***Database Administrators***	Multitasking Roles & tasks description remained unchanged.
1.4	***Computer Scientists***	Multitasking Roles & tasks description remained unchanged.
2: Evaluation	**585,000 *Computer Programmers***	
2.1	***Application Programmers***	Produce codes, test custom software including configuring package software systems.
2.2	***Programmer Analysts***	Produce codes, test custom software including configuring package software systems.
3: Construction	**697,000 *Computer Software Engineers***	
3.1	***Computer Applications Software Engineers***	Produce designs, software architects, codes specification, generate programming codes and test scripts for industry specific packaged systems and custom applications.
3.2	***Computer Systems Software Engineers***	Produce designs hardware architects, configure hardware systems' software and test scripts for organizations' overall computer infrastructures including networking and intranets.
4: Transition	**60,000 *Computer Hardware Engineers***	
4.1	***Computer Hardware Engineers***	Produce designs hardware architects, develop and test. In addition, Computer Hardware Engineer design and develop the software systems that control the hardware systems.
5: Go-Live Support	**758,000 *Computer Support Specialists & Systems Administrators***	
5.1	***Computer support Specialists***	Produce systems' problem specification, designs, structure charts and code specifications and test scripts for software, hardware, and networking and communication systems.
5.2	***Communication Systems Administrators***	Produced systems communications related technical procedures and processes.
Total	**11 specializations were implemented**	**3,014,000**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2002-03c

As the DOL's data of 2002-2003 shows above in table 41, computer engineering organizations entered the phase of fully implementing the 3 tier computer hardware systems' architectural platform. This direction can be interpreted based on the data of DOL that indicated a decrease of production process phase 1, which is the inception phase. Hence, during DOL's 2002-03 labor report, the number of *Computer Systems Analysts, Engineers & Scientists* decreased to 887,000 from the 1.5 million of the 2000-01 report, which was a decreased of 149,113. This decline suggests that there was a shift in the direction of computer engineering organizations as the invention phase of the 3 tier was completed; they were going to get into the mass production of computer hardware systems to improve the speed of production. Hence, as analyzed above, the invention of the 3 tier computer hardware architectural platform was completed by the early 2000's and the challenge for computer engineering organizations became the development and implantation of the computer hardware systems for the 3 tier architectural platform. Thus, as the DOL 2002-03 data suggests, to accomplish this goal, by 2002-03, computer engineering organization rearranged the categories of the multitasking *Division of Labor* forces into the different production process phases, implementing two of the remaining *Iterative* production process phases, which were the construction and the transition production phases and finally incorporated an additional production process phase known as go-live support.

From 1996-01, they only implemented two of the *Iterative* production process frameworks, which were the inception production process phase for the first and the evaluation production process phase for the second. As DOL labor data suggests, computer engineering organizations first rearranged the inception phase in the following order: 1) They put the *Systems Analysts* first in 2002-03 (where they had been at third step in the production process from 1996-97 to 2000-01) to handle industry specific system analysis and identify problem scenarios that they delivered as outputs to the *Computer Scientists*, so that new solutions could be invented. However, as the primary goal of the computer engineering organizations shifted to development and implementation of the mass production of computer hardware systems for the implementation of the 3 tier architectural platform to ensure that the speed of production process of the web-based computer application systems improves, *Systems Analysts*' role was placed first in the production process to continue to produce analysis, design and programming specification as outputs for *Computer Programmers*. And most importantly, the 1st layer of the 3 tier architectural platform requires the *Systems Analysts* to analyze the web interface process, design a prototype and provide a coding specification as their final outputs to be delivered to the *Computer Programmer* to produce the web interface screens. 2) Computer engineering organizations created a new multitasking *Division of Labor* force called *Network Systems & Data Communications Analysts*, as the 3 tier computer hardware architectural platform requires internet, local and wide area network connections to integrate the 3 independent computer hardware systems to communicate, accordingly. 3) The *Database Administrators* remains in the third row of the production process to configure and maintain the database logics. And most importantly, their expertise was required for the 3rd layer of the 3 tier architectural platform, which is the database server. 4) During 1996-01, the *Computer Scientists* were on the first production row to create new computer systems concepts; however, after inventing the web-based computer application systems during the late 90's and they invented the 3 tier in the early 2000's, computer engineering organizations moved them towards the last production row of the inception production process phase. This was because the primary objective of the computer engineering organizations became to mass produce the already invented computer systems and new invention was no longer a primary objective to them. As this was the restructuring of the multitasking *Division of Labor* force of the inceptive production process phase, the second production process phase remained the same as it had been since 1996-97, because the role remained the same as they continued to produce the web-based computer application systems. As the development and implementation of the 3 tier architectural platform did not require the *Programmers*, their number was decreased to 585,000 during 2002-03 from 648,000 in DOL's 2001-02 labor report. Thus, the *Programmer* number was decreased by 63,000 in 2002-03. DOL's labor data suggests that this decrease took place in 2002-03 due to the rise of the newly educated multitalented professionals in such new *Division of Labor* forces as *Computer Software Engineers*, *Computer Applications Software Engineering* and *Computer Systems Software Engineers* for the construction phase, which is the third production process phase of the *Iterative* framework. DOL's 2002-03 labor report the number of *Computer Software Engineers, Computer Applications Software Engineering and Computer Systems Software Engineers* was 697,000. Computer engineering organizations created the educated *Computer Software Engineers* for the following reasons: 1) the new professional *Computer Software Engineers* were educated during the late 90's in a very diverse curriculum to master the activities that are involved within the computer hardware systems and computer application systems. Unlike the traditional *Programmers* who possessed a formal coding skill set only, the *Computer Software Engineers* were the ideal multitalented *Division of Labor* forces that were in high demand by the computer engineering organizations to promote efficiency. 2) As the primary goal for the computer engineering organizations was to develop and implement the 3 tier architectural platform, the versatile characteristics of the *Computer Software Engineers* were in demand for the team leads and project leads to assign multitasks to a give to *Computer Software Engineers* to analyze, design, program and test. 3) Due the diverse academic backgrounds that a *Computer Software Engineer* possesses, he/she has the capability of producing both computer applications software for the web-based applications and computer systems software for the computer hardware monitoring and diagnostic troubleshooting systems. This became advantageous for computer engineering organizations as these new multitasking *Division of Labor* force were able to produce more—with a high standard quality and efficiently—production devices both for production environments and the market. In DOL's 2002-03 labor report, the roles of the *Computer Software Engineers* are described as the following:

> *Software Engineers* working in applications or systems development analyzing users' needs and design, create, and modify general computer applications software or systems. Software engineers can be involved in the design and development of many types of software including software for operating systems, network distribution, and compilers, which convert programs for faster processing. In programming, or coding, software engineers instruct a computer, line by line, how to perform a function. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 169, 2002-03c

In such ways, computer engineering organization utilized their diverse skill sets to multitask them to practice the mass production of the 3 tier architectural platform and the web-based computer application systems beginning in 2002-03. Simultaneously, in 2002-03, computer engineering organizations created a new professional work force called *Computer Hardware Engineers* for the transition production process phase, which was the 4th production phase of the *Iterative* framework. Computer hardware engineering was in demand as the main objective was to develop and implement the computer hardware systems that were used to implement the 3 tier architectural platform. In 2002-03, DOL labor report recorded in table 41 indicates that the number of *Computer Hardware Engineers* was 60,000.

As far as the 5th production process phase is concerned, with the go-live support that was incorporated to the *Iterative* production process framework, computer engineering organizations removed the *Computer Support Specialist* from the inception phase and incorporated it to the go-live support phase. Computer engineering organizations also created a new role called *Communication Systems Administrators* within the go-live support production phase to use them to perform email systems installation, configuration and integration of major web-based computer applications (for the email communication systems as the 3 tier architectural platform) and the web-based application systems that were being developed and implemented. According to DOL 2002-03 labor report, the number of *Computer Support Specialist and Communication Systems Administrators* was 758,000. Computer engineering organizations used the Computer Support Specialists as subject matter experts to master the business process of a given industry to produce a process flow as their outputs to deliver to the *Systems Analysts*. This multitasking role became in demand to improve the quality of the web-based application systems that were produced for industry specific market products without errors, using the spiral module based development methodology, in an effort to speed up the production process of the *Systems Analysts*. The *Systems Support Analysts* analyze the business process from systems and business perspective and produce outputs that can easily be converted to a design by the *Systems Analysts* or *Computer Software Engineers*.

In sum, the trends of DOL labor report of 2002-03 in table 41 on the computer engineering organizations suggests that computer engineering organizations were shifting their goal towards a mass production of the 3 tier architectural platform and the web-based computer application systems. Based on this interpretation, one of the primary purposes of this direction shift was to improve the speed of production as the newly created computer hardware systems were developed and implemented as production devices in the production environment and also for the industry specific systems in the market. In addition, by this time, they fully implemented the *Iterative* production process phase and further incorporated the go-live support production phase. Using the newly incorporated go-live support production phase, computer engineering organizations were able to have their own dedicated professional subject matter experts master any given industry's specific business processes and problems specification that were used as inputs, which are produced from a technical perspective and delivered to the *Systems Analysts*. This also sped up the production process of the *Systems Analysts*, which translates to saving production hours. As a continuous production process chain, the output from the *Systems*

Analysts can be delivered in a timely manner to the *Computer Scientist* to invent new computer systems concepts to solve industries' specific systems problems, as well as create advanced production devices for computer engineering organizations' production process. During the same time in 2002-03, computer engineering organizations rearranged the roles of the *Division of Labor* forces within the first inception production phase. The goal was shifting from invention to developing and implementing the 3 tier architectural platform and the web-based computer application system. Furthermore, computer engineering organizations began to use the diverse array of professionals that were educated during the late 90's to effectively practice multitasking activities in the production process, in order to promote efficiency by assigning a given multitalented worker multiple production activities. This also doubled the production output in their using the automated multitasking web-based computer application with 3 tier architectural platform to speed up the production process. This trend was continuous throughout the mid-2000's as DOL data suggests in table 42 below.

Table 42: Specialized *Division of Labor* force 2004-05:

Production Process Phases	**Multitasking Positions 2004-05**	**Roles & Tasks Description 2004-05**
1: Inception	**979,000 *Systems Analysts, Database Administrators & Computer Scientists***	
1.1	***Systems Analysts***	Multitasking Roles & tasks description remained unchanged.
1.2	***Network Systems & Data Communications Analysts***	Multitasking Roles & tasks description remained unchanged.
1.3	***Database Administrators***	Multitasking Roles & tasks description remained unchanged.
1.4	***Computer Scientists***	Multitasking Roles & tasks description remained unchanged.
2: Evaluation	**499,000 *Computer Programmers***	
2.1	***Application Programmers***	Multitasking Roles & tasks description remained unchanged.
2.2	***Programmer Analysts***	Multitasking Roles & tasks description remained unchanged.
3: Construction	**675,000 *Computer Software Engineers***	
3.1	***Computer Applications Software Engineers***	Multitasking Roles & tasks description remained unchanged.
3.2	***Computer Systems Software Engineers***	Multitasking Roles & tasks description remained unchanged.
4: Transition	**60,000 *Computer Hardware Engineers***	
4.1	***Computer Hardware Engineers***	Multitasking Roles & tasks description remained unchanged.
5: Go-Live Support	**758,000 *Computer Support Specialists & Systems Administrators***	
5.1	***Computer support Specialists***	Multitasking Roles & tasks description remained unchanged.
5.2	***Computer Systems Administrators***	Multitasking Roles & tasks description remained unchanged.
Total	**11 specializations were implemented**	**2,998,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2004-05c

By 2004-05, computer engineering organizations were at a point where they had to continue their mass production of the web-based computer application systems and the 3 tier architectural platforms using the multitalented professionals, who by this time, were recognized by engineering organizations as team members and associates with diverse curricular backgrounds in their education from the late 90's. As DOL's 2004-05 data indicates in table 42 above, the number of the *Systems Analysts, Database Administrators & Computer Scientists* increased to 979,000 from 887,000 in 2002-03. Their number was increased by 92,000 during 2004-05. This increase suggests that computer engineering organizations were inventing an integrated production device to engender a remotely accessible and centralized production environment. These devices were recommended advantages from the capability maturity model integrated (CMMI) best practice guidelines that aim at promoting efficiency for computer engineering organizations in the production process (Ralf Kneuper, *CMMI*: *Improving Software and Systems Development Process*, 2009). These advantages for computer engineering organizations were many: 1) Real-time production progress could be viewed by team leads and project leads, 2) workers could share development methods and outputs samples could be viewed, 3) knowledge can be shared across the production environments regardless of the locations of the workers, 4) training workers on development methods now occurred online, 5) from a management perspective, team leads and project leads were able to extract data from the shared knowledge artifacts stored as assets in the knowledge bank to perform basis of estimates (BOE) for project planning, 6) finally, as knowledge and information became available in an integrated production environment, a real-time remote production process came about in monitoring and solving production related issues in a timely manner, and most significantly, production hours were not wasted as the production devices were all integrated to be accessible during production process to speed up the process of completing production process activities (Kneuper, 2009). Hence, the *Computer Scientists* created a new concept as an integrated computer system known as MS SharePoint, which is built based upon web-based computer application systems (Kneuper, 2009). The production devices of the production environments became a vital instrument in accomplishing both the management goals and production needs to promote efficiency. At the same time, along with MS SharePoint, serving as an integrated knowledge sharing production device for production environments, computer engineering organizations demanded an advanced level 3 tier computer hardware systems' architectural platform to avoid down time if there were a computer systems crash or power outage due to unexpected occurrences. As a result, the following two crucial needs were met: 1) the first need was determining the supply and demand of products for ecommerce shopping. This became a challenge from a new trend, as the world market and transactions were now being conducted via online ecommerce. Thus, computer engineering organizations realized that the market was becoming more and more dependent upon electronic data storages to maintain customer records, to execute transactions and understand customers' ecommerce shopping behavior to determine the demand of the market and produce the supply for it. With this, the data stored in the data warehousing, which is part of the 3rd layer of the 3 tier architectural platform, must be available at all times and accessible to users (Fowler, 2003b). This was part of the CMMI best practices for reinforcing the demand of a continuously availability of computer system and to increase production to supply to meet the demand. 2) The second need is that computer engineering organizations' production environments must be available at all times in a real-time mode to access production process activities' artifacts, such as the product codes, designs and analysis artifacts and management data (WPPR, WBS and BOE). This business continuity concept was the information technology information library's (ITIL) best practice for developing and maintaining computer hardware systems' 3 tier architectural platform (Kneuper, 2009). This required a new concept to be invented by the *Computer Scientists* to modify the 3 tier architectural platform so as to meet the demand. Accordingly, the number of *Computer Scientists* of this time suggests that they were indeed intending to create the new concept that would modify the 3 tier architectural platform, so that production continues without any down time (Fowler, 2003b). Thus, during the mid-2000's, *Computer Scientists* created the concept of cluster, which modifies the 3 tier architectural platform by adding a standby, redundant architectural platform that mimics the other to avoid overall systems failure (Fowler, 2003b). The creation of this new concept was to make the production devices of the production environments available and accessible at all times, so that production could continue uninterrupted (McConnell, 2004b). Figure 13 below illustrates the cluster 3 tier computer hardware architectural system:

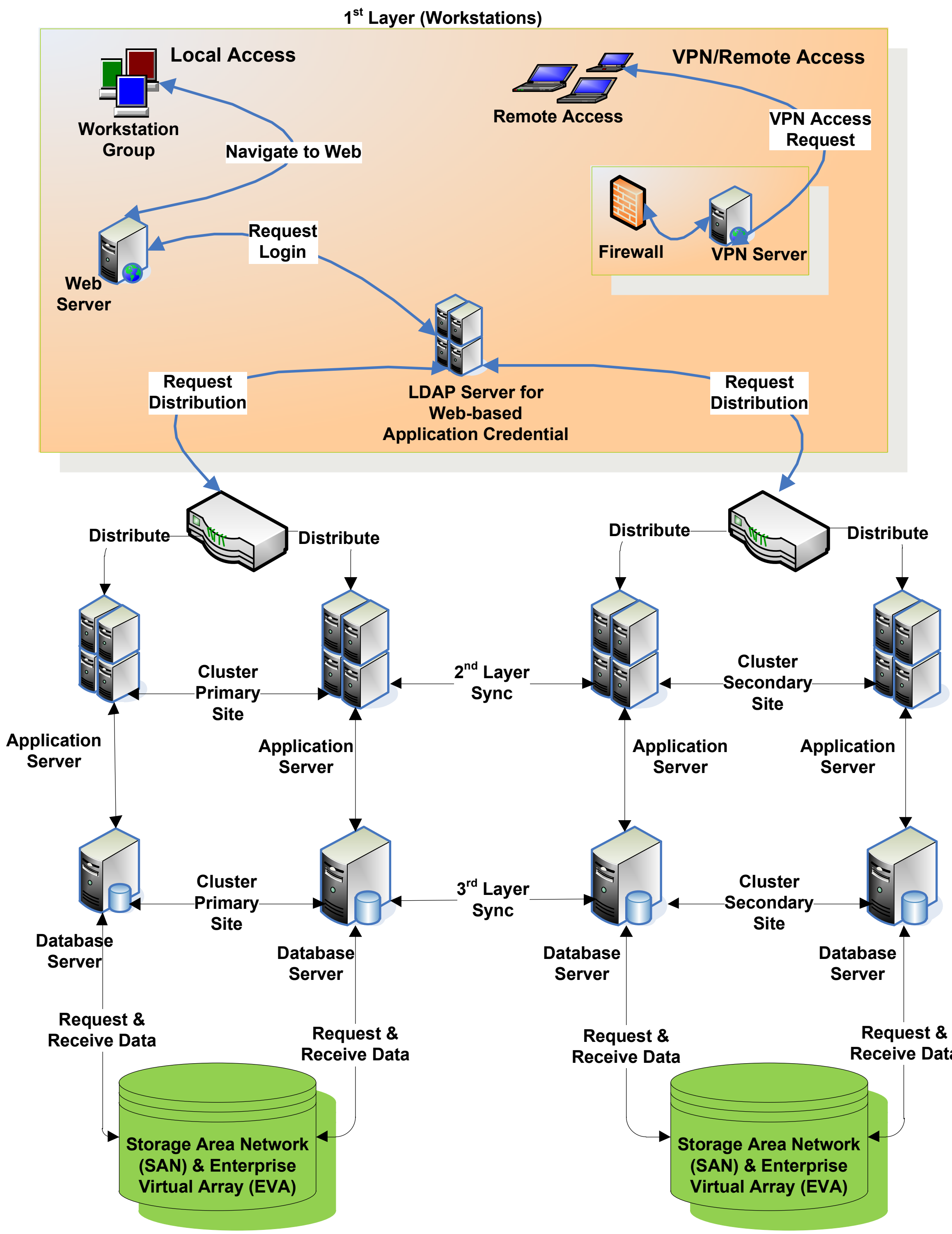


Figure 13: Cluster-Based 3 Tier Infrastructure Platform Architecture

The newly created cluster architectural solution incorporated two valuable functionalities: 1) The newly created cluster solution has an identical structure to the primary production site, which means each layer has its own exact set of replications that are synchronized with each other within the same layer to avoid single point failure of the layer, whereas the original 3 tier architectural platform did not have this capability (McConnell, 2004b). For example, the 2nd layer, which is the web-based application server, where the programming codes and business logics reside, would have its own set of exact same siblings that replicated amongst themselves and are accessible and available at all times (McConnell, 2004b). As workers are logged into the system, they are routed to the most available web-based application server (with the least amount of processing occurring/lowest number of users), and access requests to the 2nd layer are distributed accordingly (McConnell, 2004b). Workers can perform maintenance and upgrades to one of the siblings' web-based application servers of the 2nd layer while the rest of their servers are still running and accessible in production environments (McConnell, 2004b). The same principle of clustering is applied to the 3rd layer, which is the database server containing the logic where production artifacts are stored for workers, enabling uninterrupted access to the production process data. 2) The second concept created to modify the original 3 tier architectural platform was the automatic failover of the primary production site, replacing it with the secondary standby site that is built based upon clustering methodology. The automatic failover occurs when a backup operational mode (in which the functions of a system component, such as a 2^{nd} layer of the web-application server, the 3^{rd} layer of the database server, processors, network cables are assumed by secondary system components when the primary component becomes unavailable through either failure or scheduled down time) was not possible with the original 3 tier architectural platform. The original 3 tier architectural platform was built without the cluster and automatic failover concepts, which mean the 3 layers of the 3 tier architecture platform were connected through a basic one-to-one configuration, causing production down time. In such an environment, the failure or even maintenance of a single server frequently made data access impossible for a large number of workers to continue production until any of the layers of the 3 tier architect platform was restored and back online. However, as the new automatic failover concept was created to solve the limitations of the original 3 tier architectural platform, failover continues the production process instantly without the worker even noticing it. It redirects a worker's connection from the failed or down primary production environment to the secondary backup site that mimics the operations of the primary system. In this way, computer engineering organizations were able to promote efficiency and reinforce continuous productivity improvements by accessing the production environments available and accessible in real-time. Thus, as the DOL 2004-05 data shows, the number of the *Computer Scientists* increased to 979,000 from 887,000 in 2002-03. This indicates that during 2004-05, there was a demand by computer engineering organizations to create a solution that could modify the original 3 tier infrastructural platform, which indicated that the increase of the

Computer Scientists was indeed to create the cluster and automatic failover concepts, which were implemented to modify the original 3 tier infrastructural platform (McConnell, 2004b). As computer engineering organizations in 2004 were primarily focused on modification of the 3 tier architectural platform, DOL's 2004-05 labor data indicated that the number *Programmers* decreased to 499,000 from the 585,000 of 2002-03. This decrease in number by 86,000 indicates that *Programmers* were only maintaining and updating programmer codes of web-based application systems and that their roles were becoming more shared with the professional *Computer Applications Software Engineers* and *Computer Systems Software Engineers* who were educated beginning in the mid-90's through the early 2000's in diverse curricula (McConnell, 2004b).

With this trend, computer engineering organizations continued to use the *Computer Applications Software Engineers* and *Computer Systems Software Engineers* as multitasking *Division of Labor* forces, giving them titles as team members and associates. Although DOL's 2004-05 labor report shows there was a slight decrease by 22,000 in the number of *Computer Applications Software Engineers* and *Computer Systems Software Engineers*, which the 2004-05 shows (675,000 from 679,000 in 2002-03). This was due to the fact that the during 2004-05, computer engineering organizations' primary objective was focused on the *Computer Scientists* inventing the cluster concept and modifying the original 3 tier infrastructural platform, and mass production of web-based application systems development and implementation was secondary. Although this mass production of the web-based application software systems and the mass production software for the computer hardware devices were secondary objectives for computer engineering organizations, the demand for the multitalented professional *Computer Applications Software Engineers* and *Computer Systems Software Engineers* highly increased throughout the late 2000's. DOL's labor report of 2006-07 in table 43 below shows a high increase of *Computer Applications Software Engineers* and *Computer Systems Software Engineers* as the computer engineering organizations' primary goal became a mass production of both the web-based application systems and the mass production of the software for the hardware systems required to implement the new cluster based 3 tier infrastructural platform.

Table 43: Specialized *Division of Labor* force 2006-07:

Production Process Phases	Multitasking Positions 2004-05	Roles & Tasks Description 2006-07
1: Inception	**487,000 *Computer Systems Analysts***	
1.1	***Systems Analysts***	Produced systems problem related specification used as inputs to create new solutions.
2: Evaluation	**507,000 *Computer Scientists & Database Administrators***	
2.1	***Computer Scientists***	Produced systems problem related specification to conceptualize new solutions
2.2	***Database Administrators***	Multitasking Roles & tasks description remained unchanged.
2.3	***Network Systems & Data Communications Analysts***	Multitasking Roles & tasks description remained unchanged.
3: Construction	**455,000 *Computer Programmers***	
3.1	***Application Programmers***	Multitasking Roles & tasks description remained unchanged.
3.2	***Programmer Analysts***	Multitasking Roles & tasks description remained unchanged.
4: Configuration & Change Management	**800,000 *Computer Software Engineers***	
4.1	***Computer Applications Software Engineers***	Multitasking Roles & tasks description remained unchanged.
4.2	***Computer Systems Software Engineers***	Multitasking Roles & tasks description remained unchanged.
5: Transition & Go-Live Support	**797,000 *Computer Support Specialists & Systems Administrators***	
5.1	***Computer support Specialists***	Multitasking Roles & tasks description remained unchanged.
5.2	***Computer Systems Administrators***	Multitasking Roles & tasks description remained unchanged.
Total	**10 specializations were implemented**	**3,046,000 were employed.**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2006-07c

Following the best practices guidelines of CMMI and ITIL, by 2006-07, computer engineering organization incorporated a production phase called configuration & change management for the 4th production phase and moved the transition phase that was the 4th production phase in previous years and combined it with the 5th production phase calling it transition & go-live support. With this arrangements and the DOL labor report of the 2006-07 above in table 43, the indication is that computer engineering organizations' primary objective became the mass productions of web-based application systems and the mass production of software systems for the hardware devices to implement the clustered 3 tier architectural platform. During the same year, as computer engineering organizations' primary objective was the mass production, DOL's 2006-07 labor report shows that the multitasking positions in the production process phase 1 & 2 were rearranged in the following ways: 1) *Computer Scientists, Database Administrators & Network Systems & Data Communications Analysts* positions were move from the inception production phase, which is from 1 to the evaluation production phase that is 2 while keeping the *Systems Analysts* in the inception production process phase. This rearrangement is an indication that these multitasking positions were to produce ongoing computer systems' related problem specification of the newly created cluster 3 tier architectural platforms and the web-based application systems that were going through the mass production mode simultaneously. The product is then stored in the database for future references to be used by the same multitalented *Computer Scientists* as inputs to create solutions that modify existing computer systems and for enhancements purposes. Hence, their roles alternate depending on the needs of the computer organizations' objectives—between multitasking for new inventions of systems solution and systems' problems specification production process. For example, in 2006-07 table 43 above, DOL's labor report indicated that the number of *Systems Analysts* were 487,000 while the *Computer Scientists, Database Administrators & Network Systems & Data Communications* number indicated 507,000. Both of these multitasking *Division of Labor* forces possess a high number of multitasking professionals, as they were in continuous demand either to produce new solutions or to specify systems problems.

During 2006-07, DOL's labor report indicated that the number of *Computer Programmers* decreased to 455,000 from the 499,000 of the 2004-05 labor report, showing a decrease of 44,000. The decrease in the number of *Computer Programmers* in 2004-05 was 86,000. This indicates that the decrease of *Computer Programmers* had been steady throughout the course of the mid-2000's. Their role had continued to be the same in maintaining and updating programming codes. As the demand for multitasking professionals of *Computer Applications Software Engineers* and *Computer Systems Software Engineers* that were educated in diverse curricula during late to early 2000 were increasing in demand, the number of *Computer Programmers* started to decreases sharply. The *Computer Programmers* production process activities were becoming part of the *Computer Applications Software Engineers* and *Computer Systems Software Engineers* tasks, as these new multitasking professional sets of *Division of Labor* forces included the production activities of the *Computer Programmers*. In 2006-07, DOL describes the production activities of these new multitasking professionals:

> *Computer Applications Software Engineers* analyze users' needs and design, construct, and maintain general computer applications software or specialized utility programs. These workers use different programming languages, depending on the purpose of the program. The programming languages most often used are C, C++, and Java. Some software engineers develop both packaged systems and systems software or create customized applications. Bureau of Labor Statistics, *Occupational Outlook Handbook*, 111, 2006-07c

Multitasking was implemented in high scale. Thus, the production activities of the *Computer Programmer* were performed by the multitasking *Computer*

Applications Software Engineers to benefit computer engineering organizations in promoting efficiency by controlling labor wage. Similarly, computer engineering organizations combined the *Computer Hardware Engineers* production process activities with the production process activities of *Computer Systems Software Engineers*. Thus, the production activities of the *Computer Hardware Engineers* were simply performed by the diversely educated *Computer Systems Software Engineers*. Thus, DOL's 2006-07 labor report in table 43 indicates that the number of the *Computer Applications Software Engineers* and *Computer Systems Software Engineers* sharply increased to 800,000 from 675,000 of the 2004-05 DOL's labor report. This increase was substantial as it was an addition of 125,000. This indicates that the *Computer Applications Software Engineers* and *Computer Systems Software Engineers* were in demand as computer engineering organizations' objective was to develop and implement the newly created cluster 3 tier architectural platform and the web-based application systems during 2006-07. Finally, as computer engineering organizations redefined the 5th production process phase as transitional & go-live support, the production activities of the *Computer Support Specialist and Systems Administrators* production activities included systems release, systems user training and computer user manual development. These additional production activities were added on the top of their industry specific computer systems' subject matter expert of their multitasking activities. As additional production process activities were added, the demand for the *Computer Support Specialist and Systems Administrators* increased. DOL's 2006-07 labor report in table 43 above indicates that the number of *Computer Support Specialist and Systems Administrators* increased to 797,000 from the 758,000 of DOL's 2004-05 labor report, which shows an increase of 39,000. In summary, as computer engineering organizations' primary objective during 2006-07 was to develop and implement the cluster 3 tier architectural platform and the web-based application systems, the *Iterative* production process phases and the production process activities were redefined to meet the demand that was needed to promote efficiency in the production process. During this time frame, as the objective for computer engineering organization was centrally focused on mass production, DOL's labor report indicates a high number of multitasking *Division of Labor* forces in *Computer Applications Software Engineers* and *Computer Systems Software Engineers*, which were in demand to develop and implement the cluster 3 tier architectural platform created a year before by the *Computer Scientists* during 2004-05, in which they were in demand at a number of 979,000. The trend continues during 2008-09 as the DOL labor report shows in table 44 below.

Table 44: Specialized *Division of Labor* force 2008-09:

Production Process Phases	**Multitasking Positions 2008-09**	**Roles & Tasks Description 2008-09**
1: Inception	**961,200 *Computer Network, Systems & Database Administrators***	
1.1	***Network and Computer Systems Administrators***	Multitasking Roles & tasks description remained unchanged.
1.2	***Database Administrators***	Multitasking Roles & tasks description remained unchanged.
1.3	***Network Systems & Data Communications Analysts***	Multitasking Roles & tasks description remained unchanged.
2: Evaluation	**28,900 *Computer Scientists***	
2.1	***Computer Scientists***	Produce systems' problems specification to conceptualize new solutions.
3: Construction	**1.3 Million *Computer Software Engineers & Computer Programmers***	
3.1	***Application Programmers***	Multitasking Roles & tasks description remained unchanged.
3.2	***Programmer Analysts***	Multitasking Roles & tasks description remained unchanged.
4: Configuration & Change Management	**532,200 *Computer Systems & Applications Analysts***	
4.1	***Computer Applications Software Engineers***	Multitasking Roles & tasks description remained unchanged.
4.2	***Computer Systems Software Engineers***	Multitasking Roles & tasks description remained unchanged.
4.3	***Computer Systems Analysts***	Multitasking Roles & tasks description remained unchanged.
5: Transition & Go-Live Support	**565,700 *Computer Support Specialists***	
5.1	***Technical Support Specialists***	Multitasking Roles & tasks description remained unchanged.
5.2	***Help-desk Technicians***	Multitasking Roles & tasks description remained unchanged.
Total	**11 specializations were implemented**	**3,388,000**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2008-09c

By the late 2000's, computer engineering organizations continued to reinforce mass production of the web-based application systems and the implementation of cluster 3 tier architectural platforms as production devices both for computer engineering organizations' production environments and industry specific market demands. As can be seen above in table 44, DOL's labor report of 2008-09 indicates that the production process activities and the multitasking *Division of Labor* forces were rearranged to meet computer engineering organizations' primary objective of continuity in mass production process. Below is a synopsis of the rearrangement of the production process and multitasking production process activities: 1) during 2008-09, *Computer System Analysts* that were in the inception production process (phase 1) were moved to the configuration & change management production (phase 4). 2) At the time, the *Database Administrators Network Systems & Data Communications Analysts* that were in the evaluation production process (phase 2) moved to the inception production process phase; however, the *Computer Scientists* were kept in the same evaluation production process phase. 3) During the same year, the *Computer Programmers* that were in the configuration & change management production process phase were moved to the construction production process phase to be combined with the *Computer Software Engineers*. 4) Simultaneously, the *Computer Support Specialists* that were in the transition & go-live support were kept in place while the *Systems Administrators* were replaced by the *Technical Support Specialists* and *Help-desk Technicians* within the same production phase. This rearrangement reflects the primary focus of mass production process and the production of the web-based application systems and software systems to implement the cluster based 3 tier architectural platforms. Based on these arrangements, the number of multitasking *Division of Labor* forces indicated a correlation in confirming that computer engineering organizations fully took advantage of the multitalented workforce by assigning multitasks to promote efficiency. For example, in table 44 above, DOL's 2008-09 labor report on computer engineering organizations indicates that the number of *Network & Computer Systems Administrators, Database Administrators* and *Network Systems & Data Communications Analysts* increased to 961,200 from the

507,000 of the 2006-07 report, showing an increase of 454,000. This was a significant increase as they were in demand to produce the ongoing systems problem specification with scenarios that could be used as inputs to conceptualize the next generation's new computer systems solutions to ensure that systems interruptions were avoided and production devices were continuously available and remotely accessible (Kneuper, 2009). Similarly, the number of *Computer Scientists* was at 28,900 during 2008-09 and computer engineering organizations used these talented professionals to evaluate the systems' problem scenarios to use them as inputs to conceptualize systems solutions. This indicates that the production process activities for the *Computer Scientists* were kept at evaluation production process phase 2 as the primary objective of computer engineering organizations continued to be mass production. Hence, the multitalented *Computer Scientists'* production process activities were adjusted as the priorities of computer engineering organizations changed. For example, during 2004-05 *Computer Scientists* created the cluster based 3 tier architectural platform to modify the original 3 tier architectural platform, and this multitasking production process activity was changed in 2008-09 to evaluate systems problems in production process. During the same year (2008-09), DOL's labor report indicates that *Computer Software Engineering* and *Computer Programmers* were combined and this indicates that an effort was made to increase the number of multitasking *Division of Labor* forces as the primary focus was on mass production of the web-based application systems and software systems to implement the cluster based 3 tier architectural platform. Thus, according to

DOL's 2008-09 labor report in table 44, the number of *Computer Software Engineering* and *Computer Programmers* increased to 1.3 million from the 800,000 of the 2006-07 report, which showed a significant increase of 500,000. As the priority continued to be mass production, these talented professional workforces continued to be in demand because they were diversely educated and a *Computer Software Engineer* was capable of multitasking to produce all production process phase activities. With this new and thoroughly multitalented workforce in place, the need to combine the traditional *Computer Programmers* with the *Computer Software Engineer* became inevitable (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2008-09c). *Computer Programmers* were required to be up to date on contemporary object oriented programming methods and reeducate themselves with contemporary curricula to be a part of the multitasking *Division of Labor* forces (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2008-09c). Thus, based on the trends of DOL's 2008-09 labor report, computer engineering organizations' primary objective of mass production, it is evident that computer engineering organizations were attempting to implement two critical operations (systems' problems specifications and the new solutions invention production process activities) in parallel to continue their mass production improvement. As the DOL data of 2008-09 indicates, there was a remarkable increase of 500,000 for the *Computer Software Engineers* and *Computer Programmers* that totaled them to 1.3 million to reinforce the mass production efforts that were taking place. At the same time, there was an increase of 454,000 for the *Network & Computer Systems Administrators, Database*

Administrators and *Network Systems & Data Communications Analysts* that totaled them to a high of 961,200, indicating that computer engineering organizations were attempting to implement in parallel the aforementioned two critical solutions of 2008-09. As recommended by CMMI and ITIL, to accomplish a continuous mass production improvement, computer engineering organizations implemented ongoing practices of systems' problems specifications with scenarios and used them as inputs to invent new solutions using the spiral development methodology (Kneuper, 2009). This is known as continuous production improvement and optimization (Kneuper, 2009).

