UNIVERSITY OF CALGARY

Supplementing the Capability Maturity Model with the Personal Software Process

by

John Frankovich

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

JUNE, 1999

0-612-49613-9

Abstract

The Capability Maturity Model (CMM) (Paulk, 1993) is a process maturity framework designed to improve an organization's process capability. The Personal Software Process (PSP) (Humphrey, 1995a) is a technique to improve an individual engineer's performance and productivity. The underlying philosophy is that the CMM is a top-down approach to improving an organization's ability to engineer software. The PSP is a bottom-up approach that enables engineers to improve the quality of their software. The combination of these macro and micro methodologies can form the framework for an organization's standard software process.

This thesis explores the impact of supplementing the Capability Maturity Model with the Personal Software Process. The effects of combining these micro and macro methodologies in a real-world environment are examined and its impact on Schedule and Effort Predictability is elucidated.

Acknowledgements

The difference between a teacher and an educator is that the teacher simply lectures to the student, where the educator provides the environment and lets the student figure it out on their own. I would like to thank two educators that I have had the pleasure of working with at the University of Calgary: Don Bidulock and Mildred Shaw. The foundation for this thesis was laid during my undergraduate years when Don Bidulock allowed me to explore my abilities, at my pace, under his guidance. This was an important time for me and has not been forgotten. Later during the development of this thesis, many difficulties were experienced that could have easily derailed the progress of the research. It was Mildred Shaw's guidance, experience and determination that kept the thesis on track. Mildred's dedication to each student's intellectual discovery process provided the environment I required for success.

I would like to thank my mother for her unconditional support in all my endeavors (past, present, and future). I would like to thank my father for being the prime example in my life of integrity, stability, and dedication to family. Finally, I would like to thank my wife and son for all their sacrifices endured over the years. Without your support, Lina, I would never have been able to accomplish my dreams.

Table of Contents

List of Tables

List of Figures

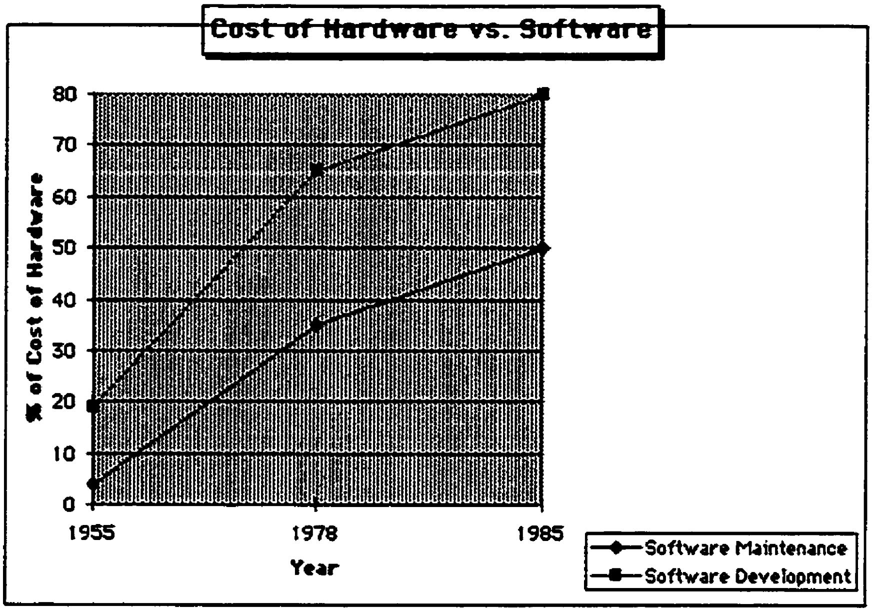
Chapter 1: Aim of Research

1.0 Introduction

The world has seen many revolutions: political revolutions, the Industrial Revolution, and the Computer Revolution. The fundamental characteristic of a revolution is the transformation of commonly held beliefs and values to a new set of more desirable beliefs and values. Currently, the computing world is in the middle of another revolution, one that will shift our focus from developing software, and move it toward the concept of engineering software.

The origins of the paradigm shift towards engineering software go back decades, to the early years of the computer revolution. In this pre-Microsoft era, projects were becoming ever increasingly complex, and their success was largely determined by hardware factors. The intrinsic complexities and cost of hardware primarily determined the success or failure of the projects (Boehm, 1981). Software was seen as an afterthought, the final packaging that allow people to use the developed system to perform the functions that it was designed to accomplish (Pressman, 1992). In this environment, organizations focused their efforts on controlling the process of developing hardware. Standards, procedures, and methodologies were applied to the act of developing hardware, with the hope that this type of disciplined approach would finally allow them to gain control over the factors that in fact controlled them. Since software was not seen as an important component of a finished product, it did not receive that same attention and thus, did not improve as quickly. Figure 1 depicts the cost of developing and maintaining software as a percentage of the cost of developing the hardware on engineering projects between 1955 and 1985.

Figure 1: Cost of Hardware vs. Software (Industrial Data)



In the early 1970's, organizations gained more control over the process of developing hardware. This allowed them to push forward with bigger and better ideas that would propel the industry forward for decades to come. As hardware platforms became more sophisticated, the need for sophisticated software that would control them developed. The approach that organizations took to develop the required software was the first real hint of the encroaching software engineering revolution.

It has been documented that out of all software development projects attempted: 16% were categorized as successful, 53% were deemed operational but less that successful, and 31% were never completed (Standish Group Industrial Data). It is difficult to precisely determine the lost revenue that organizations have faced, but the figure is unarguably astonishing.

The Capability Maturity Model (CMM) (Paulk, 1993) is a process maturity framework designed to improve an organization's process capability. The Personal Software Process (PSP) (Humphrey, 1995a) is a technique to improve an individual engineer's performance and productivity. The underlying philosophy is that the CMM is a top-down approach to improving an organization's ability to engineer software. The PSP is a bottom-up approach that enables engineers to improve the quality of their software. The combination of these macro and micro methodologies can form the framework for an organization's standard software process.

1.1 Aim of Research

The aim of this research is to investigate the impact on effort and schedule predictability in real world projects that results from supplementing the CMM with the PSP.

1.2 Objectives of Research

The purpose of this thesis is to conduct research on the impact of supplementing the CMM with the PSP. Real-world data metrics will be used to determine if there is a statistically significant impact on schedule and effort predictability as a results of combining these two software engineering methodologies. The objectives of this research are as follows:

1. To review literature that relates to the CMM and PSP and to determine the current state of the art relating to the aim of the thesis.
2. To critically analyze the current state of the art.
3. To describe the organizational environment where the research was conducted.
4. To document what data metrics were collected, how they were collected, and the project characteristics that generated the metrics.

5. To determine the statistical significance of the data presented.
6. To draw conclusions

1.3 Motivation underlying the Aim

In recent years there has been a great deal of interest in software engineering processes that claim to improve the quality of the developmental process and hence, the quality of the software product. The majority of these methodologies are targeted towards developing and sustaining an organizational software process model. Publications, research papers, and real-world data is now becoming available on these organization-wide models.

The Personal Software Process was designed and targeted towards the individual software engineer (Humphrey, 1995a). Initial Data on the PSP's impact on the engineer's performance looks promising. The underlying philosophy behind the PSP is that improving the quality and performance of individual engineers working on a software project will increase the quality and performance of the entire project. Currently, there seems to be little or no data to support this assumption.

Thus, the motivation underlying the aim of this thesis is to investigate the correlation between the engineering-level software processes and the project / organizational-level processes.

1.4 Industrial Experiments

There is a difference between an experiment that is performed in a controlled environment and one that is carried out in an industrial setting. In a classical scientific experiment, the environment is carefully controlled and measurements are collected in order to discover cause and effect relationships between independent and dependent variables. In industrial experiments in which the environment is not under the experimenters control, statistical inference methods can be used to determine the degree to which two variables are correlated.

It is very important to note that any correlation derived from industrial experiments utilizing statistical inference methods should not be used to imply cause and effect relationships. Thus, the results drawn from this thesis should be viewed as a statistical description of the industrial data collected in that environment.

1.5 Summary

This introductory chapter defines the Aim of Research that this thesis addresses. The Objectives of Research section breaks the Aim down into a set of objectives that are individually addressed by subsequent chapters. A short description concerning the motivation behind the research is presented along with a warning that the data set utilized was collected in an industrial environment and any conclusions drawn must be reviewed with care.

Chapter 2: Survey CMM and PSP

2.0 Introduction

The goal of this chapter is to review all literature that relates to the CMM and PSP and to determine the current state of the art relating to the aim of the thesis. Thus, this chapter starts with a description of the process architecture of the CMM. This is followed by a description of the PSP internal structure. The chapter concludes with a description of how PSP engineers approach planning, designing, coding, and testing individual modules.

2.1 The Capability Maturity Model

After two decades of research into productivity and quality gains from applying new software methodologies and technologies, industrial and governmental organizations are realizing the benefits that can be gained from managing their software process (Defense Science Board, 1987). In 1987, the Software Engineering Institute (SEI) designed an organizational assessment model that later developed into the Capability Maturity Model (CMM).

The CMM is a framework that characterizes an evolutionary process improvement path toward a more mature organization. A mature software organization possesses an organization wide ability to manage software development and maintenance processes (SEI, 1995). An organization can use the CMM to determine their current state of software process maturity and then to establish priorities for improvement. An organization's current state of maturity can be categorized as Initial, Repeatable, Defined, Managed, or Optimizing:

Initial: The software process is characterized as ad hoc and occasionally even chaotic. Few process are defined and success depends on individual effort.

Repeatable: Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier success on projects with similar applications.

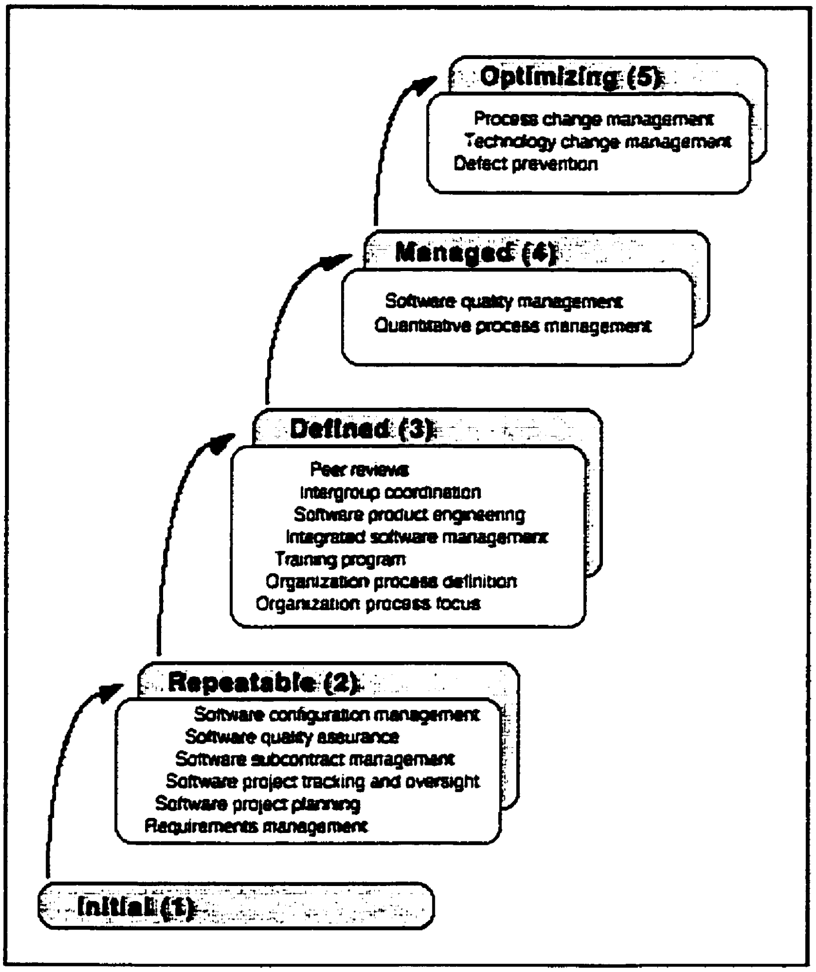
Defined: The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard process for developing and maintaining software.

Managed: Detailed measures of software process and product quality are collected. Both software process and products are quantitatively understood and controlled.

Optimizing: Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

The CMM defines five maturity levels and each level is composed of several key process areas. Figure 2 graphically shows the maturity levels along with the corresponding key process areas.

Figure 2: Maturity Levels of the CMM (SEI, 1995)



Each key process area identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capabilities (SEI, 1995). The following table describes each KPA's purpose as defined in the CMM:

Table 1: Key Process Area's Purpose

Maturity Level	Key Process Area	Purpose

2	Requirements Management	The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements to be addressed.
2	Software Project Tracking	The purpose of Software Project Tracking is to establish reasonable plans for performing the software engineering and for managing the software product.
2	Software Tracking and Oversight	The purpose of Software Tracking and Oversight is to provide adequate visibility into actual progress so that the managers can take effective actions when the software project's performance deviates significantly from the software plan.
2	Software Subcontract Management	The purpose of Software Subcontract Management is to select qualified software subcontractors and manage them effectively.
2	Software Quality Assurance	The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and the products being built.

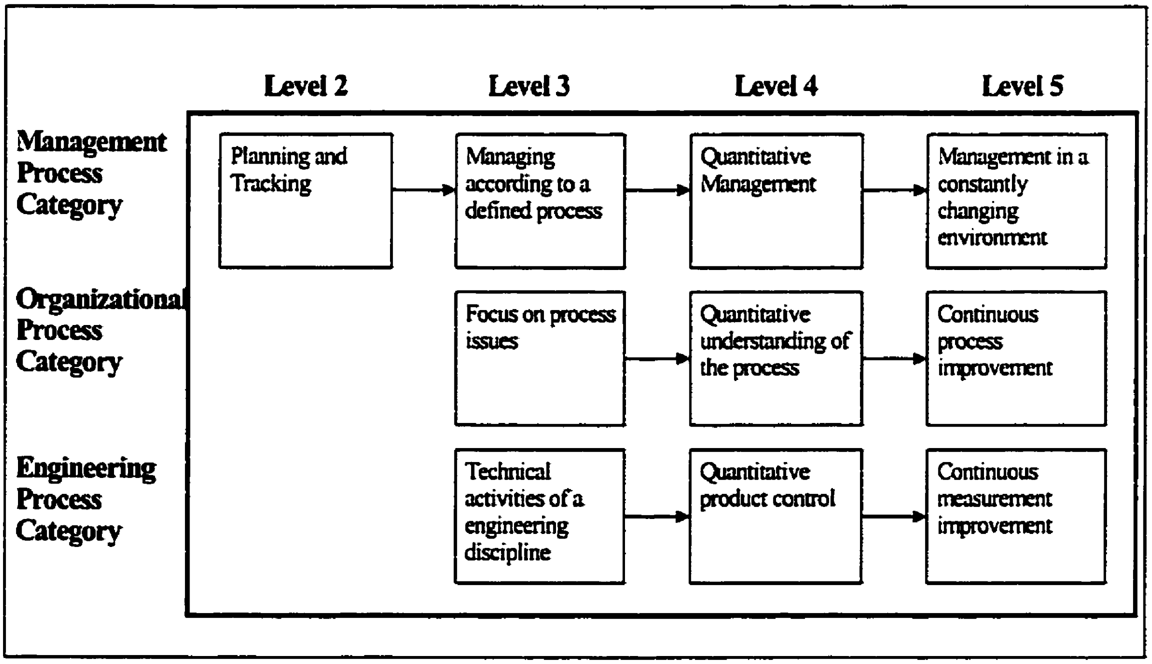
2	Software Configuration Management	The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle.
3	Organization Process Focus	The purpose of Organization Process Focus is to establish the organizational responsibilities for software process activities that improve the organization's overall software process capability.
3	Organization Process Definition	The purpose of Organization Process Definition is to develop and maintain a usable set of software process assets that improve process performance across the projects and provide the basis for cumulative, long-term benefits to the organization.
3	Training Program	The purpose of Training Program is to develop the skills and knowledge of individuals so they can perform their roles effectively and efficiently.
3	Integrated Software Management	The purpose of Integrated Software Management is to integrate the software engineering and management activities

		into a coherent, defined software process that is tailored from the organization's standard software process and related process assets, which are described in the Organization Process Definition.
3	Software Product Engineering	The purpose of Software Product Engineering is to consistently perform a well defined engineering process that integrates all the software engineering activities to produce correct, consistent software products effectively and efficiently.
3	Intergroup Coordination	The purpose of Intergroup Coordination is to establish a means for the software engineering group to participate actively with the other engineering groups so the project is better able to satisfy the customer's needs effectively and efficiently.
3	Peer Reviews	The purpose of Peer Reviews is to remove defects from the software work products early and efficiently.
4	Quantitative Process Management	The purpose of Quantitative Process Management is to control the process performance of the software project

		quantitatively.
4	of Software Quality Management	The purpose of Software Quality Management is to develop a quantitative understanding of the quality of the project's software products and achieve specific quality goals.
5	Defect Prevention	The purpose of Defect Prevention is to identify the cause of defects and prevent them from recurring.
5	Technology Change Management	The purpose of Technology Change Management is to identify beneficial new technologies (i.e., tools, methods, and processes) and transfer them to the organization in an orderly manner.
5	Process Change Management	The purpose of Process Change Management is to improve continually the software processes used in the organization with the intent of improving quality, increasing productivity, and decreasing the cycle time for product development.

Table 1 describes the Key Process Areas at each maturity level. Alternatively, the KPA's in the CMM can be broken down into 3 different process categories: the Organizational process category which contains the cross project responsibilities; the Management process category which contains project management activities; and the Engineering process category which contains technical activities. Figure 3 shows the critical issues that each process category addresses as an organization matures from one level to the next.

Figure 3: CMM Evolution of Process



Each process category describes the software processes in an organization from a different vantage point: the engineer's view, the project's view, and the organization's view. The issues encountered in attempting to implement and institutionalize certain key practices will be different depending on its process category. Certain organizational roles and groups may have considerable impact on the software processes in one process category and not in another. An example of such a group is the Software Configuration

Management Group. This group is responsible for planning, coordinating, and implementing formal configuration management activities for the software project. It is clear that the software processes in the management process category will be heavily influenced by this group but other process categories may not be impacted as directly.

The division of software processes into Management, Organization, and Engineering categories allows a unique perspective into the software processes as they mature from one level to the next.

