

**TSP (Team Software Process)****Jorawar Singh**

Computer Science And Engineering Deptt, Dronacharya College Of
Engineering, Gurgaon, Haryana, India

Mukund Sharma

Computer Science And Engineering Deptt, Dronacharya College Of
Engineering, Gurgaon, Haryana, India

Smita Srivastava

Information And Technology Deptt, Dronacharya College Of Engineering, Gurgaon,
Haryana, India

Bibhuti Bhusan

Computer Science And Engineering Deptt, Dronacharya College Of
Engineering, Gurgaon, Haryana, India

Abstract:

As quoted by the CIO journal December 2003 issue, “By the numbers available, the software quality stinks.” The Standish Group reported in 1999 that 74% of all projects were not successful.

According to a survey by Standish Group in 2002, only 34 % of the software development is successful. Around 38 billion US dollars are lost every annum due to software failure and one of the major reasons for this high failure rate is poor software quality. Typical software projects are often late, over budget, of poor quality, and difficult to track. Engineers often have unrealistic schedules dictated to them and are kept in the dark as to the business objectives and customer needs. They are required to use imposed processes, tools, and standards, and often take shortcuts to meet schedule pressures. Very few teams can consistently be successful in this environment. As software systems get larger and more complex, these problems only get worse. The best projects are an artful balance of conflicting forces. To balance these conflicting forces, teams must understand the complete context for their projects.

1.Introduction

As quoted by the CIO journal December 2003 issue, “By the numbers available, the software quality stinks.” The Standish Group reported in 1999 that 74% of all projects were not successful.⁶

According to a survey by Standish Group in 2002, only 34 % of the software development is successful. Around 38 billion US dollars are lost every annum due to software failure ^[2] and one of the major reasons for this high failure rate is poor software quality. Typical software projects are often late, over budget, of poor quality, and difficult to track. Engineers often have unrealistic schedules dictated to them and are kept in the dark as to the business objectives and customer needs. They are required to use imposed processes, tools, and standards, and often take shortcuts to meet schedule pressures. Very few teams can consistently be successful in this environment. As software systems get larger and more complex, these problems only get worse. The best projects are an artful balance of conflicting forces. To balance these conflicting forces, teams must understand the complete context for their projects.

The field of Software Engineering has studied many techniques that build software engineering skills to create better quality software. However, there has been little attention paid to the impacts of individual discipline in software development. This report studies Team Software Process, a technique to use the team skills to build better software. Software developed in this manner has extremely high quality and software projects that use disciplined methods are completed on or ahead of schedule.

The purpose of this report is to provide updated results on the use of the TSP. The report starts with an overview of the TSP to provide a context for the results reported. This is followed by a detailed description of the experiences of and benefits realized by a first-time TSP team. The main purpose of this report is to illustrate how the TSP creates an environment where skilled engineers can apply disciplined methods working on a cohesive and dedicated team. Next, I summarize the performance of more than 20 projects from 13 organizations that have used the TSP. The report concludes with a recap of the advantages that this technique could have for our organization.

⁶Webb David, Humphrey Watts S., “Using the TSP on the TaskView Project,” CrossTalk

2 Wilson Kay. Class Notes.

2.What Is TSP ?

“It is so revolutionary that I remember the exact date I was introduced to It.” – Girish Shesagiri, CEO, AIS

To have a high performance organization, you need to have high performance teams. Team Software Process, TSP as it is addressed, is designed to facilitate superior performance of the software teams. It is the next step in the quality revolution, begun more than 10 years ago by Watts Humphrey. Dave Webb, software development and integration section chief of Hill Air Force Base that managed the first TSP project, has defined TSP as “ good project management technique coupled with good personal data.”^[1].