Finally, as a result of implementing the ongoing production improvements via the two solutions in parallel, computer engineering organizations were able to implement a continuous mass production process improvement with great efficiency (Kneuper, 2009). The high numbers of increase in the multitasking *Division of Labor* were made to ensure that there would not be a labor force scarcity, in that ultimately the labor wage would not increase but would remain constant or decrease, as was practiced during the *Industrial Era* analyzed in chapter 3. Similarly, the number of *Computer Systems Analysts* increased to 532,200 from 487,000 of the 2006-07 DOL labor report, showing an increase of 45,200. During 2008-09, *Computer Systems Analysts* were moved to the configuration & change management production process phase (phase 4) from the inception production process phase (phase 1) as it was during 2006-07. As their production process phases and activities were redefined to their new categories, according to CMMI best practice guidelines, they were producing development methodology problems specification and systems change process dynamics between stakeholders and the production process (Kneuper, 2009). The outputs from this production process were used as inputs for both the *Computer Software Engineering* to fix the bugs of production devices and for *Computer Scientists* to conceptualize new inventions as solutions. During the late 2000's, *Computer Support Specialists* were not highly in demand as the computer engineering organizations' primary objective was mass production of the newly invented computer systems, and *Computer Support Specialists'* role consisted in mastering the industry specific business processes to translate them from a systems perspective. A multitalented *Computer Support Specialist* has to master the specific industry business process, the industry specific web-based application systems, perform troubleshooting, determine the causal factors, analyze, trace issues back to systems designs and write business requirements and deliver the outputs to the sharable device, such as the MS SharePoint of the production environment, making it accessible for workers to use for inputs to start their multitasking production activities. Hence all of these multitasks are to be performed by a given *Computer Support Specialist* once the newly invented systems are being developed in forms of modules based one after the other and implemented for use, but not while they are still being produced in production process. Hence, from DOL's 2008-09 labor report in table 44 above, their number decreased to 565,700 from the 797,000 of the 2006-07 report showing a reduction by 231,300. The *Computer Support Specialists* production process activities were geared towards mastering the industry specific business process as subject matter experts to translate from systems perspectives in that *Computer Software Engineers*, *Computer Scientists* and *Systems Analysts* comprehend systems related problems without wasting time to go through the process of translating from business to systems contexts. As a result, production process speeds up, in that *Computer Scientists* can directly go through the production process activities of the new solution invention, and the *Computer Software Engineers* can simply go through the production process activities of debugging the production devices.

In sum, the *Iterative* framework transformed the production process, replacing the *Waterfall* framework that could not work as a multitasking production process and did not have a modular based methodology for each production process. Equally crucial were the production process activities and the formation of the multitasking *Division of Labor* force for the production process infrastructure as inputs for the macro-matrix management infrastructure (WBS, BOE & WPPR) for planning and tracking and controlling labor wages to promote the ongoing mass production improvement in efficiency.

4.5 Triple Surplus Values: Production and Management Infrastructure

This section analyzes the manifestation of surplus value in computer engineering organizations during the timeframe of 1996-2009. This study calls the form of surplus value manifested during 1996-2009 a *Triple Surplus Value*. It asserts that the *Triple Surplus Value* replaced the classic absolute and relative surplus values that computer engineering organizations implemented during the *Waterfall* production process framework between the years of 1970 and 1995. It was during the single function production process and the individual based specialized *Division of Labor* force era of 1970-95 that the worker performed only one task. Improving the speed of the production process was the means to extracting surplus value in the following ways: 1) During this period, the surplus value mechanism was based on the accounting and project management methods of creating salary ranges and adjusting the lower end of the range in the event of a scarcity of a particular specialized worker's skill set in demand. Two different salary ranges were setup for beginners and experienced workers, and the lower ends of these salary ranges were adjusted to increase profits, whereby the bottom salaries were slightly increased to give an organization a psychological edge in attracting new employees during a scarcity. In other words, a candidate would become more willing to be hired at the lower end because of their perceiving the bump in pay at that point on the spectrum. But, since the higher end of the range would remain constant or decrease, this strategy was, in the end, profitable for computer engineering organizations. For example, as can be seen in table 20 of section 3.5 in chapter 3, the salary range for the experienced *Computer Systems Analysts* was \$12, 000 - \$22, 000 and for the beginners it was \$8, 950 - \$12, 700. As the DOL labor report indicates throughout the 70's, 80's and early to mid-90's, the lower end of the salary range was either being adjusted or remaining constant. As can be seen, both ranges had lower ends that were adjusted without consequence to the higher end. This was in an effort to promote efficiency by controlling direct labor cost (that from tasks in production process) as much as possible. 2) The second mechanism was to continually increase the workforce to keep salary ranges constant (or even lowered). This was accomplished by avoiding scarcity via the perpetual increase in specialized *Division of Labor* forces. 3) The third and most effective mechanism was doubling productivity by speeding up the production process, done by using multiple production devices each dedicated to a specific task. Although using this mechanism is often merely regarded as a means to promoting efficiency, it is really equivalent to surplus value: the speed of completing a task in the production process was fast, and the objective in applying this method was to save labor hours from the allocated budget for the task, so that the saved labor hours could be used to produce another task free of labor cost.

These were the surplus mechanisms during that time frame as the production and management infrastructures were manual based single function structures. However, computer engineering organizations needed an advanced mechanism for promoting efficiency, which is equivalent to extracting surplus labor to produce more without labor cost to enable them to extract free labor. This advantage was not limited to only one task in production process, but they started aiming to extract surplus labor hours from multiple tasks of the production process phases to maximize their profits. Thus, in the ongoing pursuit to accomplish this goal, they initiated a transformation of the production infrastructure that consisted of 4 components: 1) *Waterfall* was changed to the *Iterative* production framework method, 2) single function base production devices were replaced by multifunctional production devices, 3) individual specialization based *Division of Labor* forces and manual based individual production processes changed to multifunctional based *Division of Labor* forces, and 4) the individual based production process became a multitasking based production process. This was followed by a transformation in the hierarchy management infrastructure to a macro-matrix management infrastructure, including the replacement of the pyramid organizational structure with the upside-down and linear organizational structure. Thus, as analyzed earlier in the previous sections of this chapter, by the mid-90's, the 4 components of production infrastructure were transformed by the emergence of multitasking production devices, automation and internet communication. This gave birth to the *Informational Era*'s production infrastructure components: the *Iterative* production process framework, the multitasking based production devices, the multitasking *Division of Labor* forces and the multitasking based production process. This new advancement transformed the manual based single function production process and was meant to promote efficiency in production process—the equivalent of transforming the previous classic surplus value mechanism described above in this section, which was limited to extracting surplus labor value from only one task in the manual production process to produce a more effective mechanism of surplus value that would be compatible with the multitasking production process. Thus, applying the most advanced multitasking production devices and multitalented workforces, computer engineering organizations were able to extract surplus labor by assigning multitasking to a worker in production process, which, pre-1995, was the extraction of surplus limited to only one task in the production process. An example of the transformation of the surplus value mechanism can be seen in the trends of the DOL labor wage report in table 45 below, where in 1996-97 the salary range shows only one range (contrasting with pre-1995) that in essence constituted *two* salary ranges, whereby beginners were located towards the lower end of the range and experienced individuals enjoyed the salaries of the higher end. This was an advanced surplus value mechanism applied to control labor wage for multiple tasks in production process that enabled them to extract surplus labor from multiple tasks in production process. Computer engineering organizations no longer adjusted salary ranges in two ways, as now they were doubling their surplus labor value by multitasking a multitalented *Computer Software Engineer,* using automated based multitasking production devices, which reduced the numbers of *Analysts* and *Programmers* (cutting labor costs).

Hence, this demonstrates the manifestation of the *Triple Surplus Value*, in which computer engineering organizations used multitasking based computer systems as production devices in production environments, coupled with the multitasking production process to promote efficiency, which is equivalent to high production with much free labor cost. This becomes advantageous to the computer engineering organizations particularly when a multitalented professional *Computer Software Engineer*, who was educated during the late 90’s and early 2000’s through the mid-2000’s in diverse curricula, becomes part of the multitasking based *Division of Labor* force, capable of executing all the production process functions of the *Systems Analysts*, *Designers* and *Programmers*. At the same time, the *Computer Software Analysts* could be assigned to perform production activities with multiple products, and these production activities were monitored through the Microsoft SharePoint production environment (where production data is accessible in real-time and remotely available for a team lead), which controlled production hours. As this dissertation asserts that the *Triple Surplus Value* manifests only while the product is still in the production process, it proposes a formula that proves this fact. The *Triple Surplus Value* occurs when a multitalented professional is assigned to perform many tasks in the production process of a given project using production devises like Microsoft SharePoint as the production environment while the production progress is monitored in real-time. The *Triple Surplus Value* is formulated in the following way: 1) Find the *Offered Labor Wage* (OLW), the amount of salary offered to the worker. 2) *Historical Labor Wage* (HLW) is the labor wage for the same position during the 70's, 80's and 90's, so find the labor wage amount from the DOL historical records of 70's, 80's or the early-to-mid-90's. 3) To obtain the *Converted Labor Wage* (CLW), which accounts for inflation, enter the HLW into the DOL inflation calculator to find what that labor wage amount could buy during the same year as the time the OLW was made to the worker. 4) Take this CLW and subtract it from the OLW to find the difference, written in a mathematical form as (OS – CLW) = X. The value of X is defined in this formula as *Motivation Value* (MV). 5) This *Motivation Value* amount is the one that captures the attention of the worker during the process of getting the job. 6)

Number of Production Process Activities (NPPA) identifies all the production process activities from the position description of the job posting that defines the roles and responsibilities of the worker and finds the salary for each of the production process activities that belong to other production process phases. For example, if the position is for *Computer Systems Analysts*, then any production process activities that belong to other production phases must be counted and their labor wage must be collected from the DOL standard labor wage of the computer engineering organizations. At this point, add each of the labor wages collected and get a total dollar amount, which is ∑ NPPA = N. 7) *Gross Triple Surplus Value* (GTSV) is the *Triple Surplus Value* before the MV is deducted. Thus, the OLW value must be subtracted from the NPPA value, NPPR – OLW = X, representing the value of the GTSV. 8) *Actual Triple Surplus Value* (ATSV) is found when the MV is subtracted from the GTSV. Finally, the *Actual Triple Surplus Value* is the actual amount that goes to the computer engineering organization as a profit before the product even goes out for market. Compare the *Actual Triple Surplus Value* with the *Motivation Value* to observe the labor wage amount that translates into the profits of the computer engineering organizations, and then, the *Motivation Value* that goes to the worker in a form of a rewarding compensation.

There are two ways that the *Triple Surplus Value* methodology will be applied to measure the manifestation of *Triple Surplus Value*: 1) the first method is known as the *Motivation Value* (MV) method of measuring the manifestation of *Triple Surplus Value*. The MV is applied if the *Offered Labor Wage* (OLW) is higher than the DOL standard labor wage for a given position and this scenario requires all of the *Triple Surplus Value* variables defined above to be used in finding the *Actual Triple Surplus Value* (ATSV) that goes to the computer engineering organizations and the MV that goes to the worker. In addition, the MV is applied if the position existed during both the *Waterfall* production framework era (1970-1995) and the contemporary *Iterative* production framework because there is DOL's HLW of the position that can be computed as a converted value to get the MV. This primarily pertains to positions such as *Systems Analysts* and *Programmers*, as they have existed throughout the history of computer systems, beginning with *Waterfall* and carrying over to the contemporary *Iterative* production framework. This methodology is also applicable to the new positions incorporated during the *Iterative* framework era, which, beginning after 1996, had labor wage values higher than the DOL standard. For instance, the new positions are *Computer Applications Software Engineers*, *Computer Systems Software Engineers* and *Database Administrators*, and their HLW can only be tracked back to 1996, as they never existed before 1995. 2) The second method is known as *Gross Triple Surplus Value* (GTSV), and this is applied if the OLW is less than the DOL labor wage standard for a particular position. If the OLW is less than the DOL labor wage standard, then there is a MV and ATSV to compute for. Hence, in this is particular methodology, only the following 3 variables are required: OLW, *Number of Production Process Activities* NPPA and GTSV to find the ATSV that goes to the computer engineering organizations as a net profit in this scenario. Thus, below, table 45 and table 46 show DOL's 1996-97 and 1998-99 salary ranges for the multitasking *Division of Labor* of computer engineering organizations, respectively. Based on the trends of the DOL data, the *Triple Surplus Value* manifestation formula will be used to analyze how computer engineering organizations gained profit.

Table 45: Illustrates the salary ranges of computer engineering workers during 1996-97:

Multitasking Positions 1996-97	Annual Salary for Experienced 1996-97	Annual Salary for Beginners 1996-97
Computer Scientists	**$62,000**	**$51,000**
Computer Engineers	**$62,000**	**$45,000**
Systems Analysts	**$54,000**	**$43,500**
Database Administrators	**$62,000**	**$51,000**
Computer Support Analysts	**$55,000**	**$34,100**
Application Programmers	**$36,500**	**$29,500**
Systems Programmers	**$54,000**	**$44,000**
Programmer Analysts	**$47,000**	**$36,000**
Computer Operators	**$31,000**	**$20,000**
Computer Service Technicians	**$26,700**	**$20,000**
Computer and Office Machine Repairers	**$25,300**	**$22,400**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook,* 1996-97d

Table 46: Illustrates the salary ranges of computer engineering workers during 1998-99:

Multitasking Positions 1998-99	Annual Salary for Experienced 1998-99	Annual Salary for Beginners 1998-99
Computer Scientists	**$62,000**	**$51,000**
Computer Engineers	**$62,000**	**$45,000**
Systems Analysts	**$54,000**	**$43,500**
Database Administrators	**$67,000**	**$54,000**
Computer Support Analysts	**$55,000**	**$34,100**
Application Programmers	**$39,500**	**$32,500**
Systems Programmers	**$60,000**	**$47,500**
Programmer Analysts	**$52,000**	**$30,700**
Computer Operators	**$30,900**	**$16,600**
Computer Service Technicians	**$26,700**	**$20,000**
Computer and office machine repairers	**$25,300**	**$22,400**

Source: Department of Labor: *Occupational Outlook Handbook*, 1998-99d

As the era of the specialize *Division of Labor* was replaced by the multitasking based *Division of Labor* forces during the mid-to-later 90's, the labor wage structure was also altered from being range based for both the experienced and the beginning salary to only one range (Bureau of Labor Statistics, *Occupational Outlook Handbook*, 1998-99d). As DOL's labor reports indicates in table 45 of the 1996-97 and table 46 for 1998-99, the labor wage is structured only in one range between the experienced workers and beginners for all the positions listed. As analyzed above in this section, the two ranges were used to control labor wages as part of the classic surplus value mechanisms by adjusting the lower end of the salary ranges, which gives the workers the illusion that their range was increasing, although in practice, their high end of the salary range was unchanged. Nevertheless, based on the indications of the DOL labor wage reports, only one range was used for both the experienced and the beginners. This indicates that computer engineering organizations were forming a new surplus value mechanism, as the multitasking based production process required its own surplus mechanism such that a worker is assigned to produce multiple tasks at the same time for a given project. According to DOL's labor report, during the mid to late 90's, the *Iterative* framework's 1st inception production process phase that was newly implemented was in demand for further advancing the ".com" web based computer application systems and contributing to the development of academic curricula to create the diverse academic programs that helped produce the multitalented professionals. Thus, to make these goal a success, computer engineering organization kept the *Systems Analysts* position as it was during the specialized *Division of Labor* forces and created 4 production process activities for the 1st inception production phase with new positions, such as *Computer Scientists, Computer Engineers*, *Database Administrators* and *Computer Support*

Analysts. This new multitasking based production process started with its high salary range. For instance, during 1996-97 and 1998-99, DOI's labor wage reports indicated that the experienced *Computer Scientists* were earning $62,000 while the beginners earned $51,000 for both years. During the same years, the salary for the experienced *Computer Engineers* was $62,000 and $45,000 for the beginners for both years. Similarly, the experienced *Database Administrators* earned $67,000 and the beginners $54,000 for both years, and the experienced *Computer Support Analysts* earned $55,000 while the beginner's labor wage stood at $34,100. These labor wages were for the newly created positions within the 1st inception production process as they were in demand to expand on the ".com" web-based application systems development to contribute computer systems development methods to the academic sector to introduce the diverse curricula in the field of computer engineering, so that multitalented professionals could be trained to meet the multitasking labor demand. At the same time, computer engineering organizations kept the specialized *Systems Analysts* that were from the *Transitional Phase* 1970-1995, as they were in demand to produce analysis, design and code specification as well as business process and problems specifications. Hence, during the 1996-97 and 1998-99, the experienced *Systems Analysts* labor wage was $54,000 and for the beginners it was 43,500. With this new transformation from the individual based specialized *Division of Labor* to the multitasking based production process, the size of the multitasking based *Division of Labor* forces in *Computer Scientists, Computer Engineering, Systems Analysis, Database Administrators*, and *Computer Support Analysts* was 828,000. This number indicates that they were in demand to during the year 1996-97 and 1998-99.

From *Programmers* perspective, during the years 1996-97 and 1998-99, computer engineering organizations kept the *Application Programmers* specialized *Division of Labor* forces from the early 90's of the *Transitional Phase* and created 2 types of *Programming* positions with their own production process activities, which were *Systems Programmers* and *Programmer Analysts* positions. These additional positions were created as the computer engineering organizations were implementing the 2nd *Iterative* production process phase known as the construction phase. With this new implementation, according to DOL's 1996-97 and 1998-99, the labor wage for the experienced *Application Programmer* was $36,500 while the beginners' labor wage stood at $29,500. For the experienced *Systems Programmers* the labor wage was $54,000, and for the beginners *Systems Programmers* the labor wage was $44,000. During the same year, DOL reported that the experienced *Programmer Analysts* were earning $47,000 while the beginners were making $36,000. These labor wages were newly standardized as the computer engineering organizations were still in the process of implementing the multitasking production process based on the *Iterative* production process framework guidelines as a worker was assigned to produce multiple tasks at the same time. With this new multitasking production process in formation, DOL's labor report of the computer engineering organizations indicated that the number of *Programmers* decreased to 537,000 from the 555,000 of 1994-95.

Computer engineering organizations maintained the 3rd production process phase of deployment and the 4th of maintenance from the *Waterfall* production process framework used during the *Transitional Phase* of 1970-1995. The purpose of keeping these production process phases was to continue to produce production devices components as the market was still using the computer systems that were produced during the early to mid-90's. Hence, according to DOL's labor wage reported in 1996-97 and 1998-99, the experienced *Computer & Peripheral Equipment Operator*s were earning $31,000 and the beginners were earning $20,000. At the same time, the experienced *Computer Services Technicians* made $26,700 while the beginners were making and $20,000. Similarly, DOL's labor wage reported that the experienced *Computer & Office Machine Repairers* were paid $25,300 while the beginners were earning $22,400 during the same years. The deployment and the maintenance production process phases labor wage remained constant throughout the course of the mid to late 90's as the automation and web-based computer systems were in high demand. Hence, as analyzed in this section, towards the end of the 90's, computer engineering organizations implemented a new surplus value mechanism, which was to use only one salary range for both the experienced and the beginner instead of a range dedicated for each, which was the classic surplus value mechanism practiced during the individual based production process during 1970-95. According to the trends of DOL's labor wage data of the computer engineering organizations, after 1996-97, the data trends show that only one labor wage range was used, and this trend coincides with the implementation of the multitasking production process, suggesting that computer engineering organizations implemented only one labor wage range as a new surplus value mechanism.

Table 47: Illustrates the salary ranges of computer engineering workers during 2000-01:

Specializations 2000-2001	Annual Salary for Experienced 2000-2001	Annual Salary for Beginners 2000-2001
Systems Analysts	**$74,000**	**$40,570**
Computer Engineers	**$80,500**	**$46,240**
Computer Scientists	**$70,250**	**$34,290**
Database Administrators	**$69,920**	**$36,440**
Computer Support Analysts	**$48,810**	**$28,880**
Application Programmers	**$50,500**	**$38,000**
Systems Programmers	**$63,000**	**$49,000**
Programmer Analysts	**$70,610**	**$36,020**
Computer Technicians	**$30,900**	**$16,600**
Computer and office machine repairers	**$30,900**	**$16,600**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2000-01d

As can be seen in table 47, DOL's 2000-01 labor report, the salary of the experienced Systems Analysts increased to $74,000 from $54,000 of the 1998-99 report with $20,000 differences and the beginners salary was reduced to $40,570 from $43,500 of the 1998-99 report, which means this decrease was that the highly experienced were in demand based on the following scenario. This increase was subjected to the fact that at the beginning of the 2000's, computer engineering organizations' goal was to create a computer systems architectural platform to improve the performance of the ".com" web-based application systems that were used as production devices in the production environment and for the industry specific market as the commercial transaction trend become ecommerce based via the internet, overcoming the performance issue became a priority for computer engineering organizations. The *Systems Analysts* were producing computer systems related problems specification and it was because this aspect of their activities that they discovered the 2 tier computer systems architectural platform was causing the performance problems limiting the ".com" web-based applications performance. As the *Systems Analysts* discovered this limitation, their product outputs are used as inputs for the Computer Scientists to conceptualize new computer systems as solutions. With this scenario in hands, during the same time, DOL labor reported the experienced *Computer Scientist* salary was increased to $70,250 from $62,000 of the 1998-99 labor report showing an increase of $8,250 and for the beginners, the salary was reduced to $34,290 from $51,000 of the 1998-99 report. This suggest that both of the experienced Systems Analysts and Computer Scientists were in demand during 2000-01 as the goal was to create the 3 tier architectural platform to solve the performance problem. Similarly, during the same year, the *Computer Engineers* salary increased to $80,500 from $67,000 of the 1998-99 report indicating an increase of $13,500, and for the beginners, the salary indicated a slight increase to $46,240 from $45,000 of the 1998-99 report. This increase is subjected to the same reasoning as the *Computer Scientists* and *Systems Analysts* in that the *Computer Engineers* were in demand to produce proof of concepts of the new 3 tier architectural platform. During the same year of 2000-01 DOL's labor report, the experienced *Database Administrators* a slight salary increased to $69,920 from $67,000 of the 1998-99 labor report, and for the beginners, the salary decreased to $36,440 from $54,000 of the 1998-99 labor report. This suggests that the experienced Database Administrators were in demand during this period as computer engineering organizations goal was to invent the systems that improves the performance problem caused by the 2 tier architectural platform.

During the 2000-01, DOL labor report indicated the highest number of 1.5 million of *Systems Analysts, Computer Scientists, Computer Engineer* and *Database Administrators* that suggests these multitasking workers were in demand to invent a solution both for the production device and for specific market systems. Although during the early 2000's, the goal for the computer engineering organizations was to create the 3 tier architectural platform, DOL's 2000-01 labor report indicated an increase of the experienced *Application Programmer* to $50,500 from $39,500 of the 1998-99 report suggesting that they were part of the production process in maintaining the ".com" web-based applications. Similarly, the experienced *Systems Programmers* salary increased to $63,000 from $60,000 of the 1998-99 report as their role in production process was equally in demand as the experienced *Application Programmers* during the early 2000's. The experienced *Programmer Analysts* salary indicated an increase by $70,610 from $52,000 of the 1998-99 report showing an increase of $18,610. The beginners *Application Programmers* salary increased to $38,000 from $32,500 of the 1998-99 report. Both the beginners of the *Systems Programmers* and the *Programmer Analysts* salary showed increase during the 2000-01. While the beginners *System Programmers* showed a slight increase to $49,000 from $47,500 of the 1998-99 labor report, the beginner *Programmer Analysts* salary increased to $36,020 from $30,700 of the 1998-99 report suggesting that there was a shortage of these multitasking *Division of Labor* forces. According to the DOL labor report of 2000-01 on table 40, the number of *Computer Programmers* were 648,000, which increased only by 80,000 from the 568,000 of the 1998-99 labor report. This multitasking workers scarcity caused the salary ranges to increase during 2000-01. As analyzed earlier in this section, Computer Scientists were creating a diverse academic curricula that was meant to produce multitalented *Software Engineers* and *Application Engineers* that could be assigned to perform multitasks in production process per the of the goal of computer engineering organization during the late 90's to control the rise of the labor wage as can be seen on table 47 of the DOL's 2000-01 report indicated. These multitalented *Division of Labor* forces did not join the production process early towards the mid-2000's. According to the DOI's data of the 2000-01, it indicates that there was a success in increasing the number of the *Computer Scientists*, *Systems Analysts*, *Compute Engineers* and *Database Administrators* to create the 3 tier architectural platform to improve the performance of the ".com" web-based application. Hence, this suggests that computer engineering organizations were increasing the numbers of the workers to meet their goals depending on the type of production that required meeting their goal. Hence, the goal of during the 2002-03 was to develop and implement the new 3 tier architectural platform and continue the mass production of the ".com" web-based application systems. Although this creation of the multitalented workers was to promote efficiency by increasing the speed of production and by multitasking a worker most of the production activities to control labor wage, the scarcity of the multitasking professionals continued and the trend of labor wage increase continued as shown on the table 48 of DOI's 2002-03 labor wage data below. As can be seen below on table 48 of the 2002-03 DOL's labor report, it indicates the labor wage increases; the number of labor multitasking workers is adjusted to increase or decrease as needed to accomplish the computer engineering organizations. As DOL's 2002-03 labor wage report suggests, as the multitalented professional *Computer Application Software Engineers* and *Computer Systems Software Engineers* entered into the production process during 2002-2003, it become advantageous for the computer engineering organizations in terms of controlling labor wages because these multitalented professionals were educated in diverse curricula whereby one *Computer Applications Software Engineer* can be assigned to produce analysis & design, programming, database development, testing and deployment activities including users' training as well as maintenance of the systems servers. These are production process activities that could be produced by more than one worker that have their own salary standard. Thus, through this multitasking mechanism, the labor wage could be controlled and from DOL's 2002-03 labor wage report below on table 48, this scenario can be interpreted.

Table 48: Illustrates the salary ranges of computer engineering workers during 2002-03:

Specializations 2002-2003	Annual Salary Range for Experienced 2002-2003	Annual Salary Range for Beginners 2002-2003
Systems Analysts	**$73,210**	**$46,980**
Network Systems & Data Communications Analysts	**$69,970**	**$42,310**
Database Administrators	**$69,920**	**$34,290**
Computer Scientists	**$70,250**	**$54,700**
Application Programmers	**$50,500**	**$58,500**
Programmer Analysts	**$77,750**	**$54,000**
Computer Applications Software Engineers	**$85,490**	**$53,390**
Computer Systems Software Engineers	**$86,520**	**$54,460**
Computer Hardware Engineers	**$86,280**	**$52,960**
Computer support Specialists	**$48,810**	**$28,880**
Communication Systems Administrators	**$59,800**	**$42,800**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2002-2003d

Based on DOL's 2002-03 data on table 48 above, the trends of increasing the multitasking *Division of Labor* force depending on the specific needs of computer engineering organizations'' goals and using the multitasking of production process activities assigning to a worker to control the labor wage continued during 2002-03 and throughout the mid to late 2000's. Hence, during this period, the goal for the computer engineering organizations became to develop and implement the newly created 3 tier architectural platform and the continuation of the ".com" web-based application systems. With this scenario, DOL's 2002-03 labor wage data on table 48 above shows two new incorporations of multitalented professionals like the *Computer Applications Software Engineers* and *Computer Systems Software En*gineers who was educated during the late 90's with diverse curricula to be assigned multitasking in the production process, which in this case to produce the mass production in an efficient manner. As illustrated on table 48 above of DOL's 2002-03 labor data, the experienced *Computer Applications Software Engineers* salary was $85,490 as they entered into the production process for first time and the beginners' salary was $53,390. The salary started off with a high figure suggested that each multitalented professional was used to produce all the required production process activities as a mechanism to control for labor wages. The same report illustrates that the newly entered experienced *Computer Systems Software Engineers* who entered in the production process with previous extensive systems programming experience that became educated in the diverse curricula earning salary was $86,520 and the beginners were making $54,460 to start. According to DOL's 2002-03 labor data, the number of *Computer Applications Software Engineers* and *Computer Systems Software Engineers* for the 3rd (construction) production process phase was 697,000 to start with, which was a high number suggesting that scarcity will be an issue, which means the increase in salary range will be controlled. DOL's 2002-03 data does not include *Systems Programmers* and this suggests that this high starting salary cap for the *Computer Systems Software Engineers* was because this newly incorporated multitalented, *Computer Systems Software Engineers* were replacing the *Systems Programmers* who could only produce a specialized task with no multitasking capabilities. According to the same DOL's report, the experienced *Application Programmers* salary remained the same at $50,500 in 2002-03 as it was during 2000-01 report. This indicates that they were not in demand and their production activities were being gradually replaced by the multitalented Computer Application Software Engineers in which one multitalented worker could simply produce whereby previously took multiple *Application Programmers* to complete a module. With this change, DOL's 2002-03 labor report indicates that the number of *Application Programmers* was reduced to 585,000 from 648,000 of 2000-01 report. However, during 2002-03, the experienced Programmer Analysts showed an increase of $77,750 from $70,610 of DOL's 2000-01 report. This suggests that computer engineering organizations were utilizing the skill sets of one Programmer Analyst to perform programming, systems analysis & design, systems integration architect and coding specification in which each activity used to require individual expertise during the Waterfall production framework (1970-95). During the 2002-03, the labor wage adjustment implemented by computer engineering organizations coincides with the goal of which was the mass production the 3 tier architectural platform. As this was the primary goal, another additional multitalented professional that was the *Computer Hardware Engineers* that replaced both the *Computer Technicians* and the *Computer* and *Office Machine Repairers* activities -- and to produce the computer hardware peripherals used to implement the 3 tier architectural platform. Thus, DOL 2002-03 labor wage data indicates that the newly entered experienced *Computer Hardware Engineers* with previous expertise in computer hardware and who were formerly *Computer Technicians* and the *Computer* and *Office Machine Repairers* that then became educated in the late 90's in the diverse curricula earned $86,280 and the beginners with no previous expertise were making $52,960.

However, since the goal of the computer engineering organizations was mass production during 2002-03, DOL's report indicates a decrease of the experienced *Systems Analysts*, *Computer Scientists* and *Database Administration* of the 1st (inception) production phase was not in demand as it was during the 2000-01 year. Hence, during the 2002-03, the experienced *Computer Scientists* earning remained the same at $70,250 at it was during the 2000-01 year. However, during the same year, adjusting the salary of the beginners as a mechanism of controlling higher cap of the experienced labor wage from increasing was applied, in which, the experienced labor wage of the *Computer Scientists* remained unchanged at $70,250 and an adjustment was made to increase the salary of the beginners *Computer Scientists* to $54,700 from $34,290 of 2000-01. Further intriguing during the same time frame was that in the case of the *Systems Analysts*, the experienced salary was reduced to 73,210 from $74,000 of the 2000-01 report. The same labor wage adjustment mechanism was applied to control the experienced *Systems Analysts* salary from increasing whereby the adjustment was made to increase the beginners *Systems Analysts* salary to $46,980 from $40,570. This is labor wage controlling mechanism makes the experienced salary cap to increase slowly and at time even further reducing it. A surplus value mechanism that was complex to recognize by the average worker as the adjustment makes it seem salary is increasing, which is a form of false consciousness (Michael Burawoy, *Manufacturing Consent*, 1979).

In sum, during the 2002-03, to control labor wage, computer engineering organizations implemented a mechanism where they increased the number of multitalented professionals dependent on what their goal demands. For instance, as their goal was the mass production of the 3 tier architectural platform during the 2002-03, the number of multitalented *Computer Application Software Engineers* and the *Computer Systems Software Engineers* was 697,000 as each one was able to multitask in production process. Their labor wage was started off high in that they were multitalented in all the production process phases, each of which have standardized labor wage whereby one multitalented professional is capable of produced them free of labor cost for the computer engineering organizations, which is illustrated as part of the analysis of the *Triple Surplus Value*, on table 58 of the 2003.

In 2000-01, DOL's data indicated 1.5 million as the as the *Computer Scientists, Systems Analysts* and *Database Administrators* were in demand to invent the concept of the 3 tier as a solution to overcome the performance limitation of the 2nd tier architectural platform, which the increase coincided with the mechanism of controlling the labor wage of the *Computer Scientist*, *Systems Analysts* and *Database Administrators* because if there was a scarcity of them, then their labor wage would not have been controlled. Hence, the increase in numbers was used as a mechanism controls labor wage from increasing. However, as the goal for 2002-03 for computer engineering organization was the mass production, it was evident to see that the salary of the *Computer Scientists* and *Database Administrators* remained constant and the *Systems Analysts* salary was reduced -- and their number reduced to 887,000 from 1.5 million of the 2000-01.