2.2 The Personal Software Process

The CMM represents a substantial gain in the understanding of software engineering processes and lays the ground work to help organizations develop their own processes that will continually improve over time. Its objectives, goals, and activities are designed to institutionalize organizational structure and policies. However, it does not address what tools, techniques, and methodologies an engineer should use in order to implement and ultimately satisfy the policies. The CMM does lay the ground work for personal methodologies by providing an organizational structure but, it does not directly address the requirements of the individuals. This "Personal Gap" (Frankovich, 1997) between the CMM and individual software engineers will continue to exist until organizations can:

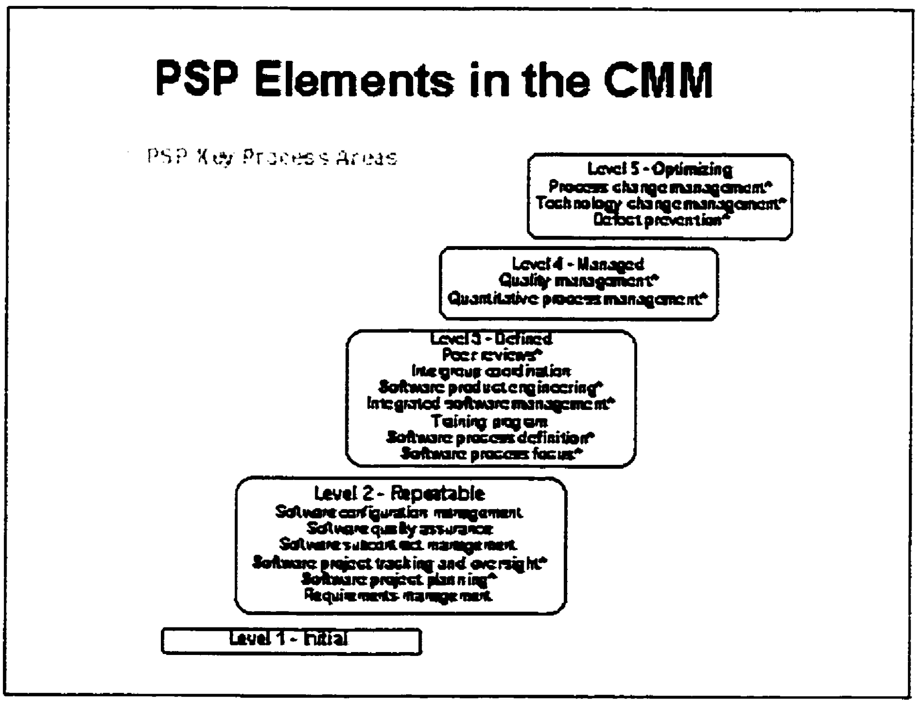
- Identify Key Process Areas that can be performed by the individual
- Define a subset of the CMM that will be useful to a small development team

The Software Engineering Institute has developed a Personal Software Process that addresses the gap that exists between the Capability Maturity Model and the individual. The PSP is a bottom-up approach to software engineering which focuses on the individual software engineers, their personal practices, and how they relate to an organization-wide software process improvement plan.

> The PSP and CMM are mutually supportive. The CMM provides the orderly support environment engineers need to do superior work, and the PSP equips engineers to do high quality work and participate in organization process improvement (Iyer, 1996).

The PSP has a process evolution framework similar to the CMM which partially addresses 12 of the 18 KPAs defined in the CMM. The following figure shows the CMM along with its Key Process Areas, and those areas that are partially addressed in the PSP have been noted with an asterisk.

Figure 4: PSP Elements in the CMM (Humphrey, 1995a)

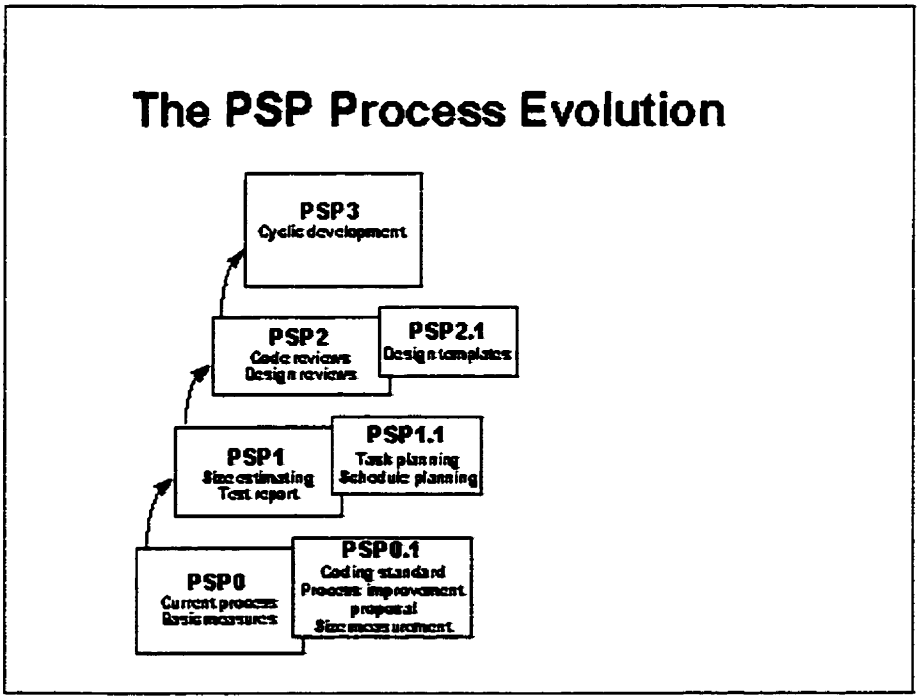


In order to develop high quality software, each individual component that is integrated into the overall product must also be of high quality. The overall strategy of the PSP is to make sure all the individual components are of high quality. The PSP accomplishes this by providing a defined personal process framework that the software engineer can use to:

- Develop a plan for every project / component
- Record their development time
- Track their defects
- Retain their data in project summary reports
- Use their data to plan future projects / components
- Analyze their data to evolve their processes
- Improve their performance

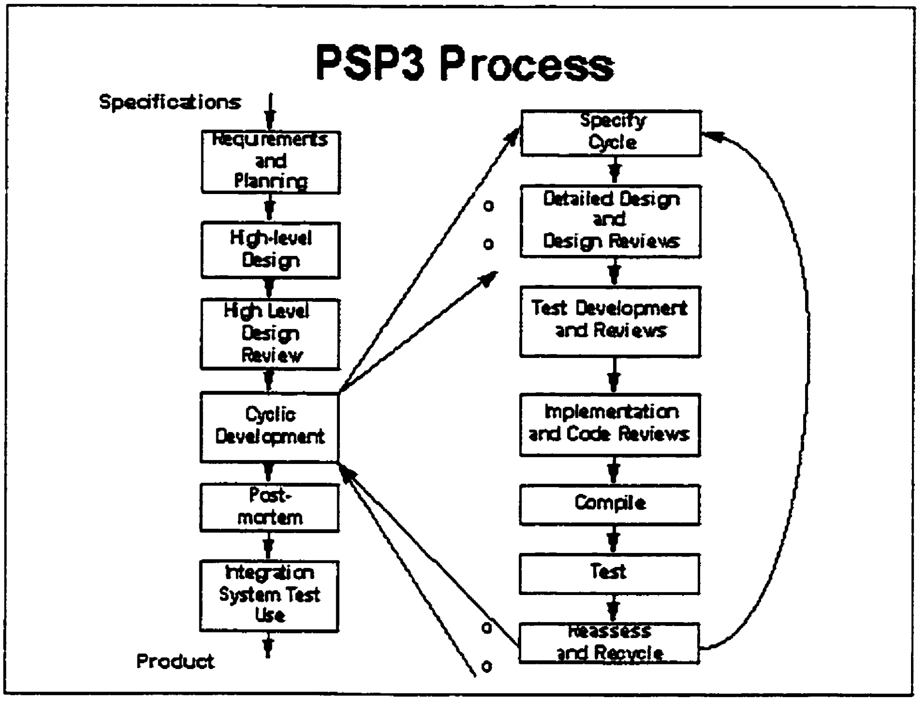
The PSP is comprised of many personal quality processes that are divided into seven sequential categories of progressing sophistication. Figure 5 depicts the seven sequential categories of the PSP.

Figure 5: The PSP Process Evolution (Humphrey, 1995a)



The fundamental objective of the initial stage of the PSP is to document the engineer's personal process as it exist today. This baseline will provide a consistent basis for measuring progress. Additionally, it provides a defined structure upon which to improve. The only modification that the engineer does to his own process at this level is to record measures of performance. The PSP1 stage extends the initial stage by adding software planning activities. These activities are geared toward elucidating the relationship between the size of programs and the amount of time that is needed to develop them. This is accomplished by developing a plan and estimating the resources required. An early PSP objective is to help engineers to deal realistically and objectively with the defects they inject. These defects are usually syntax or simple semantic errors that even the most seasoned software engineer will commit. The amount of time that is required to track down and fix these errors can become extreme in very large and complex projects. The PSP2 stage, the Personal Quality Management Process, addresses the critical issue of defect management. The defect data collection starts in initial stage and the engineers now use it to construct checklists for design and code review. The purpose of the final stage is to scale up the PSP for use on large scale projects. One of the fundamental problem solving approaches in science is Divide and Conquer, taking a complex problem and subdividing it into smaller more manageable pieces. This is the fundamental concept that is behind PSP3. The following figure illustrates this process.

Figure 6: PSP3 Process (Humphrey, 1995a)



The strength of this approach is that each sub-module is designed as a full PSP2 project. Fundamentally, the PSP was designed to aid in the development of high quality software. If each module is designed to meet this requirement, then integration of each individual component into the overall product will also satisfy the requirement.

2.2.1 PSP Planning

The PSP engineer is responsible for performing all project management activities that are required to perform Size Estimating , Resources Estimating, Task Planning, Schedule Planning, and to develop a Software Project Plan.

The PSP engineer uses linear regression techniques to help generate Size Estimates. For each new planned object, the PSP engineer estimates the object type and number of methods the object will likely contain. Then historical data is used to project the estimated LOC required to implement all the new objects. This projected LOC is then used by the PROBE method to calculate the size estimate for the project. Finally a 70 percent prediction interval is calculated for the size estimate. The Resource Estimates are calculated in a similar manar. The PROBE method is used to estimate the time required to develop the software modules, and the 70 percent prediction interval is also calculated.

The PSP engineer performs Task Planning and Schedule Planning in parallel. The Schedule Planning process requires a detailed estimate of the total project hours. Then the number of weeks required to work the estimated hours is determined with the help of historical utilization data. Task planning requires the development time estimates for each project task which were derived from the Resource Estimates. The task time estimates are then applied against the schedule to produce individual task completion dates and a final project completion date. A task baselined Earned Value distribution is generated and is used to track the progress of the project's tasks.

The PSP engineer documents all aspects of the Size Estimates, Resource Estimates, Task Planning, and Schedule Planning in a Software Project Plan. The plan also includes estimated data on productivity, test defect yield, cost of quality indexes, and defect removal efficiency. Finally, the plan specifies the data to be gathered in the Postmortem phase at the end of the project.

2.2.2 PSP Designing

In generating a High Level Design the PSP engineer produces a design and implementation strategy. This strategy includes Functional, State, and Logic Specifications, and Operational Scenarios with Test Strategies. The Functional Specification precisely describes the methods provided by the objects in the design. The State Specification describes each of the object's states and transitions among them. The logic Specification describes the pseudo-code logic for each object in the design. The Operational Scenarios describes the software system's procedural behavior in one or more scenarios. The Operational Scenarios are then used to help specify test scenarios. In addition to producing the design and implementation strategy, the PSP engineer is also responsible for conducting an effective design review.

2.2.3 PSP Coding & Test

The development tasks that the PSP engineer is responsible for start with the Module Design. For each module specified in the High Level Design, the PSP engineer constructs a Module Design and performs a Design Review on it. Next the engineer follows a coding standard to implement the Module Design followed by a code review. After the reviews, the modules are compiled and tested until they run without error. These development tasks are performed for all modules specified in the High Level Design and are integrated into a coherent system.

2.3 Summary

The CMM is a process maturity framework designed to improve an organization's process capability. The PSP is a personal methodology that empowers engineers to apply quality practices to their daily work.

The CMM lays the ground work for personal methodologies, but it does not directly address the requirements of the individuals. One of the fundamental difficulties in designing a Software Process Improvement initiative using the CMM, is that the model describes what to build but not how to build it. If we think of the CMM as a blueprint for a house it would tell the construction worker were the load bearing walls, support beams, and trusses should be placed to construct a stable house but would not tell the carpenters how to build the load bearing walls nor what constitutes a state of the art truss.

Chapter 3: Critical Analysis of Literature review

3.0 Introduction

The goal of this chapter is to critically analyze the published results of the CMM and PSP in order to determine the current state of the art relating to the aim of the thesis. Thus, this chapter starts by reviewing the results from the Hayes / Over study on the impact of the PSP on individual engineers in an educational setting. This is contrasted with a case study on the effects of the PSP on an industrial project. Both the positive and negative aspects of the PSP results are analyzed. Next, the available industrial data on CMM-based software process improvement programs is reviewed. This is also contrasted with an industrial case study utilizing data form Hughes Aircraft. Once again, both the positive and negative aspects of the CMM results are analyzed. This is followed by a brief description of the available published data concerning supplementing the CMM with the PSP. Finally, the results from this chapter are summarized.

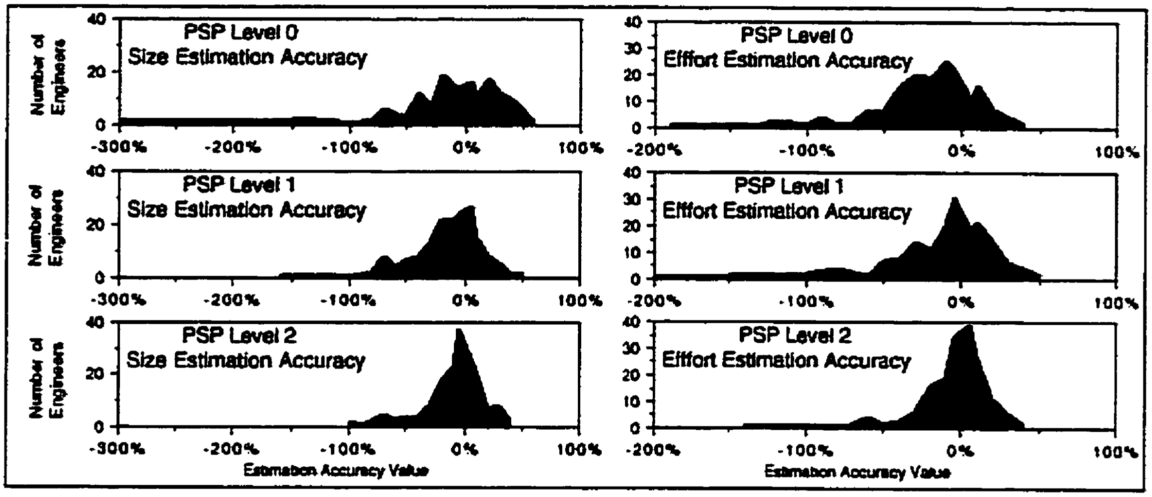
3.1 Results of the PSP

3.1.1 Class Based Empirical Study

In December 1997, Will Hayes and James W. Over of the Software Engineering Institute published a landmark report that is considered to be the first large scale statistically significant study on the impact of the PSP on individual engineers (Hayes, 1997). The data set was collected from 298 individual engineers upon the completion of 23 separate PSP training courses. The authors of the report estimate that over 300,000 lines of code we developed requiring more than 15,000 man hours.

From this data set, the study choose to analyzed five data metrics that are considered to be the foundation for personal improvement: size estimation, effort estimation, defect density, process quality, and productivity. Two of the most interesting findings involved size and effort estimation trends which are reproduced in Figure 7.

Figure 7: Empirical Data on PSP Impact(Hayes, 1997)



The data plots in Figure 7 display the estimation trends that individual engineers experienced as they progressed through and ultimately completed the PSP course. In the initial level of the PSP, both the size and effort estimation are characterized by long run off tails which indicate that the engineers were underestimating, sometimes by a factor of 300 percent. This trend is much different towards the end of the training course. The engineer's estimates are more clustered around the zero percent estimation mark and the underestimating run off tails have been shortened considerably. Additionally, the size estimation trend is approaching the point where there are a similar number of overestimation and underestimation errors. Table 2 summarizes the major findings of the study.

Table 2: Empirical Data Summary

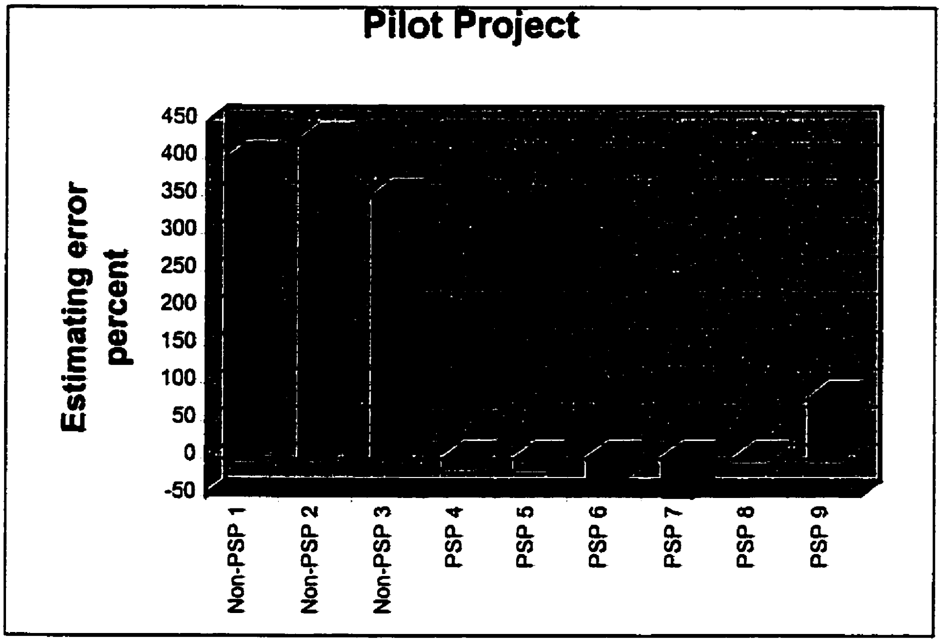
Data Metric	Results
Effort Estimation	Improved by a factor of 1.75
Size Estimation	Improved by a factor of 2.5
Estimation Error	Number of underestimation and overestimation errors were more balanced.
Defect Density	Defects found at unit test improved by a factor of 2.5
Process Quality	Defects found before compile improved by 50 percent.
Productivity	Lines of Code per hour did not change significantly.

3.1.2 Industrial Case Study

In contrast to the Hayes / Over empirical study which focused on classroom data, there is industrial PSP data available from an organization that has been applying the PSP consistently for over three years.

In June 1995, Advanced Information Services Inc. (AIS) of Peoria, Illinois, piloted the PSP on a project and later incorporated the PSP into their organizational standard software process. The project was divided up into nine different components each ranging in size from 500 to 2200 lines of code. The first three components were completed with non-PSP trained engineers who were having difficulty meeting their internal target dates. The remaining six components were re-planned and developed by PSP trained engineers. Figure 8 shows the estimating error percent for each of the nine components.

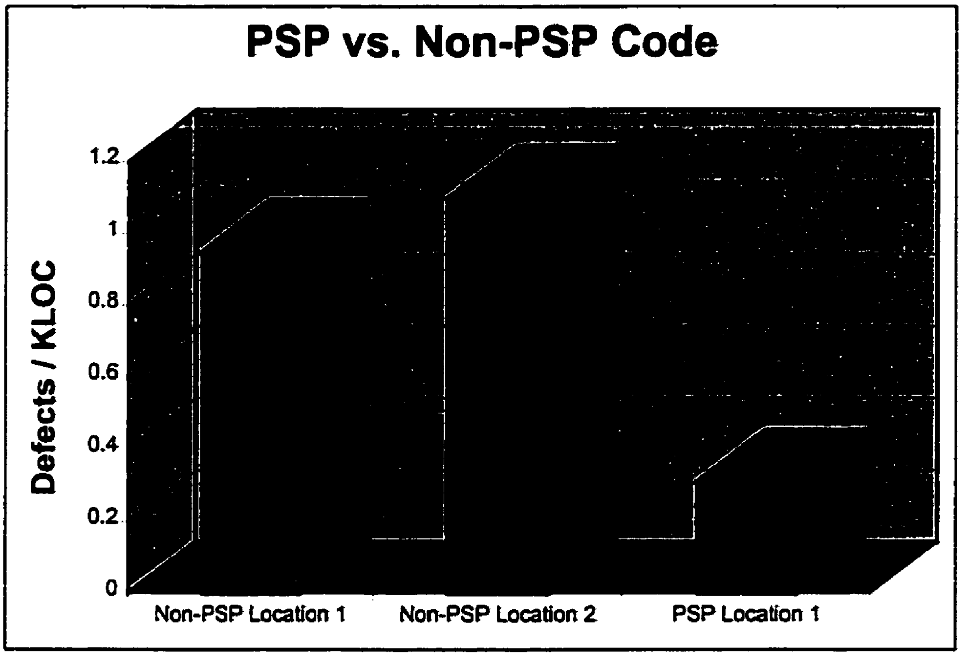
Figure 8: PSP Schedule Estimating Error on an Industrial Project (Frankovich, 1998)



The first three components had a schedule estimation error range from 350 to 425 percent. After the engineers were trained by a SEI licensed PSP trainer, the estimation error dropped to an average of -10.4 percent on the remaining six components. Similar to the Hayes / Over study, towards the end of the AIS pilot project, the engineers were drawing nearer to the point where there were a similar number of overestimation and underestimation errors.