“...[TSP is a] transparent project management paradigm—everybody has a common understanding of the plan and everyone knows what is going on in the project and where we are in the project at any time.” [Humphrey 96]

The TSP process is defined in a series of process scripts that describe all aspects of project planning and product development. The process includes team role definitions, defined measures, and the postmortem process. As will be described, the TSP is an instance of the Capability Maturity Model, (CMM) Maturity Level 5, defined for teams.

While CMM sets the standards for software quality, TSP goes a step ahead and tells you how to achieve those standards by putting down specific guidelines on how PSP-trained engineers can work as effective team members on a high performance team. CMM, PSP and TSP combined together form a winning combination for the organization producing quality software on schedule.

TSP concentrates on the people aspect of the software triangle. It emphasizes the fact that “ Together Everyone Achieves More”. TSP is a relatively new concept in the industry and till now only around 15 industrial projects have actually used it but the results are astounding winning converts in US department of Defense contract projects as well as many commercial projects.

The TSP technique handles the project “the way you eat an elephant-one bite at a time “^[1]. The projects are estimated in the top down approach and overall size and team efficiency is used to determine the overall project schedule. The entire schedule is broken into phases, which in turn are broken down into smaller tasks. Each phase is thoroughly estimated by individuals working on the different tasks using the bottom up approach. After the completion of every phase, there is a project re-launch in which all the tasks of the current phase are defined and each task is estimated using the PSP

techniques. Thus, TSP keeps a rigorous check on the software development from the beginning till the end so that small problems are tamed and solved before they become major issues.

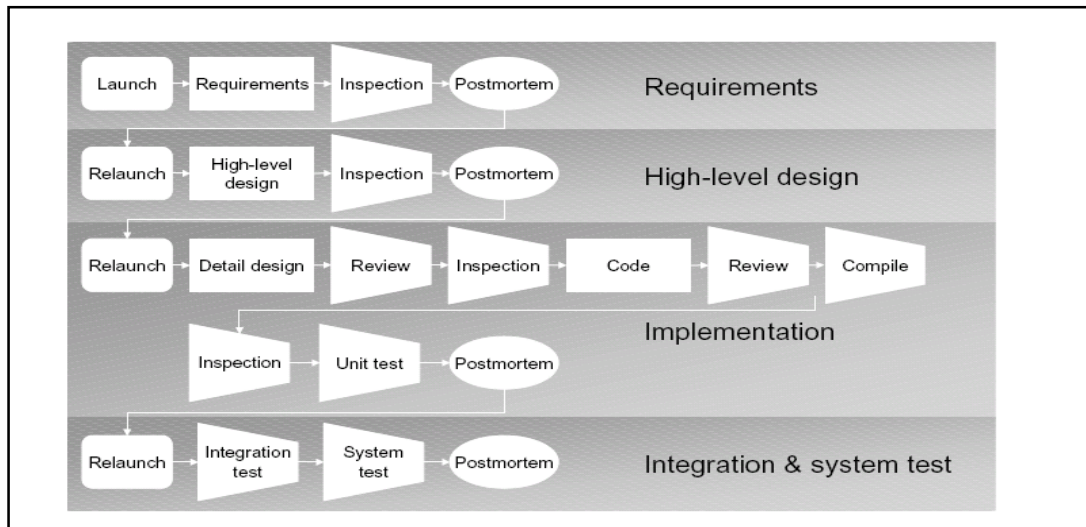


Figure 1: TSP structure Crosstalk, February 1998[3]

Figure 1 represents a standard set of phases and activities that are prescribed for the TSP. As shown in the figure, the TSP divides the software project into four typical phases of a project. Before every phase, the team goes through a complete launch or relaunch. Generally, when the team is PSP trained, a three-day launch workshop prepares them to go through entire project phase. The team then needs just two day long re-launch camps to kick off the second and subsequent phases. These launches are not training but an integral part of the project.

2. The TSP Launch Process

To start TSP process, the team goes through a launch process, which comprises of the following steps [1] :-

- Comprehensive review of the project goals in synchronization with the management.
- Defining team goals and team personnel responsibilities for each phase in the project.
- Merger of individual plans into the team plans.
- Definition of the team development plan and overall development strategy.
- Drafting of a quality plan and setting up of quality standards.