As the goal for the computer engineering organization became an invention of the cluster based 3 tier architectural platforms in 2004-05, the same trend of controlling labor wage and adjustment of the size of the multitasking based *Division of Labor* forces are applied. The table 49 below shows DOL's 2004-05 labor wage, which indicates increases a slight increase in the salary of the *Systems Analysts*, *Computer Scientists*, *Database Administrators*, as they were in demand to invent the cluster based 3 tier architectural platform, computer engineering organization increased their numbers to 979,000 in 2004-05 from 887,000 of the 2002-03 to avoid scarcity. Although they were in demand, this was a mechanism implemented to control their labor wages as they were in demand to invent the new platform.

Table 49: Illustrates the salary ranges of computer engineering workers during 2004-05:

Specializations 2004-2005	Annual Salary Range for Experienced 2004-2005	Annual Salary Range for Beginners 2004-2005
Systems Analysts	**$78,350**	**$49,500**
Network Systems & Data Communications Analysts	**$74,290**	**$44,850**
Database Administrators	**$75,100**	**$40,550**
Computer Scientists	**$78,350**	**$58,630**
Application Programmers	**$60,290**	**$51,500**
Programmer Analysts	**$77,750**	**$51,500**
Computer Applications Software Engineers	**$88,660**	**$55,510**
Computer Systems Software Engineers	**$91,160**	**$58,500**
Computer Hardware Engineers	**$86,280**	**$52,960**
Computer support Specialists	**$51,680**	**$23,060**
Computer Systems Administrators	**$69,530**	**$43,290**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2004-2005d

As described in above, by mid-2000's, computer engineering organization goal was to invent the cluster based 3 tier architectural platform to modify the original 3 tier platform, so that the business continuity and systems failover capabilities were implemented. With this goal as a priority, according to DOL's labor data of 2004-05, *Computer Scientists*, *Computer Analysts* and *Database Administrators* number increased to 979,000 from 887,000 of the 2002-03, which suggested that they became in demand to create the cluster based 3 tier architectural platform. Although these positions were in demand for the goal of the computer engineering organizations, during the same years' DOL labor data, the labor wage for the experienced Computer Scientists showed no significant increase, it slightly increased to $78,350 from $70,250 of the 2002-03 labor report showing an $8,100 increase. During the same years, Systems Analysts' salary also indicated a slight increase to $78,350 from $73,210 of the 2002-03 labor report showing a $5,140 increase. Similarly, during the same time, the Database Administrators salary slightly increased to $75,110 from $71,440 of the 2002-03 labor report. This suggest that computer engineering organizations increased the number of these professionals to 979,000 from 887,000 of the 2002-03, which was increased by 92,000 in just a year timeframe to avoid scarcity, so that they could control a high increase of the labor wage as it could have been increased if there was a scarcity of these positions as they were critical for the creation of the cluster based 3 tier architectural platform. These slight increases of labor wages became the trend right about the mid-2000 because a worker now could produce all the production process activities that have standard wages, which translates to surplus value to the computer engineering organizations by ways of encouraging efficiency and including the MV given to the worker. The surplus labor value advantages that the computer engineering organizations have by paying these amounts is very high compare to what the workers are compensated. Pertaining to this particular instance, the advantage for the computer engineering organizations can be seen on the analysis of the *Triple Surplus Value* on table 59 of the 2004 below showing where the MV is less than the value of the *Actual Profit* (AC). This surplus value mechanism was also applied to the *Computer Application Software Engineers* and the *Computer Systems Software Engineers*.

The of the same surplus value mechanism was applied in the multitalented *Computer Application Software Engineers* and the *Computer Systems Software Engineers* professionals and its manifestation is analyzed using the *Triple Surplus Value* mechanism. During the 2004-05, DOL’s labor data indicated that the experienced *Computer Application Software Engineers* slightly increased to $88,660 from $86,490 of the 2002-03 labor report, which increased by $3,170. Similarly, the experienced *Computer Systems Software Engineers* salary had shown a slight increased to $91,160 from $86,520 of the 2002-03 labor report, which showed an increase of $4,640. Although these multitalented professionals were not in a high demand as they were multitasking the mass production efforts and the goal at the time was to invent of a new platform, computer engineering organizations reduced these multitalented professionals number to 675,000 from 697,000 as each worker was able to multitask the required production process activities of the mass production of the original 3 tier architectural platform. In addition, this surplus value mechanism was applied onto the *Application Programmers* and *Programmer Analysts*.

The experienced *Application Programmers* salary increased in 2004-05 to $60,290 from $50,500, which was increased by $9,790 according to DOL's labor report of 2004-05. However, the experienced *Programmer Analysts* salary remained constant at $77,750 as it was in 2002-03. According to DOL's 2004-05 labor report, computer engineering organization did reduce the number of these professionals to 499,000 from 585,000 of the 2002-03. This suggests that the *Application Programmers* and *Programmer Analysts* were becoming more multitask oriented in that computer engineering organization could allocate multitask to one *Programmer Analyst* to perform the required production process activities and use the multitasking based surplus value mechanism to extract free labor. This trend continues throughout the remaining of the late 2000.

Table 50: Illustrates the salary ranges of computer engineering workers during 2006-07:

Specializations 2006-2007	**Annual Salary Range for Experienced 2006-2007**	**Annual Salary Range for Beginners 2006-2007**
Systems Analysts	**$82,980**	**$52,400**
Computer Scientists	**$108,440**	**$64,860**
Database Administrators	**$81,140**	**$44,490**
Network Systems & Data Communications Analysts	**$78,060**	**$46,480**
Application Programmers	**$83,250**	**$52,500**
Programmer Analysts	**$83,250**	**$52,500**
Computer Applications Software Engineers	**$92,130**	**$59,130**
Computer Systems Software Engineers	**$98,220**	**$63,150**
Computer support Specialists	**$53,010**	**$30,980**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2006-2007d

Table 51: Illustrates the salary ranges of computer engineering workers during 2008-09:

Specializations 2008-2009	**Annual Salary Range for Experienced 2008-2009**	**Annual Salary Range for Beginners 2008-2009**
Systems Analysts	**$95,810**	**$58,460**
Network and Computer Systems Administrators	**$84,110**	**$51,690**
Database Administrators	**$91,850**	**$52,340**
Network Systems & Data Communications Analysts	**$90,740**	**$54,330**
Computer Scientists	**$124,370**	**$75,340**
Application Programmers	**$89,720**	**$52,640**
Programmer Analysts	**$89,720**	**$52,640**
Computer Applications Software Engineers	**$104,870**	**$67,790**
Computer Systems Software Engineers	**$113,960**	**$73,200**
Technical Support Specialists	**$55,990**	**$33,680**
Help-desk Technicians	**$55,990**	**$33,680**

Source: Bureau of Labor Statistics, *Occupational Outlook Handbook*, 2008-2009d

DOL's labor reports of 2006-07 and 2008-09 show that there was a slight labor wage increase for the *Computer Application Software Engineers, Computer Systems Software Engineers*, *Application Programmers* and *Programmer Analysts*. These positions were those involved in the development and implementation production process activities of the *Iterative* frame. During this time, their numbers increased, indicating that they were in demand and that computer engineering organizations were attempting to avoid labor scarcity in these positions in order to prevent increases in labor wage. Hence, during 2006-

07 shown on table 50, the slight increase in the salary of the *Computer Application Software Engineers* and *Software Systems Software Engineers* suggests that computer engineering organizations' primary goal shifted towards mass production of the cluster based 3 tier architectural platform that was created during 2004-05. For instance, the experienced *Computer Applications Software Engineers*' salary was slightly increased to $92,130 from the $88,660 of 2004-05, a growth of $3,470, and the beginners' salary also showed a slight increase to $59,130 from the $55,510 of 2004-05, indicating an increase of $3,620. Also, during 2006-07, the experienced *Computer Systems Software Engineers*' salary rose to $98,220 from the $91,160 of 2004-05 labor report showing an increase of $7,060, and the beginners' salary increased to $63,150 from the $58,500 of 2004-05, showing growth of $4,650. During 2006-07, the total number of *Computer Applications Software Engineers* and *Computer Systems Software Engineers* increased to 800,000 from 675,000, a jump of 125,000, indicating that computer engineering organizations were increasing these numbers to avoid scarcity and control labor wage. However, DOL's 2006-07 data indicated that both the experienced *Application Programmers* and *Programmer Analysts*' salary increased to $83,250 from the $60,290 of 2004-05, which was increase of $22,960, and the beginners' salary increased to $52,500 from the $51,500 of 2006-07, a nuance of $1,000. This indicates that computer engineering organizations also had a goal of continuing the mass production of web-based application systems and upgraded industry specific ".com" applications developed during the late 90's and early 2000 that were in high demand. According to the

DOL labor data, the number of *Computer Programmers* was decreasing from the 697,000 of 2002-03 to 675,000 of 2004-05, and further to 455,000 in 2006-07. Although the *Computer Programmers*' number continued to go down, their salary increased, as they were trained to multitask and perform both the production activities of *Systems Analysts*, *Database Administrators* and *Programmers*. This became advantageous for the computer engineering organizations as it was used as a surplus value mechanism in production process. Although there were indications of salary increase, compared to the *Actual Triple Surplus Value* profit gains shown on the *Triple Surplus Value* table 64, a given *Applications Programmer* still suffers from the application of the surplus value mechanism.

The same trend is found during 2008-09 on table 51 above, whereby the experienced *Computer Application Software Engineers*' salary increased to $104,130 from the $88,660 of 2004-05, which increased by $15,470, and the beginners salary grew to $67,790 from the $59,130 of 2006-07 with an increase of $8,660. During the same timeframe, the experienced *Computer Systems Software Engineers*' salary increased to $113,960 from $98,220 in 2006-07, up by $15,740, and the beginners salary increased to $73,200 from the $63,150 of 2006-07 report, a growth of $10,050. This scenario suggests a continuation of computer engineering organizations' goal of mass producing the newly created cluster based 3 tier architectural platform. As computer engineering organizations utilized 800,000 *Computer Application Software Engineers* and *Computer Systems Software Engineers* in 2006-07 to implement the cluster 3 tier platform, in 2008-09, the number was reduced significantly to 532,200 from the 800,000 of the 2006-07 report. This reduction in numbers suggests that computer engineering organization were conceptualizing to invent an advance data center system known as “cloud computer service systems,” which are remote data center infrastructures (e.g., storage systems, servers and networks) that support the cluster based 3 tier architectural platform as well as web-based application systems for any industry specific market that does not have a physical data center infrastructure. The cloud computer system was meant to be developed with the wireless and internet as a foundation. Users would not even notice that they were using the remote data center infrastructure while systems administration and helpdesk support simultaneously (and easily) accessed the users’ workstations remotely via the internet to address any problems that these users might experience. This concept became advantageous for computer engineering organizations, and the industry was able to save on labor costs due to a decreased need for maintenance. They additionally reduced labor costs for systems support administration and helpdesk production process activities for the industry specific market, such as the ecommerce industry transactional systems. The new concept of remote based cloud data center infrastructure systems brought in new business opportunities for computer engineering organizations, as they had to build it for the industry specific market and provide technical assistance as needed. They preserved ongoing business through supporting the data center by using the multitalented professionals to multitask the required production process activities. With this demand, according to DOL’s 2008-09 labor data, the number of the *Computer Scientists, Database Administrators* and *Network and Computer*

Systems Administrators increased to 961,200 from the 2006-07 labor report. DOL's 2008-09 labor report indicates that the experienced *Computer Scientists*' salary increased to $124,370 from the $108,440 of 2006-07, which was a growth of $15,930, and the beginners' salary increased to $75,340 from the $64,860 of 2006-07 with an additional $10,480. The experienced *Database Administrators*' salary increased to $91,850 during 2008-09 from the $81,140 of 2006-07, up by $10,710, and the beginners' salary increased by $7,850, to $52,340 from the $44,490 of 2006-07. While computer engineering organizations' priority was to invent the cloud data center systems, at the same time, there was a continuing high demand from the industry specific market for a mass production of web-based applications, as well as upgrades to the ".com" application systems that were produced during the late 90's and early 2000's (Kneuper, 2009). DOL's 2008-09 data indicates that both the experienced *Application Programmers* and *Programmer Analysts* salary changed to $89,720 from the $83,250 of 2006-07, which was an increase of $6,470, and the beginners' salary increased to $52,640 from the $52,500 of 2006-07, a bump of $1,000. This indicates that computer engineering organizations attempted to meet demand for the mass production of web-based application systems and upgrades to the industry specific ".com" applications, as an increase in number of the *Computer Programmers* was necessary to do so. With this goal in mind, computer engineering organizations developed multitalented production teams, combining *Computer Software Engineers*, *Computer Programmers* and *Programmer Analysts* to increase the number of professionals working towards this goal to 1.5 million. By doing so, in addition to developing a production infrastructure more conducive to meeting market demands, computer engineering organizations once again avoided scarcity and consequently prevented a high end labor wage increase. And although the salaries of these multitalented professionals became slightly higher due to these workers now performing a greater number of production activities, computer engineering organizations ultimately saved on labor costs, as these production activities previously had their own labor rates that, totaled, were greater than the collective increase in the multitalented professionals' salaries. For instance, as shown in the *Triple Surplus Value* table 52 below, computer engineering organizations were able to multitask a *Programmer Analysts*, who earns $83,250, to perform most of the production process activities that previously had their own separate standard labor rate.

Recall that to analyze and measure the manifestation of surplus value between 1996 and 2009, this study proposed a *Triple Surplus Value* methodology. Applying this formula, in order to measure and analyze the manifestation of surplus value during this time period, the *Triple Surplus Value* methodology uses data from the following as inputs: 1) DOL labor wage data to get standardized labor wage values, 2) the ProQuest Historical Newspaper database that archives job postings to get the *Number of Production Process Activities* listed within the job postings for 1996-2009, 3) the job posting archival database search engine, called *Job Search Focus Group*, which was developed and is currently maintained by *Job Search Focus Group*, Avenir Technologies, LLC, 2000-2010, to get the *Number of Production Process Activities* as inputs to find the $\sum$ NPPA = N for the years 2000-2009. Based on this input data, the *Triple Surplus Value* methodology measures and analyzes the occurrence of surplus values within the computer engineering organizations' production processes while the computer systems are still in the production process. Below, in tables 52 through 64, both the MV and the GTSV *Triples Surplus Value* formulas measure the manifestation of *Triple Surplus Value* using the data and scenarios from the job postings as well as DOL labor wage standard data. The first example in tables 52a and 52b demonstrates the measurement and analysis of the *Triple Surplus Value*, using the MV *Triple Surplus Value* formula, in a scenario where a *Programmer Analyst*'s OLW is higher than the DOL standard labor wage for a *Programmer Analyst* in 1998:

Table 52a: Illustrates the OLW, HLW, CLW and MV amounts of a *Programmer Analysts* in 1998:

Offered Labor Wage (OLW) for a *Programmer Analyst* in 1998	Historical Labor Wage (HLW) for *Programmer Analyst* in 1985	Converted Labor Wage (CLW) Amount for 1998	Motivation Value (MV) for 1998
$55,195	**$23,800**	**$36,053.90**	**$19,141.10**

Table 52b: Illustrates the NPPA, OLV, GTSV and ATSV of a *Programmer Analyst* in 1998:

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Computer Scientists	Create new concepts of computer systems.	**$62,000**
Computer Engineers	Implement new concepts of computer systems and produce proof of concepts.	**$62,000**
Systems Analysts	Analyze, design, create systems specifications and develop code specifications.	**$54,000**
Database Administrators	Develop databases and maintain database spaces.	**$67,000**
Computer Support Analysts	Analyze and produce input information for programmers.	**$55,000**
Computer Operators	Translate programming codes and develop user manuals.	**$30,900**
Computer Service Technicians	Performs hardware/software systems life cycle day-to-day tech support duties.	**$26,700**
NPPA Amount		**$357,600**
Offered Labor Wage (OLW)		**$55,195**
***Gross Triple Surplus Value* (GTSV)**		**$302,405**
Motivation Value (MV)		**$19,141.10**
Actual Triple Surplus Value (ATSV)		**$283,263.90**

Resource: Classified ad 3 -- no title. (1998, *New York Times (1923-Current File)*, pp. F7.

Table 52a above shows the calculation of the OLW, HLW, CLW and MV for the 1998 job posting of an experienced *Programmer Analyst*. Based on the job posting OLW for this position in Appendix E1, the OLW ($55,195) was higher than the DOL standard salary for a *Programmer Analyst*, which was $52,000 in 1998. Hence, in this instance, since the OLW is higher than DOL's standard salary for the same position, the MV *Triple Surplus Value* method will be applied to measure and analyze the manifestation of the *Triple Surplus Value*. In this case, according to the job description in Appendix E1, the *Programmer Analyst* was assigned to multitask, performing the production activities of the *Computer Scientist, Computer Engineer, Systems Analyst, Database Administrator, Computer Support Analyst, Computer Operator* and *Computer Service Technician*'s, which made the ∑ NPPA = $357,600. To get the GTSV, the OLW is deducted from the ∑ NPPA ($357,600 - $55,195 = $302,405). To get the ATSV, subtract the MV from the GTSV ($302,405 - $19,141.10 = $283,263.90). Note that the MV amount is captured when the CLW amount is subtracted from the OLW. In the end, $283,263.90 is the ATSV amount that goes to the computer engineering organization as a profit and the $19,141.10 goes to the *Programmer Analyst* as an MV. Below, table 53a & 53b demonstrate a similar scenario where the job posting was specified for a beginner *Programmer Analysts* with a higher OLW amount than the DOL standard labor wage for the beginner *Programmer Analyst*, again requiring the application of the MV *Triple Surplus Value* method.

Table 53a: Illustrates the OLW, HLW, CLW and MV amounts of a *Programmer Analyst* in 1998:

Offered Labor Wage (OLW) for a *Programmer Analyst* in 1998	Historical Labor Wage (HLW) for *Programmer Analyst* in 1985	Converted Labor Wage (CLW) Amount for 1998	Motivation Value (MV) for 1998
$50,225	**$23,800**	**$36,053.90**	**$26,425**

Table 53b: Illustrates the NPPA, OLV, GTSV and ATSV of a *Programmer Analyst* in 1998:

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Computer Engineers	Production process activities remain the same.	**$45,000**
Systems Analysts	Production process activities remain the same.	**$43,500**
Database Administrators	Production process activities remain the same.	**$54,000**
Computer Support Analysts	Production process activities remain the same.	**$34,100**
Systems Programmer	Production process activities remain the same.	**$47,500**
Computer Operators	Production process activities remain the same.	**$16,600**
Computer Service Technicians	Production process activities remain the same.	**$20,000**
NPPA Amount		**$260,700**
Offered Labor Wage (OLW)		**$50,225**
***Gross Triple Surplus Value* (GTSV)**		**$210,475**
Motivation Value (MV)		**$26,425**
Actual Triple Surplus Value (ATSV)		**$218,950**

Resource: Classified ad 3 -- no title. (1998, *New York Times (1923-Current File)*, pp. F7.

Table 53a & 53b shows a situation in which a beginner *Programmer Analyst* with 2½ years of experience had an OLW that was $50,225, and this OLW amount is more than DOL's standard labor wage for beginners, which was $30,700. Hence, in this case, as the OLW amount was higher than the DOL standard, the MV *Triple Surplus Value* method is applied. According to the 1998 job description in Appendix E2 for this position, there were 6 required production process activities, those of the *Computer Engineer*, *Systems Analyst, Database Administrator*, *Computer Support Analyst*, *Systems Programmer*, *Computer Operator* and *Computer Service Technician*, which made the ∑ NPPA = $260,700. Following the formula, to get the GTSV, subtract the OLW from the ∑ NPPA ($260,700 - $50,225 = $210,475). Then, to get the ATSV, the MV is subtracted from the GTSV ($210,475 - $26,425 = $184,050). $184,050 goes to the computer engineering organizations as profit, and $26,425 goes to the *Programmer Analyst* as an MV amount.

Table 54a: Illustrates the OLW, HLW, CLW and MV amounts of a *Systems Programmer* in 1999:

Offered Labor Wage (OLW) for a *Systems Analyst* in 1999	Historical Labor Wage (HLW) for *Systems Analyst* in 1989-90	Converted Labor Wage (CLW) Amount for 1999	Motivation Value (MV) for 1999
$100,000	**$33,120**	**$44,498.32**	**$55,501.68**

Table 54b: Illustrates the NPPA, OLV, GTSV and ATSV of a *Systems Programmer* in 1999:

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Computer Engineers	Production process activities remain the same.	**$62,000**
Database Administrators	Production process activities remain the same.	**$67,000**
Computer Support Analysts	Production process activities remain the same.	**$55,000**
Application Programmers	Production process activities remain the same.	**$30,900**
Systems Programmers	Production process activities remain the same.	**$26,700**
Programmer Analysts	Production process activities remain the same.	**$52,000**
NPPA Amount		**$293,600**
Offered Labor Wage (OLW)		**$100,000**
***Gross Triple Surplus Value* (GTSV)**		**$193,600**
Motivation Value (MV)		**$55,501.68**
Actual Triple Surplus Value (ATSV)		**$138,098.32**

Resource: Classified ad 23 -- no title. (1999, *New York Times (1923-Current File)*, pp. W19.

As tables 54a & 54b indicate, a 1999 job posting for an experienced *Systems Analyst* showed an OLW of $100,000 that was higher than the DOL standard labor wage for *Systems Analysts*, which was $54,000 during the same year. Hence, as with the above instances, to measure the manifestation of the *Triple Surplus Value* where profit is gained by the computer engineering organizations, the MV *Triple Surplus Value* method should be applied. According to the job posting's description of qualification criteria in Appendix E3, the experienced *Systems Analyst* was required to perform the production process activities of 6 positions: *Computer Engineer, Database Administrator, Computer Support Analyst, Application Programmer, Systems Programmer* and *Programmer Analyst*, which made the ∑ NPPA = $293,600. Thus, $293,600 - $100,000 = $193,600 became the amount for the GTSV, and $193,600 - $55,501 = $138,098.32 is the ATSV that finally goes in to the computer engineering organization, while $55,501 goes to the experienced *Systems Analyst* as the MV amount.

Table 55a: Illustrates the OLW, HLW, CLW and MV amounts of a *Systems Programmer* in 2000

Offered Labor Wage (OLW) for a *Systems Programmer* in 2000	Historical Labor Wage (HLW) for *Systems Programmer* in 1996-97	Converted Labor Wage (CLW) Amount for 2000	Motivation Value (MV) for 2000
$120,000	**$54,000**	**$59,265.77**	**$60,734.23**

Table 55b: Illustrates the NPPA, OLV, GTSV and ATSV of a *Systems Programmer* in 2000

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Systems Analyst	Production process activities remain the same.	**$74,000**
Computer Engineer	Production process activities remain the same.	**$80,500**
Database Administrator	Production process activities remain the same.	**$69,920**
Computer Support Analyst	Production process activities remain the same.	**$48,810**
Application Programmer	Production process activities remain the same.	**$50,500**
Programmer Analyst	Production process activities remain the same.	**$70,610**
Computer Technician	Production process activities remain the same.	**$30,900**
Computer and Office Machine Repairer	Production process activities remain the same.	**$30,900**
NPPA Amount		**$456,140**
Offered Labor Wage (OLW)		**$120,000**
***Gross Triple Surplus Value* (GTSV)**		**$366,140**
Motivation Value (MV)		**$60,734.23**
Actual Triple Surplus Value (ATSV)		**$275,405.77**

Resource: Classified ad 563 -- no title. (2000, *New York Times (1923-Current File)*, pp. W29.

Table 55a indicates the OLW, HLW, CLW and MV calculated amounts for an experienced *Systems Programmer* using the MV *Triple Surplus Value* method. The MV *Triple Surplus Value* formula is applied because at the time of the job positing for the *Systems Programmer* in 2000, the OLW ($120,000) was higher than DOL's standard salary ($63,000) for the same year. In this case, table 55b lists the job posting's required criteria for hiring a qualified *Systems Programmer* to perform the production activities of the following 8 positions: *Systems Analyst, Computer Engineer, Database Administrator, Computer Support Analyst, Application Programmer, Programmer Analyst, Computer Technician,* and *Computer & Office Machine Repairer*, which made the ∑ NPPA = $456,140. Hence, ∑ NPPA ($456,140) – OLW ($120,000) = $366,140 finds the value of the GTSV. Thus, GTSV ($366,140) – MV ($60,734.23) = $275,405.77 gives the ATSV for the computer engineering organizations, and in this particular instance, $60,734.23 is the MV amount awarded to the *Systems Programmer*.

Table 56a: Illustrates the OLW, HLW, CLW and MV amounts of a *Programmer Analyst* in 2001:

Offered Labor Wage (OLW) for a *Programmer Analyst* in 2001	**Historical Labor Wage (HLW) for *Programmer Analyst* in 1996-97**	**Converted Labor Wage (CLW) Amount for 2001**	**Motivation Value (MV) for 2001**
$130,000	**$47,000**	**$53,050.99**	**$76,949.01**

Table 56b: Illustrates the NPPA, OLW, GTSV and ATSV of a *Programmer Analyst* in 2001:

Number of Production Process Activities (NPPA)	**Description of Production Activities**	**Total Values**
Systems Analyst	Production process activities remain the same.	**$74,000**
Computer Engineer	Production process activities remain the same.	**$80,500**
Database Administrator	Production process activities remain the same.	**$69,920**
Computer Support Analyst	Production process activities remain the same.	**$48,810**
Application Programmer	Production process activities remain the same.	**$50,500**
Systems Programmer	Production process activities remain the same.	**$63,000**
Computer Technician	Production process activities remain the same.	**$30,900**
Computer and Office Machine Repairer	Production process activities remain the same.	**$30,900**
NPPA Amount		**$448,530**
Offered Labor Wage (OLW)		**$130,000**
***Gross Triple Surplus Value* (GTSV)**		**$318,530**
Motivation Value (MV)		**$76,949.01**
Actual Triple Surplus Value (ATSV)		**$241,580.99**

Resource: Classified ad 461 -- no title. (2001, *New York Times (1923-Current File)*, pp. W17.

Table 56a shows the 2001 job posting for an experienced *Programmer Analyst*, in which the OLW was $130,000, the HLW $47,000, the CLW $53,050.99 and the MV $32,139.41. In this particular case, the MV *Triple Surplus Value* method is used to measure the manifestation of the *Triple Surplus Value* because at the time of the job positing for the experienced *Programmer Analyst* 2000, the OLW ($130,000) was higher than DOL's standard salary for the same year ($70,610). Thus, according to the job posting description listed in Appendix E5, table 56b shows this position required an expertise in the following 8 positions' daily production processes activities: *Systems Analyst, Computer*

Engineer, Database Administrator, Computer Support Analyst, Computer Programmer, Computer Technician, Computer and *Office Machine Repairer*, which made for a ∑ NPPA of $448,530. Thus, ∑ NPPA ($448,530) - OLW ($130,000) = $318,000 is the GTSV. GTSV (318,000) – MV ($76,949.01) = $241,580.99 provides the amount of the ATSV profited to the computer engineering organization, and $76,949.01 went to the *Programmer Analyst* as an MV amount.

Table 57a: Illustrates the OLW, HLW, CLW and MV amounts of a *Systems Analyst* in 2002:

Offered Labor Wage (OLW) for a *Programmer Analyst* in 2002	Historical Labor Wage (HLW) for *Programmer Analyst* in 1996-97	Converted Labor Wage (CLW) Amount for 2002	Motivation Value (MV) for 2002
$85,190.40	**$36,000**	**$41,277.25**	**$43,912.75**

Table 57b: Illustrates the NPPA, OLW, GTSV and ATSV of a *Systems Analyst* in 2002:

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Systems Analyst	Production process activities remain the same.	**$46,980**
Network Systems & Data Communications Analyst	Produce configurable network communication codes and implement wide/local area network systems.	**$42,310**
Database Administrator	Production process activities remain the same.	**$34,290**
Application Programmer	Production process activities remain the same.	**$58,500**
Computer Applications Software Engineer	Perform full life cycle development for application systems.	**$53,390**
Computer Systems Software Engineer	Perform full life cycle software development both for hardware and software systems.	**$54,460**
Computer support Specialist	Production process activities remain the same.	**$28,880**
Communication Systems Administrators	Production process activities remain the same.	**$42,800**
NPPA Amount		**$361,610**
Offered Labor Wage (OLW)		**$85,190.40**
***Gross Triple Surplus Value* (GTSV)**		**$276,419.60**
Motivation Value (MV)		**$43,912.75**
Actual Triple Surplus Value (ATSV)		**$232,506.85**

Resource: Classified ad 9 -- no title. (2002, *New York Times (1923-Current File)*, pp. J3.

Table 57a & 57b above measures the manifestation of the *Triple Surplus Value* in the 2002 scenario of a job position for a beginner *Programmer Analyst*, specifying an OLW of $85,190.40, an HLW of $36,000, a CLW of $41,277.25 and a MV of $43,912.75. In this scenario, the MV *Triple Surplus Value* formula is applied, because at the time of the job posting for the beginner *Programmer Analyst*, the OLW ($85,190.40) was higher than DOL's standard labor wage for the same position during the same year ($54,000). As described in the job posting in Appendix E6, the beginner *Programmer Analyst* was required to have expertise in 8 positions' core business process activities: *Systems Analyst*, *Network Systems & Data Communications Analysts*, *Database Administrator*, *Application Programmer*, *Computer Applications Software Engineer*, *Computer Systems Software Engineer*, *Computer Support Analyst*, *Communication Systems Administrator*, whereby ∑ NPPA = $361,610. Hence, ∑ NPPA ($361,610) – OLW ($85,190.40) = $276,419.60 provides the GTSV, and the GTSV ($276,419.60) – MV ($43,912.75) = $232,506.85 shows an ATSV that goes to the computer engineering organization, while $43,912.75 is given to the beginner *Programmer Analyst* as an MV amount.

Table 58a: Illustrates the OLW, HLW, CLW and MV amounts of a *Programmer Analyst* in 2003:

Offered Labor Wage (OLW) for a *Programmer Analyst* in 2003	**Historical Labor Wage (HLW) for *Programmer Analyst* in 1996-97**	**Converted Labor Wage (CLW) Amount for 2003**	**Motivation Value (MV) for 2003**
$79,200.40	**$36,000**	**$41,277.25**	**$37,923.15**

Table 58b: Illustrates the NPPA, OLW, GTSV and ATSV of a *Programmer Analyst* in 2003:

Number of Production Process Activities (NPPA)	**Description of Production Activities**	**Total Values**
Systems Analyst	Production process activities remain the same.	**$46,980**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$42,310**
Database Administrator	Production process activities remain the same.	**$34,290**
Application Programmer	Production process activities remain the same.	**$58,500**
Computer Applications Software Engineer	Production process activities remain the same.	**$53,390**
Computer Systems Software Engineer	Production process activities remain the same.	**$54,460**
Computer support Specialist	Production process activities remain the same.	**$28,880**
Communication Systems Administrator	Production process activities remain the same.	**$42,800**
NPPA Amount		**$361,610**
Offered Labor Wage (OLW)		**$79,200.40**
***Gross Triple Surplus Value* (GTSV)**		**$282,409.60**
Motivation Value (MV)		**$37,923.15**
Actual Triple Surplus Value (ATSV)		**$244,486.45**

Classified ad 17 -- no title. (2003, *New York Times (1923-Current File),* pp. HW3.

Table 58a above shows the calculations of the OLW, HLW, CLW and MV of a 2003 *Programmer Analyst* as part of measuring the manifestation of the *Triple Surplus Value* in this particular instance. In this instance, the MV *Triple Surplus Value* formula is applied as the OLW ($79,200.40) is higher than the HLW ($54,000) of the DOL standard labor wage for the beginner *Programmer Analyst* that year. Per the job posting description for the beginner *Programmer Analyst* in Appendix E7, he or she must possess the skill sets of 8 positions: *Systems Analyst, Network Systems & Data Communications Analyst, Database Administrator, Application Programmer, Computer Applications Software*

Engineer, Computer Systems Software Engineer, Computer Support Analyst, Communication Systems Administrator, in which ∑ NPPA = $361,610. ∑ NPPA ($361,610) – OLW ($79,200.40) = $380,409.60 gives the GTSV, and GTSV ($282,409.60) – MV ($37,923.15) = $244,486.45 provides the ATSV, which was profited by the computer engineering organization, and $37,923.15 went to the beginner *Programmer Analyst* as an MV amount.

Table 59a: Illustrates the OLW, HLW, CLW and MV amounts of a *Programmer Analyst* in 2004:

Offered Labor Wage (OLW) for a *Programmer Analyst* in 2004	**Historical Labor Wage (HLW) for *Programmer Analyst* in 1998-99**	**Converted Labor Wage (CLW) Amount for 2004**	**Motivation Value (MV) for 2004**
$87,675.12	**$30,700**	**$35,578.10**	**$52,097.02**

Table 59b: Illustrates the NPPA, OLW, GTSV and ATSV of a *Programmer Analyst* in 2004:

Number of Production Process Activities (NPPA)	**Description of Production Activities**	**Total Values**
Systems Analyst	Production process activities remain the same.	**$49,500**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$44,850**
Database Administrator	Production process activities remain the same.	**$40,550**
Application Programmer	Production process activities remain the same.	**$51,500**
Computer Applications Software Engineer	Production process activities remain the same.	**$55,510**
Computer Systems Software Engineer	Production process activities remain the same.	**$58,500**
Computer support Specialist	Production process activities remain the same.	**$23,060**
Communication Systems Administrator	Production process activities remain the same.	**$43,290**
NPPA Amount		**$366,760**
Offered Labor Wage (OLW)		**$87,675.12**
***Gross Triple Surplus Value* (GTSV)**		**$279,084.88**
Motivation Value (MV)		**$52,097.02**
Actual Triple Surplus Value (ATSV)		**$226,987.86**

Classified ad 25 -- no title. (2004, *New York Times (1923-Current File)*, pp. W3.

Table 59a above illustrates the calculated values of the OLW, HLW, CLW and MV for the beginner *Programmer Analyst* job posting in 2004. In this case, the OLW was \$87,675.12, which was higher than DOL's standard labor wage (\$77,750) for the *Programmer Analyst* during the same year. According to the job posting description in Appendix E8, the qualification for this position included the skills needed for 8 positions' production process activities: *Systems Analyst, Network Systems & Data Communications Analyst, Database Administrator, Application Programmer, Computer Applications Software Engineer, Computer Systems Software Engineer, Computer Support Analyst,* and *Communication Systems Administrator*, which made the ∑ NPPA = \$366,760. Thus, ∑ NPPA (\$366,760) – OLW (\$87,675.12) = \$279,084.88 is the GTSV. GTSV (\$279,084.88) – MV (\$52,097.02) = \$226,987.86, which became the ATSV profit gained by the computer engineering organization, and \$52,097.02 was the MV amount awarded to the beginner *Programmer Analyst*.