Other metrics that were collected include measures of defect density and productivity. Figure 9 shows the defect density of the pilot project as measured by the number of defects per thousand lines of code (KLOC).

Figure 9: PSP Defect Density on an Industrial Project (Frankovich, 1998)



In Figure 8, the left-most bar indicates that the engineers had 0.8 defects per KLOC before they were trained in the PSP. The right most bar represents the same engineers after the were trained and shows a defect density improvement of 78 percent which resulted in a significant reduction in cycle time. The middle bar represents a control group of AIS engineers and provides an organizational reference point concerning defect density.

Lastly, the pilot project collected the engineers' personal productivity rates in order to determine the overhead associated with the introduction of the PSP. It was found that there was a 7.4 percent improvement in the lines of code per hour that was produced.

3.1.3 Positive Aspects of the PSP Results

There is little doubt to the effectiveness of the PSP on the performance of individual software engineers. Hayes and Over statistically demonstrated marked improvements in effort estimation, size estimation, and defect density in an educational environment. That study is significant for a couple of reasons. Firstly, the data set was gathered from PSP training courses that were taught by SEI trained instructors. This standardized data collection environment helped to reduce the possibility of a third variable contaminating the study's results. Secondly, the study examined the improvement ratio of the individual engineers rather than the absolute value of the metric which helped to compensate for variability among individual engineers. Lastly, the large number of data points in the data set increased the reliability of the results.

AIS took the PSP to an industrial setting and documented results similar to those found in the Hayes and Over study. The results of the pilot project should not be interpreted as undeniable evidence that the PSP will be just as successful in an industrial environment as it was in the educational environment. However, AIS has added significant value to the software engineering community by demonstrating that the PSP can be utilized on real world projects with impressive results.

3.1.4 Negative Aspects of the PSP Results

As impressive as the Hayes and Over study is, it still remains a fact that the entire data set was collected within an educational environment. This theoretical and sterile environment is void of many factors that effect the performance of individual engineers on real world projects.

The number one reason given by software engineers as to why an industrial project failed is unstable requirements(Humphrey, 1989). In the PSP training course, the programming assignments are completely specified, and it is expected that each engineer will follow the specifications exactly. It is unclear how unstable requirements would effect the impact of the PSP on the performance of individual engineers. Additionally, the Hayes and Over data set was collected from the PSP course assignments which are small in nature, taking on average five to ten hours to complete. Overall, this limits the functional usability of this study to only those industrial projects that have frozen requirements, consist of modules requiring a maximum of ten hours to complete, and consist only of new development.

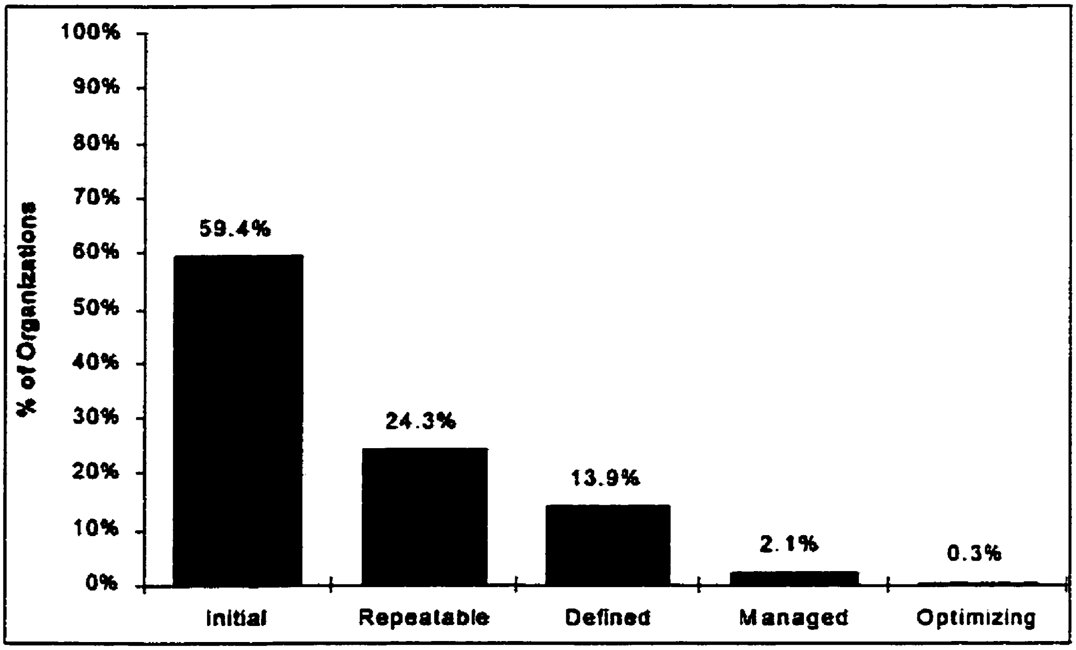
The Hayes and Over study is the only statistically significant empirical study of its kind, and it is only applicable to classroom-based assignments. The AIS pilot project can only be used as a proof of concept. Thus, no irrefutable conclusions can be drawn concerning the PSP's impact on the performance of engineers on real world projects.

3.2 Results of the CMM

3.2.1 Industry Trends

Since 1992, the Software Engineering Institute has collected data and analyzed trends related to the maturity of organizations in the software engineering community. The data comes from organizations that have conducted a software process assessment utilizing the SEI's CMM-Based Appraisal for Internal Process Improvement (CBA IPI). This unique perspective offers a cross-sectional view of the software community at a moment in time. Figure 10 displays organizational maturity broken-down by CMM levels.

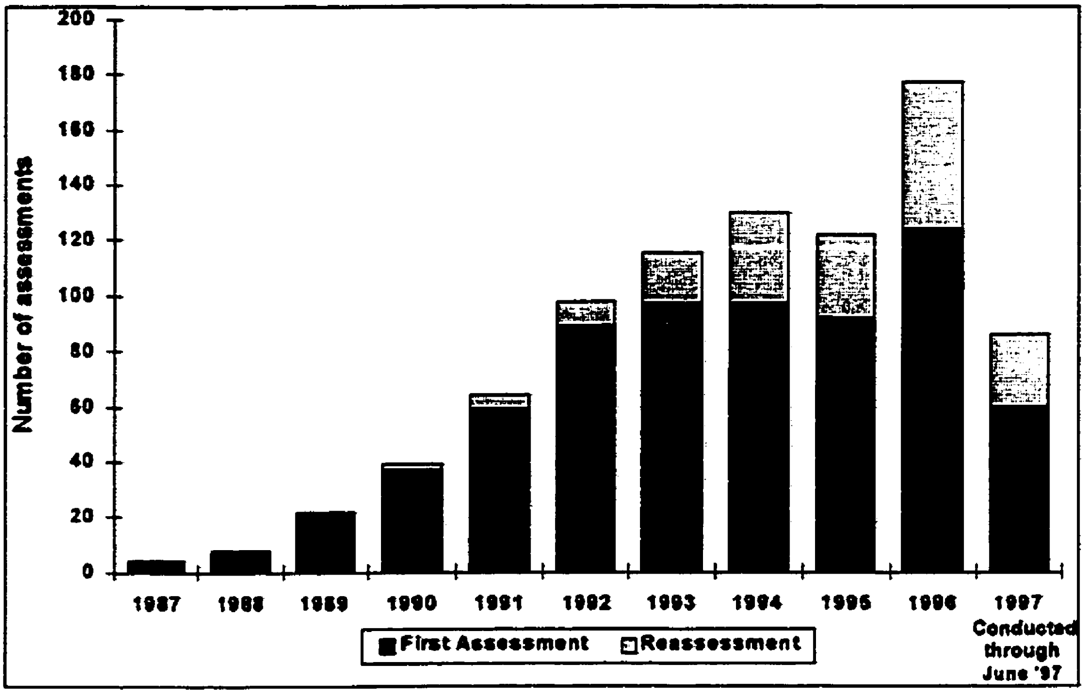
Figure 10: Organization Maturity Breakdown (SEI, 1998)



The above graph references 533 organizations that have conducted software process assessments in the last four years. This Figure shows that approximately 40% of the organizations involved have successfully moved beyond the initial level of the CMM with the remaining 60% unable to satisfy the criteria for the Repeatable level.

An important aspect in predicting the future growth of the software engineering community is to determine the diffusion rate of the industry's best practices. This indicator is numerically represented by the number of first time assessments per year, the Adoption Rate. Figure 11 displays the differences in Adoption Rates for the software community since 1987.

Figure 11: CMM Adoption Rate (SEI, 1998)



This figure shows an interesting trend that has emerge over the last 5 years. From 1992 to 1996, the adoption rate has remained constant at approximately 100 first time assessments per year. This is contrasted with an adoption rate that doubled for the first five years. The organizations that comprise the data set can be broken down in to the following industry sectors: 47% Commercial / In-house, 29.7% - DOD / Federal, 18.2 % - Military, 4.8% Other.

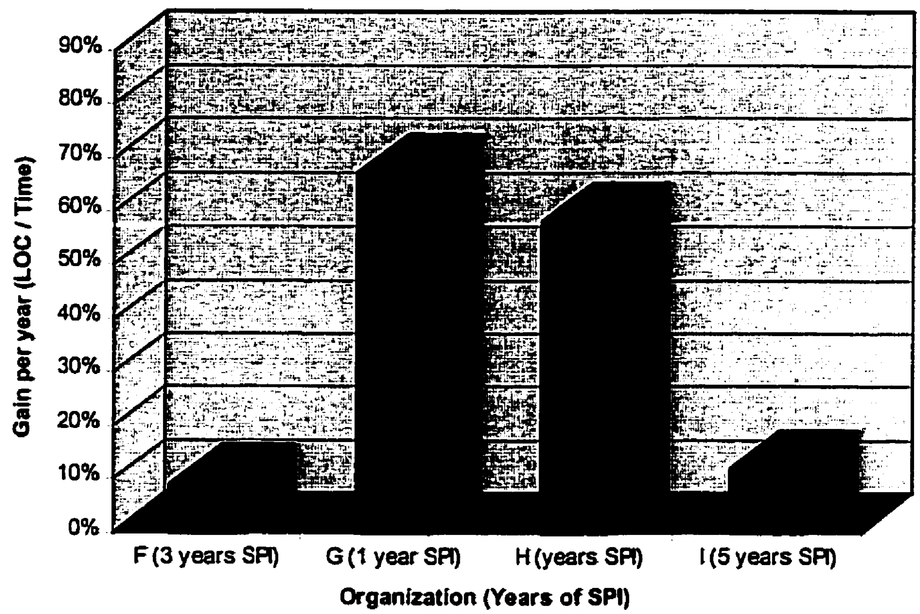
3.2.2 Industry Results

In 1993, members of the Empirical Methods Project (EMP) of the Software Engineering Institute set forth to gather and report quantitative information regarding the benefits gained from software process improvement initiatives. Members of the EMP gathered empirical data from 20 organizations that were early adapters of CMM bases software processes improvement programs. After applying stringent inclusion criteria to the data submitted, 13 organizations were chosen to be included in this first of a kind report, "Benefits of CMM-Based Software Process Improvement"(Herbsleb, 1994).

The Herbsleb Report included a very diverse group which included Department of Defense contractors, commercial organizations, and military organizations. The application domains included telecommunications, embedded real-time systems, information services, and operating systems. The authors survey the Software Engineering Process Groups of the organizations for data relating to three distinct metrics categories: impact of Software Process Improvement (SPI) on business objectives, impact of SPI on social factors, and actual performance versus projections. Two results of interest to this thesis are productivity gains and improvements in time to market.

The Herbsleb Report defined productivity to be the number of Lines of Code (LOC) developed per unit time. In order to determine the productivity gain associated with a SPI, the productivity rate before the SPI must be measured. This measure was particularly problematic for the authors since most organizations did not collect productivity rates prior to the formation of their Software Engineering Process Groups. However, four organizations provided the data which is shown in Figure 12.

Figure 12: CMM-SPI Productivity Gain per Year (Herbsleb, 1994)



Organization G experienced a 67% increase in productivity during the first year of their SPI initiative. This represents the results of two different projects developing similar applications which had a substantial overlap in resources. Overall, the productivity gains ranged from 9% to 69% averaging 35%.

Another impressive finding of the Herbsleb Report was the improvement in time to market as a result of the SPI program. Reduction in the amount of time required to develop applications represents a substantial competitive advantage. Figure 13 depicts the reduction in time to market that two organizations experienced.

Figure 13: CMM-SPI Time to Market (Herbsleb, 1994)

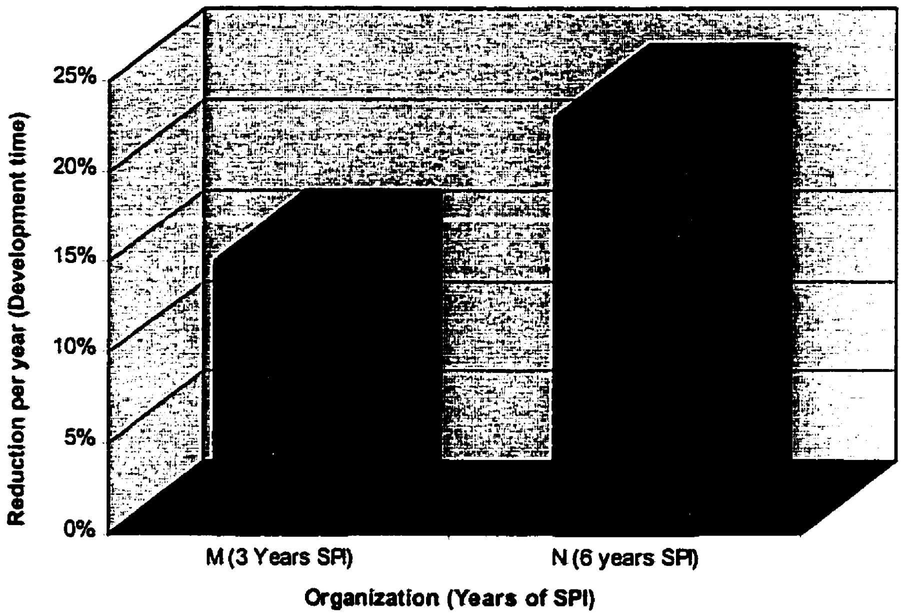


Figure 13 shows that organization N experienced a 19% reduction in time to market per year. It is important to realize that the above gains represent improvements per year, thus organization M experienced an 11% reduction in time to market for three consecutive years.

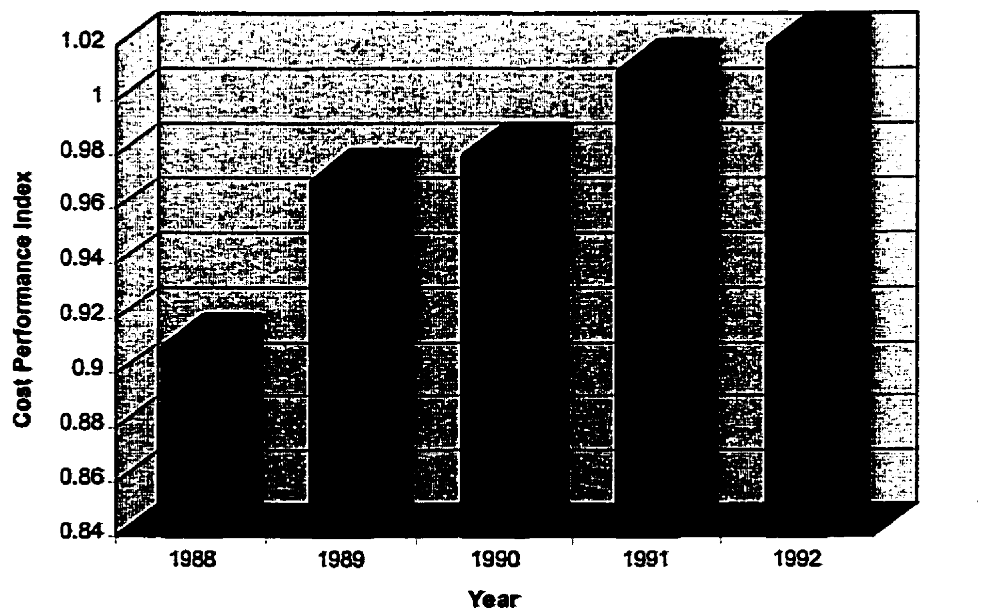
3.2.3 Industrial Case Study

Hughes Aircraft is considered to be one of the pioneers in the software engineering industry due to its long history of software process improvement. The Software Engineering Division (SED) of Hughes Aircraft started a substantial SPI program as far back as 1973. From 1973 to 1987, the SED process improvement was built on Total Quality Management (TQM) policies and practices. Project data was collected, analyzed and used for continuous process improvement, which is very similar to modern CMM Quantitative Process Management principles. Projects were tracked with Earned Value calculations and defect density was recorded. The SED process improvement efforts may have been very advanced; however, not all processes were institutionalized, and central coordination was lacking in some areas.

In 1987, the SED underwent its first SEI CMM-Based Appraisal for Internal Process Improvement (CBA-IPI) which resulted in a CMM Level 2 certification. As with any assessment, a process improvement action plan was created that addressed the weakness in their software process. The SED completed the action plan in 1990 and was reassessed, which resulted in a CMM Level 3 certification. Additionally, the 1990 assessment found that SED had many process activities in place at the Level 4 and 5 maturity levels. The last assessment took place in 1992 which gained the SED a CMM Level 4 certification.

One of the data metrics that the SED collected was the Cost Performance Index. This represents the ratio of budgeted cost of the work performed to the actual cost of the work performed. Figure 14 depicts the Cost Performance Index for projects between 1988 and 1992.