- Preparation of overall schedule
- Assessment of project risks and risk handling techniques.

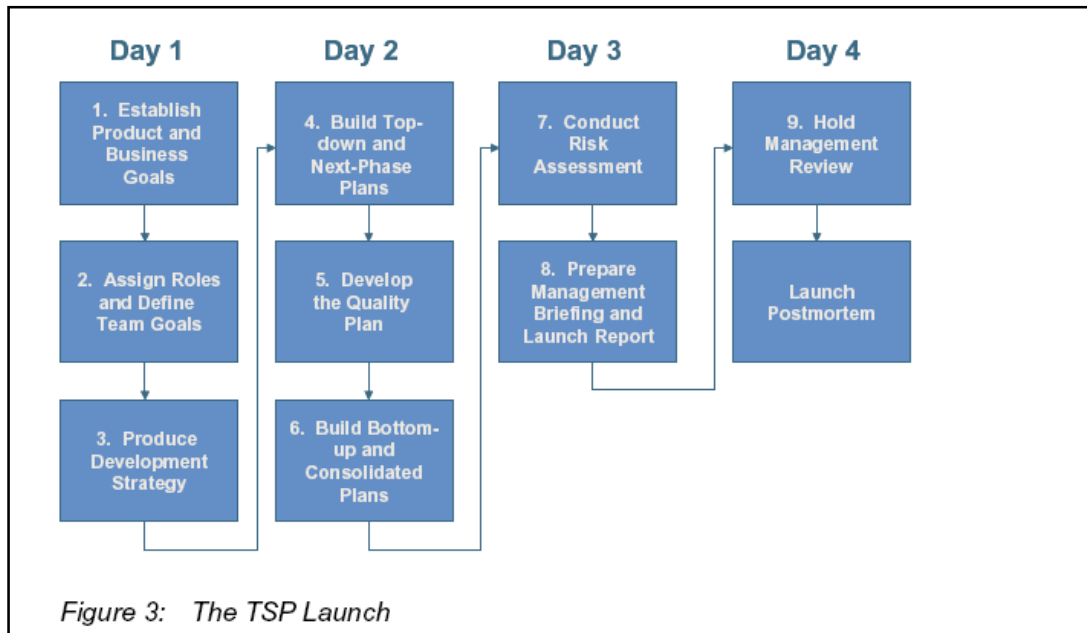


Figure 3: The TSP Launch

Figure 2: The report of result of TSP published by SEI [5]

The TSP launch consists of 9 meetings over a period of 4 days to analyze and understand the project completely. Thus, the team has a clear and concrete view of the entire project by the end of the launch phase and so they have documented support of what the project expects them to perform. Once the project begins, the teams conduct weekly meetings and periodically report their progress and status to management and customers. The TSP launch provides a team with all of the conditions necessary to become a self-directed team.