Table 60a: Illustrates the OLW, HLW, CLW and MV amounts of a *Computer Applications Software Engineer* in 2005:

Offered Labor Wage (OLW) for a *Computer Applications Software Engineer* in 2005	**Historical Labor Wage (HLW) for *Programmer Analyst* in 2002-03**	**Converted Labor Wage (CLW) Amount for 2005**	**Motivation Value (MV) for 2005**
$110,000	**$85,490**	**$92,808.21**	**$17,191.79**

Table 60b: Illustrates the NPPA, OLW, GTSV and ATSV of a *Computer Applications Software*
Engineer in 2005:

Number of Production Process Activities (NPPA)	**Description of Production Activities**	**Total Values**
Systems Analyst	Production process activities remain the same.	**$78,350**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$74,290**
Database Administrator	Production process activities remain the same.	**$75,100**
Application Programmer	Production process activities remain the same.	**$60,290**
Programmer Analyst	Production process activities remain the same.	**$77,750**
Computer Systems Software Engineer	Production process activities remain the same.	**$91,160**
Computer support Specialist	Production process activities remain the same.	**$51,680**
Communication System Administrator	Production process activities remain the same.	**$69,530**
NPPA Amount		**$578,150**
Offered Labor Wage (OLW)		**$110,000**
***Gross Triple Surplus Value* (GTSV)**		**$468,150**
Motivation Value (MV)		**$17,191.79**
Actual Triple Surplus Value (ATSV)		**$450,958.21**

Table 60a above shows the OLW, HLW, CLW and MV of an experienced *Computer Applications Software Engineer*'s job posting in 2005. The OLW ($110,000) is higher than DOL's standard labor wage ($88,660) for the position during 2005. According to the job posting in Appendix E9, the experienced *Computer Applications Software Engineer* had to possess the expertise needed for 8 positions' production process activities: *Systems Analyst, Network Systems & Data Communications Analyst, Database Administrator, Application Programmer, Programmer Analyst, Computer Applications Software Engineer,* *Computer Systems Software Engineer, Computer Support Analyst, Communication Systems Administrator*, which made the ∑ NPPA = $578,150. Thus, ∑ NPPA ($578,150) – OLW ($110,000) = $468,150 gives the GTSV. GTSV ($468,150) – MV ($17,191.79) = $450,958.21 provides the ATSV as the profit gained by the computer engineering organization, and the experienced *Computer Applications Software Programmer* takes $17,191.79 as the MV.

Table 61: Illustrates the NPPA, OLV and GTSV amount of a *Computer Applications Software Engineer* in 2006:

Number of Production Process Activities (NPPA) 2006	Description of Production Activities	Total Values 2006
Systems Analyst	Production process activities remain the same.	**$82,980**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$78,060**
Database Administrator	Production process activities remain the same.	**$81,140**
Application Programmer	Production process activities remain the same.	**$83,250**
Programmer Analyst	Production process activities remain the same.	**$83,250**
Computer Systems Software Engineer	Production process activities remain the same.	**$98,220**
Computer Support Specialist	Production process activities remain the same.	**$53,010**
NPPA Amount		**$559,910**
Offered Labor Wage (OLW) for *Computer Applications Software Engineer*		**$65,000**
***Gross Triple Surplus Value* (GTSV)**		**$494,910**

Classified ad 7 -- no title. (2006, *New York Times (1923-Current File)*, pp. I2.

Table 61 presents a scenario of an underpaying job posting for an experienced *Computer Applications Software Engineer* and the application of the GTSV *Triple Surplus Value* formula in such instances. According to the 2006 job posting in Appendix E10, the OLW ($65,000) was lower than DOL’s standard labor wage ($92,130) for the position during 2006. The job posting description required the *Computer Applications Software Engineer* candidate to have expertise in the following 7 positions’ production processes: *Systems Analyst*, *Network Systems & Data Communications Analyst, Database Administrator, Application Programmer, Programmer Analyst, Computer Systems Software Engineer, Computer Support Analyst,* which made the ∑ NPPA = $559,910. Thus, ∑ NPPA ($559,910) – OLW (65,000) = $494,910 gives the GTSV that goes to the computer engineering organization as a profit with, in this case, no MV going to the *Computer Applications Software Engineer*, as the labor rate was already lower than DOL's standard for 2006.

Table 62: Illustrates the NPPA, OLV and GTSV amount of a *Computer Applications Software*
Engineer in 2007:

Number of Production Process Activities (NPPA) 2007	Description of Production Activities	Total Values 2007
Systems Analysts	Production process activities remain the same.	**$82,980**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$78,060**
Database Administrator	Production process activities remain the same.	**$81,140**
Application Programmer	Production process activities remain the same.	**$83,250**
Programmer Analysts	Production process activities remain the same.	**$83,250**
Computer Systems Software Engineer	Production process activities remain the same.	**$98,220**
Computer support Specialist	Production process activities remain the same.	**$53,010**
NPPA Amount		**$559,910**
Offered Labor Wage (OLW) for *Computer Applications Software Engineer*		**$60,000**
Actual Triple Surplus Value (ATSV)		**$494,905**

Classified ad 4 -- no title. (2007, *New York Times (1923-Current File)*, pp. C20.

Table 62 presents the ∑ NPPA, OLW and GTSV for the *Computer Applications Software Engineer* from the 2007 job posting. The GTSV *Triple Surplus Value* formula is selected, as this was a scenario of underpayment given that the OLW ($60,000) was less than DOL's standard labor wage ($104,870) for an experienced *Computer Applications Software Engineer* that year. According to Appendix E11, the job posting description required the *Computer Applications*

Software Engineer candidate to have expertise in the following 8 positions' production process activities: *Systems Analyst, Network Systems & Data Communications Analyst, Database Administrator, Application Programmer, Programmer Analyst, Computer Systems Software Engineer, Computer Support Specialist,* which made the ∑ NPPA = $559,910. Thus, ∑ NPPA ($559,910) – OLW ($60,000) = $499,910 gives the GTSV that goes to the computer engineering organizations as a profit, again with no MV to the *Computer Applications Software Engineer*, as the labor rate was already lower than DOL's standard for 2007.

Table 63: Illustrates the NPPA, OLV and GTSV amount of an *Applications Programmer* in 2008:

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Systems Analyst	Production process activities remain the same.	**$95,810**
Network and Computer Systems Administrators	Production process activities remain the same.	**$84,110**
Database Administrator	Production process activities remain the same.	**$91,850**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$90,740**
Programmer Analyst	Production process activities remain the same.	**$89,720**
Computer Applications Software Engineer	Production process activities remain the same.	**$104,870**
Computer Systems Software Engineer	Production process activities remain the same.	**$113,960**
Technical support Specialist	Production process activities remain the same.	**$55,990**
NPPA Amount		**$727,050**
Offered Labor Wage (OLW) for *Applications Programmer*		**$50,000**
Actual Triple Surplus Value (ATSV)		**$677,050**

Table 63 presents the∑ NPPA, OLW and GTSV for the *Applications Programmer* from the 2008 job posting. The GTSV *Triple Surplus Value* formula is selected, as this was a scenario of underpayment, since the OLW ($50,000) was less than DOL's standard labor wage ($89,720) for an experienced *Applications Programmer* for 2008. According to Appendix E12, the job posting description required the candidate for this position to have expertise in the following 8 positions' production processes: *Systems Analyst, Network and Computer Systems Administrators, Network Systems & Data Communications Analyst, Database Administrator, Programmer Analyst, Computer Applications Software Engineer, Computer Systems Software Engineer and Technical Support Specialist* which made the ∑ NPPA = $727,050. Thus, ∑ NPPA $727,050) – OLW ($50,000) = $677,050 gives the GTSV that goes to the computer engineering organizations as a profit with no MV to the *Applications Programmer* since, again, the labor rate was already lower than DOL's standard for 2008.

Table 64: Illustrates the NPPA, OLV and GTSV amount of an *Applications Programmer* in 2009:

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Systems Analyst	Production process activities remain the same.	**$95,810**
Network and Computer Systems Administrator	Production process activities remain the same.	**$84,110**
Database Administrator	Production process activities remain the same.	**$91,850**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$90,740**
Programmer Analyst	Production process activities remain the same.	**$89,720**
Computer Applications Software Engineer	Production process activities remain the same.	**$104,870**
Computer Systems Software Engineer	Production process activities remain the same.	**$113,960**
Technical support Specialist	Production process activities remain the same.	**$55,990**
NPPA Amount		**$727,050**
Offered Labor Wage (OLW) for *Applications Programmer*		**$70,000**
Actual Triple Surplus Value (ATSV)		**$677,050**

Table 64 presents the∑ NPPA, OLW and GTSV for the 2009 *Applications Programmer* job posting. The GTSV *Triple Surplus Value* formula is selected because the OLW ($70,000) was less than DOL's 2009 standard labor wage ($89,720) for an experienced *Applications Programmer*. According to Appendix E13, the job posting description required the *Applications Programmer* to have expertise in the following 8 positions' production process activities: *Systems Analyst, Network Systems & Data Communications Analyst, Database Administrator, Network Systems & Data Communications Analyst, Programmer Analyst, Computer Applications Software Engineer, Computer Systems Software Engineer and Technical Support Specialist* which made the ∑ NPPA = $727,050. Thus, ∑ NPPA ($727,050) – OLW ($70,000) = $657,050 gives the GTSV that goes to the computer engineering organizations as profit. No MV went to the *Applications Programmer* because the labor rate was already lower than DOL's standard for 2009.

4.5 Conclusion

In conclusion, chapter 4 analyzed the development of the Age of Information's modes of production (upside-down organizational structure, macro-matrix management infrastructure and the production infrastructure), which transformed the computer engineering organizations' industrial based, pre-1995 modes of production (pyramid organizational structure, hierarchy based management infrastructure and the top-down manual based production infrastructure). By the mid-90's, computer engineering organizations sought a replacement for the manual and top-down modes of production that they used from 1970-95, as they found it to be limited and inefficient infrastructure for doubling their mass production efforts while controlling labor costs. For example, during the early part of the 70's, computer engineering organizations' production environment was centralized in one facility commonly known as the "computer room," where there was a large mainframe computer that workers shared as a production device for the 3 different shifts. This delayed the production process and provided very limited surplus labor gains. So, by the early to mid-80's, computer engineering organizations were preparing to transform the shared based production process to the individual based production process, and the large mainframe production devices were replaced by the single function based minicomputer systems to promote efficiency. This production infrastructure enabled each specialized worker to produce individually, as a production device was allocated for every individual worker to increase the speed of the production process, which was not possible under the large mainframe system. As this production infrastructure was being implemented, computer engineering organizations began to gain surplus labor by increasing the speed of production and number of workers to decrease labor wages as a mechanism to control labor costs in production process. However, computer engineering organizations found this surplus mechanism insufficient, as it was manual based and cumbersome to go through the process of increasing the number of specialized workers and adjusting labor wage ranges constantly to control labor costs to gain free labor that could be used for other projects' production processes. Thus, beginning in the late 80's and through the mid-90's, computer engineering organizations developed new modes of production that completely transformed the entire modes of production that existed pre-1995. Computer engineering organizations were creating the internet and wide area networks and multifunctional capable computer systems in parallel as they were conceptualizing the type of modes of production that were going to be developed using the internet, wide area networks and automated based multifunctional workstation computer systems. Hence, computer engineering organizations no longer needed the top-down rigid organizational structure, the hierarchy management infrastructure or the single functioned manual based computer systems. Thus, computer engineering organizations replaced the pyramid organizational structure with the upside-down pyramid organizational structure.

This new organizational structure was created with a foundation of an open communication structure, which set the base for the development of a new macro-matrix management and automated multifunctional based computer systems. Using the advanced computer systems, computer engineering organizations automated the macro-matrix management infrastructure mechanisms (WBS, Gantt Chart and BOE), which were used to plan, track production statuses and control labor costs. The emergence of this new structure enabled management mechanisms to be practiced remotely, whereby a team lead or a project lead could simply monitor production progress from anywhere, as the traditional facility based computer-room production environment was no longer needed. However, to make this management a success, the production infrastructure had to be developed with the advanced computer systems (internet, wide area networks and multifunctional computer systems) as a foundation, meaning that the appropriate production framework, production process activities and *Division of Labor* forces had to be created, so that these could be used as inputs for the management WBS, Gantt Chart and BOE for planning purposes.

With this in demand, computer engineering organization started developing a production infrastructure that was compatible with the macro-based management infrastructure. Their 1st result was the development of the *Iterative* production process framework. The *Iterative* framework replaced the *Waterfall* framework that existed pre-mid 90's. The *Iterative* framework defined the production phases and the production activities, and based on these, it set the foundation for the development of the diverse academic curricula to train the multitalented professionals that were needed for the new multitasking based production process. The production framework and production activities were used as inputs for the WBS to specify the production activities assigned to the multitalented professional and allocate labor costs, all in an integrated and automated management system (like Microsoft Project).

This constitutes the new modes of production that were the linear based, upside-down pyramid organizational structure, macro-matrix management and the automated multifunction based production infrastructure (*Iterative* production framework, multitasking production process and multitasking based *Division of Labor*), which could all be pronounced as the modes of production for the Age of Information beginning in the mid-90's. With this new modes of production availability, computer engineering organizations began to use the modes of production to promote efficiency by assigning a multitalented professional to multitask the positions' production processes of multiple positions' activities. Each position had its own standard labor wage, and with this new system in place, many production activities were being performed free of labor costs, as the new multitalented professional were executing production process activities of multiple positions without receiving proportional compensation. In addition, the speed of production process was still part of the surplus mechanism to increase mass production using the automated based multifunctional production devices. This study calls this combination of surplus mechanism a *Triple Surplus Value*. To measure and analyze the manifestation of this form of surplus value, this dissertation formulated a *Triple Surplus Value* formula, which was illustrated in section 4.5. Using DOL's 39 years of records on computer engineering organizations' *Division of Labor* and labor wage records, 12 job posting records, illustrated on tables 52 through 64, were analyzed both by the MV and GTSV *Triple Surplus Value* formulas to find the manifestation of *Triple Surplus Value*. The results yielded that the *Triple Surplus Value* has been practiced since the mid-90's and is still in practice today, indicating that computer engineering organizations have been successful in promoting efficiency, which is equivalent to the practice of the contemporary surplus value mechanism of doubling up their mass production and securing profits while controlling labor wage in production process. As analyzed throughout this chapter, computer engineering organizations have continued to transform their modes of production, generating effective mechanisms that best control labor wage in production process, which workers perceive as promoting efficiency and improving productivity.

CHAPTER 5

CONCLUSION

Computer engineering organizations have created three modes of production (organizational structures, management infrastructure and production infrastructure) to define their own two surplus value mechanisms that enabled them to benefit from free labor in production process. This study categorizes the creation of these two modes of production in two historical events: The *Transitional Phase* (1970-95) and Age of Information (1996-09). During the *Transitional Phase*, computer engineering organizations formed the 1st formal modes of production by adopting the industrial pyramid organizational structure, which was based on a top-down structure, whereby each department within the organization followed chains of command. Computer engineering organizations adopted the industrial organizational structure because it was the only available organizational structure they could implement, as the field of computer engineering was new to the world.

With the pyramid organizational structure as a foundation, computer engineering organizations started forming their management infrastructure, and they realized that this must be a mechanism that works in collaboration with the top-down based pyramid organizational structure. Thus, they adopted the industrial hierarchy based management structure with its mechanisms. The hierarchy management structure had two phases with their respective mechanisms: top-down administration and micromanagement oriented for production supervisions. The top-down administration used chains of command to make decisions, in which the owner or shareholders passed the ruling policies to the general manager to implement them in practice (Schermerhorn, Hunt and Osborn, *Organizational Behavior*, 2008b). The general manger provides the organization's strategy to the senior executives, and they use provide instructions regarding this strategy to the middle managers (Schermerhorn, Hunt and Osborn, 2008b).

At this point, the middle manager used the micromanagement production supervision mechanisms to control production process and labor wages: sequential scientific management method, work breakdown structure (WBS), Gantt-charts and bases of estimates (BOE) to plan out tasks to be carried out (Paulk, Weber, and Curtis, *Capability Maturity Model*, 2000b). Computer engineering organizations' middle managers used the following sequential management methods: 1) project initiation and planning stages of the project management tasks, 2) product development tracking, monitoring and controlling stages of the production process, 3) measurement and analysis, 4) taking corrective actions, and 5) project completion stages of the project management process (Paulk, Weber, and Curtis, 2000b). Using this mechanism, the middle manager guides the tasks' production process supervisor to use the WBS to define the tasks, start date, and end date and to control labor cost. The accounting principle known as "cost control" is included in the WBS method.

However, in order for this management infrastructure to be successfully implemented, computer engineering organizations had to create a production infrastructure that could work in collaboration with the hierarchy management mechanisms. The components of the production infrastructure were to be used as production process task specification inputs into the WBS. Once the production process task specifications are inserted, then the WBS has production process tasks as values to measure production outputs and the speed of production process, the latter of which reduces labor costs. The production infrastructure has three elements: the production process framework, production device and *Division of Labor* forces. As there was never a standardized production framework for the computer systems field, in 1970, Winston Royce introduced the *Waterfall* production framework. The *Waterfall* production framework had a top-down structure and was a critical methodology with 3 major dimensions: The 1st component was to define the production process phases that matched one-to-one with the management mechanism as illustrated in table 4 of chapter 3. Phase #1 "project initiation & planning" mapped to "staff coordination," Phase #2 "project tracking & control" mapped to "business requirements development," Phase #2.1 "project tracking & control" mapped to "product analysis & design," Phase #2.2 "project tracking & control" mapped to "product programming & testing," phase #3 "project transitioning" mapped to "product deployment" and finally, phase #4 "project closing" mapped to "product maintenance tasks," all of which set the foundation for both the management infrastructure and the production infrastructure to work in collaboration.

These production process phases required their production process task specifications to be defined. Thus, Winston Royce (1970) created production process tasks based on a similar top-down structure as illustrated in table. The 6 production process tasks were: requirements for computer systems development initiations, analysis & design for computer systems specifications, implementation for programming, verification for computer systems testing, deployment for computer hardware & software systems integration & releases and maintenance for continuous troubleshooting and support of the computer systems. As the production process tasks were created and standardized by the early to mid-70's, computer engineering organizations began to use the newly created *Waterfall* framework, with its standardized production process tasks specification, to create the appropriate *Division of Labor* forces for the new computer engineering field. The appropriate *Division of Labor* forces had to work collaboratively for the top-down management and production infrastructure. Hence, the creation of the *Division of Labor* forces was based on specializations. Each individual worker had to be trained to specialize in a given task that was outlined and standardized in the *Waterfall* production process framework. As a result, specialization based *Division of Labor* forces became the standard from 1970-95. Computer engineering organizations management also needed to create the type of production process and production devices to be used as inputs for the WBS, so that management would measure outputs and control labor wages in production process.

Thus, the 2nd dimension of the production infrastructure was the creation of computer systems used as production devices for the production environment. During the early 70's, the computer system was the standalone mainframe computer system that was used as a production device. The production environment known as the "computer room" was based locally within the facilities. Hence, the large mainframe computer system had its peripherals setup within the same computer room. There were no network connection distribution devices created, so all the mainframe peripherals were stationed in the same production environment. The mainframe device was large and all the peripherals were also large to be compatible.

As the production devices were all located in the same computer room, users had to make requests to get their transactional data processed and printed for them by specialized workers and delivered to the users' designated offices. As described in chapter 3, the specialized workers were all confined within the same production environment. The most common specialized workers during the 70's that used the standalone mainframe production device were: the *Systems Analysts* to produce the design of the systems, *Programmers* to produce codes, the *Key Punchers* to produce the transferable codes into the devices, the *Console Operators* to translate codes into manuals and deliver them to *Code Converters* to convert codes into magnates and the *Tape Librarian* to produce the backup data on tapes every night, so that transactional data records were backed up and could be recovered in a daily basis.

With the production framework, standalone mainframe production devices and process tasks implemented, the computer engineering organizations' goal became creating a production process that would complete the production infrastructure; this process was the shared base production process. The shared based production process was compatible with the standalone mainframe computer system because the large mainframe system was located in the computer room and shared by all the workers. Hence, computer engineering organizations implemented 3 shifts around the clock in the production environment. Standardly, the *Systems Analysts* were scheduled to analyze particular industry specific business processes and design a system solution while the users were still available during the 1st shift. From the design, the *Systems Analysts* then produced code specifications to be delivered to the *Programmers*, who worked mostly during the 2nd shift. Once the *Programmers* completed the coding, the *Console Operators* translated the program into a user manual during the 3rd shift. Also during the 3rd shift, the *Key Punchers* were assigned to produce codes in the coding cards by keying into the mainframe system. The same 3rd shift was used by the *Converter Operators* to transform the coding cards to the magnates of the mainframe system devices. Lastly, the 3rd shift was also used by the *Tape Librarian* to produce data backups on tapes throughout the course of the night, so that transactional data was ready for the next business day. In this way, throughout the course of the 70's, computer engineering organizations were able to create the management and production infrastructure of the new computer field.

The production infrastructure was standardized and consisted of 3 components: the *Waterfall* production process framework, the specialized based *Division of Labor* forces and the shared based production process. These production infrastructure elements were required as inputs for the management's WBS, so that the production supervisors could specify the tasks, assign each task to a specialized worker, schedule the starting date and end date and allocate labor costs associated to the assigned specialized worker. Once the production data was inserted into the WBS template, the production supervisor became responsible to micromanagement the production process.

The 3 shifts had their own designated shift supervisors. The production supervisor monitored the task of each specialized worker producing in the production process. As the production infrastructure called for a computer room, shift based production environment and a shared base production process, computer engineering organizations had to implement a surplus value mechanism that worked for these scenarios. For the shift based production environment, the surplus value mechanism became increasing the number of specialized *Division of Labor* forces, so that a scarcity in specialized based labor was avoided.

Hence, computer engineering organizations applied the classical surplus value mechanism to increase the labor force to control the labor wage from increasing when a particular specialized labor was in demand. If a particular computer system was in demand and there were not enough *Programmers* or

Systems Analysts to produce the system in demand, then their labor wage increases. When this happened, computer engineering organizations increased the number of the *Programmers* and *Systems Analysts* to prevent their labor wages from increasing. Accounting principles as a surplus value mechanism were also incorporated to prevent labor wage from increasing, keeping it constant or reducing it by creating the concept of salary ranges, which were created for both the experienced and beginner specialized workers. For example, the experienced *Programmers* had a salary range that had a lower end and a higher end cap. The salary of the specialized *Programmers* could be controlled using this salary range concept as a surplus value mechanism. If the *Programmers* became in demand, computer engineering organizations could prevent the salary range from increasing, keeping it at the same rate or decreasing it.

Furthermore, the lower end of the range is at times increased, so that the higher end of the salary range is decreased or remains constant. The beginners' salary range also is subjected to be adjusted. For example, if the specialized *Systems Analysts* were in demand to develop a computer system, then the management could simply and shrewdly decide to increase the higher end of the beginners' salary range, since it is lower than the experienced *Systems Analysts* salary range. Increasing the specialized *Division of Labor* forces, coupled with the accounting principle of adjusting salary ranges, became a very effective surplus value mechanism for computer engineering organizations from 1970-95.

By the early 80's firms were ready to reengineer their production devices to speed up production process. With this motive, organizations meant to transform the shared based production process to individual based production process and to advance the standalone mainframe production device to create a terminal based minicomputer system that could be used by each specialized worker. The shared based production process was less productive as workers shared the standalone mainframe computer system that was used as a production device. For instance, a *Systems Analyst* had to wait for the *Business Systems Analyst* to complete the workflow design before using the standalone mainframe computer system to start using it for production. While the *Systems Analyst* used the system, the *Business Systems Analyst* went to the design table to sketch a workflow manually until the *Systems Analyst* completed the design specification production, which was produced using the standalone mainframe system. In the process of waiting to share the production device, there were ample production hours wasted during the 1st shift. Similarly, the *Programmer* had to wait for the *Console Operator* to complete the code translation of the programs and for the *Key Puncher* to complete the entry of the coding card into the devices before the mainframe system was freed up for the *Programmer* to start coding the program instructions into the mainframe system. As this was causing a major production time loss, computer engineering organizations were not able to extract as much surplus labor as they would want to. Hence, they realized the need to create a new individual based production process to increase levels of productivity in production process.

The new terminal based minicomputer was created at a personal computer level for each specialized worker to use as a production device. This terminal based minicomputer was created to process only one task. It had only a single application to process a task. For instance, a specific application for systems analysis and design that could process that task was created. Similarly, a computer system that could process programming codes was created, and so forth and so on. These terminal based minicomputers became available during the early to mid-80's. At this point, the shared based production process was transformed by the individual based production process and production. Computer engineering organizations now did not have to lose production hours that were otherwise wasted during the shared based production process in the process of waiting to share the standalone mainframe computer system.

As the new terminal based minicomputer systems became available as a production device, the production supervisors were able to increase productivity by increasing the production process of each specialized worker. With this, increasing the speed of production became the daily work of the production supervisor. The production supervisor micromanaged workers to ensure that the specialized worker produced with continuous effort using the allocated production device to complete the task in a timely manner, so that the production hours left from the initial production process were effectively utilized to perform the task of another project. By this time, production supervisors were tracking the weekly product progress reports of each individual worker as the speed of workers' production was the critical success factor of the surplus value mechanism in this individual based production process. With this individual based production process and the availability of the terminal based computer systems, computer engineering organizations became determined to increase the size of specialized based *Division of Labor* forces, because the higher the size, the more production could increase.

As the number of specialized *Division of Labor* forces increased, production also increased and scarcity was avoided to control labor wages. Hence, computer engineering organizations began taking advantage of both the individual based production process and the terminal based minicomputer system to speed up the production process, which translated to a free surplus value, where the extracted labor hours from one project's production process were used in another project's production process to reduce labor costs. Although this advantage brought a positive result in the increase of the number the specialized *Division of Labor* forces, it required a proportional increase in the production of terminal based minicomputer devices for production environments, and a burdensome micromanagement of workers to speed up the production process and save labor hours (extracting surplus labor value).

By the late 80's to mid-90's, computer engineering organizations began conceptualizing a new idea of automating the manual production devices to replace the terminal based minicomputer systems with multifunctional workstation based computer systems. With this transition, a multitasking based production process replaced the individual based production process. The ultimate goal of computer engineering organizations was assigning multiple tasks to one multitalented *Computer Application Software Engineer* to produce what used to require several specialized workers. This was in an effort to have most of the production process activities produced free of labor cost, as each position's production process activity had its own labor value standardized by DOL. Hence, by paying a salary to *Computer Application Software Engineers* that was less than the *Number Production Process Activity* (NPPA) that computer engineering organizations took from this surplus mechanism, an *Actual Triple Surplus Value* (ATSV) was achieved.

Thus, in order to make this objective a success, computer engineering organizations realized that they had to create new modes of production that would enable them to accomplish their goal, as the top-down based modes of production was insufficient (the pyramid based organizational structure, the hierarchy based micromanagement infrastructure and the *Waterfall* based production process framework). For instance, the *Waterfall* production process framework required a complete system, test and deployment in sequential mode, all of which had to be produced by specialized workers individually assigned per task, and workers used a single functional minicomputer system to perform the task. This type of individual based production process using the single function minicomputer system, and requiring a specialized worker per task, became unproductive to computer engineering organizations, and it was burdensome to continuously increase the number of specialized *Division of Labor* forces to control the labor wage.

Furthermore, as the *Waterfall* production process framework requires the individual production of all modules before they are integrated into one complete system and released to users, it took a long period of time to complete a project, which became costly. The users of the system have to wait for a long period of time to get the completed computer systems. An additional problem with the *Waterfall* production process framework was its product quality, whereby unknown programming defects could only be discovered during the systems integration and user acceptance testing after all the modules of the system were completed. Thus, using this production framework was not profitable enough for computer engineering organizations; they did not save on labor costs and profit from unpaid labor, as the framework was not suitable for speedier forms of production process. With this limitation of the *Waterfall* production process framework, computer engineering organizations decided to create a framework that absolved the limitations of the *Waterfall*. With this as a priority, by 1995, computer engineering organizations adopted methods from the study being conducted by computer engineering methodologists, James Rumbaugh, Grady Booch and Ivar Jacobson, who were creating a production process framework called "*Iterative*." As the *Iterative* production framework was built with foundations of best practice process, engineering models and open communication capabilities of the newly introduced internet communication and advanced wide area network systems, it needed a compatible modes of production to make it a successful production framework. Hence, simultaneously with the creation of the *Iterative* production process framework, scholars of Organizational Behavior (OB) were creating the upside-down pyramid organizational structure and the macro-matrix management infrastructure.

The upside-down pyramid organizational structure was developed with open communication as a foundation, which was distinct from the pyramid organizational structure, in which communications were closed and only functioned under a chain of command. Hence, the upside-down pyramid organizational structure made it possible for different departments within organizations to communicate without a chain of command and created the possibility of unconfined production environments, which enabled workers to telecommute from wherever they were located in the production environment, remotely. Furthermore, this open communication enabled computer engineering organizations to work in collaboration with other firms to share resources and achieve common goals.

In addition, facility based production environment were no longer needed, as the upside-down pyramid organizational structure brought about remote based production environments, whereby a *Systems Analyst* could produce designs, systems integration architects and code specification and test scripts remotely via an internet connection to the production environment. This was the advantage of implementing the upside-down organizational structure, but computer engineering organizations learned that this new system was not going to integrate effectively with the hierarchy management structure, which was a top-down micromanagement based structure. Hence, computer engineering organizations started looking into the macro-matrix based management infrastructure that was contemporaneously being formed by the OB scholars.

The macro-matrix management structure was formed with best practice process engineering at its foundation. This was possible due to the emergence of the internet and wide area networks during this time, as well as the multifunctional based workstations as production devices. With these advanced communications and computer systems available, computer engineering organizations automated the previous hierarchy management infrastructure's mechanisms (WBS, Gantt-Chart, WPPR & BOE). Microsoft introduced the first Microsoft Project system that automated management mechanisms, including project planning, project tracking & oversight, measurements and taking corrective actions, all of which were automated into the Microsoft Project.

With this automated management system available, computer engineering organizations began focusing on the production process activities of the *Iterative* productions framework, and its implementation became necessary to make the macro-matrix management infrastructure's mechanism effective. This led to the invention of the production infrastructure, which encompasses the production process framework, the type of production process and the production device of the production environment. Thus, the *Iterative* production framework became the 1st component to be developed, so that it would define the production process activities that would be used to create the type of multitalented *Division of Labor* forces required, and further be used as inputs for the WBS to control production process.

The *Iterative* production framework was built on linear and spiral methods as a foundation in transitioning from the *Waterfall* production framework. The linear method consisted of inceptions, evaluations, construction and transitions, which are the production process phases enabling computer engineering organizations to produce each system's module separately. Analysis, programming, testing and deployment to the market were completed for each module without having to wait for all the systems components to be completed. Furthermore, the linear based structure was created to synchronize well with the macro-matrix management infrastructure, as the linear production process phases enabled a team lead or a project lead to assign production process activities across the production phases to a multitalented professional.

The spiral method defined the production process activities for each production phase. At this point, both the linear and the spiral methods had an automated production device called the "unified modeling language" (UML). UML became the standard production device of the production environment, and it could be accessed remotely by workers regardless of their location. Since the spiral method was used to train multitalented professionals, the UML type of production device enabled one multitalented worker to perform all of the required production activities of a project, which reduced labor costs for computer engineering organizations.

Thus, the team lead or project lead would simply assign a worker in accordance with the spiral method. The spiral method was created in and around the linear production phases, and a team lead could identify the required production activities horizontally and insert production process activity specifications into the Microsoft Project system, as the UML and Microsoft

Project could interface with each other. The spiral method has the following components: a workflow & use cases requirements development technique, an object oriented analysis & design (OOA&D) technique, an object oriented programming (OOP) technique and test cases & scenarios. These components of the spiral methodology are applied to produce each module of the system separately, and test and deploy them to their designated industry specific market as they are. This is to say that each module is developed independently and each is added one-by-one to previously released modules as building blocks to avoid the old problem of discovering integration defects only after the entire system had been completed. This became very beneficial to computer engineering organizations, as they could now assign just one multitalented professional to produce a module using automated production devices, such as UML, to complete a module. Furthermore, by the mid-2000's, with the invention of the Microsoft SharePoint production device, macro-managing the production status also greatly improved as the team lead or the project lead could monitor the real-time status of the production remotely. Hence, the automated based multitasking modes of production that the computer engineering organizations created during the mid-90's had its own unique macro-matrix management infrastructure and production infrastructure consisting of a multitasking production process, multitasking *Division of Labor* forces and multifunctional production devices.

As has been shown, computer engineering organizations transformed the modes of production to their advantage twice within the span of 40 years, and they also created a surplus value mechanism that worked well with the modes of production. As increasing the number of specialized based *Division of Labor* forces and speeding up the production process were the two major dimensions of the classical surplus value mechanism from 1970-95, a new surplus value mechanism was developed for the automated based multifunction modes of production, which has been in practice since 1996. Hence, with the availability of the automated multitasking based production devices, multitalented professionals and multifunction based production processes, computer engineering organizations created a surplus value mechanism in which they could assign multiple tasks to a multitalented *Computer Applications Software Engineer* to efficiently produce the positions' production process activities. In this scenario, these positions have their own standardized labor wages, which the worker is not paid for, as his/her labor wage offered is specifically allocated to the position title. This multitasking based production process required longer hours to get all the production process activities completed, which meant it required longer production hours, and the speed of production process continued to be the crucial mechanism for reducing labor wages, as has always been the case. This means the multitasking, longer production hours and speed of production process became the core of the surplus value mechanism of this time: the *Triple Surplus Value* (a concept and term coined by the present study). Its degree is tested by applying the *Triple Surplus Value* formula. In chapter 4, there were 13 job postings records that the *Triple Surplus Value* formula tested and results confirmed the manifestation of *Triple Surplus Value* practices, indicating that computer engineering organizations have benefited from its implementation since the mid to late 90's, as this trend continues today. Below, the *Triple Surplus Value* formula is applied to 2 more job postings records from 2010-2011.