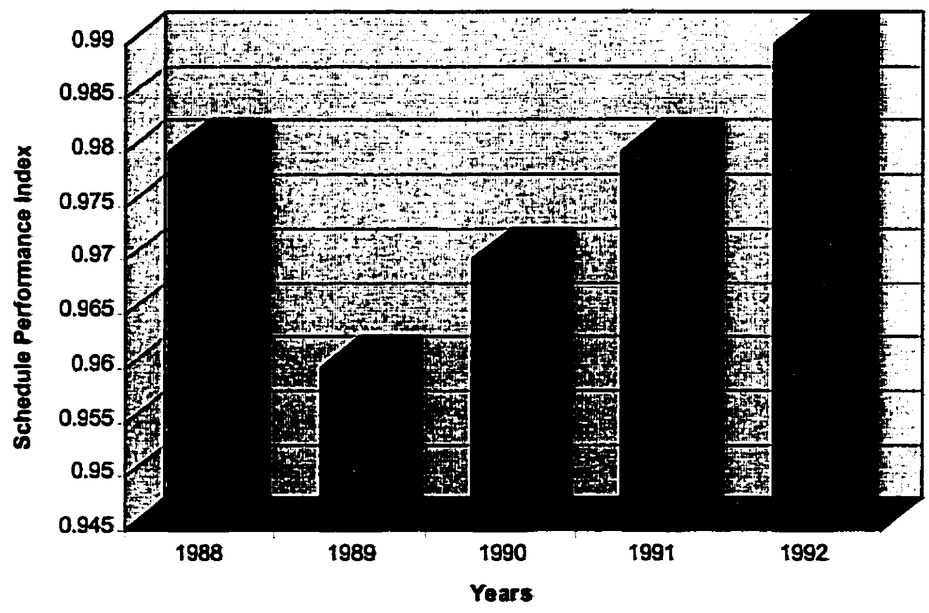
Figure 14: Hughes Cost Performance Index (Herbsleb, 1994)



A Cost Performance Index of 1.00 indicates budgeted cost matching actual cost, and an index below 1.00 indicates a budget overrun in which the actual cost of the project was greater than the budgeted resources. Figure 14 shows that the SED has continuously improved their cost estimating abilities. In fact, in 1992 an overestimation of only 2% had occurred.

The SED also collected the Schedule Performance Index over the same period of time. The Schedule Performance Index is collected in a similar manner to the Cost Performance Index. Figure 15 depicts the Cost Performance Index for projects between 1988 and 1992.

Figure 15: Hughes Schedule Performance Index (Herbsleb, 1994)



The SED experienced a modest dip in the Schedule Performance Index during 1989 which correlates to the time frame where the SED was implementing the findings of the

1987 assessment. After the implementation of the process improvement action plan, the SED experienced a steady increase in its ability to predict the schedule of developmental efforts.

3.2.4 Positive Aspects of CMM Results

There is a definite momentum in the software engineering industry towards adopting CMM based software process improvement initiatives. This is evident in the Software Engineering Institutes published adoption rate for First-Time assessments. Of the organizations currently pursuing process improvements, 40% have achieved a CMM Level 2 or higher certification. This is welcome news for a software industry that has been plagued by low quality and massive budget overruns. Fundamentally, this demonstrates a paradigm shift from developing software towards engineering software.

The Hersleb Report documents the process related improvements experienced by 13 different organizations. On average, the organizations experienced a 35% increase in productivity per year, with one group experiencing an incredible 69% increase in productivity per year. Additionally, a couple of organizations experienced a marked decreased in the time to market metric for their developmental efforts. Success stories like the ones presented in the Hersleb Report help propel the process improvement industry forward.

The Hughes Aircraft case study helps put to rest some old myths about software process improvements frameworks. Many falsely believe that to improve quality means to sacrifice schedule and increase the overall cost of a project. The SED group found that their Schedule Performance index remained relatively constant during the entire implementation phase of the software process improvement program. The Cost Performance Index actually experienced a substantial gain representing an improved ability to predict the overall budget of their projects.

3.2.5 Negative Aspects of CMM Results

The Software Engineering Institute annual survey of industry maturity levels demonstrates the forward momentum of CMM based process improvement efforts, but the focus is not on the empirical benefits of these programs. This results in a lack of hard evidence for many of the improvement claims made by the process community.

There exist studies like the Herbsleb Report that can provide useful insight into the state of the industry, but they lack the large scale scope of a statistically significant industry study. The Herbsleb Report utilized data from 13 organizations, many of which were military and Department of Defense contractors. Therefor this report cannot be generalized to all organizations since approximately half of the first time assessments in 1996 are for commercial software organizations. These organizations undoubtedly face environments and implementation issues different from those in the military world. Additionally, the productivity gains reported are for organizations that previously maintained productivity rates before implementing a CMM based SPI effort. This implies a raw competency or latent ability towards software process that may not be present in other organizations. Thus, the productivity gains may be overstated by utilizing organizations predisposed to process improvements.

The Hughes Aircraft case study suffers the same process disposition bias as the Herbsleb Report. The SED group was certified CMM Level 2 in 1987 but the schedule and cost data reported started in 1988. The data reported more accurately reflects the improvement gains realized after achieving maturity Level 3. Only 16.3% of organizations are operating at a Level 3 or higher, thus it would be inappropriate to generalize these results to the majority of organizations beginning a software process improvement program. Additionally, the calculations used to establish the Schedule and Cost Performance Indexes is unknown. It is unclear if the results are the summation of many projects over a year or represent a single program as it is developed over many years. It is dangerous to draw conclusions from data whose origins are unknown.

3.3 Supplementing the CMM with the PSP

There is no published data available on how to supplement the CMM with the PSP to form a cohesive organizational software process.

One possible source of information that will soon be available is the Team Software Process (TSP). Watts Humphrey is leading a team at the Software Engineering Institute that is developing the TSP which is schedule to publish results sometime in 1999. Currently there is very little information available about the structure of the TSP. The following is an excerpt from an article in which Watts Humphrey is introducing the TSP:

> "Software development is generally taught as a solo activity but when engineers go into industry, most of them work on teams. The transition from solo to team behavior is not obvious and engineers rarely get guidance on how to work as team members. The Team Software Process (TSP)SM provides such guidance. The TSP's principal objective is to show PSP-trained engineers how to run a team-based project. The TSP also creates an environment that fosters continued use of disciplined PSP methods.(Humphrey, 1998)"

3.4 Summary

The literature reviewed in this chapter suggests that CMM based SPI programs do not significantly impact project schedule and cost predictability. However, performance gains are realized in increased productivity and reduced time to market measures. The Hayes / Over study demonstrates that in a class room environment there exists a statistically significant increase in an engineer's ability to plan and schedule his / her work. Finally, there exists no published data on the topic of supplementing the CMM with the PSP.

Chapter 4: Research Environment

4.0 Introduction

The goal of this chapter is to describe the organizational environment where the research was conducted. Thus, this chapter starts by describing the organization's process history from their initial Declaration of Improvement through to their current software process improvement initiative. This is followed by a description of the organization's standard lifecycle. Finally, the mechanism used by the organization to integrate the PSP into the CMM process architecture is described in detail.

Due to confidentiality issues, the organization does not wish to be named and will be referred to as the XYZ organization. Records concerning the organization's authenticity and the verification of the data collected have been affirmed by the University of Calgary's Software Engineering Research Chairperson, Dr. Mildred Shaw.

4.1 The Organizational process history

The XYZ organization is an independent software contractor with facilities in the United States and India. XYZ's application domain includes software development, consulting, and software process training. The software development group accounted for 33% of the revenue in 1997 and varied in size between 20 and 30 software professionals. The software projects in the development group typically ranged in size from one to eight software professionals.

The XYZ organization started their software process improvement program in January 1992. They used the Software Engineering Institute's CMM as the process maturity framework to improve their organizational process capability. Additionally, the PSP was later selected as the enabling technology to improve each individual engineer's performance and productivity. The organization publicly stated its Declaration of Improvement, which included three main points.

1. Improve profitability of development projects by meeting cost estimates and schedule commitments with reasonable consistency.
2. Provide a continuing management focus on the progress and visibility of each project from initial commitment to orderly progression through the development lifecycle phases and customer acceptance.
3. Enable continuous improvement of the development process through a changed organizational culture biased towards rapid implementation of many small incremental improvements as opposed to a few large changes.

In April 1996, XYZ participated in an internal CMM Based Appraisal for Internal Process Improvement (CBA IPI) with a SEI licensed assessor. The objectives of the appraisal were to identify software process weaknesses and strengths, identify highest priority issues for software process improvements, and provide a framework to focus process improvement actions. The scope was to assess five projects where XYZ had responsibility for project and process management, and to assess the software process for the CMM Repeatable and Defined Levels. The following table summarizes the major findings of the assessment.

Figure 16: XYZ Process Assessment Findings

KPA Fully Satisfied	Improvement Opportunities
Requirements Management	Software Configuration Management
Software Project Planning	Software Quality Management
Software Project Planning & Tracking	Subcontract Management

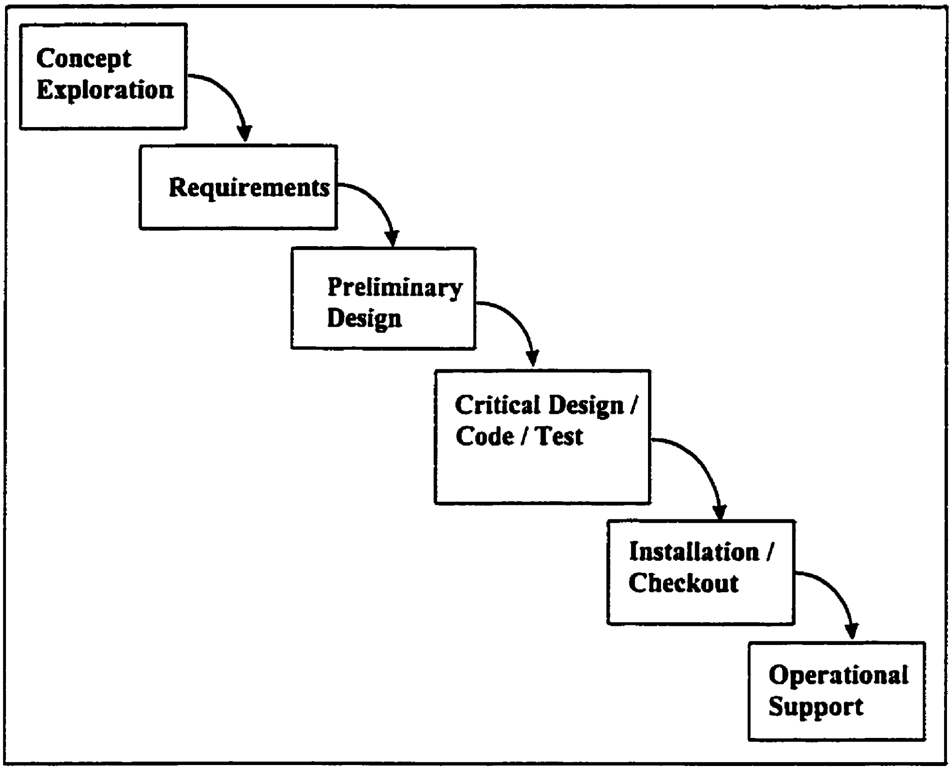
Organization Process Focus	Organizational Process Definition
Training Program	
Integrated Software Management	
Intergroup Coordination	
Peer Reviews	

A process improvement action plan based on the improvement opportunities identified in the assessment was presented to the development group in June 1996. The action plan identified the tasks to be completed, planned schedule for implementation, and resources assigned. The Action plan was successfully completed in December 1997, and the organization is currently preparing for another assessment sometime in the first quarter of 1999.

4.2 The Organizational Standard Lifecycle

The XYZ Organizational Standard Lifecycle represents a six step sequential approach to the development of software. It is based on the classic waterfall model, a systematic approach in which the software development effort progresses through a series of discrete phases. Figure 17 depicts the phases of the XYZ lifecycle.

Figure 17: XYZ Lifecycle



In the Concept Exploration phase, the customer's high level business needs and the scope of the project are determined and recorded in the Preliminary Analysis document. A contract called the Statement of Work (SOW) is developed for the requirements phase. The SOW contains a description of the tasks to be completed, estimated hours, estimated schedule, and the phase deliverables. Deliverables from this phase include:

1. Preliminary Analysis Document
2. Statement of Work for the Requirements Phase

In the Requirements phase, the activities of gathering, analyzing, and documenting the requirements occurs within a Joint Requirements Planning sessions (JRP).
All requirements that are identified during this phase are documented in a Software Requirements Specification (SRS). The goal of the SRS is to describe all externally observed behaviors and characteristics expected of the software system. The contents of a SRS include specifications for the inputs, outputs, system behaviors, and externally

observable characteristics. A Software Quality Assurance Plan is constructed that describes the quality activities that are necessary in order to ensure compliance to XYZ standards. Deliverables from this phase include:

1. Software Requirements Specification (IEEE Std 830-1993)
2. Software Quality Assurance Plan (IEEE Std 730-1993)
3. Statement of Work for the Preliminary Design phase

In the Preliminary Design phase, a high level design of the system is developed and documented in the Preliminary Software Design Document (PSDD). The contents of the PSDD include Entity Relationship Diagrams, Data Dictionary, Data Flow Diagrams, Process Descriptions, and File Definitions. A Prototype may be used to gain a better understanding of the interface design. A Software Configuration Management Plan is constructed that describe the activities for managing the creation and evolution of the software work products. Additionally, a system and acceptance test plans are developed cooperatively with the customer. Deliverables of this phase include:

1. Prototype
2. Preliminary Software Design Document
3. Software Configuration Management Plan (IEEE 828-1990)
4. System / Acceptance Test Plan (IEEE 829-1983)
5. Statement of Work for the Critical Design / Code / Test phase

In the Critical Design / Code / Test phase, each component identified during Preliminary Design is refined and expanded, the tests are designed, the code is written, and the components are tested. Components are integrated and Integration Tests are executed. The Systems and Acceptance Tests are executed on the completed system and the Preliminary Users Documentation is written. Deliverables of this phase include:

1. Critical Software Design Description
2. Component Test Plan (IEEE Std 1008-1987)
3. Integration Test Plan (IEEE Std 829-1983)
4. System Test Design (IEEE Std 829-1983)

5. Product Executable and Files
6. Preliminary User Documentation
7. Statement of Work for the Installation / Checkout phase

In the Installation / Checkout phase, the product is installed in the customer's environment and acceptance testing is performed. The User Documentation is completed based on the feedback of acceptance testing. Deliverable of this phase include:

1. User Documentation
2. Verification and Validation Report
3. Statement of Work for the Operational Support Phase

The Operational Support phase is the final phase in the XYZ life cycle. An electronic copy of the project's work products are delivered to the customer. If required, resources are made available for ongoing maintenance. If user training is required, a training program is developed and delivered. Deliverables of this phase include:

1. Electronic copies of work products
2. Maintenance support
3. Training Program

4.3 Integrating PSP into CMM

The aim of this thesis is to investigate the impact that supplementing the CMM with the PSP has on effort and schedule predictability on real-world industrial projects. This investigation must include a detailed description of the research environment in order to gain perspective on the data metrics that were collected. It is sufficient to describe only those process areas that have a direct impact on effort and schedule predictability, specifically the CMM key process areas of Software Project Planning and Software Project Tracking and Oversight. It is also necessary to show how the PSP processes can be integrating into the CMM process framework, to form a coherent process architecture. Describing this integration shows that the results apply equally to the CMM and the PSP. Thus, the description of the research environment includes:

1. A description of the PSP-CMM process integration for Software Project Planning
2. A technical description of how the PSP processes map to each activity with the key process area of Software Project Planning
3. A description of the PSP-CMM process integration for Software Project Tracking and Oversight
4. A technical description of how the PSP processes map to each activity with the key process area of Software Project Tracking and Oversight

4.3.1 PSP-CMM Integration: Software Project Planning

At the end of Preliminary Design, a Statement of Work is developed for the Critical Design / Code / Test phase. This contract includes the estimated schedule and effort that is required to translate the High Level Design into an executable system. The estimates are created by breaking the High Level Design into smaller modules. The size and complexity of the modules are chosen in such a way that an engineer could reasonably implement them during a standard work week. These components are then given to PSP engineers who uses their own defined personal processes to estimate size, time, and effort required for implementation of the component. After all components have been estimated in this fashion, the estimation data is passed back to the project manager. The Statement of Work estimates are then developed by combining the individual estimates for all of the components of the High Level Design. Figure 18 graphically displays the relationship between the PSP and the CMM based Software Project Planning processes:

Figure 18: XYZ Planning Process

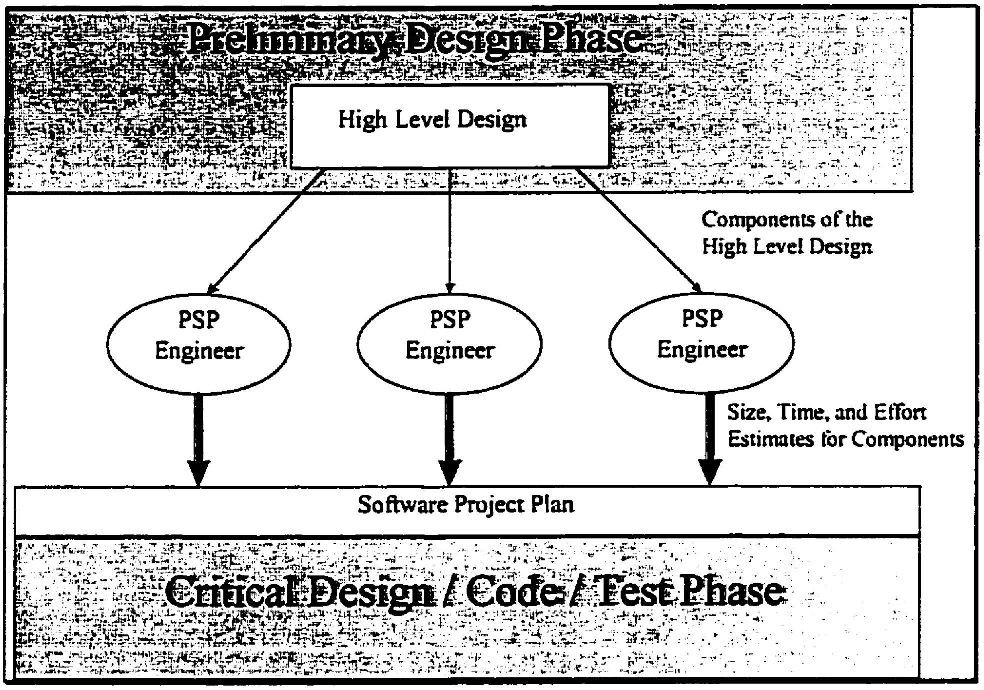


Figure 18 shows multiple PSP engineers involved with generating the estimates for Critical Design / Code / Test. It is important to note that the PSP module estimates typically account for 75 to 85 percent of the total estimates in the Software Project Plan. The Project Manager is responsible for estimating the time and effort of other non-PSP activities like developing the module descriptions, management overhead, and developing the Component, Integration, and System Test Plans.

4.3.2 PSP-CMM Mapping: Software Project Planning

The overall goal of the Key Process Area of Software Project Planning is to institutionalize planning processes that ensure that the organization has the ability to make plans that accurately reflect what it can reasonably accomplish. Table 3 reproduces the three goals of this Key Process Area as stipulated in the CMM.

Table 3: KPA Software Project Planning Goals and Activities

	As defined in the CMM	Activity Map
Goal 1	Software estimates are documented for use in planning and tracking the software project.	9, 10, 11, 12, 15
Goal 2	Software project activities and commitments are planned and documented.	2, 5, 6, 7, 8, 13, 14
Goal 3	Affected groups and individuals agree to their commitments related to the software project.	1, 3, 4

The PSP-CMM planning process structure described in Figure 18, can be used as a framework for satisfying the goals listed in Table 4. The techniques the PSP utilizes become part of the organizational planning process to form a single homogeneous process rather than disjointed extensions.