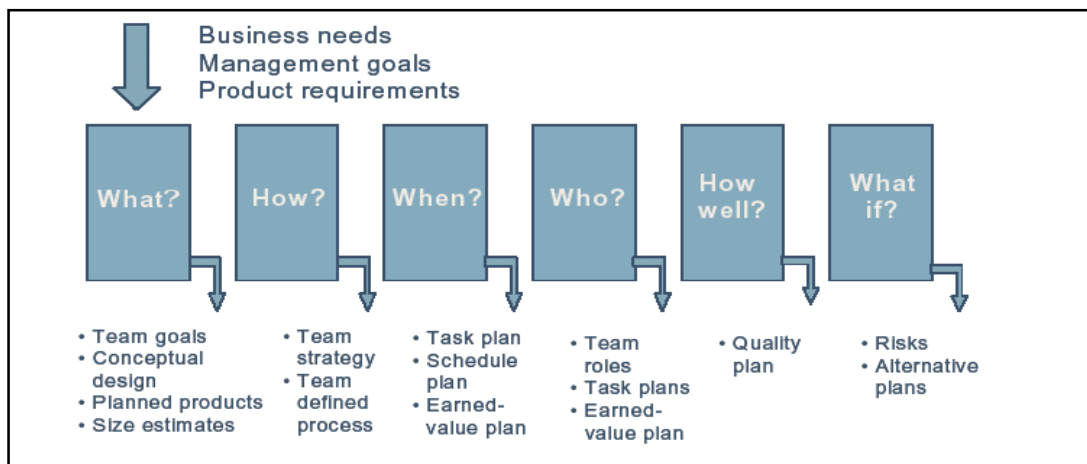


Figure 3: TSP launches Process deliverables [5]

The current TSP uses 23 scripts, 14 forms and three standards. The TSP scripts define 173 launches and development steps, each step describing in detail how engineers can do what they are supposed to do [3].

3. Why Is TSP Used?

The TSP has five main objectives [3]: -

- Building self-directing teams that plan and track their own work.
- Show managers how to coach and manage their team and help them sustain peak performance.
- Accelerate software process improvement by making CMM level 5 behaviors normal and expected.
- Provide improvement guidance to high-maturity organizations
- Facilitate university teaching of industrial-grade team skills.

“The principal benefit of TSP is that it shows engineers how to produce quality products for planned costs and on aggressive schedule by managing their work and owning their own processes and plans.”³

Thus, TSP formulates guidelines for the engineers explicitly laying down the path to successful software. The TSP improves schedule and quality management by creating an environment where individuals and teams can routinely do excellent work.

The TSP guides self-directed teams in addressing critical business needs of better cost and schedule management, effective quality management, and cycle-time reduction. It defines a whole product framework of customizable processes and an introduction strategy that includes building management sponsorship, training for managers and engineers, coaching, mentoring, and automated tool support. The TSP can be used for all aspects of software development: requirements elicitation and definition, design, implementation, test, and maintenance. The TSP can support multidisciplinary teams that range in size from two engineers to over a hundred engineers. It can be used to develop various kinds of products, ranging from real-time embedded control systems to commercial desktop client-server applications. [5].

4. TSP Improves Quality

One team at Embry Riddle Aeronautical University (ERAU) removed over 99 percent of development defects before system test entry. Their defect-removal profile is shown in Figure 2. [3]

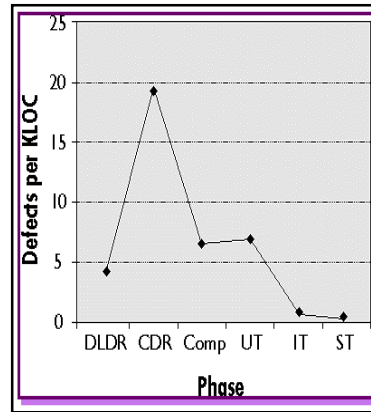


Figure 4: Defects per thousand lines of code (KLOC) removed by phase. [3]

The TSP technique emphasizes collection of data at personal level. All this data can be analyzed in each re-launch phase. Thus, engineers can determine which components are likely to have errors before they start system integration. By re-working the defect-prone components before test-entry, they can save a subsequent amount of test time and produce higher-quality products.

Measure	TSP Projects Average Range	Typical Projects Average
System test defects (defects/KLOC)	0.4 0 to 0.9	15
Delivered defects (defects/KLOC)	0.06 0 to 0.2	7.5
System test effort (% of total effort)	4% 2% to 7%	40%
System test schedule (% of total duration)	18% 8% to 25%	40%
Duration of system test (days/KLOC)	0.5 0.2 to 0.8	NA ⁷
Failure COQ	17% 4% to 38%	50%

Table 2: system defects source

5.TSP Makes The Teams Work Professionally

“Successful team-building programs typically expose a group to a challenging situation that requires cooperative behavior of the entire group [Morgan 93].