Table 65: Illustrates the offered annual salary of a *Computer Applications Software Engineer* in 2010:

Number of Production Process Activities (NPPA)	**Description of Production Activities**	**Total Values**
Systems Analyst	Production process activities remain the same.	**$95,810**
Network and Computer Systems Administrator	Production process activities remain the same.	**$84,110**
Database Administrator	Production process activities remain the same.	**$91,850**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$90,740**
Applications Programmer	Production process activities remain the same.	**$89,720**
Programmer Analyst	Production process activities remain the same.	**$89,720**
Computer Systems Software Engineer	Production process activities remain the same.	**$113,960**
Technical support Specialist	Production process activities remain the same.	**$55,990**
NPPA Amount		**$744,160**
Offered Labor Wage (OLW) for *Computer Applications Software Engineer*		**$93,175**
Actual Triple Surplus Value (ATSV)		**$650,985**

© 1996-2011 The Washington Post, Job Posting ID: 1009019. Posted on 10/27/2010

Table 65 presents the∑ NPPA, OLW and GTSV for a *Computer Applications Software Engineer* from the 2010 job posting. The GTSV *Triple Surplus Value* formula is selected, as this was a scenario of underpayment, whereby the OLW ($93,175) was less than DOL's standard labor wage ($104,870) for an experienced *Computer Applications Software Engineer*. According to Appendix E14a and E14b, the job posting description required the candidate for this position to have an expertise in the following 8 positions' production processes: *Systems Analyst, Network and Computer Systems Administrator*, *Database Administrator*, *Network Systems & Data Communications Analyst*, *Applications Programmer*, *Programmer Analyst*, *Computer Systems Software Engineer and Technical Support Specialist*, which made the ∑ NPPA = $744,160. Thus, ∑ NPPA ($744,160) – OLW ($93,175) = $650,985 gives the GTSV that goes to the computer engineering organizations as a profit with no MV given to the *Computer Applications Software Engineer*, as the labor rate was already lower than DOL's standard for 2010.

Table 66: Illustrates the offered annual salary of a *Computer Systems Software Engineer* in 2011:

Number of Production Process Activities (NPPA)	Description of Production Activities	Total Values
Systems Analyst	Production process activities remain the same.	**$95,810**
Network and Computer Systems Administrator	Production process activities remain the same.	**$84,110**
Database Administrator	Production process activities remain the same.	**$91,850**
Network Systems & Data Communications Analyst	Production process activities remain the same.	**$90,740**
Applications Programmer	Production process activities remain the same.	**$89,720**
Programmer Analyst	Production process activities remain the same.	**$89,720**
Computer Applications Software Engineer	Production process activities remain the same.	**$104,870**
Technical support Specialist	Production process activities remain the same.	**$55,990**
NPPA Amount		**$702,810**
Offered Labor Wage (OLW) for *Computer Systems Software Engineer*		**$100,000**
Actual Triple Surplus Value (ATSV)		**$602,810**

 Job Posting ID: XYRAT. Posted on 04/12/2011

Table 66 presents the∑ NPPA, OLW and GTSV for the *Computer Systems Software Engineer* from the 2009 job posting. The GTSV *Triple Surplus Value* formula is selected as this again was a scenario of underpayment, in which the OLW ($100,000) was less than DOL's standard labor wage ($113,960) for an experienced *Computer Systems Software Engineer* this year. According to Appendix E15a and E15b, the job posting description required that the *Computer Systems Software Engineer* have expertise in the following 8 positions' production processes: *Systems Analyst, Network and Computer Systems Administrators, Database Administrator, Network Systems & Data*

Communications Analyst, Programmer Analyst, Computer Applications Software Engineer, Computer Systems Software Engineer and Technical Support Specialist, which made the ∑ NPPA = \$702,810. Thus, ∑ NPPA (\$702,810) – OLW (\$70,000) = \$602,810 gives the GTSV that goes to the computer engineering organizations as a profit, while no MV goes to the *Computer Systems Software Engineer*, as the labor rate was already lower than DOL's standard for 2011.

5.1 Recommendation for Future Studies

Based on these findings, this study recommends future study of the following: 1) There should be a sequential follow up study on the implementation of the *Triple Surplus Value* formula to educate the multitalented professionals, so that workers can make educated decisions on accepting job offers. Following this path, the academic community and institutions should incorporate these studies into their curricula, so that the future generation is prepared to undertake the challenge of creating their own methods of production process. 2) There ought to be a study on the distribution of labor wage calculated based on the merit of the worker. 3) An exploration of the idea of creating computer systems as production devices in the context of fair distribution of workers' earnings will demonstrate the need for mutual benefits in profit amongst the workers in the production process and the owner(s) of computer engineering organizations.

First, as the manifestation of *Triple Surplus Value* and its mechanisms causes a labor wage disadvantage to multitalented professionals has demonstrated, this study contributes a *Triple Surplus Value* formula to contemporary workforces, so that they are aware of what they are worth as multitalented members of the *Division of Labor* forces, and they are able to take an educated measurement of the value of their production process activities. Any forthcoming study on this subject must outline the implementation processes and procedures of the *Triple Surplus Value* formula in such ways that can be understood by the laymen who are already in the workforce. This outline must provide step-by-step guidance to the multitalented worker in identifying his/her talent and appropriate labor wages, preparing him/her for negotiations with employers. Furthermore, the outline must help the worker to development awareness of his or her worth, with the ultimate goal of enlightening the workforce and help them reach their full potential as creative workers. This approach must not only bridge the gap between profit gains and unbalanced labor wage distribution, but also enable workers to create their own reality to contrast with the false consciousness of the production process. As illustrated in this study, the *Motivation Value* (MV) in the equation of the *Triple Surplus Value* formula is the amount that is higher than DOL's standard labor wage for a given position; however, it is disproportionally and unfairly distributed as less than the *Actual Triple Surplus Value* (ATSV) amount that goes to the computer engineering organizations. Under this circumstance, the MV is used to motivate the worker and the average worker perceives it as something to be motivated by. With the implementation the *Triple Surplus Value* method, a worker can make his/her own objective decision in measuring self-worth, and this realization contributes to encouraging a worker to master his/her own methods of production process per his/her capabilities to produce with self-fulfillment, leading to the generation of an independently productive workforce.

Second, as discussed in this study, computer engineering organizations have benefited from free labor wages in production process. They created operational modes of production infrastructures with surplus value mechanisms that have facilitated their gaining profits from the production process. Currently, a multitalented professional in the field of computer systems is charged to perform the production activities of multiple positions, whereby each position has its own standardized labor wage. As this has become a major profiting mechanism, it is a disadvantageous to the worker, because the gap between the earnings of the worker and the profit enjoyed by computer engineering organizations is very wide. A future study that focuses on how to go about creating a method and process that ensure workers are paid according to what they produce based upon their merit is desirable. Such a methodology should include 1) a measurable distribution of production hours to calculate the rate of each position's production activities, 2) management mechanisms that work collaboratively with the methodology of distributing production hours for the planning purposes, and 3) to measure the output of the production and track & monitor the production process of each production activity assigned to the multitalented professional. In this way, each position's production process activities will be measured, in that the multitasked worker is paid for each position's production process activities to reduce the current high gap between what the worker earns and the profit gains of the computer engineering organizations.

The third recommendation for future study is how to create computer systems as production devices considering fair distribution of workers' earnings. The implementation of any recommendations of such a study should provide benefits to both the workers and the owners to promote mutual interests. Its primary focus should be consists in the following 2 components: 1) creating computer systems development methodology that uses production devices that require high human involvement in such ways that would create jobs for professionals in the computer systems field, and 2) developing production frameworks that mutually benefit the workers and owners.

Such a framework should utilize production devices that would involve more human inputs to boost job availability for workers. It is evident that contemporary production devices are automated and multifunctional with the internet in the background, causing disadvantages for workforces. Currently, computer systems are automated, and the computer systems are supported by a multitalented professional covering most of the day-to-day production activities, or by an off-shore technical support from abroad. Both are present disadvantages for workers. On one hand, the implementation of multifunctional computer devices and multitasking production process makes their job obsolete and causes domestic unemployment, and on the other hand, it unfairly increases employment abroad, giving these jobs to off-shore technicians. A future study that will create alternative computer systems must take these factors into consideration. As analyzed in this study, the automation initiative enables a worker to produce all automated tasks, which used to require a worker for each manual task. But future computer systems have to be created that require human participation to cultivate the creativity and cognitive capabilities of the labor force, which would also benefits the owner(s) of computer engineering organizations, as there would be more creative labor forces available. This might require revisiting labor wage standards that are fair both for the worker in terms of earnings and computer engineering organizations' in terms of profit gains. Creating this future alternative computer system of automation and manual computer systems will promote sustainable job availability in the computer systems field.

Appendix A

DESKTOP COMPUTER SYSTEMS

The historical development of terminal based desktop computers is highlighted below:
Resource: http://en.wikipedia.org/wiki/Mainframe_computer

Microprocessors

On November 15, 1971, Intel released the world's first commercial microprocessor, the 4004. It was developed for a Japanese calculator company, Busicom, as an alternative to hardwired circuitry, but computers were developed around it, with much of their processing abilities provided by a single small microprocessor chip. Coupled with one of Intel's other products - the RAM chip, based on an invention by Robert Dennard of IBM, (kilobits of memory on a single chip) - the microprocessor allowed fourth generation computers to be smaller and faster than previous computers. The 4004 was only capable of 60,000 instructions per second, but its successors, the Intel 8008, 8080 (used in many computers using the CP/M operating system), and the 8086/8088 family (the IBM PC and compatibles use processors still backwards-compatible with the 8086) brought ever-increasing speed and power to the computers. Other manufacturers also produced microprocessors which were widely used in microcomputers.

Supercomputers

At the other end of the computing spectrum from the microcomputers, the powerful supercomputers of the era also used integrated circuit technology. In 1976 the Cray-1 was developed by Seymour Cray, who had left Control Data in 1972 to form his own company. This machine, the first supercomputer to make vector processing practical, had a characteristic horseshoe shape, to speed processing by shortening circuit paths. Vector processing, which uses a single instruction to perform the same operation on many arguments, has been a fundamental supercomputer processing method ever since. The Cray-1 could calculate 150 million floating point operations per second (150 megaflops). 85 were shipped at a price of $5 million each. The Cray-1 had a CPU that was mostly constructed of SSI and MSI ECL ICs.

Mainframes and Minicomputers

Before the introduction of the microprocessor in the early 1970s, computers were generally large, costly systems owned by large institutions: corporations, universities, government agencies, and the like. Users—who were experienced specialists—did not usually interact with the machine itself, but instead prepared tasks for the computer on off-line equipment, such as card punches. A number of assignments for the computer would be gathered up and processed in batch mode. After the jobs had completed, users could collect the output printouts and punched cards. In some organizations it could take hours or days between submitting a job to the computing center and receiving the output. A more interactive form of computer use developed commercially by the middle 1960s. In a time-sharing system, multiple teletype terminals let many people share the use of one mainframe computer processor. This was common in business applications and in science and engineering.

A different model of computer use was foreshadowed by the way in which early, pre-commercial, experimental computers were used, where one user had exclusive use of a processor.[2] Some of the first computers that might be called "personal" were early minicomputers such as the LINC and PDP-8, and later on VAX and larger minicomputers from Digital Equipment Corporation (DEC), Data General, Prime Computer, and others. They originated as peripheral processors for mainframe computers, taking on some routine tasks and freeing the processor for computation. By today's standards they were physically large (about the size of a refrigerator) and costly (typically tens of thousands of US dollars), and thus were rarely purchased by individuals. However, they were much smaller, less expensive, and generally simpler to operate than the mainframe computers of the time, and thus affordable by individual laboratories and research projects. Minicomputers largely freed these organizations from the batch processing and bureaucracy of a commercial or university computing center.

In addition, minicomputers were more interactive than mainframes, and soon had their own operating systems. The minicomputer Xerox Alto (1973) was a landmark step in the development of personal computers, because of its graphical user interface, bit-mapped high resolution screen, large internal and external memory storage, mouse, and special software.[3]

Microprocessor and cost reduction

The minicomputer ancestors of the modern personal computer used integrated circuit (microchip) technology, which reduced size and cost, but processing was carried out by circuits with large numbers of components arranged on multiple large printed circuit boards before the introduction of the microprocessor. They were consequently physically large and expensive to manufacture. After the "computer-on-a-chip" was commercialized, the cost to manufacture a computer system dropped dramatically. The arithmetic, logic, and control functions that previously occupied several costly circuit boards were now available in one integrated circuit which was very expensive to design but very cheap to manufacture in large quantities. Concurrently, advances in the development of solid state memory eliminated the bulky, costly, and power-hungry magnetic core memory used in prior generations of computers. There were a few researchers at places such as SRI and Xerox PARC who were working on computers that a single person could use and could be connected by fast, versatile networks: not home computers, but personal ones.

Altair 8800 and IMSAI 8080

Development of the single-chip microprocessor was an enormous catalyst to the popularization of cheap, easy to use, and truly personal computers. The Altair 8800, introduced in a *Popular Electronics* magazine article in the January 1975 issue, at the time set a new low price point for a computer, bringing computer ownership to an admittedly select market in the 1970s. This was followed by the IMSAI 8080 computer, with similar abilities and limitations. The Altair and IMSAI were essentially scaled-down minicomputers and were incomplete: to connect a keyboard or teletype to them required heavy, expensive "peripherals". These machines both featured a front panel with switches and lights, which communicated with the operator in binary. To program the machine after switching it on the bootstrap loader program had to be entered, without error, in binary, then a paper tape containing a BASIC interpreter loaded from a paper-tape reader. Keying the loader required setting a bank of eight switches up or down and pressing the "load" button, once for each byte of the program, which was typically hundreds of bytes long. The computer could run BASIC programs once the interpreter had been loaded.

The MITS Altair, the first commercially successful microprocessor kit, was featured on the cover of *Popular Electronics* magazine in January 1975. It was the world's first mass-produced personal computer kit, as well as the first computer to use an Intel 8080 processor. It was a commercial success with 10,000 Altairs being shipped. The Altair also inspired the software development efforts of Paul Allen and his high school friend Bill Gates who developed a BASIC interpreter for the Altair, and then formed Microsoft.

The MITS Altair 8800 effectively created a new industry of microcomputers and computer kits, with many others following, such as a wave of small business computers in the late 1970s based on the Intel 8080, Zilog Z80 and Intel 8085 microprocessor chips. Most ran the CP/M-80 operating system developed by Gary

Kildall at Digital Research. CP/M-80 was the first popular microcomputer operating system to be used by many different hardware vendors, and many software packages were written for it, such as WordStar and dBase II.

Many hobbyists during the mid-1970s designed their own systems, with various degrees of success, and sometimes banded together to ease the job. Out of these house meetings the Homebrew Computer Club developed, where hobbyists met to talk about what they had done, exchange schematics and software, and demonstrate their systems. Many people built or assembled their own computers as per published designs. For example, many thousands of people built the Galaksija home computer later in the early 80s.

It was arguably the Altair computer that spawned the development of Apple, as well as Microsoft which produced and sold the Altair BASIC programming language interpreter, Microsoft's first product. The second generation of microcomputers — those that appeared in the late 1970s, sparked by the unexpected demand for the kit computers at the electronic hobbyist clubs, were usually known as home computers. For business use these systems were less capable and in some ways less versatile than the large business computers of the day. They were designed for fun and educational purposes, not so much for practical use. And although you could use some simple office/productivity applications on them, they were generally used by computer enthusiasts for learning to program and for running computer games, for which the personal computers of the period were less suitable and much too expensive. For the more technical hobbyists home computers were also used for electronics interfacing, such as controlling model railroads, and other general hobbyist pursuits.

Microcomputer Emerges

The advent of the microprocessor and solid-state memory made home computing affordable. Early hobby microcomputer systems such as the Altair 8800 and Apple I introduced around 1975 marked the release of low-cost 8-bit processor chips, which had sufficient computing power to be of interest to hobby and experimental users. By 1977 pre-assembled systems such as the Apple II, Commodore PET, and TRS-80 (later dubbed the "1977 Trinity" by *Byte* Magazine)[11] began the era of mass-market personal computers; much less effort was required to obtain an operating computer, and applications such as games, word processing, and spreadsheets began to proliferate First on CP/M small business systems, then IBM introduced the IBM-PC, which was heavily cloned, which lead to the current monoculture of architecturally identical personal *computers, then Windows was released, resulting in the current situation.*

Third generation Hardware Systems

According to the history of third genertion hardware systems, the mass increase in the use of computers accelerated with 'Third Generation' computers (*Free Encyclopidia*). While large 'mainframes' such as the System/360 increased storage and processing capabilities, the integrated circuit also allowed the development of much smaller computers. The minicomputer was a significant innovation in the late 1960s and 1970s. It brought computing power to more people, not only through more convenient physical size but also through broadening the computer vendor field. Digital Equipment Corporation became the number two computer company behind IBM with their popular PDP and VAX computer systems. Smaller, affordable hardware also brought about the development of important new operating systems like Unix.

In 1966, Hewlett-Packard entered the general purpose computer business with its HP-2116, offering a computational power formerly found only in much larger computers. It supported a wide variety of languages, among them BASIC, ALGOL, and FORTRAN.

In 1969, Data General shipped a total of 50,000 Novas at $8000 each. The Nova was one of the first 16-bit minicomputers and led the way toward word lengths that were multiples of the 8-bit byte. It was first to employ medium-scale integration (MSI) circuits from Fairchild Semiconductor, with subsequent models using large-scale integrated (LSI) circuits. Also notable was that the entire central processor was contained on one 15-inch printed circuit board.

In 1973, the TV Typewriter, designed by Don Lancaster, provided electronics hobbyists with a display of alphanumeric information on an ordinary television set. It used $120 worth of electronics components, as outlined in the September 1973 issue of Radio Electronics magazine. The original design included two memory boards and could generate and store 512 characters as 16 lines of 32 characters. A 90-minute cassette tape provided supplementary storage for about 100 pages of text. His design used minimalistic hardware to generate the timing of the various signals needed to create the TV signal. Clive Sinclair later used the same approach in his legendary Sinclair ZX80.

Appendix B:

COMPUTER NETWORK SYSTEMS

The historical development of computer network systems is highlighted below:
Resource:
http://www.nethistory.info/History%20of%20the%20Internet/pcnets.html

At the same time as the academic and research communities were creating a network for scientific purposes; a lot of parallel activity was going on elsewhere building computer networks as well.

A lot of the West Coast hackers belonged to the Homebrew Computer Club, founded by Lee Felsenstein. Lee had actually begun networking computers before the development of the PC, with his Community Memory project in the late 1970s. This system had dumb terminals (like computer screens with keyboards connected to one large computer that did the processing). These were placed in laundromats, the Whole Earth Access store, and community centres in San Francisco. This network used permanent links over a small geographical area rather than telephone lines and modems.

The first public bulletin board using personal computers and modems was written by Ward Christensen and Randy Seuss in Chicago in 1978 for the early amateur computers. It was about 1984 that the first bulletin boards using the IBM (Bill Gates/Microsoft) operating system and Apple operating systems began to be used. The most popular of these was FidoNet.

At that time the Internet technologies were only available on the UNIX computer operating system, which wasn't available on PCs. A piece of software called ufgate, developed by Tim Pozar, was one of the first bridges to connect the Fidonet world to the Internet world. An alternative approach undertaken by Scott Weikart and Steve Fram for the Association for Progressive Communications saw UNIX being made available on special low cost PCs in a distributed network.

In the community networking field early systems included PEN (Public Electronic Network) in Santa Monica, the WELL (Whole Earth 'Lectronic Link) in the Bay area of San Francisco, Big Sky Telegraph, and a host of small businesses with online universities, community bulletin boards, artists networks, seniors clubs, womens networks etc. ..

Gradually, as the 1980s came to a close, these networks also began joining the Internet for connectivity and adopted the TCP/IP standard. Now the PC networks and the academic networks were joined, and a platform was available for rapid global development.

By 1989 many of the new community networks had joined the Electronic Networkers Association, which preceded the Internet Society as the association for network builders. When they met in San Francisco in 1989, there was a lot of activity, plus some key words emerging - connectivity and interoperability. Not surprisingly in the California hippy culture f the time, the visions for these new networks included peace, love, joy, Marshall McLuhan's global village, the paperless office, electronic democracy, and probably Timothy Leary's Home Page. However, new large players such as America on Line (AOL) were also starting to make their presence felt, and a more commercial future was becoming obvious. Flower power gave way to communications protocols, and Silicon Valley just grew and grew.

PEN (The Public Electronic Network) in Santa Monica, may be able to claim the mantle of being the first local government based network of any size. Run by the local council, and conceived as a means for citizens to keep in touch with local government, its services included forms, access to the library catalogue, city and council information, and free email.

PEN started in February 1989, and by July 1991 had 3,500 users. One of the stories PEN told about the advantages of its system was the consultations they had with the homeless people of Santa Monica. The local council decided that it would be good to consult the homeless to find out what the city government could do for them. The homeless came back via email with simple needs - showers, washing facilities, and lockers. Santa Monica, a city of 96000 people at the time, was able to take this on board and provide some basic dignity for the homeless - and at a pretty low cost. This is probably the first example of electronic democracy in action.

Meanwhile, back in the academic and research world, there were many others who wanted to use the growing network but could not because of military control of Arpanet. Computer scientists at universities without defence contracts obtained funding from the National Science Foundation to form CSNet (Computer Science Network). Other academics who weren't computer scientists also began to show interest, so soon this started to become known as the "Computer and Science Network". In the early days, however, only a few academics used the Internet at most universities. It was not until the1990s that the penetration of Internet in academic circles became at all significant.

Because of fears of hackers, the Dept of Defence created a new separate network, MILNet, in 1982. By the mid-1980s, ARPANET was phased out. The role of connecting university and research networks was taken over by CSNet, later to become the NSF (or national science foundation) Network.

The NSFnet was to become the U.S. backbone for the global network known as the Internet, and a driving force in its early establishment. By 1989 ARPANet had disappeared, but the Information Superhighway was just around the corner.

Appendix C:

WEEKLY PROJECT PROGRESS REPORT (WPPR)

The WPPR template is tailored based on the CMMI guidelines for computer engineering organizations management and it is illustrated below:

<Project Name>

Weekly Product Progress Report

Version <#>

Author: *<Author Name>*
Template Developed by: <John Smith>

<MM/DD/YYY>

REVISION HISTORY

VERSION	DATE	DESCRIPTION	AUTHOR	TEMPLATE DEVELOPED BY	DOCUMENT MAINTAINED BY
#	MM/DD/Y Y	First draft for review		John Smith	John Doe

How to Use this Template

Prject's Weekly Progress Report (WPPR) template represents an aggregated summary status report of the assignments assigned by the designated project manager to the team members as they join the team Even though assignments are defined and assigned to team members throughout the course of the Project Development Life Cycle, the project's progress status is tracked through the Project Tracking & Oversight phase of the Project Management Life Cycle. Thus, the Project Manager is responsible to receive the Weekly Product Progress Report (WPPR) from team members to use as an input to populate this template. The Weekly Project Progress Report template offers a consistent and repeatable method to track, measure and report the overall project progress status: Weekly, Monthly and Quarterly to respective stakeholder.

Note: Team members must submit this report every Monday by 12:00pm to their designated Project Manager.

Please follow the italicized instructions below to complete each section of the template in as much detail as possible. ***When you have completed the section, delete the italicized instructions including the section How to Use this Template****.*

Employee Name	*Joe Smith*	**Estimated Hours**	***XX*** *hours*
Employee ID	*XXXXX*	**Actual Hours**	*XX hours*
Assigned by	*John Smith*	**Variance**	*XX hours*
Quarter	*First/Second/Third/Fourth*	**Date**	*XX/XX/XX-XX/XX/XX*

ASSIGNMENT DESCRIPTION

This field should already be filled; do not change it. The Work Assignment Authorization Number (WAA#) is a combination of project phase-task and assignment codes. If you do not have the complete codes, please submit whatever codes you available for your team under the WAAC field.

WAAC	Assignment Name	Assignment Description	Date Assigned

PROGRESS WITH PERCENT COMPLETED

Detail the progress you made during this reporting period on the project(s) you were assigned to. Detail the work you performed on each of your project(s) with the percent completed. Address every project assigned to you in the Assignment Description section above. If you have more than five projects, add them accordingly with the percent completed.

To add additional projects, highlight an entire table by clicking on the four-pointed arrow that appears on the top left corner of a table when your cursor is hovering over it. Copy the table. Paste the copied table. Be sure to insert a carriage return between each new table. The first two tables contain examples that to define the content of the fields. Thus, once you comprehend the content meaning, please delete them and populate the fields with your own assignment content.

WAA #: <*XXXXXXXXXXXXXXXXX*>

Assignment Name	<*Data Modeling*>	Percent Completed	<*%*>
Assignment Description	Describe the type of data modeling assignment that was given to you and the task that the assignment associated with.		
Work Performed	Provide detailed status report regarding the stated assignment given to you.		

WAA #: <*XXXXXXXXXXXXXXXXX*>

Assignment Name	<*Requirement Development* >	Percent Completed	<*%*>
Assignment Description	Describe the type of Requirements Development assignment that was given to you and the task that the assignment associated with.		
Work Performed	Provide detail status report regarding the stated assignment given to you.		

WAA #: <*XXXXXXXXXXXXXXXXX*>

Assignment Name	***<Assignment Name>***	**Percent Completed**	*<%>*
Assignment Description			
Work Performed			

WAA #: <*XXXXXXXXXXXXXXXXX*>

Assignment Name	***<Assignment Name>***	**Percent Completed**	*<%>*
Assignment Description			
Work Performed			

WAA#: <*XXXXXXXXXXXXXXXXX*>

Assignment Name	***<Assignment Name>***	**Percent Completed**	*<%>*
Assignment Description			
Work Performed			

ISSUES

List any issues that have come up in this reporting period. Give a clear, detailed explanation of the issues so they can be resolved in a timely fashion. Sort issues by: Logistics, Capacity, and Resources.

*To add more issues, place your cursor in one of the middle rows of the table. Go to the main toolbar and click **Table>Insert>Rows Below**.*

Logistics	Capacity	Resources
1) *Methods, processes, procedures, standards, SDLC, and frameworks.*	**1)** *Network, technology, slow server performance, negative change in the physical working environment, or weather problems.*	**1)** *Labor, supplies, cost, etc.*
2)	**2)**	**2)**
3)	**3)**	**3)**
4)	**4)**	**4)**
5)	**5)**	**5)**

PROJECTED PLAN FOR NEXT WEEK

Explain what you plan to work on during the oncoming week.

COMMENTS AND/OR QUESTIONS

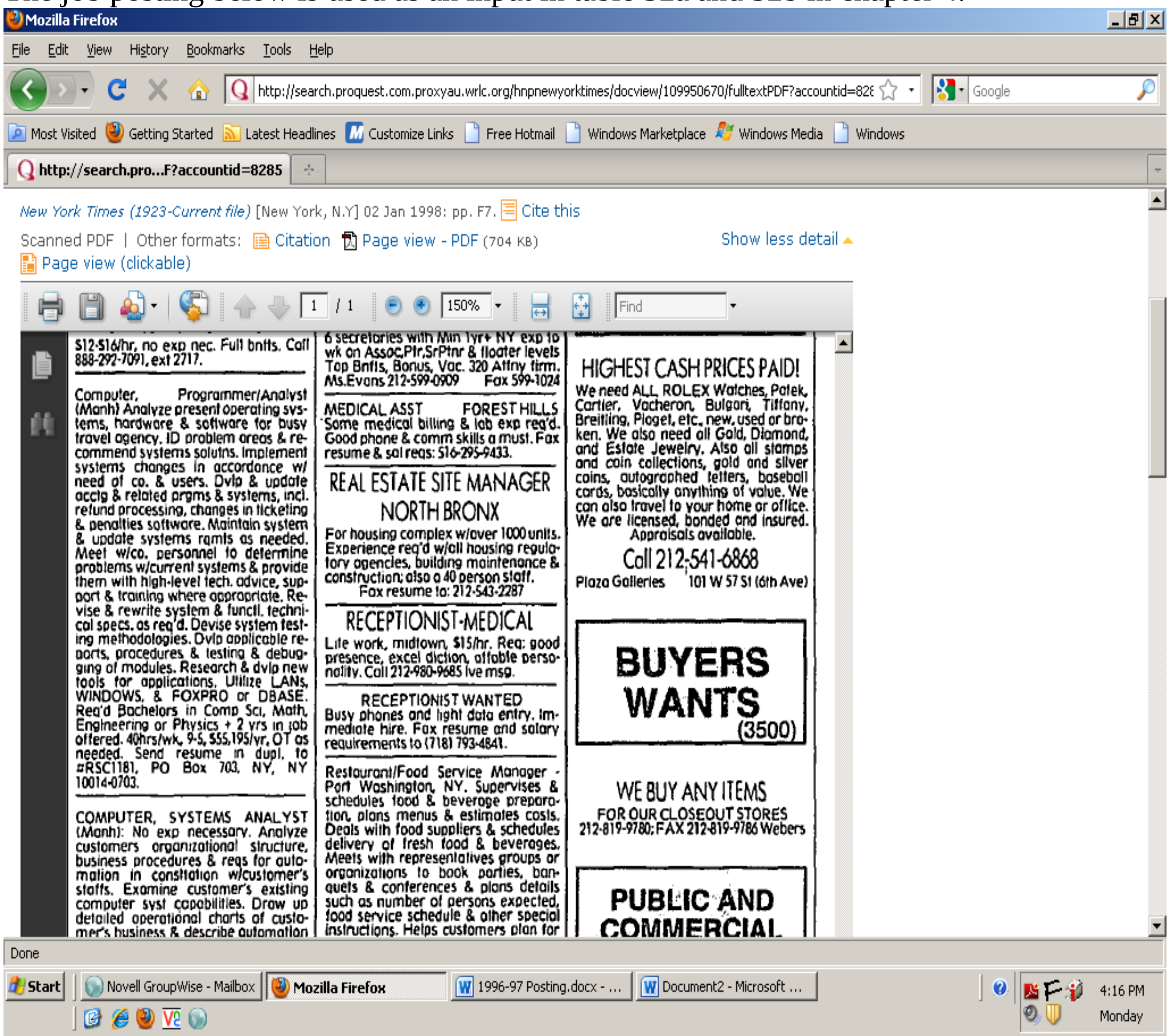
List any comments and/or questions you have about your assignment(s), team members, etc.

Note: When you have completed the Weekly Product Progress Report, be sure to update the Contents. ***Right click*** *on the Contents and choose* ***Update Field****. Select* ***Update entire table*** *and click* ***OK****.*

Appendix E1:

JOB POSTINGS 1998

The job posting below is used as an input in table 52a and 52b in chapter 4.



$12-$16/hr, no exp nec. Full bnfts. Call 888-292-7091, ext 2717.

Computer, Programmer/Analyst (Manh) Analyze present operating systems, hardware & software for busy travel agency. ID problem areas & recommend systems solutns. Implement systems changes in accordance w/ need of co. & users. Dvlp & update acctg & related prgms & systems, incl. refund processing, changes in ticketing & penalties software. Maintain system & update systems rqmts as needed. Meet w/co. personnel to determine problems w/current systems & provide them with high-level tech. advice, support & training where appropriate. Revise & rewrite system & functl. technical specs. as req'd. Devise system testing methodologies. Dvlp applicable reports, procedures & testing & debugging of modules. Research & dvlp new tools for applications. Utilize LANs, WINDOWS, & FOXPRO or DBASE. Req'd Bachelors in Comp Sci, Math, Engineering or Physics + 2 yrs in job offered. 40hrs/wk, 9-5, $55,195/yr, OT as needed. Send resume in dupl. to #RSC1181, PO Box 703, NY, NY 10014-0703.

COMPUTER, SYSTEMS ANALYST (Manh): No exp necessary. Analyze customers organizational structure, business procedures & reqs for automation in constation w/customer's staffs. Examine customer's existing computer syst capabilities. Draw up detailed operational charts of customer's business & describe automation

6 secretaries with Min 1yr+ NY exp to wk on Assoc,Ptr,SrPtnr & floater levels Top Bnfts, Bonus, Vac. 320 Attny firm. Ms.Evans 212-599-0909 Fax 599-1024

MEDICAL ASST FOREST HILLS
Some medical billing & lab exp req'd. Good phone & comm skills a must. Fax resume & sal reqs: 516-295-9433.

REAL ESTATE SITE MANAGER
NORTH BRONX
For housing complex w/over 1000 units. Experience req'd w/all housing regulatory agencies, building maintenance & construction; also a 40 person staff.
Fax resume to: 212-543-2287

RECEPTIONIST-MEDICAL
Lite work, midtown, $15/hr. Req: good presence, excel diction, affable personality. Call 212-980-9685 lve msg.

RECEPTIONIST WANTED
Busy phones and light data entry. Immediate hire. Fax resume and salary requirements to (718) 793-4841.