The following is a description of how the PSP maps to the Key Process Area of Software Project Planning. The activities of the KPA are listed along with a short description of how the PSP component of the PSP-CMM planning process structure helps to satisfy the activity. Each activity is classified as one of the following:

- **Fully Supported.** The PSP component of the PSP-CMM planning structure is a major component in supporting the activity.
- **Partially Supported.** The PSP component of the PSP-CMM planning structure plays a supporting role in supporting the activity.
- **Not Supported.** The PSP component of the PSP-CMM planning structure plays no role in supporting the activity.

It is important to note that this mapping is only for the PSP component of the PSP-CMM planning structure in the Critical Design / Code / Test phase of a project. Thus, when activities are categorized as Partially Supported or Not Supported, it should only be interpreted to mean that the PSP does not full address the activity. This does not preclude other organizational processes supporting the activities.

Activity 1 **The software engineering group participates on the project proposal team.**

In a PSP-CMM project, the proposed commitments are fully documented in the software project plan. The proposal includes schedules, required resources, module decomposition, and design strategies. The PSP engineer directly participates in constructing the commitments by using their own defined personal process to estimate size, time, and effort required for implementation of the component.

The PSP fully supports this activity.

Activity 2 **Software project planning is initiated in the early stages of, and in parallel with, the overall project plan.**

The PSP does not support this activity.

Activity 3 **The software engineering group participates with other affected groups in the overall project planning throughout the project's life.**

In a PSP-CMM project, the software engineering group is composed of PSP engineers. Within the scope of the Critical Design / Code / Test phase, the software engineering group also makes up the "other affected groups".

The PSP fully supports this activity.

Activity 4 **Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure.**

The PSP does not support this activity.

Activity 5 **A software lifecycle with predefined stages of manageable size is identified or defined.**

In a PSP-CMM project, the software lifecycle includes the PSP lifecycle. The engineers follow the cyclic developmental lifecycle of the PSP. The following scripts and instruction sheets define the cyclic developmental life cycle:

- PSP3 Process Script
- PSP3 Planning Script
- PSP3 Development Script
- PSP3 Postmortem Script
- Cycle Summary Instructions

The PSP fully supports this activity.

Activity 6 **The project's software development plan is developed according to a documented procedure.**

In a PSP-CMM project, the engineers develop their own software development plan which are rolled up into the overall project plan. The PSP software development plan is developed according to the documented procedure specified in the following script:

- PSP3 Planning Script

The PSP fully supports this activity.

Activity 7 The plan for the software project is documented.

In a PSP-CMM project, the engineer's software development plan is specified in the following scripts, forms, and templates:

- PSP3 Project Plan Summary
- Cycle Summary Form
- Size Estimating Template
- Task Planning Template
- Schedule Planning Template

The PSP fully supports this activity.

Activity 8 Software work products that are needed to establish and maintain control of the software project are identified.

The PSP process identifies the following work product that are needed to establish and maintain control of the software project:

- PSP3 Project Plan Summary
- PSP3 Issue Tracking Log
- Task Planning Template
- Schedule Planning Template
- Operation Scenario Template
- Functional Specification Template
- State Specification Template
- Logic Specification Template

The PSP fully supports this activity.

Activity 9 Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure.

In a PSP-CMM project, the engineer derives size estimates according to documented procedures and then rolls them up to the project level. The engineer derives estimates according to the procedures defined in the following scripts, instruction sheets, and templates:

- PROBE Estimating Script
- PSP3 Project Plan Summary Instructions
- Size Estimating Template Instructions
- Size Estimating Template

The PSP fully supports this activity.

Activity 10 Estimates for the software project's effort and cost are derived according to a documented procedure.

In a PSP-CMM project, the engineer derives the resource estimates according to the documented procedure defined in the following script:

- PROBE Estimating Script
- PSP3 Project Plan Summary Instructions
- Resource Size Estimating Template Instructions
- Resource Estimating Template

The PSP fully supports this activity.

Activity 11 Estimates for the project's critical computer resources are derived according to a documented procedure.

The PSP does not supports this activity.

Activity 12 The projects software schedule is derived according to a documented procedure.

In a PSP-CMM project, the engineer derives the schedule according to the documented procedure defined in the following templates, instruction sheets, and scripts:

- Task Planning Template Instructions
- Schedule Planning Template Instructions
- Task Planning Template
- Schedule Planning Template
- PSP3 Planning Script

The PSP fully supports this activity.

Activity 13 The software risks associated with the cost, resource, schedule, and technical aspects of the project are identified, assessed, and documented.

The PSP does not supports this activity.

Activity 14 Plans for the project's software engineering facilities and support tools are prepared.

The PSP does not supports this activity.

Activity 15 Software planning data is recorded.

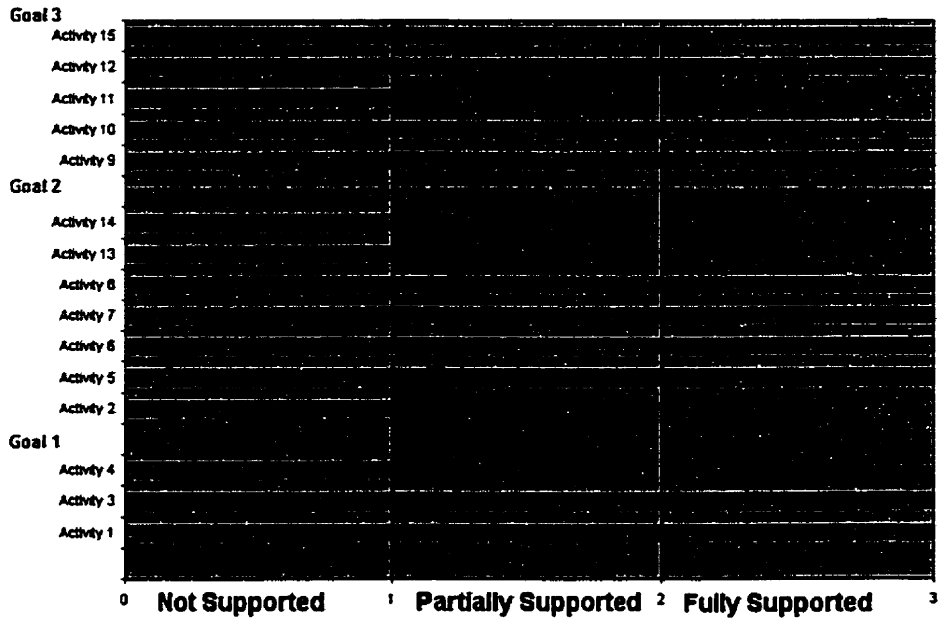
The PSP engineer records all data associated with the planning of project in the following scripts:

- PSP3 Project Plan Summary
- Cycle Summary Form
- Size Estimating Template
- Task Planning Template
- Schedule Planning Template

The PSP fully supports this activity.

The above mapping illustrates how the PSP can supplement the CMM with regards to Software Project Planning. The results of the detailed PSP-CMM planning process mapping are displayed graphically in Figure 19.

Figure 19: Software Project Planning Activities Supported by PSP



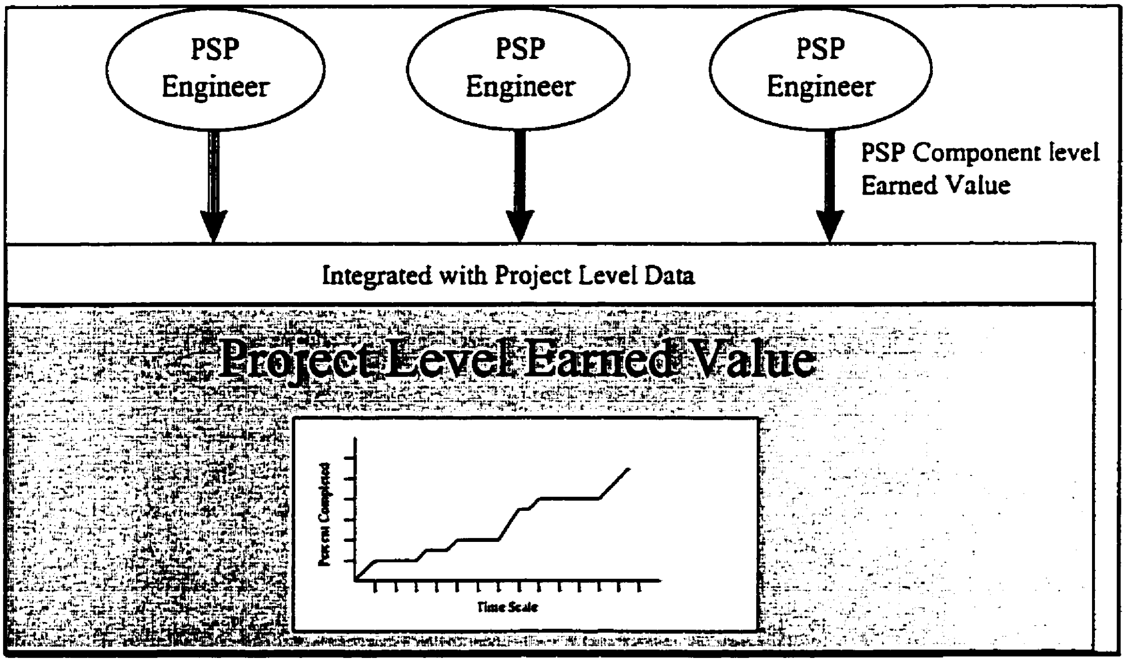
4.3.3 PSP-CMM Integration: Software Project Tracking and Oversight

The Key Process Area of Software Project Tracking and Oversight focus is on management's ability to determine project status. In order to provide adequate visibility into the process of developing software, actual results and performances must be tracked against the Software Project Plan. When the actual results and the plan significantly differ, corrective action should be taken. One way to perform this type of detailed tracking is with earned value project scheduling.

Earned value (EV) tracking is a mechanism to evaluate the progress of a project (Boehm, 1981). EV works by establishing a value for each task in the software project plan. This value represents the percent of effort required to complete the task relative to the overall project effort. The EV tracking mechanism provides a common value scale for each task regardless of the type of work involved. As each task is completed, the project is awarded that task's earned value. When the project reaches 100% earned value, all of the planned tasks are completed.

The following chart graphically displays the relationship between the PSP and the CMM based Software Project Tracking and Oversight processes:

Figure 20: XYZ Tracking and Oversight Process



As mentioned earlier, the Software Project Plan is developed by integrating the individual estimates for all of the components of the high level design. In the Critical Design / Code / Test phase, each of these components is implemented by engineers performing full PSP2.1 cycles on the components. Each engineer tracks to completion each of the implementation tasks with earned value. This low level earned value is then passed back to the project level via weekly or monthly project status update meetings.

Project level earned value can be constructed in one of two ways depending on the granularity of tracking that is required by the project: Component Level earned value or Engineer Level earned value.

Component Level Earned Value

The project manager can track component level earned value by defining an earned value task to be the completion of a PSP 2.1 Cycle. Once the engineer has completed the implementation of a component (when the engineers earned value is 100%), the project is awarded the earned value associated with the component.

Engineer Level Earned Value

The project manager can track the engineering level earned value by defining an earned value task to be the completion one of the PSP phases of the PSP2.1 cycle (Planning, Design, Implementation, Testing, Postmortem). Once the engineer has completed a PSP phase, the project is awarded the earned value associated with the phase.

This type of low level earned value tracking can greatly increase the visibility into the process of developing software. This type of approach allows the project manager to track earned value not only at the project level but also at the individual engineer level. The complexity of defining a software process to calculate earned value is masked by the PSP since the process is already defined for the engineering level.

4.3.4 PSP-CMM Mapping: Software Project Tracking and Oversight

The overall goal of the Key Process Area of Software Project Tracking and Oversight is to institutionalize tracking processes that ensure that the organization has adequate visibility into the process and progress of the organizations projects. Table 4 reproduces the three goals of this Key Process Area as stipulated in the CMM.

Table 4: KPA Software Project Tracking and Oversight Goals and Activities

	As defined in the CMM	**Supporting Activities**
Goal 1	Actual results and performance are tracked against the software plans.	1, 5, 6, 7, 8, 9, 10, 11, 12, 13
Goal 2	Corrective actions are taken and manages to closure when actual results and performance deviate significantly from the software plans.	2, 5, 6, 7, 8, 9, 11
Goal 3	Changes to software commitments are agreed to by the affected groups and individuals.	3, 4

The following is a description of how the PSP maps to the Key Process Area of Software Project Tracking and Oversight. The activities of the KPA are listed along with a short description of how the PSP component of the PSP-CMM tracking process structure satisfies the activity. Each activity is classified as one of the following:

- **Fully Supported.** The PSP component of the PSP-CMM tracking structure is a major component in supporting the activity.
- **Partially Supported.** The PSP component of the PSP-CMM tracking structure plays a supporting role in supporting the activity.
- **Not Supported.** The PSP component of the PSP-CMM tracking structure plays no role in supporting the activity.

It is important to note that this mapping is only for the PSP component of the PSP-CMM tracking structure in the Critical Design / Code / Test phase of a project. Thus, when activities are categorized as Partially Supported or Not Supported, it should only be interpreted to mean that the PSP does not full address the activity. This does not preclude other organizational processes supporting the activities.

Activity 1 **A documented software development plan is used for tracking the software activities and communicating status.**

In a PSP-CMM project, the engineers track their work according to their software developmental plan, which is specified and documented in the following scripts, forms, and templates:

- PSP3 Project Plan Summary
- Cycle Summary Form
- Size Estimating Template
- Task Planning Template
- Schedule Planning Template

The PSP fully supports this activity.

Activity 2 **The project's software development plan is revised according to a documented procedure.**

The PSP does not support this activity.

Activity 3 **Software project commitments and changes to commitments made to individuals and groups external to the organization are review with senior management according to a documented procedure.**

The PSP does not support this activity.

Activity 4 **Approved changes to commitments that affect the software project are**

communicated to the members of the software engineering group and other software related groups.

The PSP does not support this activity.

Activity 5 **The sizes of the software work products (or sizes of the changes to the software work products) are tracked, and corrective actions are taken as necessary.**

In a PSP-CMM project, the engineer produces a software project plan that contains detailed size estimates for all modules to be developed. The process of tracking the actual size of each module is defined in the following forms and instruction sheets:

- Cycle summary Instructions
- Cycle Summary

The PSP fully supports this activity.

Activity 6 **The project's software effort and cost are tracked, and corrective actions are taken as necessary.**

In a PSP-CMM project, the engineer tracks actual effort from which cost can be derived. The following instruction sheets and templates define the effort tracking procedure:

- Schedule Planning Template Instructions
- Schedule Planning Template
- Cycle Summary Instructions

- Cycle Summary

The PSP fully supports this activity.

Activity 7 **The project's critical computer resources are tracked and corrective actions are taken as necessary.**

The PSP does not support this activity.

Activity 8 **The project's software schedule is tracked, and corrective actions are taken as necessary.**

In a PSP-CMM project, the engineer tracks the project's schedule with actual and estimated earned value. The following instruction sheets and templates define the schedule tracking procedure:

- Schedule Planning Template Instructions
- Schedule Planning Template

The PSP fully supports this activity.

Activity 9 **Software engineering technical activities are tracked, and corrective actions are taken as necessary.**

In a PSP-CMM project, the engineer tracks the project's technical activities with actual and estimated earned value. The following instruction sheets and templates define the task tracking procedure:

- Task Planning Template Instructions
- Task Planning Template
- Cycle Summary Instructions
- Cycle Summary

The PSP fully supports this activity.

Activity 10 **The software risks associated with cost, resource, schedule, and technical aspects of the project are tracked.**

The PSP does not support this activity.

Activity 11 **Actual measurement data and re-planning data for the software project are recorded.**

In a PSP-CMM project, the engineer records all actual measurement and re-planning data in the following scripts and templates:

- PSP3 Project Plan Summary
- Test Report Template
- Cycle Summary
- Task Planning Template
- Schedule Planning Template
- Time Recording Log
- Defect Recording Log

The PSP fully supports this activity.

Activity 12 **The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan.**

In a PSP-CMM project, the engineer documents, tracks and manages issues related to the development of software. The data that is reviewed is contained in the following scripts, forms and templates

- PSP3 Project Plan Summary
- Cycle Summary Form
- Size Estimating Template
- Task Planning Template
- Schedule Planning Template
- Issue Tracking Log

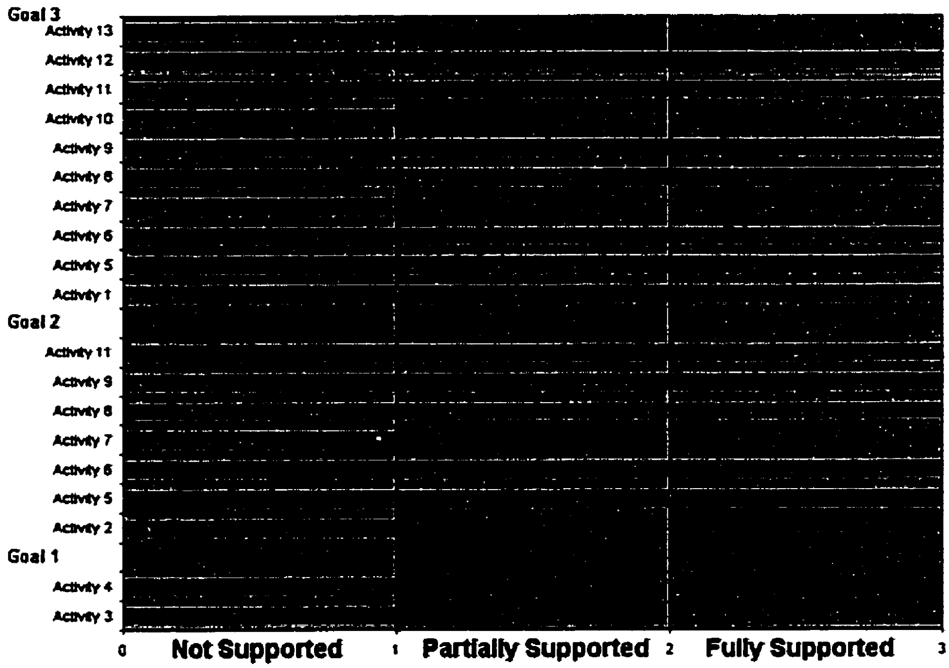
The PSP fully supports this activity.

Activity 13 Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure.

The PSP does not support this activity.

The above mapping illustrates how the PSP can supplement the CMM with regards to Software Project Tracking and Oversight. The results of the detailed PSP-CMM tracking process mapping is displayed graphically in Figure 21.

Figure 21: Software Project Tracking and Oversight Activities Supported by PSP



4.4 Summary

The data set utilized in this thesis was collected in an environment that can be characterized as mature, institutionalized, and process focused. Additionally, its organizational standard lifecycle contains well defined phases, standardized deliverables, and quality processes. Finally, the process mechanism used by the organization for Software Project Planning and Software Project Tracking and Oversight were formed from supplementing the CMM framework with the PSP.

Chapter 5: Research Implementation

5.0 Introduction

The goal of this chapter is to document what data metrics were collected, how they were collected, and the project characteristics that generated the metrics. Thus, this chapter starts with a description of the project characteristics and the resulting data set. This is followed by a description of the XYZ metric program and the organizational process database. Schedule and effort predictability are defined and organizational data is presented. Finally, positive and negative aspects of the data set are elucidated.

5.1 Project Characteristics

The majority of the projects that XYZ has undertaken have been information technology projects were XYZ has maintained project management control of the environment. There have been implementations from fortune 100 organizations, media companies, and transportation companies.