Restaurant/Food Service Manager - Port Washington, NY. Supervises & schedules food & beverage preparation, plans menus & estimates costs. Deals with food suppliers & schedules delivery of fresh food & beverages. Meets with representatives groups or organizations to book parties, banquets & conferences & plans details such as number of persons expected, food service schedule & other special instructions. Helps customers plan for

HIGHEST CASH PRICES PAID!
We need ALL ROLEX Watches, Patek, Cartier, Vacheron, Bulgari, Tiffany, Breitling, Piaget, etc., new, used or broken. We also need all Gold, Diamond, and Estate Jewelry. Also all stamps and coin collections, gold and silver coins, autographed letters, baseball cards, basically anything of value. We can also travel to your home or office. We are licensed, bonded and insured. Appraisals available.
Call 212-541-6868
Plaza Galleries 101 W 57 St (6th Ave)

BUYERS WANTS
(3500)

WE BUY ANY ITEMS
FOR OUR CLOSEOUT STORES
212-819-9780; FAX 212-819-9786 Webers

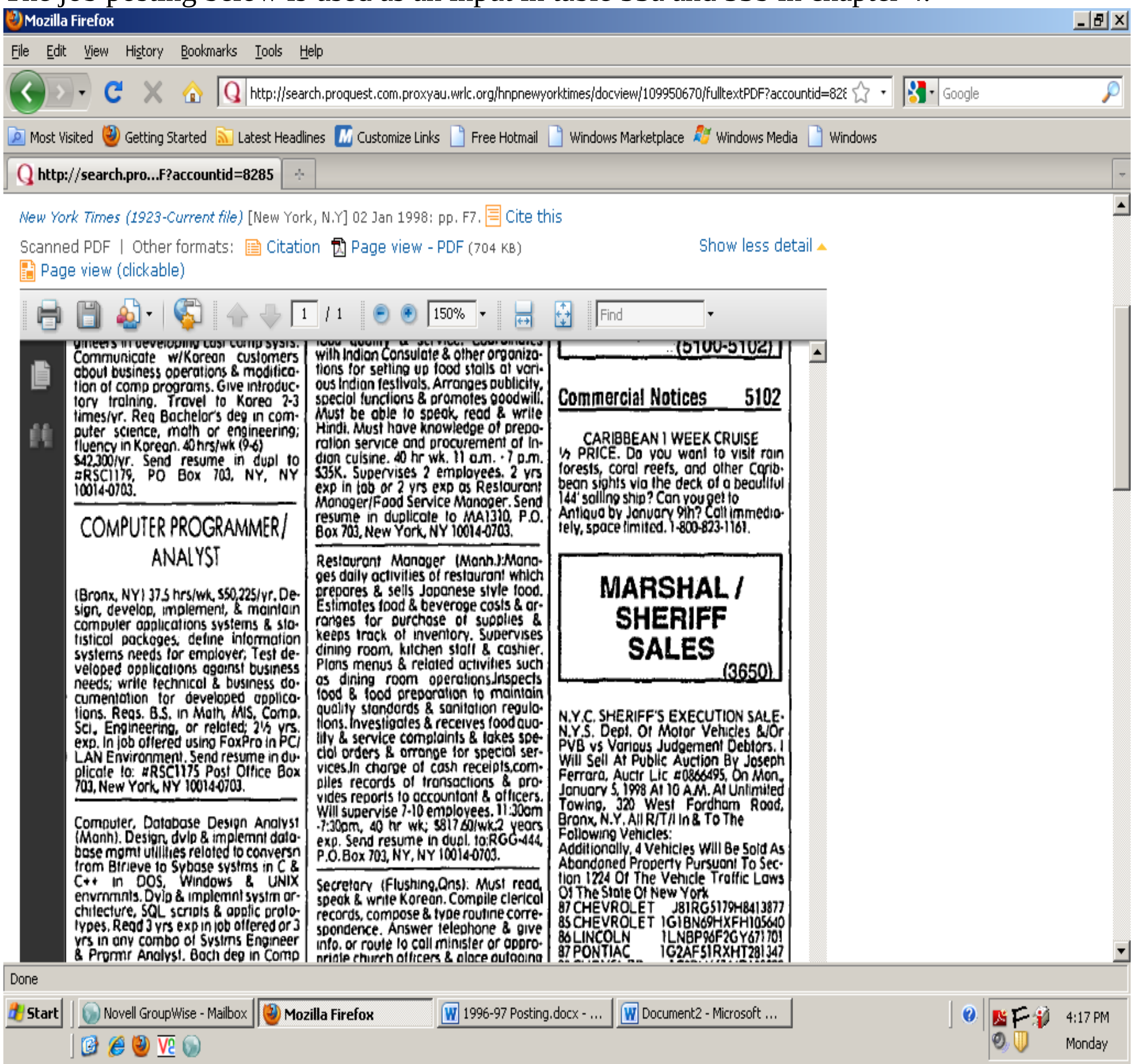
PUBLIC AND COMMERCIAL

Classified ad 3 -- no title. (1998, *New York Times (1923-Current File)*, pp. F7. Retrieved from http://search.proquest.com/docview/109950670?accountid=8285

Appendix E2:

JOB POSTING 1998

The job posting below is used as an input in table 53a and 53b in chapter 4.



New York Times (1923-Current file) [New York, N.Y] 02 Jan 1998: pp. F7.

gineers in developing cust comp systs. Communicate w/Korean customers about business operations & modification of comp programs. Give introductory training. Travel to Korea 2-3 times/yr. Req Bachelor's deg in computer science, math or engineering; fluency in Korean. 40 hrs/wk (9-6) $42,300/yr. Send resume in dupl to #RSC1179, PO Box 703, NY, NY 10014-0703.

COMPUTER PROGRAMMER/ ANALYST

(Bronx, NY) 37.5 hrs/wk, $50,225/yr. Design, develop, implement, & maintain computer applications systems & statistical packages, define information systems needs for employer; Test developed applications against business needs; write technical & business documentation for developed applications. Reqs. B.S. in Math, MIS, Comp. Sci., Engineering, or related; 2½ yrs. exp. in job offered using FoxPro in PC/LAN Environment. Send resume in duplicate to: #RSC1175 Post Office Box 703, New York, NY 10014-0703.

Computer, Database Design Analyst (Manh). Design, dvlp & implemnt database mgmt utilities related to conversn from Btrieve to Sybase systms in C & C++ in DOS, Windows & UNIX envrnmnts. Dvlp & implemnt systm architecture, SQL scripts & applic prototypes. Reqd 3 yrs exp in job offered or 3 yrs in any combo of Systms Engineer & Prgmr Analyst. Bach deg in Comp

food quality & service. Coordinates with Indian Consulate & other organizations for setting up food stalls at various Indian festivals. Arranges publicity, special functions & promotes goodwill. Must be able to speak, read & write Hindi. Must have knowledge of preparation service and procurement of Indian cuisine. 40 hr wk. 11 a.m. - 7 p.m. $35K. Supervises 2 employees. 2 yrs exp in job or 2 yrs exp as Restaurant Manager/Food Service Manager. Send resume in duplicate to MA1310, P.O. Box 703, New York, NY 10014-0703.

Restaurant Manager (Manh.):Manages daily activities of restaurant which prepares & sells Japanese style food. Estimates food & beverage costs & arranges for purchase of supplies & keeps track of inventory. Supervises dining room, kitchen staff & cashier. Plans menus & related activities such as dining room operations.Inspects food & food preparation to maintain quality standards & sanitation regulations. Investigates & receives food quality & service complaints & takes special orders & arrange for special services.In charge of cash receipts,compiles records of transactions & provides reports to accountant & officers. Will supervise 7-10 employees. 11:30am -7:30pm, 40 hr wk; $817.60/wk;2 years exp. Send resume in dupl. to:RGG-444, P.O.Box 703, NY, NY 10014-0703.

Secretary (Flushing,Qns): Must read, speak & write Korean. Compile clerical records, compose & type routine correspondence. Answer telephone & give info. or route to call minister or appropriate church officers & place outgoing

(5100-5102)

Commercial Notices 5102

CARIBBEAN 1 WEEK CRUISE ½ PRICE. Do you want to visit rain forests, coral reefs, and other Caribbean sights via the deck of a beautiful 144' sailing ship? Can you get to Antigua by January 9th? Call immediately, space limited. 1-800-823-1161.

MARSHAL / SHERIFF SALES (3650)

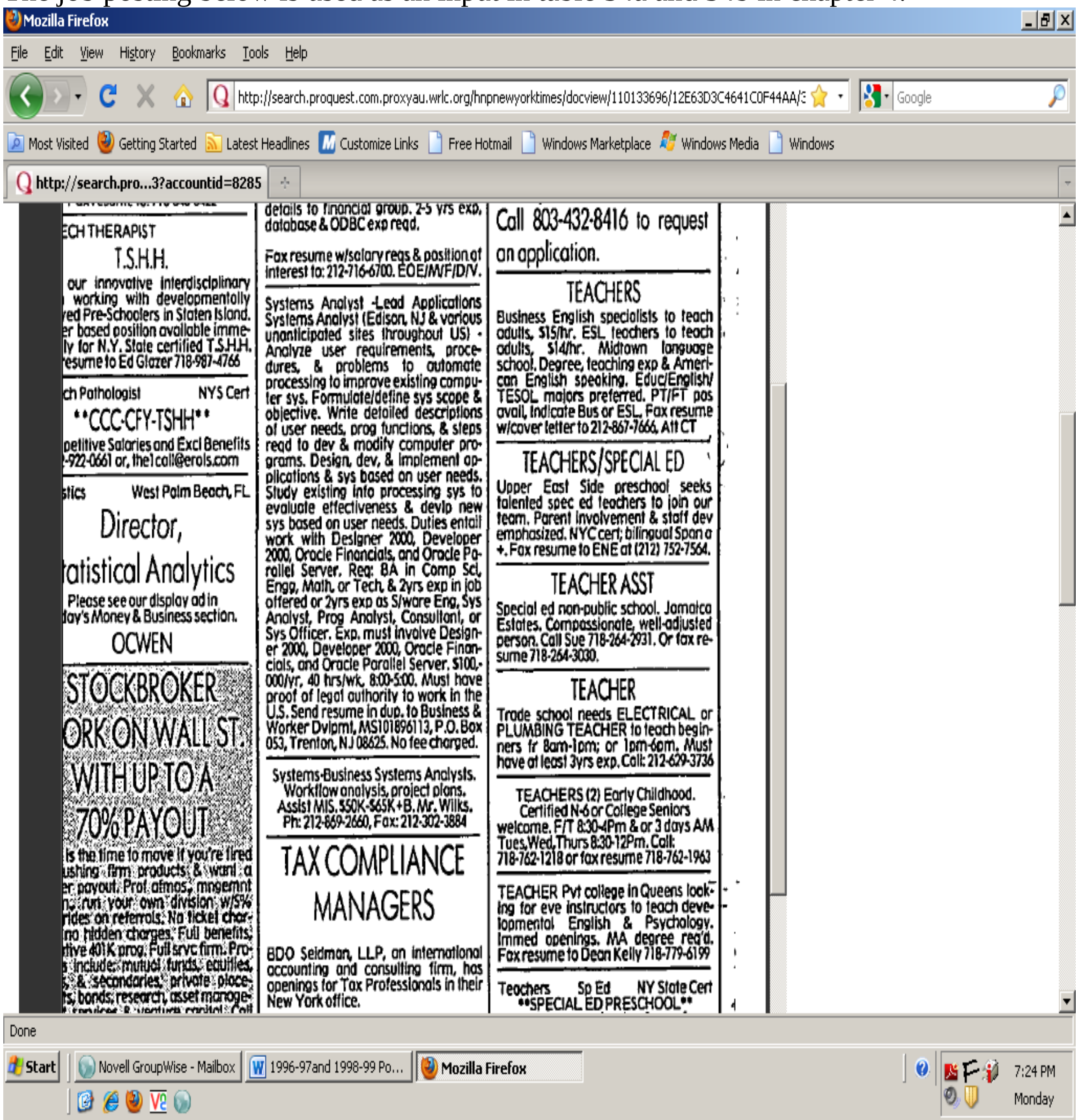
N.Y.C. SHERIFF'S EXECUTION SALE-N.Y.S. Dept. Of Motor Vehicles &/Or PVB vs Various Judgement Debtors. I Will Sell At Public Auction By Joseph Ferrara, Auctr Lic #0866495, On Mon., January 5, 1998 At 10 A.M. At Unlimited Towing, 320 West Fordham Road, Bronx, N.Y. All R/T/I In & To The Following Vehicles:
Additionally, 4 Vehicles Will Be Sold As Abandoned Property Pursuant To Section 1224 Of The Vehicle Traffic Laws Of The State Of New York
87 CHEVROLET J81RG5179H8413877
85 CHEVROLET 1G1BN69HXFH105640
86 LINCOLN 1LNBP96F2GY671701
87 PONTIAC 1G2AF51RXHT281347

Classified ad 3 -- no title. (1998, *New York Times (1923-Current File)*, pp. F7. Retrieved from http://search.proquest.com/docview/109950670?accountid=8285

Appendix E3:

JOB POSTING 1999

The job posting below is used as an input in table 54a and 54b in chapter 4.



ECH THERAPIST

T.S.H.H.

our innovative interdisciplinary working with developmentally ed Pre-Schoolers in Staten Island. er based position available imme- ly for N.Y. State certified T.S.H.H. esume to Ed Glazer 718-987-4766

ch Pathologist NYS Cert

CCC-CFY-TSHH

petitive Salaries and Excl Benefits -922-0661 or, the1call@erols.com

stics West Palm Beach, FL

Director,

tatistical Analytics

Please see our display ad in day's Money & Business section.

OCWEN

STOCKBROKER ORK ON WALL ST. WITH UP TO A 70% PAYOUT

is the time to move if you're tired ushing firm products & want a er payout. Prof. atmos., mngemnt n. run your own division w/5% rides on referrals. No ticket char- no hidden charges. Full benefits, tive 401K prog. Full srvc firm. Pro- s include mutual funds, equities, s & secondaries, private place- s, bonds, research, asset manage-

details to financial group. 2-5 yrs exp, database & ODBC exp read.

Fax resume w/salary reqs & position of interest to: 212-716-6700. EOE/M/F/D/V.

Systems Analyst -Lead Applications Systems Analyst (Edison, NJ & various unanticipated sites throughout US) - Analyze user requirements, procedures, & problems to automate processing to improve existing computer sys. Formulate/define sys scope & objective. Write detailed descriptions of user needs, prog functions, & steps read to dev & modify computer programs. Design, dev, & implement applications & sys based on user needs. Study existing info processing sys to evaluate effectiveness & devlp new sys based on user needs. Duties entail work with Designer 2000, Developer 2000, Oracle Financials, and Oracle Parallel Server. Req: BA in Comp Sci, Engg, Math, or Tech. & 2yrs exp in job offered or 2yrs exp as S/ware Eng, Sys Analyst, Prog Analyst, Consultant, or Sys Officer. Exp. must involve Designer 2000, Developer 2000, Oracle Financials, and Oracle Parallel Server. $100,000/yr, 40 hrs/wk, 8:00-5:00. Must have proof of legal authority to work in the U.S. Send resume in dup. to Business & Worker Dvlpmt, MS101896113, P.O. Box 053, Trenton, NJ 08625. No fee charged.

Systems-Business Systems Analysts. Workflow analysis, project plans. Assist MIS. $50K-$65K+B. Mr. Wilks. Ph: 212-869-2660, Fax: 212-302-3884

TAX COMPLIANCE MANAGERS

BDO Seidman, LLP, an international accounting and consulting firm, has openings for Tax Professionals in their New York office.

Call 803-432-8416 to request an application.

TEACHERS

Business English specialists to teach adults, $15/hr. ESL teachers to teach adults, $14/hr. Midtown language school. Degree, teaching exp & American English speaking. Educ/English/TESOL majors preferred. PT/FT pos avail. Indicate Bus or ESL. Fax resume w/cover letter to 212-867-7666, Att CT

TEACHERS/SPECIAL ED

Upper East Side preschool seeks talented spec ed teachers to join our team. Parent involvement & staff dev emphasized. NYC cert; bilingual Span a +. Fax resume to ENE at (212) 752-7564.

TEACHER ASST

Special ed non-public school. Jamaica Estates. Compassionate, well-adjusted person. Call Sue 718-264-2931. Or fax resume 718-264-3030.

TEACHER

Trade school needs ELECTRICAL or PLUMBING TEACHER to teach beginners fr 8am-1pm; or 1pm-6pm. Must have at least 3yrs exp. Call: 212-629-3736

TEACHERS (2) Early Childhood. Certified N-6 or College Seniors welcome. F/T 8:30-4Pm & or 3 days AM Tues,Wed,Thurs 8:30-12Pm. Call: 718-762-1218 or fax resume 718-762-1963

TEACHER Pvt college in Queens looking for eve instructors to teach developmental English & Psychology. Immed openings. MA degree req'd. Fax resume to Dean Kelly 718-779-6199

Teachers Sp Ed NY State Cert

SPECIAL ED PRESCHOOL

Classified ad 23 -- no title. (1999, *New York Times (1923-Current File)*, pp. W19. Retrieved from http://search.proquest.com/docview/110133696?accountid=8285

Appendix E4:

JOB POSTING 2000

The job posting below is used as an input in table 55a and 55b in chapter 4.

Help Wanted 2600

PRINTING

HIGHLY QUALIFIED PRESSMEN WANTED

Coral Graphic Services, specialists in high-quality, multi-color printing for major corps is now recruiting skilled craftsmen for pressroom positions running 28" & 40" sheetfed presses

To qualify you must have a strong work ethic & a commitment to excellence that drives you to be the best in your field. Great work environment, generous compensation & bnfts avail for the select few who are capable of working in our state-of the-art facility

Experience on Komori's a + but not a necessity. Positions avail on all 3 shifts for 1st & 2nd Pressmen & Operators.

Contact David Spielberg @ 516-349-5187 or Fax your resume to my attention at 516-576-2153 or mail to
CORAL GRAPHIC SERVICES
840 South Broadway
Hicksville, NY 11801
All inquiries held in strict confidence
Equal Opportunity Employer M/F

Printing & Packaging
Materials Coordinator
Project Management

This is a new important position for a high energy, analytical, computer skilled indiv. National company offers a challenge, excel pay and full benefits. Your reply is strictly confidential. Fax resumes to 516-883-0604 Attn. Kathi Grantel

Printing

Help Wanted 2600

Prog BONUS

pro DP

The Professional Data Processing Search Firm

(212) 684-4069 FAX 684-2245
E-Mail - prodp@aol.com

UNIX/C++ $80-100K

Analysis, development & coding of an Object Oriented Market Data application consisting of a component oriented architecture. 3+ years UNIX/C++ and C. New Development Migrating to JAVA. Will Train

UNIX Systems Admin To$125K

2-5 years UNIX Systems Administration with SUN/Solaris or HP/UX exp NFS/NIS and Shell Scripting exp

JAVA Script/UNIX To$120K

Major North Jersey Pharmaceutical Co seeking JAVA Script developers in a UNIX environment. Any Oracle or Broadvision a plus. Train BroadVision

JAVA $80-120K

Currently seeking JAVA server side developers with any J2EE, EJB, RMI, JSP or Swing to develop a Securities Lending Trading system. Stock Options and Bonus. NYC & Connecticut available

Visual Foxpro $70-100K

3+ years Visual Foxpro & hands on Project Management with any SQL or Xbase Database. Any UNIX, DB2, Or

Help Wanted 2600

PROGRAMMER/VP Apps Devel

VP APPS DEV

$150-180,000

Major Financial Services Firm has embarked on a major application development effort to enhance its E Commerce, & B2B initiatives while strengthening its Client Server migration/integration from an aging Legacy environ. Significant changes remain to be architected & implemented. Ideal candidate will have come from a strong development and production systems environment with demonstrated management leadership and have a strong enterprise wide architecture & planning capability as well as a thorough first hand Web-enabled technologies implementation bkgd. We require proven technical & managerial leadership capable of fulfilling the IT solution to the business requirement. Outstanding benefits and financial packages. Contact in confidence:

Herb Douglas (212) 683-8255
FAX (212) 686-0762

DONAHUE & MOORE
(We hunt tigers.)

WEB SITE: www.DandM-Assoc.com
E-M: herbdouglas@DandM Assoc.com

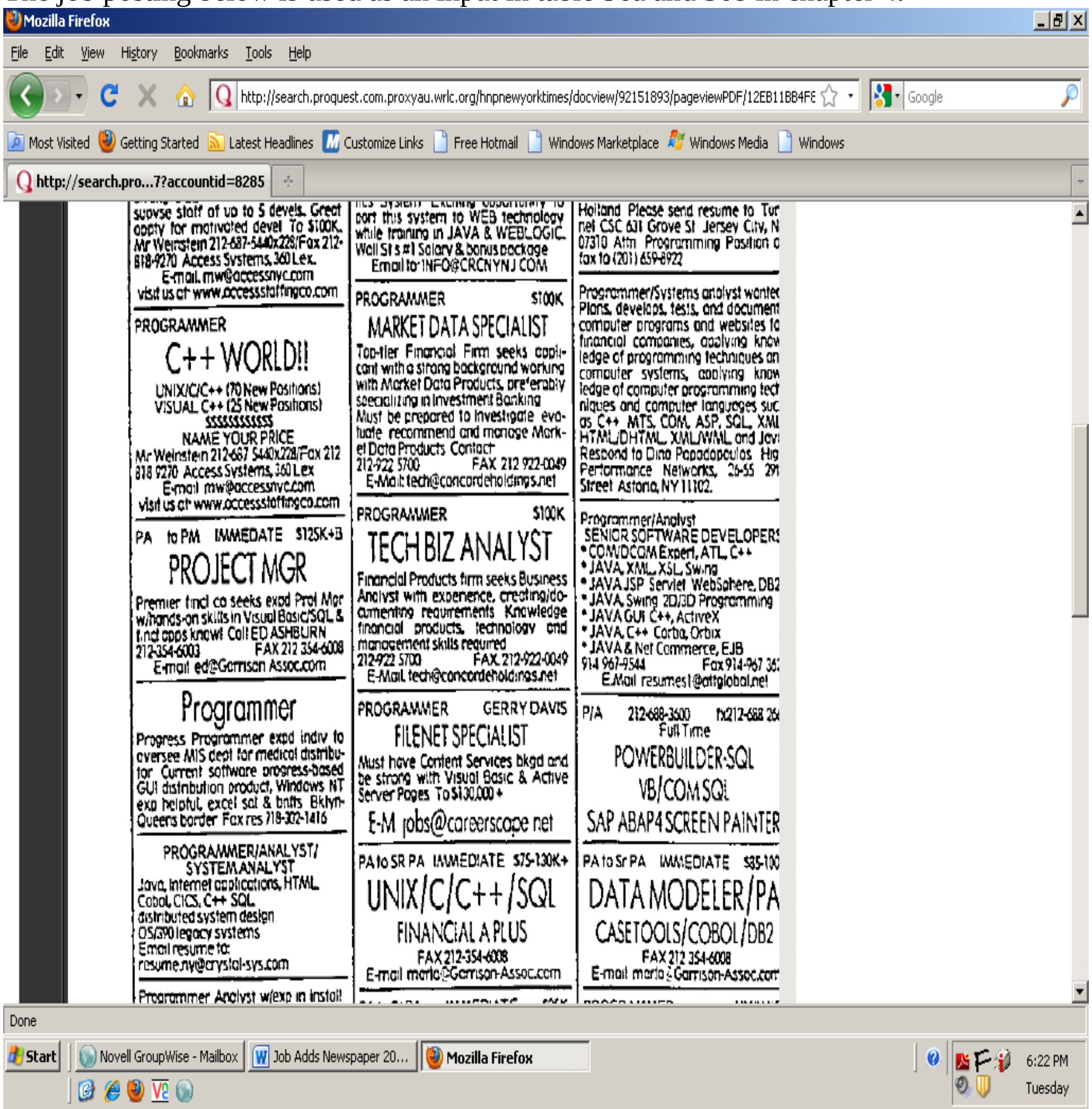
PROGRAMMER NY/NJ/WESTCH

Classified ad 563 -- no title. (2000, *New York Times (1923-Current File)*, pp. W29. Retrieved from http://search.proquest.com/docview/91447626

Appendix E5:

JOB POSTING 2001

The job posting below is used as an input in table 56a and 56b in chapter 4.



supvse staff of up to 5 devels. Great opty for motivated devel To $100K. Mr Weinstein 212-687-5440x228/Fax 212-818-9270 Access Systems, 360 Lex. E-mail. mw@accessnyc.com visit us at www.accessstaffingco.com

PROGRAMMER
C++ WORLD!!
UNIX/C/C++ (70 New Positions)
VISUAL C++ (25 New Positions)
$$$$$$$$$$$
NAME YOUR PRICE
Mr Weinstein 212-687 5440x228/Fax 212 818 9270 Access Systems, 360 Lex
E-mail mw@accessnyc.com
visit us at www.accessstaffingco.com

PA to PM IMMEDIATE $125K+B
PROJECT MGR
Premier fincl co seeks exsd Proj Mgr w/hands-on skills in Visual Basic/SQL & fincl apps know! Call ED ASHBURN 212-354-6003 FAX 212 354-6008
E-mail ed@Garrison Assoc.com

Programmer
Progress Programmer exsd indiv to oversee MIS dept for medical distributor. Current software progress-based GUI distribution product, Windows NT exp helpful, excel sal & bnfts Bklyn-Queens border Fax res 718-302-1416

PROGRAMMER/ANALYST/ SYSTEM ANALYST
Java, Internet applications, HTML, Cobol, CICS, C++ SQL, distributed system design OS/390 legacy systems
Email resume to: resume.ny@crystal-sys.com

Programmer Analyst w/exp in install

port this system to WEB technology while training in JAVA & WEBLOGIC. Wall St s #1 Salary & bonus package
Email to: INFO@CRCNYNJ.COM

PROGRAMMER $100K
MARKET DATA SPECIALIST
Top-tier Financial Firm seeks applicant with a strong background working with Market Data Products, preferably specializing in Investment Banking. Must be prepared to investigate, evaluate, recommend and manage Market Data Products. Contact 212-922 5700 FAX 212 922-0049
E-Mail: tech@concordeholdings.net

PROGRAMMER $100K
TECH BIZ ANALYST
Financial Products firm seeks Business Analyst with experience, creating/documenting requirements. Knowledge financial products, technology and management skills required 212-922 5700 FAX 212-922-0049
E-Mail tech@concordeholdings.net

PROGRAMMER GERRY DAVIS
FILENET SPECIALIST
Must have Content Services bkgd and be strong with Visual Basic & Active Server Pages To $130,000+
E-M jobs@careerscope.net

PA to SR PA IMMEDIATE $75-130K+
UNIX/C/C++/SQL
FINANCIAL A PLUS
FAX 212-354-6008
E-mail marlo@Garrison-Assoc.com

Holland Please send resume to Turnel CSC 631 Grove St Jersey City, NJ 07310 Attn Programming Position or fax to (201) 659-8922

Programmer/Systems analyst wanted Plans, develops, tests, and documents computer programs and websites for financial companies, applying knowledge of programming techniques and computer systems, applying knowledge of computer programming techniques and computer languages such as C++ MTS, COM, ASP, SQL, XML HTML/DHTML, XML/WML and Java. Respond to Dino Papadopoulos High Performance Networks, 26-55 29th Street Astoria, NY 11102.

Programmer/Analyst
SENIOR SOFTWARE DEVELOPERS
• COM/DCOM Expert, ATL, C++
• JAVA, XML, XSL, Swing
• JAVA JSP Servlet WebSphere, DB2
• JAVA, Swing 2D/3D Programming
• JAVA GUI C++, ActiveX
• JAVA, C++ Corba, Orbix
• JAVA & Net Commerce, EJB
914 967-9544 Fax 914-967 36
E.Mail resumes1@attglobal.net

P/A 212-688-3500 fx212-688 26
Full Time
POWERBUILDER-SQL
VB/COM SQL
SAP ABAP4 SCREEN PAINTER

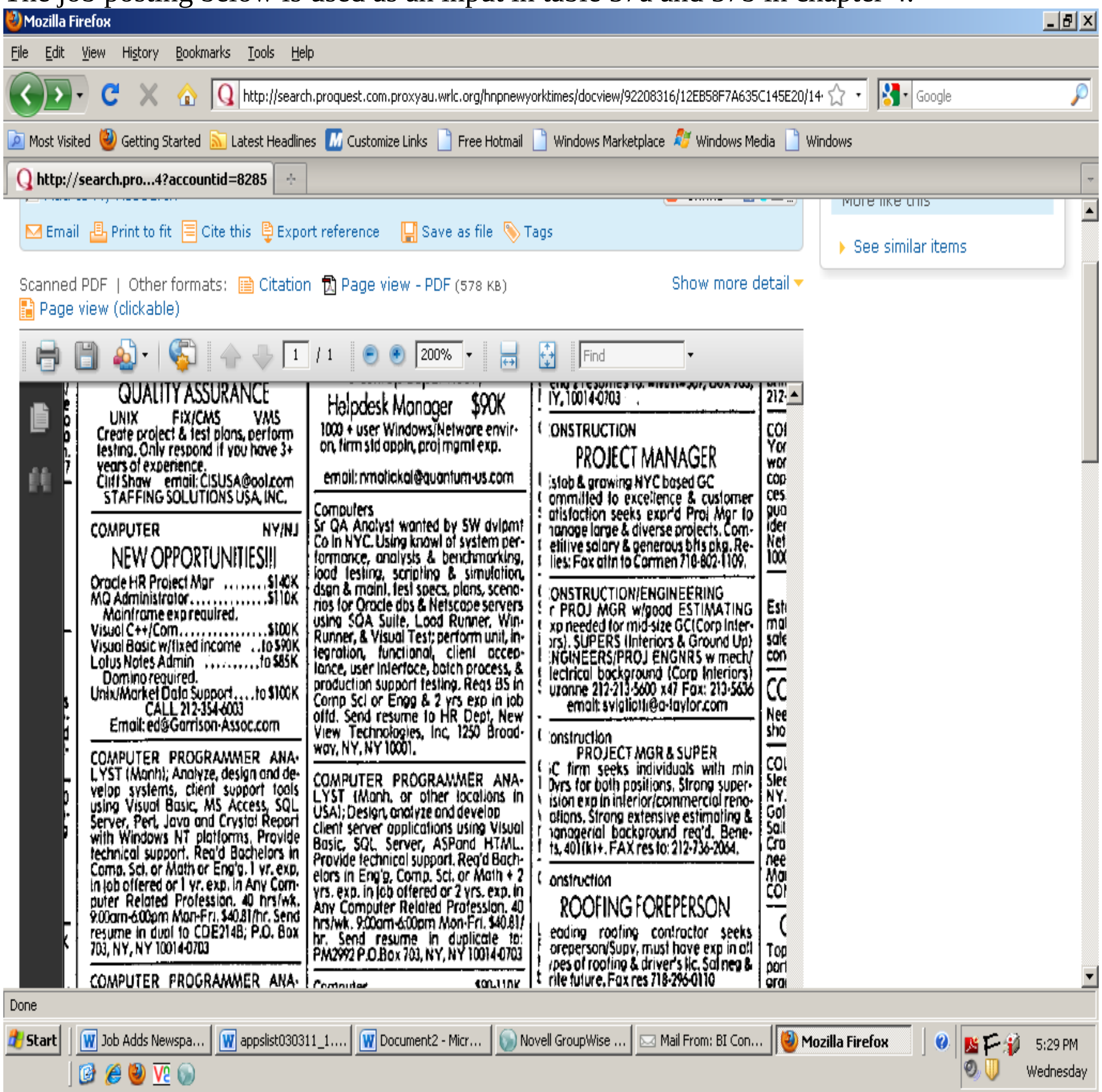
PA to Sr PA IMMEDIATE $85-100
DATA MODELER/PA
CASE TOOLS/COBOL/DB2
FAX 212 354-6008
E-mail marlo@Garrison-Assoc.com

Classified ad 461 -- no title. (2001, *New York Times (1923-Current File)*, pp. W17. Retrieved from http://search.proquest.com/docview/92151893?accountid=8285

Appendix E6:

JOB POSTING 2002

The job posting below is used as an input in table 57a and 57b in chapter 4.



QUALITY ASSURANCE

UNIX FIX/CMS VMS

Create project & test plans, perform testing. Only respond if you have 3+ years of experience.

Cliff Shaw email: CISUSA@aol.com

STAFFING SOLUTIONS USA, INC.

COMPUTER NY/NJ

NEW OPPORTUNITIES!!!

Oracle HR Project Mgr $140K

MQ Administrator..............$110K

Mainframe exp required.

Visual C++/Com................$100K

Visual Basic w/fixed income ..to $90K

Lotus Notes Admin ....to $85K

Domino required.

Unix/Market Data Support....to $100K

CALL 212-354-6003

Email: ed@Garrison-Assoc.com

COMPUTER PROGRAMMER ANALYST (Manh); Analyze, design and develop systems, client support tools using Visual Basic, MS Access, SQL Server, Perl, Java and Crystal Report with Windows NT platforms. Provide technical support. Req'd Bachelors in Comp. Sci. or Math or Eng'g. 1 yr. exp. in job offered or 1 yr. exp. in Any Computer Related Profession. 40 hrs/wk. 9:00am-6:00pm Mon-Fri. $40.81/hr. Send resume in dual to CDE214B; P.O. Box 703, NY, NY 10014-0703

Helpdesk Manager $90K

1000 + user Windows/Network environ, firm std appln, proj mgmt exp.

email: rmatickal@quantum-us.com

Computers

Sr QA Analyst wanted by SW dvlpmt Co in NYC. Using knowl of system performance, analysis & benchmarking, load testing, scripting & simulation, dsgn & maint. test specs, plans, scenarios for Oracle dbs & Netscape servers using SQA Suite, Load Runner, WinRunner, & Visual Test; perform unit, integration, functional, client acceptance, user interface, batch process, & production support testing. Reqs BS in Comp Sci or Engg & 2 yrs exp in job offd. Send resume to HR Dept, New View Technologies, Inc, 1250 Broadway, NY, NY 10001.

COMPUTER PROGRAMMER ANALYST (Manh. or other locations in USA); Design, analyze and develop client server applications using Visual Basic, SQL Server, ASP and HTML. Provide technical support. Req'd Bachelors in Eng'g, Comp. Sci. or Math + 2 yrs. exp. in job offered or 2 yrs. exp. in Any Computer Related Profession. 40 hrs/wk. 9:00am-6:00pm Mon-Fri. $40.81/hr. Send resume in duplicate to: PM2992 P.O.Box 703, NY, NY 10014-0703

CONSTRUCTION

PROJECT MANAGER

Estab & growing NYC based GC committed to excellence & customer satisfaction seeks exp'd Proj Mgr to manage large & diverse projects. Competitive salary & generous bfts pkg. Resumes: Fax attn to Carmen 718-802-1109.

CONSTRUCTION/ENGINEERING

Sr PROJ MGR w/good ESTIMATING exp needed for mid-size GC (Corp Interiors). SUPERS (Interiors & Ground Up) ENGINEERS/PROJ ENGNRS w mech/electrical background (Corp Interiors) Suzanne 212-213-5600 x47 Fax: 213-5636 email: svigliotti@a-taylor.com

Construction

PROJECT MGR & SUPER

GC firm seeks individuals with min 5yrs for both positions. Strong supervision exp in interior/commercial renovations. Strong extensive estimating & managerial background req'd. Benefits, 401(k)+. FAX res to: 212-736-2064.

Construction

ROOFING FOREPERSON

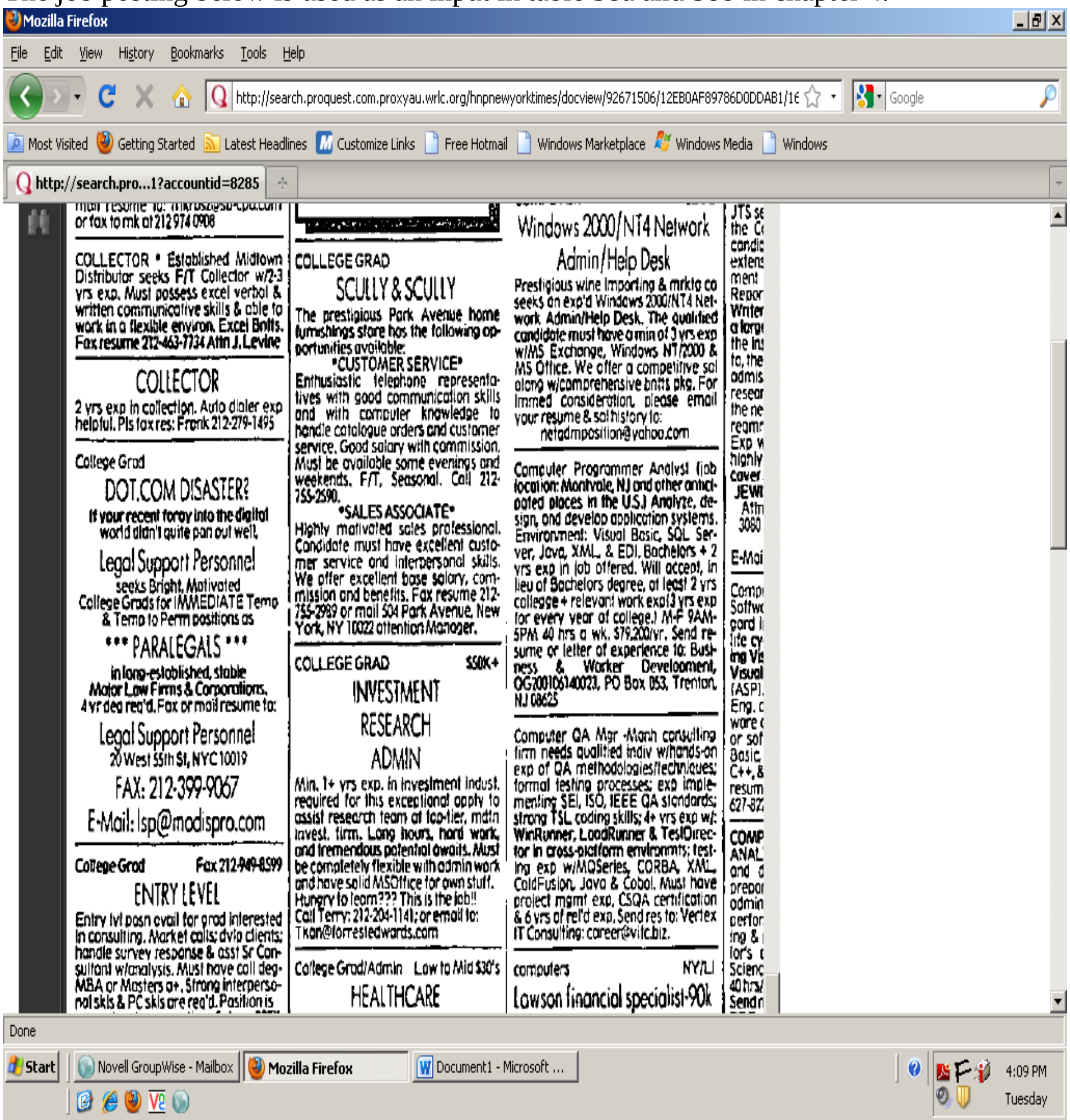
Leading roofing contractor seeks foreperson/Supv, must have exp in all types of roofing & driver's lic. Sal neg & a rite future. Fax res 718-296-0110

Classified ad 9 -- no title. (2002, *New York Times (1923-Current File)*, pp. J3. Retrieved from http://search.proquest.com/docview/92208316?accountid=8285

Appendix E7:

JOB POSTING 2003

The job posting below is used as an input in table 58a and 58b in chapter 4.



or fax to mk at 212 974 0908

COLLECTOR • Established Midtown Distributor seeks F/T Collector w/2-3 yrs exp. Must possess excel verbal & written communicative skills & able to work in a flexible environ. Excel Bnfts. Fax resume 212-463-7734 Attn J. Levine

COLLECTOR

2 yrs exp in collection. Auto dialer exp helpful. Pls fax res: Frank 212-279-1495

College Grad

DOT.COM DISASTER?

If your recent foray into the digital world didn't quite pan out well,

Legal Support Personnel

seeks Bright, Motivated College Grads for IMMEDIATE Temp & Temp to Perm positions as

*** PARALEGALS ***

in long-established, stable Major Law Firms & Corporations. 4 yr deg req'd. Fax or mail resume to:

Legal Support Personnel

20 West 55th St, NYC 10019

FAX: 212-399-9067

E-Mail: lsp@modispro.com

College Grad Fax 212-949-8599

ENTRY LEVEL

Entry lvl posn avail for grad interested in consulting. Market calls; dvlp clients; handle survey response & asst Sr Consultant w/analysis. Must have coll deg-MBA or Masters a+. Strong interpersonal skls & PC skls are req'd. Position is

COLLEGE GRAD

SCULLY & SCULLY

The prestigious Park Avenue home furnishings store has the following opportunities available:

CUSTOMER SERVICE

Enthusiastic telephone representatives with good communication skills and with computer knowledge to handle catalogue orders and customer service. Good salary with commission. Must be available some evenings and weekends. F/T, Seasonal. Call 212-755-2590.

SALES ASSOCIATE

Highly motivated sales professional. Candidate must have excellent customer service and interpersonal skills. We offer excellent base salary, commission and benefits. Fax resume 212-755-2989 or mail 504 Park Avenue, New York, NY 10022 attention Manager.

COLLEGE GRAD $50K+

INVESTMENT RESEARCH ADMIN

Min. 1+ yrs exp. in investment indust. required for this exceptional opty to assist research team at top-tier, mdtn invest. firm. Long hours, hard work, and tremendous potential awaits. Must be completely flexible with admin work and have solid MSOffice for own stuff. Hungry to learn??? This is the job!! Call Terry: 212-204-1141; or email to: Tkon@forrestedwards.com

College Grad/Admin Low to Mid $30's

HEALTHCARE

Windows 2000/NT4 Network Admin/Help Desk

Prestigious wine importing & mrktg co seeks an exp'd Windows 2000/NT4 Network Admin/Help Desk. The qualified candidate must have a min of 3 yrs exp w/MS Exchange, Windows NT/2000 & MS Office. We offer a competitive sal along w/comprehensive bnfts pkg. For immed consideration, please email your resume & sal history to: netadmposition@yahoo.com

Computer Programmer Analyst (job location: Montvale, NJ and other anticipated places in the U.S.) Analyze, design, and develop application systems. Environment: Visual Basic, SQL Server, Java, XML, & EDI. Bachelors + 2 yrs exp in job offered. Will accept, in lieu of Bachelors degree, at least 2 yrs college + relevant work exp(3 yrs exp for every year of college.) M-F 9AM-5PM 40 hrs a wk. $79,200/yr. Send resume or letter of experience to: Business & Worker Development, OG200106140023, PO Box 053, Trenton, NJ 08625

Computer QA Mgr -Manh consulting firm needs qualified indiv w/hands-on exp of QA methodologies/techniques; formal testing processes; exp implementing SEI, ISO, IEEE QA standards; strong TSL coding skills; 4+ yrs exp w/: WinRunner, LoadRunner & TestDirector in cross-platform environmts; testing exp w/MQSeries, CORBA, XML, ColdFusion, Java & Cobol. Must have project mgmt exp, CSQA certification & 6 yrs of rel'd exp. Send res to: Vertex IT Consulting: career@vitc.biz.

computers NY/LI

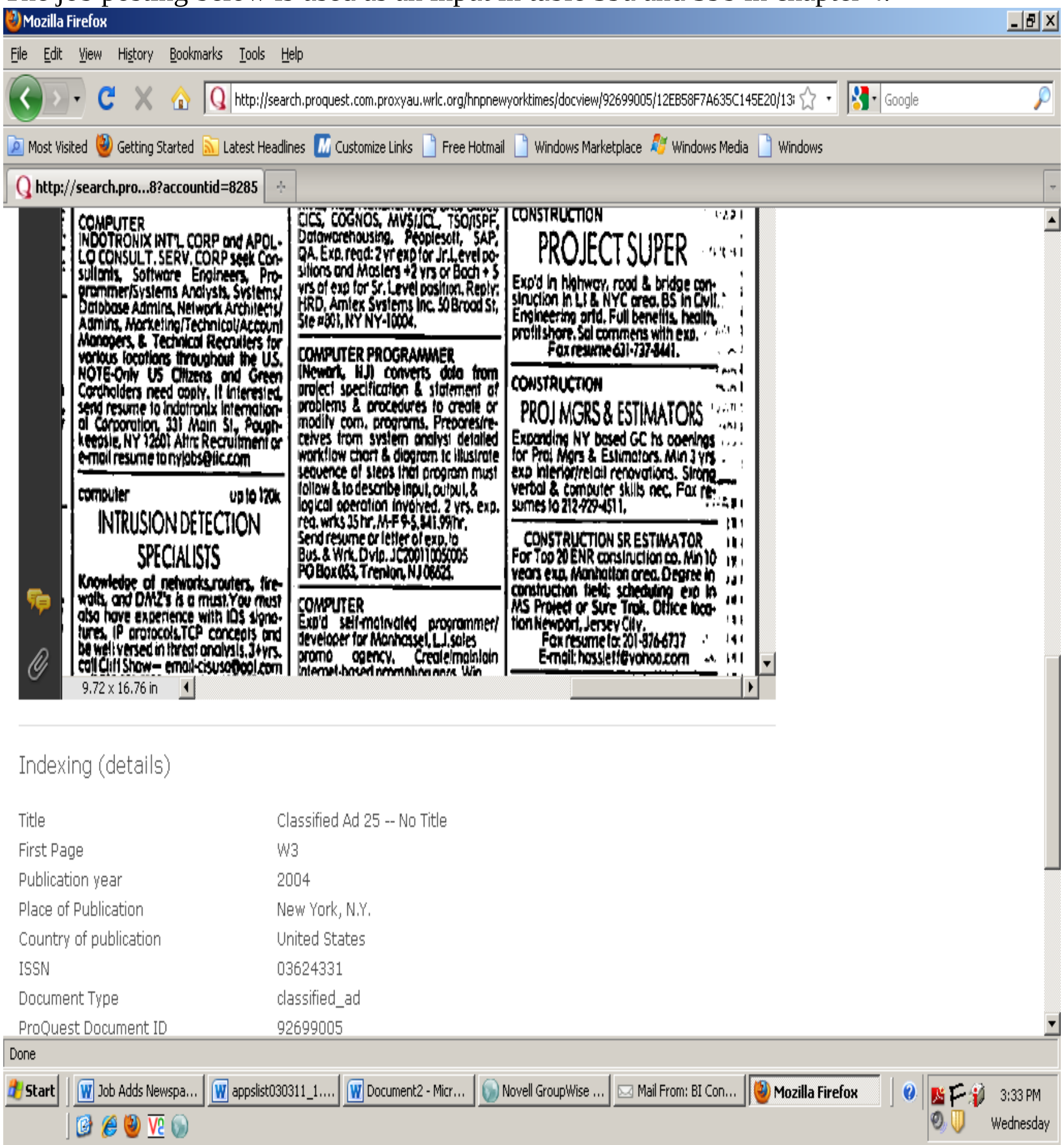
Lowson financial specialist-90k

Classified ad 17 -- no title. (2003, *New York Times (1923-Current File)*, pp. HW3. Retrieved from http://search.proquest.com/docview/92671506?accountid=8285

Appendix E8:

JOB POSTING 2004

The job posting below is used as an input in table 59a and 59b in chapter 4.



COMPUTER
INDOTRONIX INT'L CORP and APOLLO CONSULT. SERV. CORP seek Consultants, Software Engineers, Programmer/Systems Analysts, Systems/Database Admins, Network Architects/Admins, Marketing/Technical/Account Managers, & Technical Recruiters for various locations throughout the U.S. NOTE-Only US Citizens and Green Cardholders need apply. If interested, send resume to Indotronix International Corporation, 331 Main St., Poughkeepsie, NY 12601 Attn: Recruitment or e-mail resume to nyjobs@iic.com

computer up to 120k

INTRUSION DETECTION SPECIALISTS

Knowledge of networks, routers, firewalls, and DMZ's is a must. You must also have experience with IDS signatures, IP protocols, TCP concepts and be well versed in threat analysis, 3+yrs. call Cliff Show— email-cisusa@aol.com

CICS, COGNOS, MVS/JCL, TSO/ISPF, Datawarehousing, Peoplesoft, SAP, QA. Exp. reqd: 2 yr exp for Jr. Level positions and Masters +2 yrs or Bach + 5 yrs of exp for Sr. Level position. Reply: HRD, Amtex Systems Inc. 50 Broad St, Ste #801, NY NY-10004.

COMPUTER PROGRAMMER
(Newark, NJ) converts data from project specification & statement of problems & procedures to create or modify com. programs. Prepares/receives from system analyst detailed workflow chart & diagram to illustrate sequence of steps that program must follow & to describe input, output, & logical operation involved. 2 yrs. exp. req. wrks 35 hr, M-F 9-5, $41.99/hr. Send resume or letter of exp. to Bus. & Wrk. Dvlp. JC20011005005 PO Box 053, Trenton, NJ 08625.

COMPUTER
Exp'd self-motivated programmer/developer for Manhasset, L.I. sales promo agency. Create/maintain internet-based promotional ...

CONSTRUCTION

PROJECT SUPER

Exp'd in highway, road & bridge construction in LI & NYC area. BS in Civil Engineering prfd. Full benefits, health, profit share. Sal commens with exp. Fax resume 631-737-8441.

CONSTRUCTION

PROJ MGRS & ESTIMATORS

Expanding NY based GC hs openings for Proj Mgrs & Estimators. Min 3 yrs exp interior/retail renovations. Strong verbal & computer skills nec. Fax resumes to 212-929-4511.

CONSTRUCTION SR ESTIMATOR
For Top 20 ENR construction co. Min 10 years exp. Manhattan area. Degree in construction field; scheduling exp in MS Project or Sure Trak. Office location Newport, Jersey City.
Fax resume to: 201-876-6737
E-mail: hasslett@yahoo.com

Indexing (details)

Title	Classified Ad 25 -- No Title
First Page	W3
Publication year	2004
Place of Publication	New York, N.Y.
Country of publication	United States
ISSN	03624331
Document Type	classified_ad
ProQuest Document ID	92699005

Classified ad 25 -- no title. (2004, *New York Times (1923-Current File)*, pp. W3. Retrieved from http://search.proquest.com/docview/92699005?accountid=8285

Appendix E9:

JOB POSTING 2005

The job posting below is used as an input in table 60a and 60b in chapter 4.
[jsfg_cinti] Job Posting: Peoplesoft EPM Developer

- *From*: JSFG@xxxxxxxx
- *To*: JSFG@xxxxxxxx, jsfg_cinti@xxxxxxxxxxxxx
- *Date*: Mon, 31 Jan 2005 15:45:33 -0600

The following job opportunity has been posted to the JSFG E-mail Mailing List by JSFG@xxxxxxxx on January 31st, 2005 at 03:45PM (CST).

Date: 01/31/05
JobTitle: Peoplesoft EPM Developer
JobClass: Engineering/Technical
JobDescript: Peoplesoft EPM Developer - Rutherford, NJ

A client of ours in Rutherford, NJ is seeking a Peoplesoft EPM Developer. The **position pays up to $110K annually** and they are only looking for local candidates.

Description:
Provide development and integration role across various vendor products to provide BASEL II reporting on a large risk and finance data warehouse-tools including Peoplesoft EPM. Work with business SMEs to understand requirements and develop solution. This provides a challenging opportunity to work in one of the largest IT projects currently underway to implement BASEL II reporting solution on a large data warehouse for GCIB senior management.

If you are interested in this position, please forward
your Word formatted resume, cover note and salary requirement to
resumes@xxxxxxxxxxxxxxxxxxxxxxxxx Please include the specific position to which
you are applying in the subject line of your email. This will ensure your resume is routed to the correct recruiter as quickly as possible.

Please note, only qualified candidates will be contacted for interviews.
Receipt of your resume will be confirmed if it is sent as an attachment.

Location: Rutherford, NJ
Requirements_Qualif: Qualifications:
4 years of experience with Peoplesoft tools: Peoplecode, peopletools, EPM 8.8, application engine, business interlinks, data modeling, data marts. Experience with Peoplesoft modules such as RWC and Global Consolidation is plus. Good communication skills and working experience in finance project is plus.
WorkType: Full-time Regular
Travel: No
PositionState: Existing
SalaryRange: $100,000-110,000
HowApply_Email: Yes
Company: Work Wonders Staffing, LLC
Person_Requesting_Profiles: Michael Wilmarth
Contact_StreetAddress: 4107 East Woodstock Rd
Contact_City: Cave Creek
Contact_State: AZ
Contact_ZipCode: 85331
Contact_WorkPhone: 6022185133
e-mail: resumes@xxxxxxxxxxxxxxxxxxxxxxx
email: JSFG@xxxxxxxx
bgcolor1: #FFFFFF
Cutoff: 03/30/05
Submit: Submit Form

-------- Env Report --------
REMOTE_HOST:
REMOTE_ADDR: 68.3.149.1
HTTP_USER_AGENT: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET
CLR 1.1.4322)

You can unsubscribe from the list by sending email to jsfg_cinti-request@xxxxxxxxxxxxx with 'unsubscribe' in the Subject field.

Web archive: http://www.freelists.org/archives/jsfg_cinti

Questions to: jsfg@xxxxxxxx

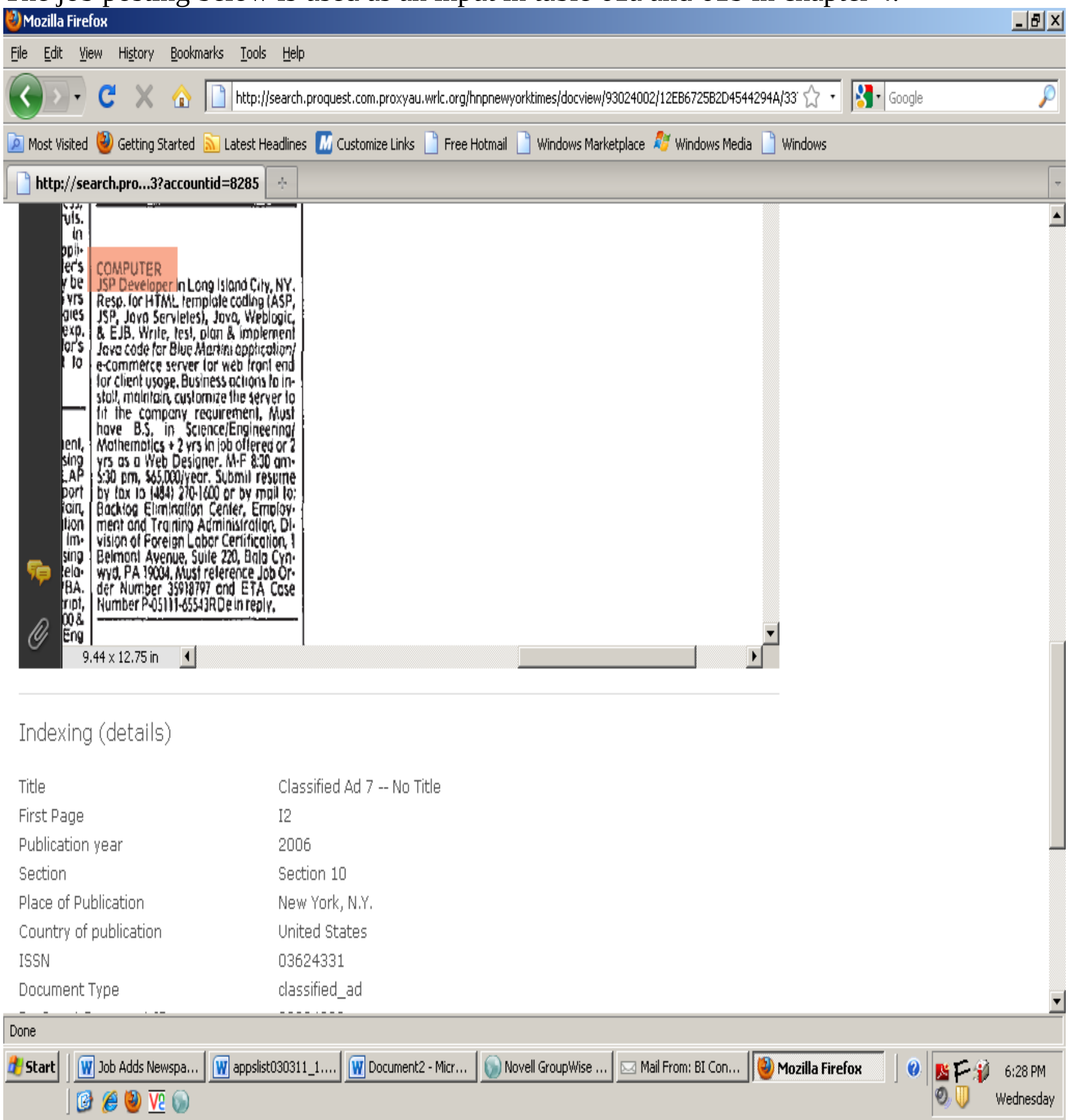
Other related posts:

- » [jsfg_cinti] Job Posting: Peoplesoft EPM Developer http://www.freelists.org/post/jsfg_cinti/Job-Posting-Peoplesoft-EPM-Developer

Appendix E10:

JOB POSTING 2006

The job posting below is used as an input in table 61a and 61b in chapter 4.

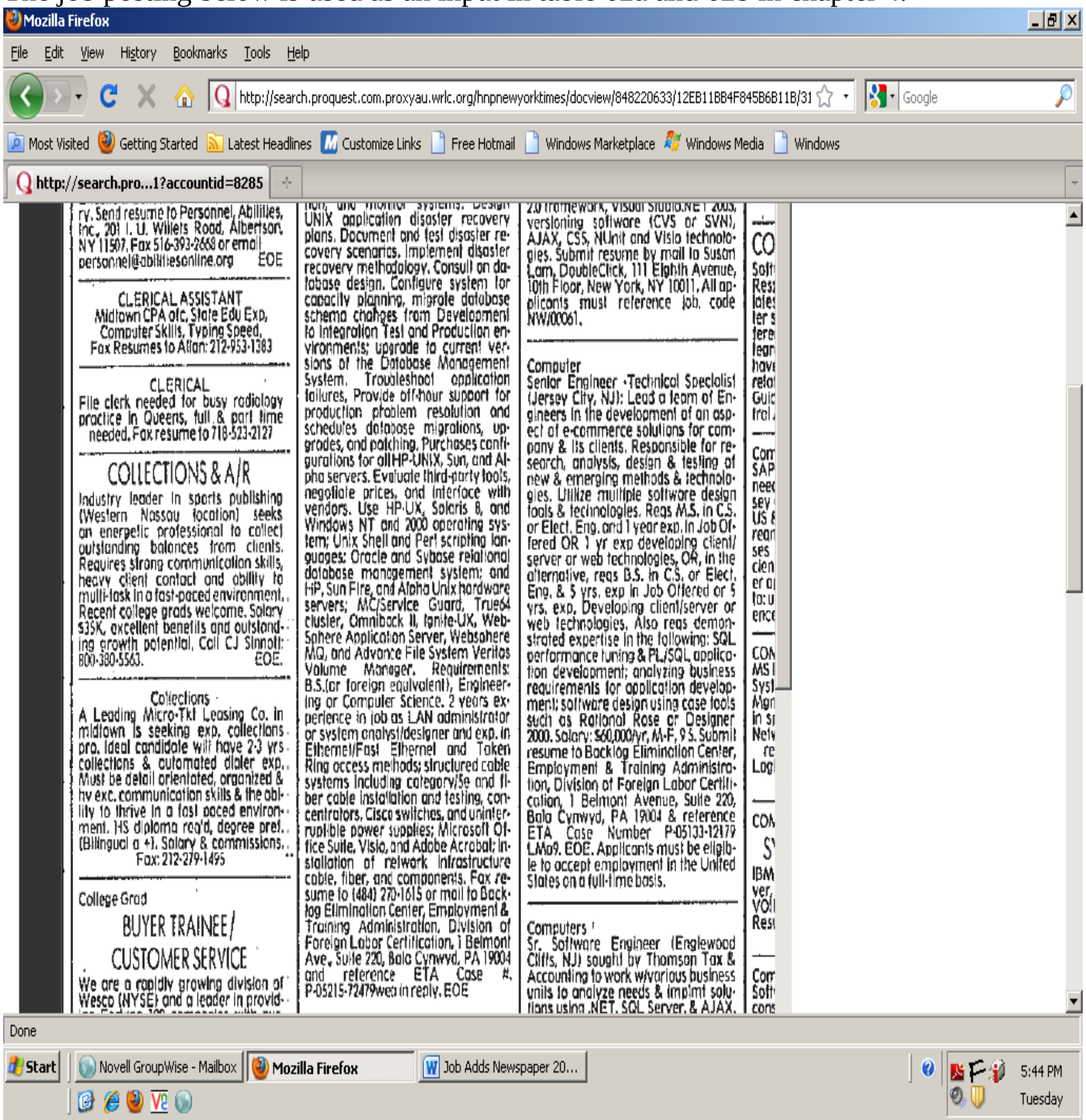


Classified ad 7 -- no title. (2006, *New York Times (1923-Current File)*, pp. I2. Retrieved from http://search.proquest.com/docview/93024002?accountid=8285

Appendix E11:

JOB POSTING 2007

The job posting below is used as an input in table 62a and 62b in chapter 4.



ry. Send resume to Personnel, Abilities, Inc., 201 I. U. Willets Road, Albertson, NY 11507, Fax 516-393-2668 or email personnel@abilitiesonline.org EOE

CLERICAL ASSISTANT
Midtown CPA ofc. State Edu Exp, Computer Skills, Typing Speed, Fax Resumes to Allan: 212-953-1383

CLERICAL
File clerk needed for busy radiology practice in Queens, full & part time needed. Fax resume to 718-523-2127

COLLECTIONS & A/R
Industry leader in sports publishing (Western Nassau location) seeks an energetic professional to collect outstanding balances from clients. Requires strong communication skills, heavy client contact and ability to multi-task in a fast-paced environment. Recent college grads welcome. Salary $35K, excellent benefits and outstanding growth potential. Call CJ Sinnott: 800-380-5563. EOE.

Collections
A Leading Micro-Tkt Leasing Co. in midtown is seeking exp. collections pro. Ideal candidate will have 2-3 yrs collections & automated dialer exp. Must be detail orientated, organized & hv exc. communication skills & the ability to thrive in a fast paced environment. HS diploma req'd, degree pref. (Bilingual a +). Salary & commissions. Fax: 212-279-1495

College Grad
BUYER TRAINEE/
CUSTOMER SERVICE
We are a rapidly growing division of Wesco (NYSE) and a leader in provid-

UNIX application disaster recovery plans. Document and test disaster recovery scenarios. Implement disaster recovery methodology. Consult on database design. Configure system for capacity planning, migrate database schema changes from Development to Integration Test and Production environments; upgrade to current versions of the Database Management System. Troubleshoot application failures. Provide off-hour support for production problem resolution and schedules database migrations, upgrades, and patching. Purchases configurations for all HP-UNIX, Sun, and Alpha servers. Evaluate third-party tools, negotiate prices, and interface with vendors. Use HP-UX, Solaris 8, and Windows NT and 2000 operating system; Unix Shell and Perl scripting languages; Oracle and Sybase relational database management system; and HP, Sun Fire, and Alpha Unix hardware servers; MC/Service Guard, True64 cluster, Omniback II, Ignite-UX, WebSphere Application Server, Websphere MQ, and Advance File System Veritas Volume Manager. Requirements: B.S.(or foreign equivalent), Engineering or Computer Science. 2 years experience in job as LAN administrator or system analyst/designer and exp. in Ethernet/Fast Ethernet and Token Ring access methods; structured cable systems including category/5e and fiber cable installation and testing, concentrators, Cisco switches, and uninterruptible power supplies; Microsoft Office Suite, Visio, and Adobe Acrobat; installation of network infrastructure cable, fiber, and components. Fax resume to (484) 270-1615 or mail to Backlog Elimination Center, Employment & Training Administration, Division of Foreign Labor Certification, 1 Belmont Ave., Suite 220, Bala Cynwyd, PA 19004 and reference ETA Case #, P-05215-72479wea in reply. EOE

2.0 framework, Visual Studio.NET 2005, versioning software (CVS or SVN), AJAX, CSS, NUnit and Visio technologies. Submit resume by mail to Susan Lam, DoubleClick, 111 Eighth Avenue, 10th Floor, New York, NY 10011. All applicants must reference job code NW/00061.

Computer
Senior Engineer -Technical Specialist (Jersey City, NJ): Lead a team of Engineers in the development of an aspect of e-commerce solutions for company & its clients. Responsible for research, analysis, design & testing of new & emerging methods & technologies. Utilize multiple software design tools & technologies. Reqs M.S. in C.S. or Elect. Eng. and 1 year exp. in Job Offered OR 1 yr exp developing client/server or web technologies, OR, in the alternative, reqs B.S. in C.S. or Elect. Eng. & 5 yrs. exp in Job Offered or 5 yrs. exp. Developing client/server or web technologies. Also reqs demonstrated expertise in the following: SQL performance tuning & PL/SQL application development; analyzing business requirements for application development; software design using case tools such as Rational Rose or Designer 2000. Salary: $60,000/yr, M-F, 9-5. Submit resume to Backlog Elimination Center, Employment & Training Administration, Division of Foreign Labor Certification, 1 Belmont Avenue, Suite 220, Bala Cynwyd, PA 19004 & reference ETA Case Number P-05133-12179 LMa9. EOE. Applicants must be eligible to accept employment in the United States on a full-time basis.

Computers
Sr. Software Engineer (Englewood Cliffs, NJ) sought by Thomson Tax & Accounting to work w/various business units to analyze needs & implmt solutions using .NET, SQL Server, & AJAX.

Classified ad 4 -- no title. (2007, *New York Times (1923-Current File)*, pp. C20. Retrieved from http://search.proquest.com/docview/848220633?accountid=8285

Appendix E12:

JOB POSTING 2008

The job posting below is used as an input in table 63a and 63b in chapter 4.

The following job opportunity has been posted to the JSFG E-mail Mailing List.

Date:
JobTitle: Advanced Web Application Programmer - Developer
JobClass: Information Systems
JobDescript: Must have solid understanding of Object Oriented Programming concepts and internet technologies: PHP, MySQL, HTML, XML JavaScript, CSS, DNS
and Apache.

Will be able to develop and maintain all project deliverables including scope, statement of work, schedules, project quality and budget. Must be able to quote programming projects and complete projects in a timely manner with minimal assistance. Should know how to build a PHP web application from the ground up, as well as how to modify and build on existing code. Will manage and enhance web content management systems. Will develop and support e-commerce & payment
processing web sites.

We are looking for an enthusiastic & communicative team member with a positive
attitude; dedication and willingness to work hard often under high production pressure.

If you would like to be part of a fast growing team of professional web designers and programmers, please send us your resume.

Candidates applying must send examples of recent coding samples. Please state what application you used to create your example, how much time was spent on the production of the example, and the purpose of the example.

Please state your availability time and your expectation of starting earnings for this position. Those candidates applying should be local to the Cincinnati/NKY area.

Location: Cincinnati
Requirements_Qualif: - 2 to 5 yrs. Web Application Development experience.
- 2 to 5 yrs. experience working in a Linux, Apache, MySQL, PHP environment.
- Must have strong knowledge of Internet-related systems & protocols (HTTP, HTTPS, DNS, TCP/IP, Proxies, FTP, Shell etc).
- Ecommerce experience necessary.
- Excellent written and oral communication skills.
- Ability to work in a fast-paced, team environment with minimal supervision.
- College degree or equivalent experience required.
Candidates must be able to pass a drug and background check
WorkType: Full-time Regular
Travel: No
PositionState: Existing
SalaryRange: $$50,000.00
HowApply_Email: Yes
Company: Pixels & Dots
Person_Requesting_Profiles: Monte Davis
Contact_StreetAddress: 3181 Linwood Avenue
Contact_City: Cincinnati
Contact_State: OH
Contact_ZipCode: 45208
Contact_WorkPhone: 513-651-9322
Contact_FAX:
email: monte@xxxxxxxxxxxxxxxxx
Industry: Internet Technologies
Comments:

You can unsubscribe from the list by going to the JSFG E-mail Mailing List page (http://www.jsfg.com/listserv.htm), entering your e-mail address, and selecting "Unsubscribe" instead of "Subscribe"

Web archive: http://www.freelists.org/archives/jsfg_cinti

Questions to: jsfg@xxxxxxxx

Appendix E13:

JOB POSTING 2009

The job posting below is used as an input in table 64a and 64b in chapter 4.

-----Original Message-----
From: McNeal, Keith [mailto:kmcneal@xxxxxxx]
Sent: Thursday, January 08, 2009 10:51 AM
To: JSFG
Subject: RE:

Ruth,

Here is what I am looking for here in the city for two of our clients.

1.) Java Developer. (Salary range-60/70k) so this would probably be a 4-7 year person.

3 month right to hire...

java developer with good OO skills that can work in a fast paced environment. they're using java enterprise edition but not using beans etc just java. They also use a lot of open source tools like jboss,, JEDI webserver, freemarker for UI and use restlets and not servlets. so person has to be open to this and quick learner. They're primarily a Linux shop for deployment but don't care if candidate is a windows java developer. Just wants good developer with good OO skills.

2.) Web Developer with a Microsoft background (C#/ASP.Net) Salary open. This is a straight contract position.

Local company has a immediate need for a web development professional to perform a contracted web ad integration service on one of their websites. The site is built on C sharp with community server. This assignment will probably only be 2 to 3 weeks but will blossom into more. The person has to integrate advertisements into community server.

Let me know if you need anything else but I think this should get it.

From: JSFG [mailto:jsfg@xxxxxxxx]
Sent: Thursday, January 08, 2009 10:30 AM
To: McNeal, Keith
Subject:

A blank email,
From Ruth

Best to you, and I know that we have lots of good candidates for you
- Notice
This email and any files sent with it are confidential and are for the use of the addressee only. If you have received this email in error, please notify the sender immediately.

It is the responsibility of the recipient to virus scan the information provided before loading onto any computer.

This message contains views or opinions of the sender and not necessarily those of Interactive Business Systems.