The data set used in this thesis consists of 37 distinct projects representing over 100,000 man hours. Of these, 64.9% of the projects were unique development efforts with the remaining 35.1% being enhancements to existing software. Additionally, the average length of the projects was 11 months and the average assigned resources were just over three. All projects were internally managed by XYZ using the organizational standard software process that was present in the organization at the time.

5.2 Metrics Program

The Software Process Engineering Group (SEPG) was created in 1992 and has been assigned the responsibility of ownership and updates to the XYZ defined software process, including collecting and maintaining project related data metrics. During the Phase Review process, the Project Manager submits to the SEPG the Project Data Collection form which includes the following data items:

1. Project Name
2. Phase Name
3. Project Manager
4. Current Date
5. Estimated Hours
6. Actual Hours
7. Start Date
8. Planned End Date
9. Actual End Date
10. Average Number of Resources
11. Acceptance Test Defects
12. Post Delivery Defects
13. Customer Satisfaction
14. Re-planned? (Y/N)

Only after the submission of the SEPG Project Data Collection form can a project be officially close, and the Project Manager relieved from its responsibility. Thus, the data metrics are record immediately after phase completion, which has a direct impact on the accuracy of the data. The SEPG group then inputs the data metrics into the organizational process database.

Since the SEPG group was formed in 1992, the data metrics from 1988 to 1992 had to be collected in a different fashion. Last year, XYZ was nominated for the IEEE Computer Society Award for Software Process Improvement. This required submitting a detailed report on the improvement trends that the organization had experienced since its foundation. Four individuals, the founder, the president, the process manager, and the director of training combed the organization's records and reconstructed the metrics from 1988 to 1992 which were later assimilated into the organizational database.

The XYZ organization process database currently contains over 850 data items representing over 100 different project phases from 1988 to 1998. The Thesis Data Set is shown in Appendix A Table K and lists the data metrics that are used in this thesis, a subset of the XYZ organizational process database. The data set used in this thesis consists of all the Critical Design / Code / Test project phases, since it is this phase of the lifecycle where the PSP is most integrated into CMM architecture.

5.3 Schedule Predictability

The Estimated Schedule is the total projected time between the project's start date and its estimated end date. The Estimated Schedule is calculated as follows:

Estimated Schedule = f (Estimated End Date - Start Date)

Where $f(X)$ = Converts X to a whole number.

The Actual Schedule is the total elapsed time between the project's start date and its end date. The Actual Schedule is calculated as follows:

Actual Schedule = f (Actual End Date - Start Date)

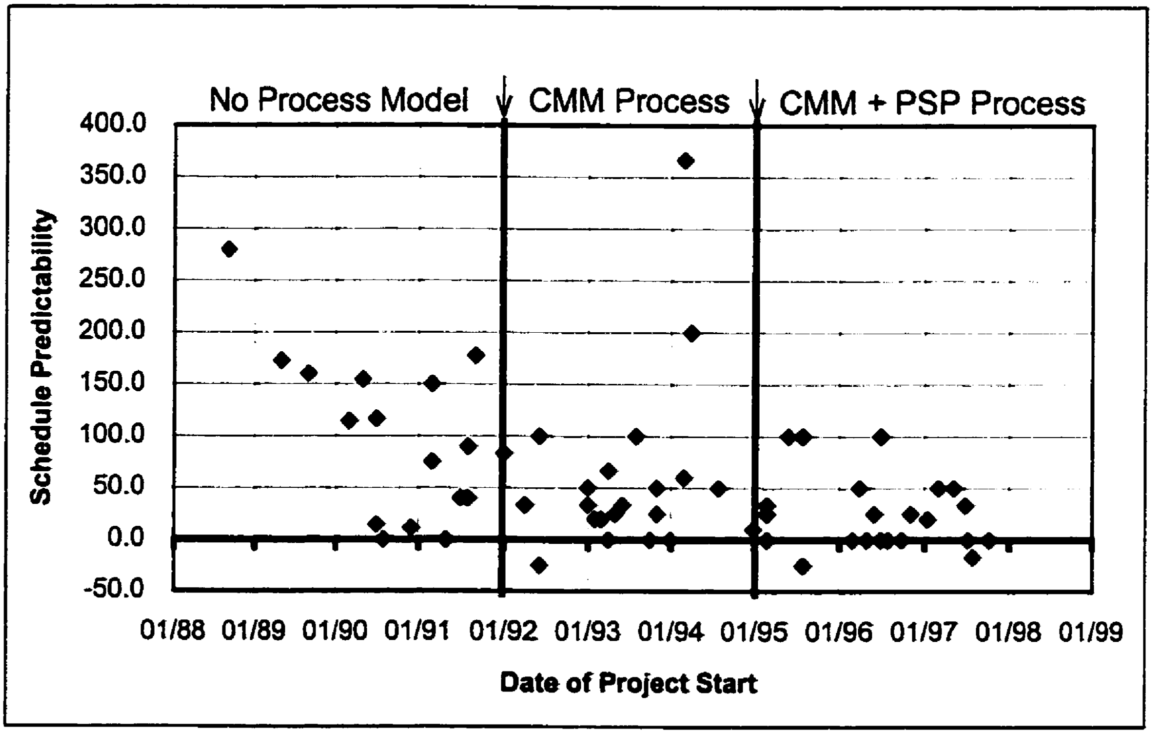
Where $f(X)$ = Converts X to a whole number.

Schedule predictability refers to how closely the Estimated Schedule matches the Actual Schedule. This can be numerically represented as a percentage by calculating the Schedule Predictability as follows:

Schedule Predictability = (Actual Schedule /Estimated Schedule)*100)-100

The following figure graphically represents the Schedule Predictability experienced by XYZ over ten years of project activity. The dates represented along the horizontal axis are from July 1988 though December 1998. The date the project phase started was selected as the horizontal coordinate because the process maturity of the project phase is primarily influenced by the maturity of the organization when the phase started. The three methodology milestones are represented as labels across the top of the graph: when the organization followed no process model (1988 - 1992), implementation of a CMM process architecture (1992 - 1995), and the integration of the PSP into the CMM framework (1996 - 1998).

Figure 22: XYZ Schedule Predictability (Frankovich, 1998)



The graph appears to show a steady decline in the schedule predictability through the three methodology milestones. Any conclusions concerning the effectiveness of the methodologies on schedule predictability should be deferred to chapter six, when statistical inference methods are used to determine if any statistically significant conclusions can be drawn from this data set.

5.4 Effort Predictability

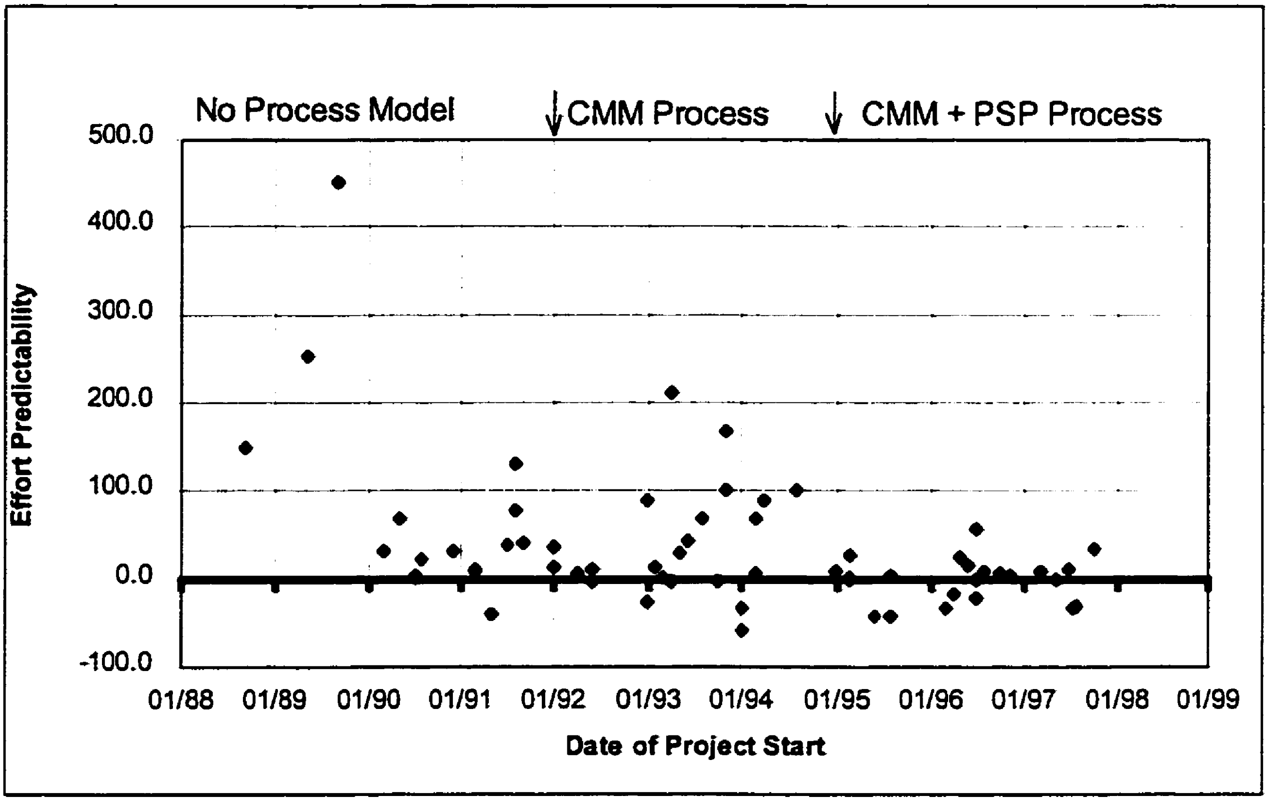
The Estimated Effort is the total projected number of hours that are planned to be worked in order to complete the project. The Actual Effort is the actual number of hours worked on the project. Both data items were retrieved from the organizational process database.

Effort Predictability refers to how closely the Estimated Effort matches the Actual Effort. This can be numerically represented as a percentage by calculating the Effort Predictability as follows:

Effort Predictability = (Actual Effort /Estimated Effort) * 100) - 100

The following figure graphically represents the Effort Predictability experienced by XYZ over ten years of project activity. It contains the same format as the Schedule Predictability graph: methodology milestones across the top and project phase start dates along the horizontal axis.

Figure 23: XYZ Effort Predictability (Frankovich, 1998)



Once again, there appears to be a noticeable decline in the effort predictability but no conclusions should be drawn at this time. Additionally, there appears to be evidence of organizational learning which the Statistical Testing Procedures of this thesis will explore.

5.5 Data Comparison

5.5.1 Positive Aspects of the Data Set

One of the major findings from the 1996 CMM Based Appraisal for Internal Process Improvement (CBA IPI) was the fact that the XYZ organizational standard software process was highly institutionalized. This means that process improvement had become a part of the organization culture and had been fully accepted as the standard way of conducting business. This high level of buy in and cooperation resulted in the software methodologies being applied very consistently across all projects. Thus, the data set collected from the projects can be considered representative of the process. In other words, there is a very low probability that the projects that submitted data to this study did not follow the methodology.

Another positive aspect of the data set is that 76% of the data collected was generated by projects contracted by a single client. This common environment, culture, and application domain adds reliability to the data and helps reduce the impact of variability in this study.

5.5.2 Negative Aspects of the Data Set

One of the most difficult problems that organizations face is retaining those employees that form their knowledge base. The loss of this base can have a direct impact on the performance of the organization. XYZ has experienced a 10.5 % average turnover rate over the last six years. Although this may be considered low by some organizations, it still can have a direct impact on the performance of individual projects, hence stunting the visible benefits of the process improvement program.

As with any organization that is involved with software process improvement, there is the possibility that the improvements may be caused by organizational growth in general and may not be directly related to the "new" process that has been implemented. This has an inseparable impact on the results of the data.

5.6 Summary

The Thesis Data Set contains data items representing 37 distinct project phase that were retrieved from the XYZ organizational process database. The majority of projects that contributed to the process database shared a common client and, according to a recent CBA IPI assessment, followed institutionalized software methodologies. From five main data items (Estimated End Date, Actual End Date, Actual Effort, Estimated Effort, and Start Date) the composite metrics of Schedule Predictability and Effort predictability can be derived.

Chapter 6: Evaluation

6.0 Introduction

The goal of this chapter is to determine the statistical significance of the data presented in this thesis. Thus, this chapter starts with a discussion of the difficulties associated with real-world experiments. This is followed by a description of the research design and the statistical testing procedure that was used to analyze the thesis data set. Finally, each test is described and statistical results are presented.

6.1 Industrial Real-world Experiments

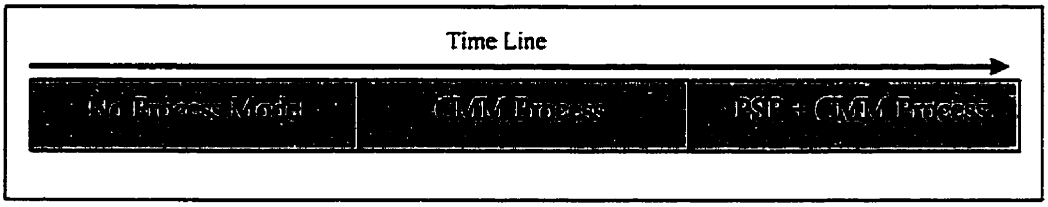
There is a difference between an experiment that is performed in a controlled environment and one that is carried out in an industrial setting. The major difference is in the control of the environmental variables. In the XYZ working environment, it was impossible to identify and utilize a control group since its environment is dynamic and is in a continuous state of flux. This had the effect of producing test results that may be open to other explanations and interpretations than the ones offered in this thesis.

As with any experiment, it is difficult to state unequivocally the cause and effect relationship between the independent and dependant variables. The best that can be offered is an explanation of the relationship the exists between them. One must always consider the possible existence of a third variable that is operating between two variables. Thus, any conclusions offered in this thesis should be understood to reflect characteristics of the relationship and not hard facts.

6.2 Research Design

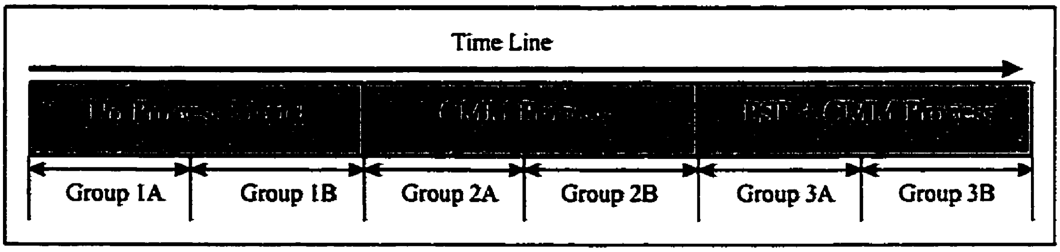
The aim of this research is to investigate the impact on effort and schedule predictability in real world projects that results from supplementing the CMM with the PSP. The data collected on schedule and effort predictability can be represented as follows:

Figure 24: Thesis Data Set



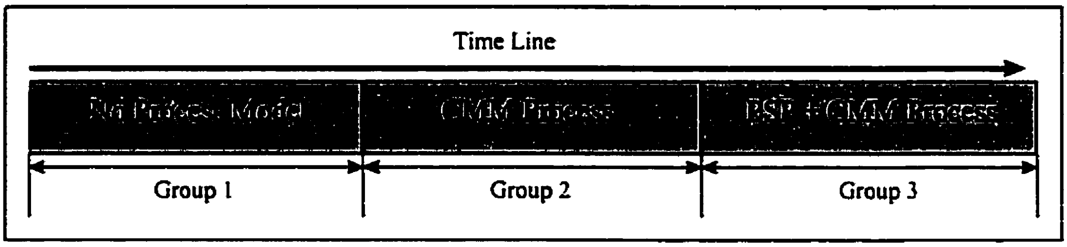
In order to investigate any changes in predictability from one data set to the next, it is necessary first to demonstrate that the three data sets are independent of each other. If the data sets are not independent then any improvement trends experienced when CMM process is introduced may be carried over to the PSP + CMM data set. If this were the case, it would be impossible to determine if the improvement was caused by integrating the PSP into the CMM or if it was a continuation of the improvement trend initiated in the CMM Process data set. Under the same premise, the organization may be naturally learning over time how to make more accurate estimates, regardless of the methodology introduced. Thus, the first step in the statistical analysis of the data is to demonstrate independence of the three data sets. This is accomplished by breaking the data sets into the following groupings:

Figure 25: Data Set Testing Subgroups



To demonstrate that there exists no improvement trend contamination, it is sufficient to show that the A groups are drawn from the same sample population as the B groups (1A = 1B, 2A = 2B, 3A = 3B). If there was an improvement trend, it would be identified in these subgroups and the A groups would not be equivalent to the B groups. After data set independence is established, the data set will be grouped as follows:

Figure 26: Data Set Testing Groups



To investigate the impact on schedule and effort predictability, the Wilcoxon Statistical Inference test for two independent samples (Segel, 1998) will be used to determine if there exists convincing evidence of the method's impact on predictability.

6.3 Statistical Testing Procedure

All statistical tests conducted in this thesis follow the same basic testing procedure. The following table describes the testing procedure:

Table 5: Testing Procedure

Step	Description
Null Hypothesis	State the null hypothesis (Ho) and its alternative (H1).
Statistical Test	Select the test that satisfies the research design requirements.
Significance level	Determine a significance level (α) and a sample size.
Sample Distribution	Define the sample distribution.

Rejection Reason	Define the region of rejection.
Decision	Compute test and decide whether to reject Ho.

The null hypothesis (Ho) is sometimes referred to as the "no effect" hypothesis. When evaluating two independent groups, it states that the groups are equivalent and the applied process had no measurable effect. If the null hypothesis is rejected then the alternative hypothesis (H1) is supported.

The Wilcoxon Statistical Test was selected as the Statistical test for all analysis that appears in thesis. The Wilcoxon test is used to test whether two independent groups have been drawn from the same population. This comparison is accomplished by first letting m = number of data items in the smaller group, n = number of data items in the larger group, and N = m + n. Next, both groups are combined and the numeric scores of all the data items are systematically ranked. The factor Wx = the summation of the ranks of group m. Equation 1.0 is used to calculate the normal approximation of the data set.

Equation 1.0

$$z = \frac{W_x + 0.5 - m(N+1)/2}{\sqrt{mn(N+1)/12}}$$

Finally, the probability (p) that the results are statistically significant is assessed by evaluating the data shown in Appendix A: Table L (Segel, 1988) : "Probabilities associated with the upper tail of the normal distribution". If the resulting probability is equal to or is less than the significance level determined in the research design, then reject Ho in favor of H1.

6.4 Statistical Tests Performed

The first six tests were designed to determine if the three data sets are independent of each other. The remaining four tests analyze the impact on Schedule and Effort Predictability resulting from supplementing the CMM with the PSP. The following table lists all statistical tests that were performed on the data set presented in Appendix B Table K: Thesis Data Set.

Table 6: Tests Performed

Row	Test Name	Test Result	Significance Level	Appendix Reference
1	Effort Predictability and Organizational Learning No Process Group	Date set independence	No	Table A
2	Schedule Predictability and Organizational Learning No Process Group	Date set independence	No	Table B
3	Effort Predictability and Organizational Learning CMM Group	Date set independence	No	Table C
4	Schedule Predictability and Organizational Learning CMM Group	Date set independence	No	Table D
5	Effort Predictability and Organizational Learning CMM-PSP Group	Date set independence	No	Table E
6	Schedule Predictability and Organizational Learning CMM-PSP Group	Date set independence	No	Table F
7	Effort Predictability No Process vs. CMM	No statistically significant impact	No	Table G

		on Effort predictability detected		
8	Schedule Predictability No Process vs. CMM	No statistically significant impact on Schedule predictability detected	No	Table H
9	Effort Predictability CMM vs. CMM-PSP	Statistically significant impact on Effort predictability detected	Yes	Table I
10	Schedule Predictability CMM vs. CMM-PSP	Statistically significant impact on Schedule predictability detected	Yes	Table J

For complete description of tests performed, see Appendix A: Statistical Test Performed.