You can unsubscribe from the list by going to the JSFG E-mail Mailing List page (http://www.jsfg.com/listserv.htm), entering your e-mail address, and selecting "Unsubscribe" instead of "Subscribe"

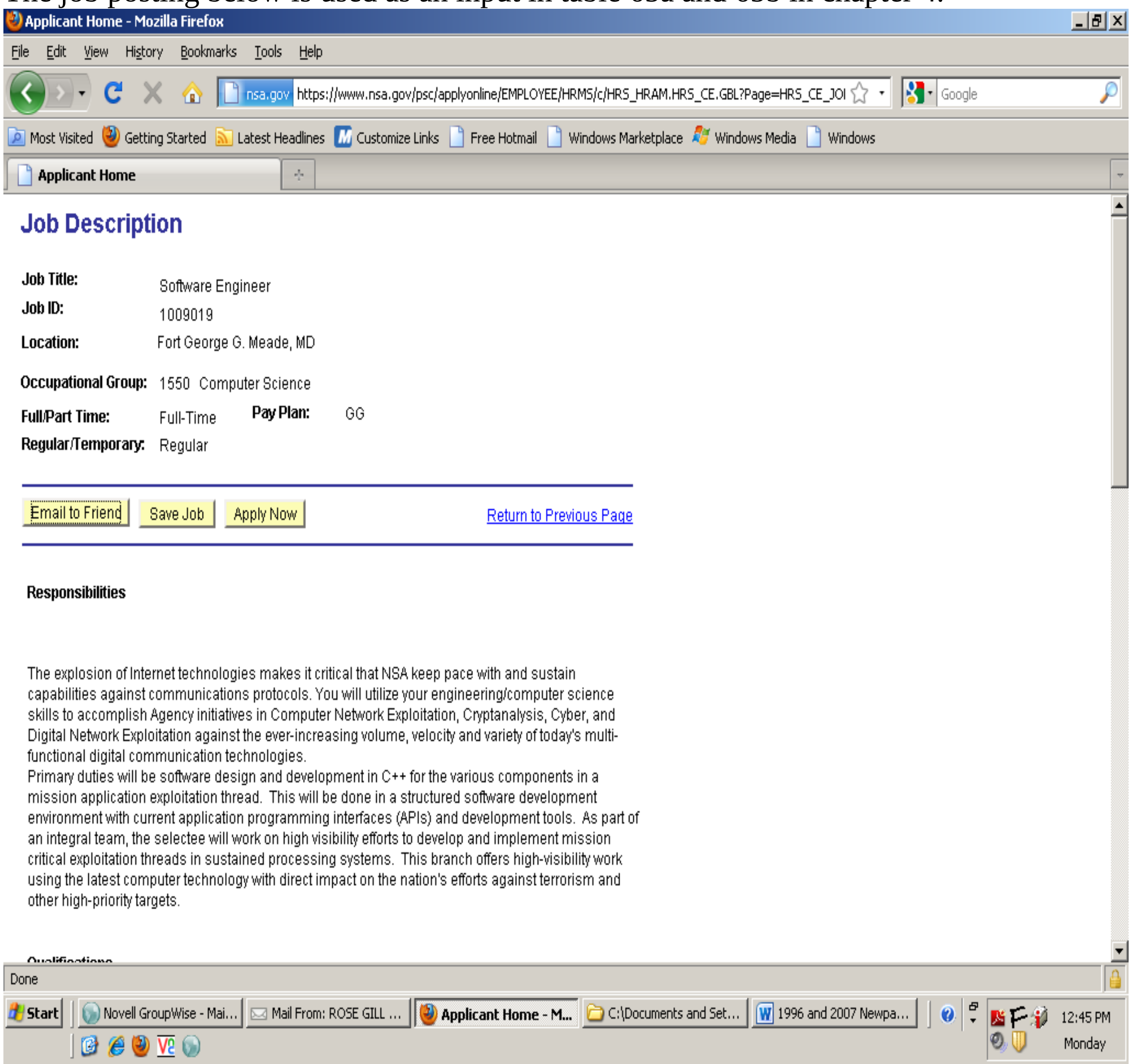
Web archive: http://www.freelists.org/archives/jsfg_cinti

Questions to: jsfg@xxxxxxxx

Appendix E14a:

JOB POSTING 2010

The job posting below is used as an input in table 65a and 65b in chapter 4.


Applicant Home - Mozilla Firefox

File Edit View History Bookmarks Tools Help

nsa.gov https://www.nsa.gov/psc/applyonline/EMPLOYEE/HRMS/c/HRS_HRAM.HRS_CE.GBL?Page=HRS_CE_JOI

Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows

Applicant Home

Job Description

Job Title:	Software Engineer
Job ID:	1009019
Location:	Fort George G. Meade, MD
Occupational Group:	1550 Computer Science
Full/Part Time:	Full-Time
Pay Plan:	GG
Regular/Temporary:	Regular

Email to Friend Save Job Apply Now Return to Previous Page

Responsibilities

The explosion of Internet technologies makes it critical that NSA keep pace with and sustain capabilities against communications protocols. You will utilize your engineering/computer science skills to accomplish Agency initiatives in Computer Network Exploitation, Cryptanalysis, Cyber, and Digital Network Exploitation against the ever-increasing volume, velocity and variety of today's multi-functional digital communication technologies.
Primary duties will be software design and development in C++ for the various components in a mission application exploitation thread. This will be done in a structured software development environment with current application programming interfaces (APIs) and development tools. As part of an integral team, the selectee will work on high visibility efforts to develop and implement mission critical exploitation threads in sustained processing systems. This branch offers high-visibility work using the latest computer technology with direct impact on the nation's efforts against terrorism and other high-priority targets.

Done

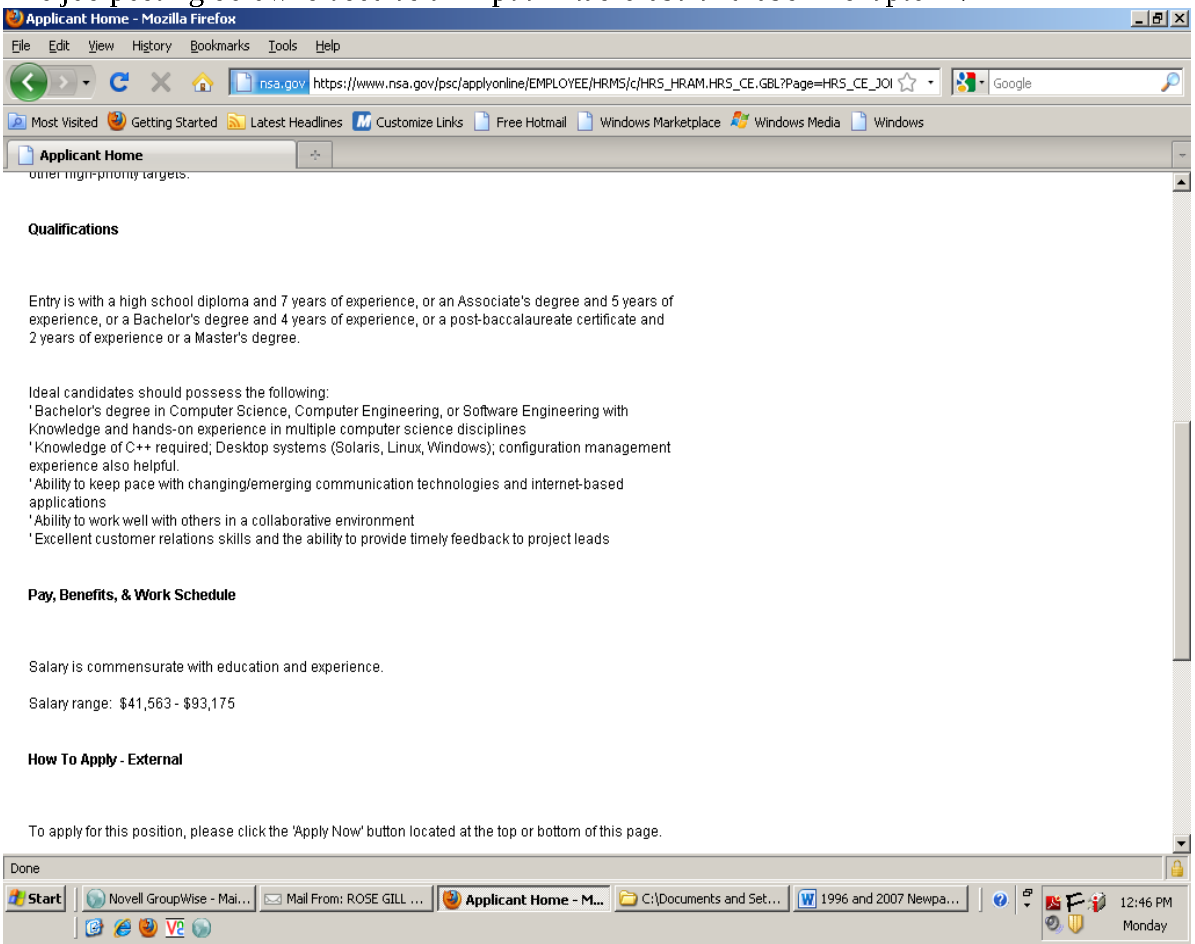
Start Novell GroupWise - Mai... Mail From: ROSE GILL ... Applicant Home - M... C:\Documents and Set... 1996 and 2007 Newpa... 12:45 PM Monday

posted on 10/27/2010
https://www.nsa.gov/psc/applyonline/EMPLOYEE/HRMS/c/HRS_HRAM.HRS_CE.GBL?Page=HRS_CE_JOB_DTL&Action=A&JobOpeningId=1009019&SiteId=1

Appendix E14b:

JOB POSTING 2010

The job posting below is used as an input in table 65a and 65b in chapter 4.



other high-priority targets.

Qualifications

Entry is with a high school diploma and 7 years of experience, or an Associate's degree and 5 years of experience, or a Bachelor's degree and 4 years of experience, or a post-baccalaureate certificate and 2 years of experience or a Master's degree.

Ideal candidates should possess the following:
' Bachelor's degree in Computer Science, Computer Engineering, or Software Engineering with Knowledge and hands-on experience in multiple computer science disciplines
' Knowledge of C++ required; Desktop systems (Solaris, Linux, Windows); configuration management experience also helpful.
' Ability to keep pace with changing/emerging communication technologies and internet-based applications
' Ability to work well with others in a collaborative environment
' Excellent customer relations skills and the ability to provide timely feedback to project leads

Pay, Benefits, & Work Schedule

Salary is commensurate with education and experience.

Salary range: $41,563 - $93,175

How To Apply - External

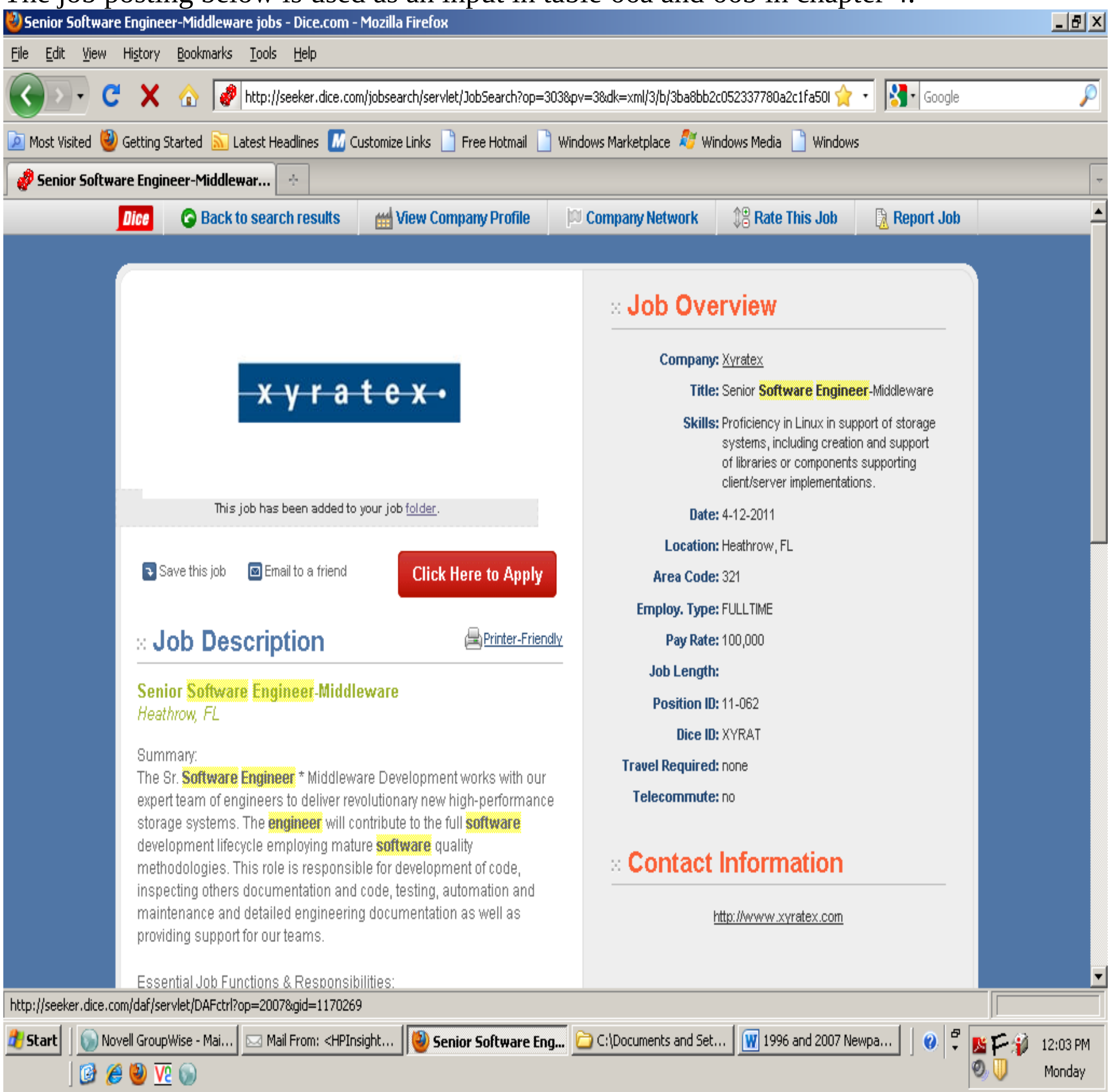
To apply for this position, please click the 'Apply Now' button located at the top or bottom of this page.

https://www.nsa.gov/psc/applyonline/EMPLOYEE/HRMS/c/HRS_HRAM.HRS_CE.GBL?Page=HRS_CE_JOB_DTL&Action=A&JobOpeningId=1009019&SiteId=1

Appendix E15a:

JOB POSTING 2011

The job posting below is used as an input in table 66a and 66b in chapter 4.


Senior Software Engineer-Middleware jobs - Dice.com - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://seeker.dice.com/jobsearch/servlet/JobSearch?op=303&pv=3&dk=xml/3/b/3ba8bb2c052337780a2c1fa50

Most Visited | Getting Started | Latest Headlines | Customize Links | Free Hotmail | Windows Marketplace | Windows Media | Windows

Senior Software Engineer-Middlewar...

Dice | Back to search results | View Company Profile | Company Network | Rate This Job | Report Job

xyratex

This job has been added to your job folder.

Save this job | Email to a friend | Click Here to Apply

Job Description

Printer-Friendly

Senior Software Engineer-Middleware
Heathrow, FL

Summary:
The Sr. Software Engineer * Middleware Development works with our expert team of engineers to deliver revolutionary new high-performance storage systems. The engineer will contribute to the full software development lifecycle employing mature software quality methodologies. This role is responsible for development of code, inspecting others documentation and code, testing, automation and maintenance and detailed engineering documentation as well as providing support for our teams.

Essential Job Functions & Responsibilities:

Job Overview

Company: Xyratex
Title: Senior Software Engineer-Middleware
Skills: Proficiency in Linux in support of storage systems, including creation and support of libraries or components supporting client/server implementations.
Date: 4-12-2011
Location: Heathrow, FL
Area Code: 321
Employ. Type: FULLTIME
Pay Rate: 100,000
Job Length:
Position ID: 11-062
Dice ID: XYRAT
Travel Required: none
Telecommute: no

Contact Information

http://www.xyratex.com

http://seeker.dice.com/daf/servlet/DAFctrl?op=2007&gid=1170269

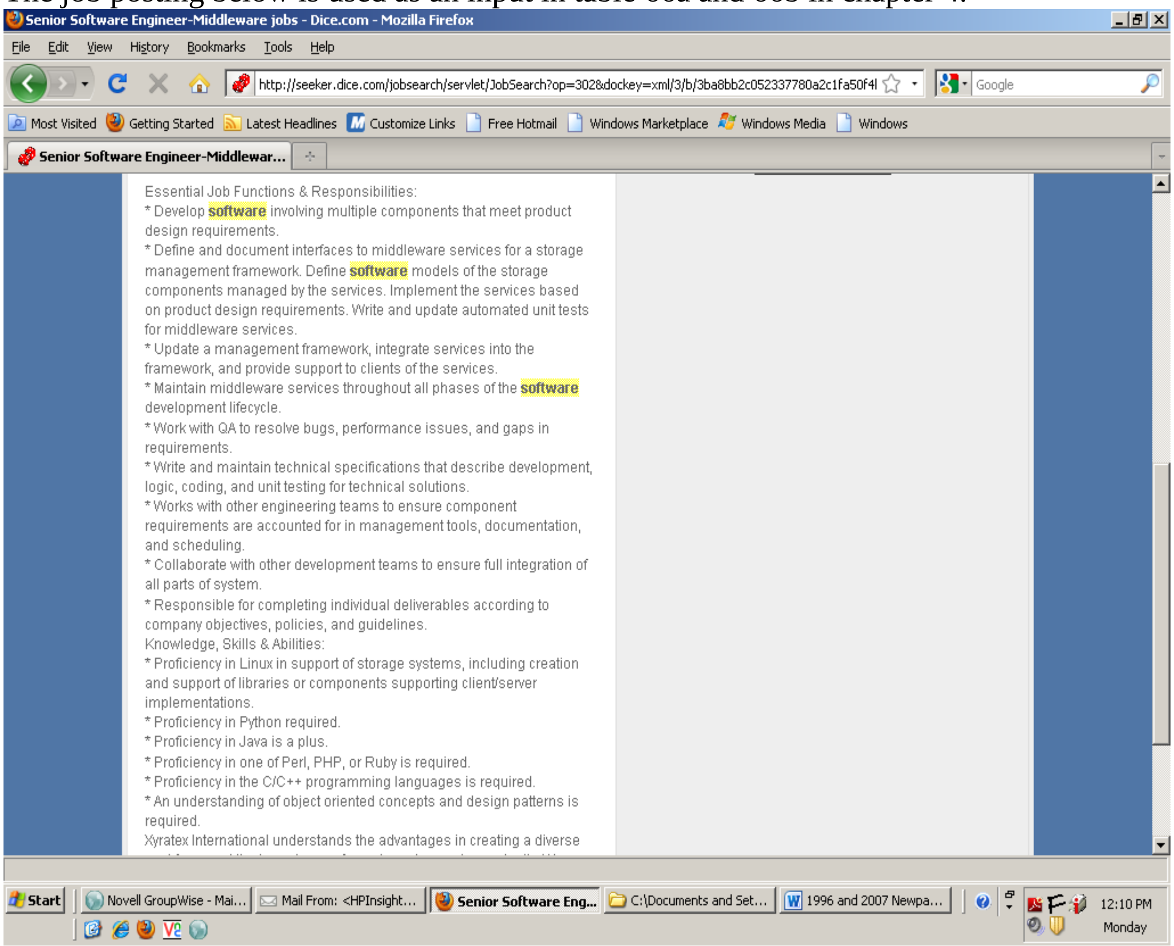
Start | Novell GroupWise - Mai... | Mail From: <HPInsight... | Senior Software Eng... | C:\Documents and Set... | 1996 and 2007 Newpa... | 12:03 PM Monday

http://seeker.dice.com/jobsearch/servlet/JobSearch?op=303&pv=3&dk=xml/3/b/3ba8bb2c052337780a2c1fa50f4bb839@endecaindex&jt=Senior+Software+Engineer-Middleware&source=19&r=2618

Appendix E15b:

JOB POSTING 2011

The job posting below is used as an input in table 66a and 66b in chapter 4.


Essential Job Functions & Responsibilities:
* Develop software involving multiple components that meet product design requirements.
* Define and document interfaces to middleware services for a storage management framework. Define software models of the storage components managed by the services. Implement the services based on product design requirements. Write and update automated unit tests for middleware services.
* Update a management framework, integrate services into the framework, and provide support to clients of the services.
* Maintain middleware services throughout all phases of the software development lifecycle.
* Work with QA to resolve bugs, performance issues, and gaps in requirements.
* Write and maintain technical specifications that describe development, logic, coding, and unit testing for technical solutions.
* Works with other engineering teams to ensure component requirements are accounted for in management tools, documentation, and scheduling.
* Collaborate with other development teams to ensure full integration of all parts of system.
* Responsible for completing individual deliverables according to company objectives, policies, and guidelines.
Knowledge, Skills & Abilities:
* Proficiency in Linux in support of storage systems, including creation and support of libraries or components supporting client/server implementations.
* Proficiency in Python required.
* Proficiency in Java is a plus.
* Proficiency in one of Perl, PHP, or Ruby is required.
* Proficiency in the C/C++ programming languages is required.
* An understanding of object oriented concepts and design patterns is required.
Xyratex International understands the advantages in creating a diverse

http://seeker.dice.com/jobsearch/servlet/JobSearch?op=303&pv=3&dk=xml/3/b/3ba8bb2c052337780a2c1fa50f4bb839@endecaindex&jt=Senior+Software+Engineer-Middleware&source=19&r=2618

REFERENCES

Allen, Thomas J. (1994). *Information technology and the corporation of the 1990s: research studies*.

Alvesson, Mats. (1987). *Organization theory and technocratic consciousness: Rationality, Ideology, and quality of work*. Berlin, Germany: Walter De Gruyter & Co.

Armbruster-Sandoval, Ralph. (1999). "Globalization and Cross-Border Labor Organization". *Latin American Perspectives, 26.* 108-128.

Badham, Richard J. (1986). *Theories of industrial society*. London, England:

Biddles Ltd, Guildford and King's Lynn,.

Bell, Daniel. (1974a). *The Coming of Post-Industrial Society*: A Venture in Social Forecasting, London, England: Heinemann,

Bell, Daniel. (1976b). *The Cultural Contradictions of Capitalism*. London, England: Heinemann

Bell, Daniel. (1980). *The Social Framework of the Information Society*: *The Microelectronics Revolution*. Oxford: Blackwell.

Bertalanffy, Ludwig. (1968). General Systems Theory. New York, NY: Ludwig Von Bertalanffy.

Bhattacher and Hirscheim. (1997). "IT and organizational change". *Lesson From Client/Server Technology Implementation.* Pp. *23.* 31.

Booch, Grady. (1994). Object-Oriented Analysis and Design. Menlo Park, CA. Addison Wesley Longman, Inc.

Braun, Jerome. (1995). *Humanized workplace: a psychological, historical, and Practical*. London, UK. Praeger.

Brown John and Duguid Paul. (2000). *The Social life of Information*. Harvard College.

Brown, Richard K. (1992). *Understanding industrial Organizations: Theoretical*

Perspectives in Industrial Sociology. New York, NY: Routledge.

Burawoy, Michael. (1979). *Manufacturing consent*: *changes in the labor process under monopoly capitalism*. Chicago, IL: The University of Chicago Press.

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1970-71a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1972-73a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1974-75a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1976-77a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1978-79a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1980-81a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1982-83a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1984-85a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1986-87a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1988-89a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1990-91a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1992-93a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1994-95a. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1970-71b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1972-73b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1974-75b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1976-77b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1978-79b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1980-81b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1982-83b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1984-85b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1986-87b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1988-89b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1990-91b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1992-93b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1994-95b. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1996-97c. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook *Handbook*, 1998-99c. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook *Handbook*, 2000-01c. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2002-03c. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2004-05c. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2006-07c. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2008-09c. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1996-97d. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 1998-99d. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2000-01d. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook *Handbook*, 2002-03d. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2004-05d. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2006-07d. Washington, DC

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2008-09d. Washington, DC

Cantor, Murray. (2002). *Software leadership*: *A guide to successful software*. *Development*.

Castells, Manual. (1999a). *End Of Millennium*. UK: Blakwell Publishers Inc.

Castells, Manuel. (1999b). *The Rise of The Network Society*. UK: Blackwell Publishers Inc.

Durkheim, Emile. (1893) translated (1984) by Coser, Lewis and Hall, W.D. *The*

Division of Labor in Society. Macmillian Publishers Ltd.

Eicher, Theo S. (2009). *Information technology and productivity growth*: *German trends and OECD comparisons*.

Eldridge, John. (1991). *Industrial sociology and economic crisis.*

Emam, Khaled El. (1999). *Elements of software process assessment and improvement*.

Engels, Friedrich. (1868). *On Marx's Capital*. Republished (1940). Moscow, Russia: Progress Publishers.

Etzioni, Amitai. (1964). Modern Organizations. Engelwood Cliffs, New Jersey: Prentice-Hall, Inc.

Fowler, Martin. (1997a). *Analysis Patterns: Reusable Object Models*. Menlo Park, CA. Addison-Wesley.

Fowler, Martin. (2003b). Pattern of Enterprise Architecture. Boston, MA. Pearson Education, Inc.

Gill, Tyson. (2002). *Planning Smarter*: Creating Blueprint Quality Software Specifications.

Ginac, Frank P. (2000). Creating high performance software development teams.

Garmus, David. (2001). *Function point analysis*: *Measurement practices for successful software projects*.

Gospel, Howard F. (1991). *Industrial training and technological innovation*: A Comparative and Historical Study.

Gouldner, Alvin. (1976). *The Dialectic of Ideology and Technology*. NY: Oxford University Press.

Hauben, Michael. (1997). *On the History and Impact of Usenet and the Internet*: Los Alamitos.

Hempell, Thomas. (2006). Computers and productivity: *How Firms Make a General Purpose Technology Work.*

Hirszowicz, Maria. (1982). *Industrial sociology*: New York, NY: St. Martin’s Press, Inc.

Harrer, Julie. (2008). *Internal Control Strategies: A Mid to Small Business Guide*.

Humphrey, Watts S. (1997a). *Managing technical people*: *Innovation, teamwork, and the software process*.

Humphrey, Watts S. (2000b). *Introduction to the team software process*.

Humphrey, Watts S. (2006c). *TSP*: *Leading a development team*.

Humphrey, Watts S. (2006d). *TSP*: *Coaching development teams*.

Jacobson, J. Rumbaugh and G. Booch. (1998). The Unified Modeling Language Reference Manual: New York, NY. Addison Wesle.

Keen, Peter G.W. (1981). "Information systems and organizational change". *Communication of the A.C.M, 24.*

Kling, Rob. (1997a). *Computerization and Controversy*. CA: Academic Press, Inc.

Kling, Rob. (1987b), *Technology and the Transformation of White-Collar Work*.

Kneuper, Ralf: (2009). *CMMI*: *Improving software and systems development processes using capability maturity model integration.*

Kolawa, Adam. (2009). *Next Leap in Productivity*: *What Top Managers Really Need to know about Information Technology*.

Li, Eldon. (1997). "Perceived Importants of Information System Success Factor". *A Meta-Analysis of Group Difference*, 32. 15-28.

Lopp, Michael. (2007). *Managing humans*: *Biting and humorous Tales of a Software Engineering Manager*.

Maathai, Wangari. (1999). "Let Nature's Harvest Continue". *Review of African Polictical Economy, 25.* pp. 529-531.

Martin, James and McClure, Carma. (1988). *Structured Techniques: The Basis for Case*. Englewood Cliffs, New Jersey. Prentice Hall.

Marx, Karl. (1867a). Capital: *A Critique of Political Economy* Volume Three. Translated (1978) by Fernback, David Cambridge, MA: Cambridge University Press

Marx, Karl. (1867b). *Capital: A Critique of Political Economy*. Volume 1: *The*

Process of Produciton of Capital. Trans. (1887), Edited by Frederick (1890). Republished (1954) by Samuel Moore and Edward Aveling. Moscow, Russia: Progress Publishers

Marx, Karl. (1849) Translated and Republished (1884) by Engels, Frederick and Republished (1933). *Wage-labour and capital*. New York, NY: International Publishers Co., INC.

Marx, Karl. (1858). *Pre-capitalist economic formations*. Translated (1965) by Jack Cohen. New York, NY: International Publishers

Masuda, Yoneji. (1981). *The Information Society: As Post-Industrial Society*. Tokyo Japan: Institute of the Information Society.

Mathiassen, Lars. (2002). Improving Software Organizations: From principles to Practice.

McConnell, Steve (1996a). Rapid Development: Taming Wild Software Schedules. Microsoft Press

McConnell, Steve (2004b). *Code Complete, 2nd edition*. Microsoft Press.

McConnell, Steve (2006c). *Software Estimation: Demystifying the Black Art*. Microsoft Press.

Morrison, Ken. (1995). *Marx, Durkheim and Weber: Formations of Modern Social Thought*. UK: Sage Publication, Inc.

M. Paulk, C. Weber and B. Curtis. (1994a). *The Capability Maturity Model: Guidelines for Improving the Software Process*: Carnegie Mellon Univ. Software Engineering Inst.

M. Paulk, R. Hunter and R. Thayer. (2001b). *Software Process Improvement*, La Alamitos, CA. The Computer Society.

Prais, S. J. (1995). *Productivity, Education and Training*: An International.

Reynolds, George and Sair, Ralph. (1998). *Information Systems: A Managerial Approach*. Cambridge, MA. International Thomson Publishing Company.

Ritzer, George. (2000). “Exchange, Network and Rational Choice Theories.” *Modern Sociological Theory*. USA: McGraw Hill.

Ritzer, George. (2000). “George Simmel.” *Classical Sociological Theory*. USA: McGraw Hill.

Ritzer, George. (2000). “Varieties of Neo-Marxian Theory.” *Modern Sociological Theory*. USA: McGraw Hill.

Rob, Peter and Coronel, Carlos. (1997). *Database Systems: Design, Implementation and Management*. Cambridge, MA. International Thomson Publishing Company.

Robbins, P. Stephen, (2007), *Organizational Behavior*. New Jersey, Pearson/Prinston Hall

Robertson, James and Robertson, Suzanne. (1999). *Mastering the Requirements Process*. New York, NY. Addison-Wesley.

Roger Pressman. (2001). Software Engineering. London, England. McGraw-Hill.

Roehl-Anderson, M. Jaince (2004), *Controllership*. USA: John Wiley & Sons Inc.

Royce, Winston (1970), "*Managing the Development of Large Software Systems*", *Proceedings of IEEE WESCON* 26 (August): 1–9

Saint-Simon, Henri, comte de. (1824). *Social organization, the Science of Man and Other Writings [by] Henri de Saint-Simon*. Translated by Felix Markham.

Saint-Simon, Henri, comte de. (1822) Trans. (1958) by Georg G. Iggers. *Doctrine of Saint-Simon: An Exposition First Year*.

Saint-Simon, Henri, comte de. (1817). Trans. (1956) by Emanuel, Frank. The New World of Henri Saint-Simon. Cambridge, MA. Harvarad University Press.

Saint-Simon, Henri, comte de. (1821). Trans. (1976) by Ghita Ionescu. *Political thought of Saint-Simon*. London, UK. Oxford University Press.

Sanderson, Stephen. (1995). *Social Transformations: A General Theory of Historical Development*. Massachusetts: Blackwell Publishers, Inc.

Schermerhorn, J., Hunt, J., & Osborn, R. (1997a). *Organizational Behavior*. NY: John Wiley and Son, Inc.

Schermerhorn, J., Hunt, J., & Osborn, R. (2008b). *Organizational Behavior*.

NY: John Wiley and Son, Inc.

Smith, Adam. (1776a) trans. (1957) by George J. Stigler. *Selections from The Wealth of Nations*. Arlington Heights, IL. AHM Publishing Corporation.

Smith, Adam. (1776b) trans. (1997) by Skinner, Andrew. *Wealth of nations. Books I-III*. London, UK. Penguin Books

So, Alvin. (1990). "The Modernization School." *Social Change and Development: Modernization, Dependency, and World-System Theories.* CA: Sage Publication.

Stellman, Andrew. (2009). *Beautiful teams: Inspiring and Cautionary Tales from Veteran Team Leaders*.

Stinchcombe, Arthur L. (1990). Information and organizations. CA. University of California Press.

Trauth, Eileen. (1999). "Leapfrogging and IT Labor Force". *Journal of Global Information Management, 7*. 22-32.

Turner, Jonathan. (1998). *The Emergence of Sociological Theory*. CA: Wadsworth Publishing Company.

Wallace, Michael. (1998). *Global Inequalities*. CA: Pine Forge Press.

Wallerstein, ImmanueL. (2000). *The Essential Wallerstein*. NY: New Press.

Wardell, Mark. (1999). *Rethinking the labor Process*. New York, NY: State University of New York, Albany.

Weatherby N. Joseph. (1999). *The Other World*. NY: Longman Inc.

Weber, Max. (1909). *Essential Weber*. Reedited (2004) by Sam Whimster. London, England: Routledge Taylor & Francis Group

Weber, Max. (1904-5a). *On Capitalism, Bureaucracy and Religion*. Trans. (1983) by Stanislav Andreski. London, England: George Allen & Unwin (Publishers) Ltd.

Weber, Max. (1904-5b), *The Iron Cage: A Historical Interpretation*. Translated by Arthur Mitzman (1985).

Weber, Max. *Closing the Iron Cage: The Scientific Management of Work & Leisure,* trans. by Ed Andrew (1999)

Williams, Steve. (2007). *Profit impact of business intelligence*.

Wright, Erik Olin. (1997). "The Class Structure of Capitalism and its Transformations." *Class Counts: Comparative Studies in Class Analysis. NY: Cambridge University Press.*

Young, Lawrence. (1989). Decision Support and Idea Processing Systems. Dubuque, IA. Wm. C. Brown Publishers.

Yourdon, Edward. (1996a). *Rise and Resurrection of the American Programmer*. NJ: Prentice-Hall, Inc.

Yourdon, Edward. (1979b). *Classics In Structural Design*. New Jersey: NJ: Princton Hall-Inc.

Yourdon, Edward. (1986c), *Managing the Structured Development Techniques*. New Jersey: NJ: Princton Hall-Inc.

Zeitlin, Irvin. (1997). *Ideology and the Development of Sociological Theory*. New Jersey, NJ: Prentice Hall.