6.5 Statistical Results

6.5.1 Data Set Independence

The results from the first six statistical tests supports the claim that the three thesis data sets have not experienced organizational learning and in fact can be considered independent data sets. This supports the selection of the Wilcoxon-Mann-Whitney test by satisfying its requirements in demonstrating independence of the selected data sets. Additionally, the four remaining statistical tests can be viewed without fear of contamination from improvement trends introduced in an earlier period.

This finding can best be explained by the realization that the CMM is a methodology framework with different categories of software processes being introduced at different levels of maturity. The CMM does have process improvement threads throughout all levels, but it is not until levels 4 and 5 that an organization focuses on continuous process improvement. At the time of data collection, XYZ had only introduced levels 2 and 3 processes so it is not surprising that there is no evidence of ongoing organizational learning or improvement trends that continue on after the initial introduction of the methodology.

6.5.2 No Process vs. CMM

The two statistical tests design to investigate the impact on effort and schedule predictability within the CMM data set yielded the following result:

> **There exists no evidence that XYZ experienced a statistically significant impact on their ability to predict Schedule or Effort required for the Critical Design / Code / Test phase as a result of introducing CMM Level 2 & 3 processes.**

It is very important to note that the above statement is only valid for the thesis data set that was collected in XYZ's working environment. There does exist the possibility of a third unknown variable, contamination from unusual business factors, or even unintentional errors in the original data entry. Additionally, it is equally important to note that the statement does not imply that:

1. XYZ did not experience an impact on predictability over the entire lifecycle of its project.
2. Organizations should expect the same results in their environment.

The CMM does include Levels 2 and 3 software methodologies that improve the organization's ability to engineer software in a predictable way. These processes are primarily reflected in the Key Process Areas of: Requirements Analysis, Project Planning, Project Tracking and Oversight, and Integrated Software Management. An organization may experience improvements in predictability over the entire lifecycle by implementing CMM processes but their effectiveness in the Critical Design / Code / Test phase can be enhanced by utilizing the Personal Software Process.

The location of Acceptance Testing within the XYZ lifecycle may influence the results presented above. Acceptance Testing occurs with the client and is primarily responsible for ensure that the final product satisfies their requirements. In the XYZ lifecycle, Accepting Testing occurs after the Critical Design / Code / Test phase thus any improvements experienced within its constructs would not be visible in the Thesis Data Set. This is significant since the organizational and project level processes of Levels 2 and 3 would likely have a positive impact on Acceptance Testing.

6.5.3 CMM vs. CMM-PSP

The two statistical tests designed to investigate the impact on effort and schedule predictability within the CMM-PSP data set yielded the following result:

> **There does exists evidence that XYZ experienced a statistically significant impact on their ability to predict the Schedule and Effort required for the Critical Design / Code / Test phase as a result of supplementing the CMM with the PSP.**

Once again, it is important to note that the above statement is only valid for the thesis data set that was collected with in the XYZ environment.

The PSP was designed to improve the engineer's capability to develop software, and the methodology's ability to improve the predictability of individual engineers in a classroom environment is well documented (Over, 1998). The findings from the analysis of the Thesis Data Set suggest that XYZ has successfully generated similar results in a working environment and has translated the improvements trends from the engineering level up to the project level.

The CMM Levels 2 and 3 organizational and project level processes were supplemented with the low level PSP engineering process to form the mechanism that facilitated the migration of the results from the engineer to the project. This is a significant finding since it would appear that this combination of macro and micro methodology finally addresses the Personal Gap that exist between the organizational framework and the individual that actually does the work.

6.6 Summary

This chapter described the research design that was used for the ten statistical tests that analyzed the Thesis Data Set, which yielded three findings. Firstly, the three data set groupings (No Process, CMM Process, and CMM-PSP Processes) were determined to be independent samples that possessed no organizational learning contamination. Secondly, there was no evidence in the CMM Process data set that XYZ experienced a significant improvement in predictability of the Critical Design / Code / Test phase. Finally, XYZ did experience predictability improvements when the CMM was supplemented with the PSP.

Chapter 7: Conclusions

7.0 Introduction

The goal of this chapter is to review the six objectives of the thesis and determine if they have been satisfied. Thus, this chapter starts by summarizing the findings as they relate to each objective. The chapter concludes with a discussion on possible future directions for this line of research.

7.1 Thesis Objectives

The fundamental aim of this research is to investigate the impact on effort and schedule predictability in real world projects that result from supplementing the CMM with the PSP. In order to systematically explore this aim, six objectives were defined. The following lists these objectives along with descriptions on how this thesis addressed them.

To review all literature that relates to the CMM and PSP and to determine the current state of the art relating to the aim of the thesis.

A detailed literature review was completed which focused mainly on the process architectures of the CMM and the PSP. It was concluded that the CMM is a process maturity framework designed to improve an organization's process capability and the PSP is a personal methodology that introduces quality processes into the engineer's daily work environment.

To critically analyze the current state of the art.

Available published reports were critically analyzed in order to determine the effectiveness of the two methodologies as related to the aim of the thesis. Results from the Hayes / Over study and an industrial case study (AIS) were used to determine the state of the art for the PSP. Additionally, CMM industrial data from Hughes Aircraft and the Software Engineering Institute were assessed in detail. One surprising conclusion reached form this analysis was that a CMM based SPI program did not significantly impact schedule or cost predictability, but rather, improved productivity and time to market measures. Additionally, the PSP demonstrated a statistically significant impact on predictability in a classroom based environment which was reproduced by at lease one industrial case study.

To describe the organizational environment where the research was conducted.

The research environment was documented by describing the organization's process history from their initial Declaration of Improvement though to their current software process improvement initiatives. An in-depth discussion was included concerning the organization's standard lifecycle and the mechanism used to integrate the PSP into the CMM process architecture. It was concluded that this integration formed a coherent process architecture that could be characterized as mature, institutionalized and process focused.

To document what data metrics were collected, how they were collected, and the project characteristics that generated the metrics.

Both the data metrics and the characteristics of the projects that they represent were described in detail. The processes by which the organizations Software Engineering Process Group used to collect, store, and review the data metrics were also described. Schedule and effort predictability were defined and the organization's data presented.

To determine the statistical significance of the data presented.

In order determine the statistical significance of the thesis data set, it was first necessary to partition the data set into three subsets and demonstrate that they represented three independent data sets. Using the Wilcoxon-Mann-Whitney Test for Small Samples, data set independence was confirmed which suggested that improvement trend contamination had not occurred. Finally, four statistical test were performed that evaluated the impact on schedule and effort predictability which yielded the two following results:

1. There exists no evidence that XYZ experienced a statistically significant impact on their ability to predict Schedule or Effort required for the Critical Design / Code / Test phase as a result of introducing CMM Level 2 & 3 processes.
2. There does exists evidence that XYZ experienced a statistically significant impact on their ability to predict the Schedule and Effort required for the Critical Design / Code / Test phase as a result of supplementing the CMM with the PSP.

To draw conclusions

Once again, the aim of this research is to investigate the impact on effort and schedule predictability in real world projects that result from supplementing the CMM with the PSP. It was found that XYZ did not increase their ability to estimate their Critical Design / Code / Test phase by introducing CMM Level 2 & 3 processes. However, supplementing the CMM with the PSP did have a statistically significant impact on effort and schedule predictability in this real world project.

7.2 Future Directions

An interesting extension of this thesis would be to develop a process methodology that offers an individual engineer all the benefits of the PSP within the context of the CMM. This concept differs slightly from the Team Software Process (TSP) that Watt Humphrey developed, which attempts to supplement the PSP with additional processes in order to scale it up from the individual to the project level. On the contrary, the proposed thesis extension is to scale the CMM down to the individual level. Thus, I suggest that an interesting assertion to investigate is:

> **By supplementing the PSP with additional personal processes, it may be possible to define a One Person Project (OPP) that addresses all Level 2 & 3 Key Process Areas of the CMM.**

The OPP could be defined as the set of activities, methods, and practices used by individual engineers to manage all software engineering and project management activities associated with the development and maintenance of software. This new methodology would contain all of the script, forms, and templates that are defined in the PSP, in addition to supplementary software processes that are designed to support the following Key Process Areas:

- Software Configuration Management
- Software Quality Assurance
- Requirements Management
- Organization Process Definition
- Organization Process Focus

The OPP would be used by individual engineers who are working on one person projects, who do not have the support of a mature organization, and are working in a process hostile environment. In other words, this methodology would be intended for consultants that are sent to the "client site". The PSP assumes that the organization provides the engineer with stable requirements, performs quality audits, and is responsible for Configuration Management processes (which is usually not the case for consultants). If the fundamental philosophy of the above 5 Key Process Areas could be represented in the OPP, then I believe that the OPP would represent an important step forward in the Software Process Improvement movement.

References

Boehm, B. W. (1981). **Software Engineering Economics.** Englewood Cliffs, NJ: Prentice Hall.

Defense Science Board (1987). **Report of the Defense Science Board Task Force on Military Software.** Washington, DC: Office of the Under Secretary of Defense Acquisition.

Frankovich, J. F. (1997). **The One Person Project Software Process.** Course Paper. Calgary, AB: University of Calgary.

Frankovich, J. F (1998). Supplementing the CMM with the PSP. **1998 Software Engineering Process Conference.** Chicago, IL. The Software Engineering Institute.

Hayes, W. and Over, J.W. (1997). **The Personal Software Process: An Empirical Study on the Impact of PSP on Individual Engineers.** Technical Report CMU/SEI-97-TR-001. Software Engineering Institute. Pittsburgh, PA: Carnegie Mellon University.

IEEE Std 730-1993. **Standard for Software Quality Assurance Plans.** Software Engineering: IEEE Standard Collection - 1994. New York, NY: The Institute for Electrical and Electronics Engineers, Inc.

IEEE 828-1990. **Standard for Software Configuration Plans.** Software Engineering: IEEE Standard Collection - 1994. New York, NY: The Institute for Electrical and Electronics Engineers, Inc.

IEEE 829-1983. **Standard for Software Test Documentation.** Software Engineering: IEEE Standard Collection - 1994. New York, NY: The Institute for Electrical and Electronics Engineers, Inc.

IEEE Std 830-1993. **Recommended Practice for Software Requirements Specification.** Software Engineering: IEEE Standard Collection - 1994. New York, NY: The Institute for Electrical and Electronics Engineers, Inc.

IEEE Std 1008-1987. **Standard for Software Unit Testing.** Software Engineering: IEEE Standard Collection - 1994. New York, NY: The Institute for Electrical and Electronics Engineers, Inc.

Industrial Data (Currently unavailable): hamlet.cogsci.umassd.edu/CoursePages/SoftwareEngineering/LectureMat/whyse.html

Iyer, S. (1996). **PSP and the Capability Maturity Model.** Technical Report. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., Zubrow, D. (1994). **Benefits of CMM Bases Software Process Improvement.** Technical Report CMU/SEI-94-TR-013. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Humphrey, W. S. (1998). **Why Don't They Practice What We Preach?** SEI Article. Pittsburgh, PA:
Software Engineering Institute, Carnegie Mellon University.

Humphrey, W. S. (1995a). **A Discipline for Software Engineering.** Reading, MA: Addison-Wesley.

Humphrey, W. S. (1989). **Managing the Software Process.** Reading, MA: Addison-Wesley.

Paulk, M., Curtis B. Chrissis, M. and Weber, C. (1993). **Capability Maturity Model for Software Version 1.1.** Technical Report CMU/SEI-93-TR-24. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Pressman, R. (1992). **Software Engineering: A Practitioner's Approach.** New York, NY: McGraw-Hill, Inc.

Segel, S., Casterllan, N. (1988). **Nonparametric Statistics: For the Behavioral Sciences.** New York, NY: McGraw-Hill, Inc.

SEI (1995). **Capability Maturity Model: Guidelines for improving the software process.** Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

SEI (1998). The Software Community Process Maturity Profile - May 1997. **Proceeding of the 1998 Software Engineering Process Group Conference.** Chicago, IL. The Software Engineering Institute.

Appendix A: Statistical Test Performed.

Table A: Effort Predictability and Organizational Learning - No Process Group

Test Steps / Procedures	**Description**
Null Hypothesis	Ho: The Percent Effort Deviation for the Critical Design / Code / Test phase of industrial projects that do not follow a defined process remains the same over time. H1: Industrial projects that do not follow a defined process will improve their Percent Effort Deviation for the Critical Design / Code / Test phases over time.
Statistical Test	Wilcoxon-Mann-Whitney Test for Small Samples
Significance Level	Let m = The number of data items -1 in the first half of the No Process Group. Let n = The number of data items in the second half of the No Process Group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 7$, $n = 8$
Sampling Distribution	The probability associated with the occurrence under Ho of values as extreme as an observed W_x may be determined by reading the probability associated with the W_x value from Reference Table L
Rejection Reason	Since H1 predicts the direction of the difference, the region of rejection is one-tailed. It consists of all values W_x which

	are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than $\alpha = 0.05$.
Decision	The data and calculations for Percent Effort Deviation related to Organizational Learning - No Process Group are shown in Appendix B, Table K. The results are as follows: $m = 7$ $n = 8$ $W_x = 66$ The probability when Ho is true is $p < 0.1405$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The data only weakly supports the hypothesis that industrial projects that do not follow a defined process will improve their Percent Effort Deviation for the Critical Design / Code / Test phases over time. Thus, Ho cannot be rejected based on this data.

Table B: Schedule Predictability and Organizational Learning - No Process Group

Test Steps / Procedures	**Description**
Null Hypothesis	Ho: The Percent Schedule Deviation for the Critical Design / Code / Test phase of industrial projects that do not follow a defined process remains the same over time. H1: Industrial projects that do not follow a defined process will improve their Percent Schedule Deviation for the Critical Design / Code / Test phases over time.
Statistical Test	Wilcoxon-Mann-Whitney Test for Small Samples
Significance Level	Let m = The number of data items -1 in the first half of the No Process Group. Let n = The number of data items in the second half of the No Process Group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 7$, $n = 8$
Sampling Distribution	The probability associated with the occurrence under Ho of values as extreme as an observed W_x may be determined by reading the probability associated with the W_x value from Reference Table L
Rejection Reason	Since H1 predicts the direction of the difference, the region

	of rejection is one-tailed. It consists of all values W_x which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than $\alpha = 0.05$.
Decision	The data and calculations for Percent Schedule Deviation related to Organizational Learning - No Process Group are shown in Appendix B, Table L. The results are as follows: $m = 7$ $n = 8$ $W_x = 66.5$ The probability when Ho is true is $p < 0.1405$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The data only weakly supports the hypothesis that industrial projects that do not follow a defined process will improve their Percent Schedule Deviation for the Critical Design / Code / Test phases over time. Thus, Ho cannot be rejected based on this data.

Table C: Effort Predictability and Organizational Learning - CMM Group

Test Steps / Procedures	Description
Null Hypothesis	Ho: The Percent Effort Deviation for the Critical Design / Code / Test phase of industrial projects that follow the CMM framework remains the same over time. H1: Industrial projects that follow the CMM framework will improve their Percent Effort Deviation for the Critical Design / Code / Test phases over time.
Statistical Test	Wilcoxon-Mann-Whitney Test for Small Samples
Significance Level	Let m = The number of data items -1 in the first half of the CMM Group. Let n = The number of data items in the second half of the CMM Group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 5$, $n = 7$
Sampling Distribution	The probability associated with the occurrence under Ho of values as extreme as an observed W_x may be determined by reading the probability associated with the W_x value from Reference Table L.
Rejection Reason	Since H1 predicts the direction of the difference, the region of rejection is one-tailed. It consists of all values W_x which

	are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than $\alpha = 0.05$.
Decision	The data and calculations for Percent Effort Deviation related to Organizational Learning - CMM Group are shown in Appendix B, Table M. The results are as follows: $m = 5$ $n = 7$ $W_x = 29$ The probability when Ho is true is $p < 0.7348$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The data only weakly supports the hypothesis that Industrial projects that follow the CMM framework will improve their Percent Effort Deviation for the Critical Design / Code / Test phases over time. Thus, Ho cannot be rejected based on this data.

Table D: Schedule Predictability and Organizational Learning - CMM Group

Test Steps / Procedures	**Description**
Null Hypothesis	Ho: The Percent Schedule Deviation for the Critical Design / Code / Test phase of industrial projects that follow the CMM framework remains the same over time. H1: Industrial projects that follow the CMM framework will improve their Percent Schedule Deviation for the Critical Design / Code / Test phases over time.
Statistical Test	Wilcoxon-Mann-Whitney Test for Small Samples
Significance Level	Let m = The number of data items -1 in the first half of the CMM Group. Let n = The number of data items in the second half of the CMM Group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 5$, $n = 7$
Sampling Distribution	The probability associated with the occurrence under Ho of values as extreme as an observed W_x may be determined by reading the probability associated with the W_x value from Reference Table L.
Rejection Reason	Since H1 predicts the direction of the difference, the region

	of rejection is one-tailed. It consists of all values W_x which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than α = 0.05.
Decision	The data and calculations for Percent Schedule Deviation related to Organizational Learning - CMM Group are shown in Appendix B, Table N. The results are as follows: m = 5 n = 7 $W_x = 27.5$ The probability when Ho is true is $p < 0.8283$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The only weakly supports the hypothesis that industrial projects that follow the CMM framework will improve their Percent Schedule Deviation for the Critical Design / Code / Test phases over time. Thus, Ho cannot be rejected based on this data.

Table E: Effort Predictability and Organizational Learning - CMM-PSP Group

Test Steps / Procedures	Description
Null Hypothesis	Ho: The Percent Effort Deviation for the Critical Design / Code / Test phase of industrial projects that supplement the CMM with the PSP remains the same over time. H1: Industrial projects that supplement the CMM with the PSP will improve their Percent Effort Deviation for the Critical Design / Code / Test phases over time.
Statistical Test	Wilcoxon-Mann-Whitney Test for Small Samples
Significance Level	Let m = The number of data items -1 in the first half of the CMM-PSP Group. Let n = The number of data items in the second half of the CMM-PSP Group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 4$, $n = 6$
Sampling Distribution	The probability associated with the occurrence under Ho of values as extreme as an observed W_x may be determined by reading the probability associated with the W_x value from Reference Table L.
Rejection Reason	Since H1 predicts the direction of the difference, the region

	of rejection is one-tailed. It consists of all values W_x which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than $\alpha = 0.05$.
Decision	The data and calculations for Percent Effort Deviation related to Organizational Learning - CMM-PSP Group are shown in Appendix B, Table O. The results are as follows: $m = 4$ $n = 6$ $W_x = 13$ The probability when Ho is true is $p < 0.9905$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The data only weakly supports the hypothesis that industrial projects that supplement the CMM with the PSP will improve their Percent Effort Deviation for the Critical Design / Code / Test phases over time. Thus, Ho cannot be rejected based on this data.

Table F: Schedule Predictability and Organizational Learning - CMM-PSP Group

Test Steps / Procedures	**Description**
Null Hypothesis	Ho: The Percent Schedule Deviation for the Critical Design / Code / Test phase of industrial projects that supplement the CMM with the PSP remains the same over time. H1: Industrial projects that supplement the CMM with the PSP will improve their Percent Schedule Deviation for the Critical Design / Code / Test phases over time.
Statistical Test	Wilcoxon-Mann-Whitney Test for Small Samples
Significance Level	Let m = The number of data items -1 in the first half of the CMM-PSP Group. Let n = The number of data items in the second half of the CMM-PSP Group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 4$, $n = 6$
Sampling Distribution	The probability associated with the occurrence under Ho of values as extreme as an observed W_x may be determined by reading the probability associated with the W_x value from Reference Table L.
Rejection Reason	Since H1 predicts the direction of the difference, the region

	of rejection is one-tailed. It consists of all values W_x which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than α = 0.05.
Decision	The data and calculations for Percent Schedule Deviation related to Organizational Learning - CMM-PSP Group are shown in Appendix B, Table P. The results are as follows: M = 4 N = 6 W_x = 10 The probability when Ho is true is $p < 0.1000$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The data only weakly supports the hypothesis that industrial projects that supplement the CMM with the PSP will improve their Percent Schedule Deviation for the Critical Design / Code / Test phases over time. Thus, Ho cannot be rejected based on this data.

Table G: Effort Predictability - No Process vs. CMM

Test Steps / Procedures	Description
Null Hypothesis	Ho: The Percent Effort Deviation for the Critical Design / Code / Test phase of industrial projects is the same for projects that follow the CMM framework as for projects that have no defined processes. H1: Industrial projects that follow the CMM framework will have a lower Percent Effort Deviation for the Critical Design / Code / Test phases than projects that follow no process.
Statistical Test	Wilcoxon-Mann-Whitney Test for Large Samples
Significance Level	Let m = The number of data items in the CMM Process group. Let n = The number of data items in the No Process group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 12$, $n = 15$
Sampling Distribution	Equation 1.0 yields values of z. The probability associated with the occurrence under Ho of values as extreme as an observed z may be determined by reading the probability associated with the z value from

	Reference Table L.
Rejection Reason	Since H1 predicts the direction of the difference, the region of rejection is one-tailed. It consists of all values z which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than $\alpha = 0.05$.
Decision	The data and calculations for Effort Deviation - No Process vs. CMM are shown in Appendix B, Table Q. The results are as follows: $m = 12$ $n = 15$ $W_x = 147$ $z = -1.00$ The probability when Ho is true is $p < 0.1587$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The data only weakly supports the hypothesis that industrial projects that follow the CMM framework will have a lower Percent Effort Deviation for the Critical Design / Code / Test phases than projects that follow no process. Thus, Ho cannot be rejected based on this data.

Table H: Schedule Predictability- No Process vs. CMM

Test Steps / Procedures	**Description**
Null Hypothesis	Ho: The Percent Schedule Deviation for the Critical Design / Code / Test phase of industrial projects is the same for projects that follow the CMM framework as for projects that have no defined processes. H1: Industrial projects that follow the CMM framework will have a lower Percent Schedule Deviation for the Critical Design / Code / Test phases than projects that follow no process.
Statistical Test	Wilcoxon-Mann-Whitney Test for Large Samples
Significance Level	Let m = The number of data items in the CMM process group. Let n = The number of data items in the No Process group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 12$, $n = 15$
Sampling Distribution	Equation 1.0 yields values of z. The probability associated with the occurrence under Ho of values as extreme as an observed z may be determined by reading the probability associated with the z value from Reference Table L.

Rejection Reason	Since H1 predicts the direction of the difference, the region of rejection is one-tailed. It consists of all values z which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than $\alpha = 0.05$.
Decision	The data and calculations for Schedule Deviation - No Process vs. CMM are shown in Appendix B, Table R. The results are as follows: $m = 12$ $n = 15$ $W_x = 155$ $z = -0.61$ The probability when Ho is true is $p < 0.2709$ which does not satisfy rejecting Ho under the significance level previously set.
Conclusion	The data only weakly supports the hypothesis that industrial projects that follow the CMM framework will have a lower Percent Schedule Deviation for the Critical Design / Code / Test phases than projects that follow no process. Thus, Ho cannot be rejected based on this data.

Table I: Effort Predictability - CMM vs. CMM-PSP

Test Steps / Procedures	**Description**
Null Hypothesis	Ho: The Percent Effort Deviation for the Critical Design / Code / Test phase of industrial projects is the same for projects that follow the CMM framework as for projects that supplement the CMM with the PSP. H1: Industrial projects that supplement the CMM with the PSP will have a lower Percent Effort Deviation for the Critical Design / Code / Test phases than projects that follow the CMM solely.
Statistical Test	Wilcoxon-Mann-Whitney Test for Large Samples
Significance Level	Let m = The number of data items in the CMM-PSP process group. Let n = The number of data items in the CMM group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 10$, $n = 12$
Sampling Distribution	Equation 1.0 yields values of z. The probability associated with the occurrence under Ho of values as extreme as an observed z may be determined by reading the probability associated with the z value from

	Reference Table L.
Rejection Reason	Since H1 predicts the direction of the difference, the region of rejection is one-tailed. It consists of all values z which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than α = 0.05.
Decision	The data and calculations for Effort Deviation - CMM vs. CMM-PSP are shown in Appendix B, Table S. The results are as follows: m = 10 n = 12 $W_x = 83$ z = -2.08 The probability when Ho is true is $p < 0.0188$ which does satisfy rejecting Ho under the significance level previously set.
Conclusion	The data does support the hypothesis that industrial projects that supplement the CMM with the PSP will have a lower Percent Effort Deviation for the Critical Design / Code / Test phases than projects that follow the CMM solely.

Table J: Schedule Predictability- CMM vs. CMM-PSP

Test Steps / Procedures	**Description**
Null Hypothesis	Ho: The Percent Schedule Deviation for the Critical Design / Code / Test phase of industrial projects is the same for projects that follow the CMM framework as for projects that supplement the CMM with the PSP. H1: Industrial projects that supplement the CMM with the PSP will have a lower Percent Schedule Deviation for the Critical Design / Code / Test phases than projects that follow the CMM solely.
Statistical Test	Wilcoxon-Mann-Whitney Test for Large Samples
Significance Level	Let m = The number of data items in the CMM-PSP process group. Let n = The number of data items in the CMM group. Let α = The Significance Level for rejecting Ho. Thus: $\alpha = 0.05$, $m = 10$, $n = 12$
Sampling Distribution	Equation 1.0 yields values of z. The probability associated with the occurrence under Ho of values as extreme as an observed z may be determined by reading the probability associated with the z value from

	Reference Table L.
Rejection Reason	Since H1 predicts the direction of the difference, the region of rejection is one-tailed. It consists of all values z which are so extreme (in the predicted direction) that the associated probability under Ho is equal to or less than $\alpha = 0.05$.
Decision	The data and calculations for Schedule Deviation - CMM vs. CMM-PSP are shown in Appendix B, Table T. The results are as follows: $m = 10$ $n = 12$ $W_x = 68$ $z = -3.07$ The probability when Ho is true is $p < 0.0011$ which does satisfy rejecting Ho under the significance level previously set.
Conclusion	The data does support the hypothesis that industrial projects that supplement the CMM with the PSP will have a lower Percent Schedule Deviation for the Critical Design / Code / Test phases than projects that follow the CMM solely.

Appendix B: List of Data and Results tables.

Table K: Effort Predictability and Organizational Learning - No Process Group

X = First half of the No Process Group

Y = Second half of the No Process Group

m = 7

n = 8

$W_x = 66$

p = 0.1405

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
149.3	30.5	-39.4	Y	1
253	11	4.4	X	2
451.1	9	9	Y	3
31	-39.4	11	Y	4
68.3	38.3	22.4	X	5
4.4	129.7	30.5	Y	6
22.4	77.8	31	X	7
	40.4	38.3	Y	8
		40.4	Y	9
		68.3	X	10
		77.8	Y	11
		129.7	Y	12
		149.3	X	13
		253	X	14
		451.1	X	15

Table L: Schedule Predictability and Organizational Learning - No Process Group

X = First half of the No Process Group

Y = Second half of the No Process Group

m = 7

n = 8

$W_x = 66.5$

p = 0.1405

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
280	11.1	0	X	0.5
172.7	75	0	Y	0.5
160	150	11.1	Y	3
114.3	0	14.3	X	4
154.5	40	40	Y	5.5
14.3	90	40	Y	5.5
0	40	75	Y	7
	177.8	90	Y	8
		114.3	X	9
		150	Y	10
		154.5	X	11
		177.8	Y	12
		160	X	13
		172.7	X	14
		280	X	15

Table M: Effort Predictability and Organizational Learning - CMM Group

X = First half of the CMM Process Group

Y = Second half of the CMM Process Group

$m = 5$

$n = 7$

$W_x = 29$

$p = 0.7348$

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
36.4	2.1	-3.5	X	1
13	-3.1	-3.1	Y	2
-3.5	67.3	2.1	Y	3
25.6	7	7	Y	4
88.1	68.5	13	X	5
	89.2	25.6	X	6
	100.8	36.4	X	7
		67.3	Y	8
		68.5	Y	9
		88.1	X	10
		89.2	Y	11
		100.8	Y	12

Table N: Schedule Predictability and Organizational Learning - CMM Group

X = First half of the CMM Process Group

Y = Second half of the CMM Process Group

$m = 5$

$n = 7$

$W_x = 27.5$

$p = 0.8283$

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
83.3	20	-25	X	1
83.3	0	0	Y	2
-25	100	20	Y	3
33.3	366.7	33.3	X	4
50	60	50	X	5.5
	200	50	Y	5.5
	50	60	Y	7
		83.3	X	8.5
		83.3	X	8.5
		100	Y	10
		200	Y	11
		366.7	Y	12

Table O: Effort Predictability and Organizational Learning - CMM-PSP Group

X = First half of the CMM-PSP Process Group

Y = Second half of the CMM Process Group

m = 4

n = 6

$W_x = 13$

p = 0.9905

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
8.4	-0.4	-41.4	X	1
-41.4	-22.4	-33.9	X	2
-33.9	2.9	-32.8	Y	3
25.1	-32.8	-22.4	Y	4
	0	-0.4	Y	5
	34.5	0	Y	6
		2.9	Y	7
		8.4	X	8
		25.1	X	9
		34.5	Y	10

Table P: Schedule Predictability and Organizational Learning - CMM-PSP Group

X = First Half of the CMM-PSP Process Group

Y = Second half of the CMM Process Group

m = 4

n = 6

$W_x = 10$

p = 1.000

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
10	0	-25	X	1
-25	0	0	X	4.5
0	25	0	X	4.5
0	0	0	Y	4.5
	20	0	Y	4.5
	0	0	Y	4.5
		0	Y	4.5
		10	X	8
		20	Y	9
		25	Y	10

Table Q: Effort Predictability - No Process vs. CMM

X = CMM Process Group

Y = No Process Group

m = 12

n = 15

$W_x = 147$

z = -1.00

p = 0.1587

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
36.4	149.3	-39.4	Y	1
13	253	-25.6	X	2
-3.5	451.1	-3.5	X	3
-25.6	31	-3.1	X	4
88.1	68.3	2.1	X	5
2.1	4.4	4.4	Y	6
-3.1	22.4	7	X	7
67.3	30.5	9	Y	8
7	11	11	Y	9
68.5	9	13	X	10
89.2	-39.4	22.4	Y	11
100.8	38.3	30.5	Y	12
	129.7	31	Y	13
	77.8	36.4	X	14

	40.4	38.3	Y	15
		40.4	Y	16
		67.3	X	17
		68.3	Y	18
		68.5	X	19
		77.8	Y	20
		88.1	X	21
		89.2	X	22
		100.8	X	23
		129.7	Y	24
		149.3	Y	25
		253	Y	26
.		451.1	Y	27

Table R: Schedule Predictability - No Process vs. CMM

X = CMM Process Group

Y = No Process Group

m = 12

n = 15

$W_x = 155$

z = -0.61

p = 0.2709

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
83.3	280	-25	X	1
83.3	172.7	0	X	2
-25	160	0	Y	3
33.3	114.3	0	Y	4
50	154.5	11.1	Y	5
20	14.3	14.3	Y	6
0	0	20	X	7
100	11.1	33.3	X	8
366.7	75	40	Y	9
60	150.0	40.0	Y	10
200	0	50	X	11
50	40.0	50	X	12
	90	60	X	13
	40	75	Y	14

	177.8	83.3	X	15
		83.3	X	16
		90	Y	17
		100	X	18
		114.3	Y	19
		150.0	Y	20
		154.5	Y	21
		160	Y	22
		172.7	Y	23
		177.8	Y	24
		200	X	25
		280	Y	26
		366.7	X	27

Table S: Effort Predictability - CMM vs. CMM-PSP

X = CMM-PSP Process Group

Y = CMM Process Group

m = 10

n = 12

$W_x = 83$

z = -2.08

p = 0.0188

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
8.4	36.4	-41.4	X	1
-41.4	13	-33.9	X	2
-33.9	-3.5	-32.8	X	3
25.1	-25.6	-25.6	Y	4
-0.4	88.1	-22.4	X	5
-22.4	2.1	-3.5	Y	6
2.9	-3.1	-3.1	Y	7
-32.8	67.3	-0.4	X	8
0	7	0	X	9
34.5	68.5	2.1	Y	10
	89.2	2.9	X	11
	100.8	7	Y	12
		8.4	X	13
		13	Y	14

		25.1	X	15
		34.5	X	16
		36.4	Y	17
		67.3	Y	18
		68.5	Y	19
		88.1	Y	20
		89.2	Y	21
		100.8	Y	22

Table T: Schedule Predictability - CMM vs. CMM-PSP

X = CMM-PSP Process Group

Y = CMM Process Group

m = 10

n = 12

$W_x = 68$

z = -3.07

p = 0.0011

CMM Process Group (X)	No Process Group (Y)	Combined Ordered Score	Group	Rank
10	83.3	-25	X	1
-25	83.3	-25	Y	2
0	-25	0	X	3
0	33.3	0	X	4
0	50	0	X	5
0	20	0	X	6
25	0	0	X	7
0	100	0	X	8
20	366.7	0	Y	9
0	60	10	X	10
	200	20	X	11
	50	20	Y	12
		25	X	13

		33.3	Y	14
		50	Y	15
		50	Y	16
		60	Y	17
		83.3	Y	18
		83.3	Y	19
		100	Y	20
		366.7	Y	21
		200	Y	22

Table U: Thesis Data Set

Project Number	Phase	Estimated Hours	Actual Hours	Start Date	Original End Date	Actual End Date	Schedule Deviation	Effort Deviation
1	CD/I/T	3419	8524	09/88	06/89	10/91	280.0	149.3
2	CD/I/T	2250	7942	05/89	03/90	10/91	172.7	253.0
3	CD/I/T	1500	8266	09/89	06/90	10/91	160.0	451.1
4	CD/I/T	4389	5748	03/90	09/90	05/91	114.3	31.0
5	CD/I/T	7029	11830	05/90	03/91	08/92	154.5	68.3
6	REQ/PD	1556	1586	07/90	12/90	07/91	116.7	1.9
7	CD/I/T	1140	1190	07/90	01/91	02/91	14.3	4.4
8	CD/I/T	1600	1958	08/90	07/91	07/91	0.0	22.4
9	CD/I/T	488	637	12/90	08/91	09/91	11.1	30.5
10	CD/I/T	1522	1689	03/91	06/91	09/91	75.0	11.0
11	CD/I/T	730	796	03/91	08/91	05/92	150.0	9.0
12	CD/I/T	763	462	05/91	11/91	11/91	0.0	-39.4
13	CD/I/T	2667	3688	07/91	11/91	01/92	40.0	38.3
14	CD/I/T	3004	6899	08/91	05/92	02/93	90.0	129.7
15	CD/I/T	2755	4898	08/91	10/92	04/93	40.0	77.8
16	CD/I/T	5950	8356	09/91	05/92	09/93	177.8	40.4
17	CD/I/T	220	300	01/92	06/92	11/92	83.3	36.4
18	CD/I/T	2652	2998	01/92	06/92	11/92	83.3	13.0
19	REQ	379	399	04/92	06/92	07/92	33.3	5.3
20	CD/I/T	375	362	06/92	09/92	08/92	-25.0	-3.5
21	REQ	962	1075	06/92	09/92	01/93	100.0	11.7

22	CD/I/T	676	503	01/93	03/93	04/93	33.3	-25.6
23	CD/I/T	420	790	01/93	04/93	06/93	50.0	88.1
24	REQ	464	529	02/93	06/93	07/93	20.0	14.0
25	CD/I/T	519	530	03/93	07/93	08/93	20.0	2.1
26	CD/I/T	319	309	04/93	06/93	06/93	0.0	-3.1
27	REQ	220	685	04/93	06/93	08/93	66.7	211.4
28	PD	755	971	05/93	08/93	09/93	25.0	28.6
29	PD	844	1201	06/93	08/93	09/93	33.3	42.3
30	CD/I/T	410	686	08/93	10/93	01/94	100.0	67.3
31	REQ	302	293	10/93	02/94	02/94	0.0	-3.0
32	REQ	448	1196	11/93	06/94	08/94	25.0	167.0
33	REQ	83	167	11/93	12/93	01/94	50.0	101.2
34	PD	265	110	01/94	03/94	03/94	0.0	-58.5
35	PD	183	123	01/94	03/94	03/94	0.0	-32.8
36	CD/I/T	867	928	03/94	05/94	04/95	366.7	7.0
37	CD/I/T	864	1456	03/94	07/94	10/94	60.0	68.5
38	CD/I/T	415	785	04/94	06/94	12/94	200.0	89.2
39	CD/I/T	2168	4353	08/94	03/95	07/95	50.0	100.8
40	CD/I/T	4242	4600	01/95	10/95	11/95	10.0	8.4
41	REQ	572	731	03/95	06/95	07/95	25.0	27.8
42	REQ/PD	1069	1081	03/95	08/95	08/95	0.0	1.1
43	REQ	254	251	03/95	05/95	06/95	33.3	-1.2
44	IN/CK	141	80	06/95	08/95	11/95	100.0	-43.3
45	PD	408	419	08/95	11/95	03/96	100.0	2.7
46	CD/I/T	2248	1318	08/95	03/96	01/96	-25.0	-41.4
47	CD/I/T	443	293	03/96	05/96	05/96	0.0	-33.9
48	REQ/P	242	200	04/96	05/96	06/96	50.0	-17.4

	D							
49	CD/I/T	1639	2050	05/96	09/96	09/96	0.0	25.1
50	REQ/P D	432.7	494.3	06/96	09/96	10/96	25.0	14.2
51	CD/I/T	749	746	07/96	08/96	08/96	0.0	-0.4
52	CD/I/T	116	90	07/96	08/96	08/96	0.0	-22.4
53	REQ	210	330	07/96	08/96	10/96	100.0	57.1
54	PD	708	773	08/96	10/96	10/96	0.0	9.2
55	CE	118	116.5	05/97	06/97	07/97	50.0	-1.3
56	CDIT	2910	2994.6	11/96	02/97	03/97	25.0	2.9
57	REQ/P D	1320.6	1427.5	03/97	06/97	08/97	50.0	8.1
58	CD/I/T	2696.1	1812.6	07/97	09/97	09/97	0.0	-32.8
59	CD/IT		669.3	01/97	05/97	06/97	20.0	
60	REQ	520	578.8	07/97	09/97	10/97	33.3	11.3
61	CD/I/T	225.3	303	10/97	12/97	12/97	0.0	34.5
62	PD	862.3	919.7	09/96	02/97	02/97	0.0	6.7
63	REQ	908	626.8	07/97	12/97	11/97	-16.7	-31.0