Perhaps the most important consequence of TSP is the way it helps the teams to manage their working environment. Each member of the team is included and empowered. It helps the teams to combat the unreasonable schedule pressure and thus, make useful plans. TSP provides data to the team that can be used to draft aggressive but realistic plans which can be then re-negotiated with the management.

6. But My Company Is CMM Certified!!!

The intent of CMM is for some key practices to be addressed at the organizational level, some at project level and some at both the levels. CMM has 18 key process areas and TSP guides engineers in addressing all of them. Thus, PSP, TSP and CMM provide an integrated three-dimensional framework for the process improvement. [3]

The TSP represents an operational process for teams that may be used as a strategy for implementing the CMM framework on teams. The TSP as defined addresses nearly all of the key process areas (KPAs) in the CMM framework. Table illustrates the key process areas addressed by the TSP.

Level	Focus	Key Process Area	TSP
5 Optimizing	Continuous Process Improvement	Defect Prevention	X
		Technology Change Management	X
		Process Change Management	X
4 Managed	Product and Process Quality	Quantitative Process Management	X
		Software Quality Management	X
3 Defined	Engineering Process	Organization Process Focus	X
		Organization Process Definition	X
		Training Program	
		Integrated Software Management	X
		Software Product Engineering	X
		Intergroup Coordination	X
		Peer Reviews	X
2 Repeatable	Project Management	Requirements Management	X
		Software Project Planning	X
		Software Project Tracking	X
		Software Quality Assurance	X
		Software Configuration Management	X
		Software Subcontract Management	

Table 2: TSP coverage of CMM key process areas. [3]

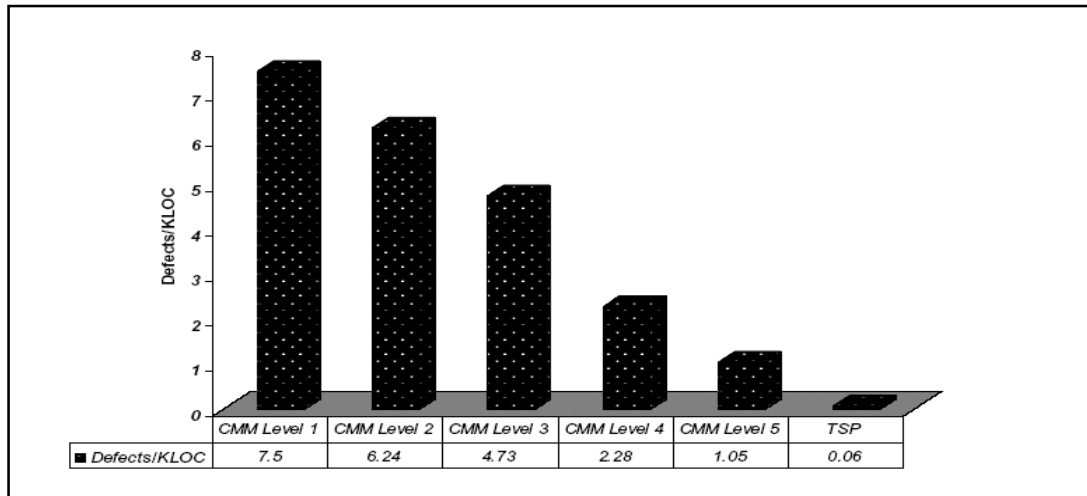


Figure 5

7. Average Defect Density of Delivered Software source [5]

Figure shows the quality of delivered software classified by CMM Level [Jones 00], compared to the TSP teams presented in this report. These data show that TSP teams produced software an order of magnitude higher in quality than projects from organizations rated at CMM Level 5.

PSP-trained engineers need help in applying the advanced process methods to their project. The TSP guides these teams in launching their projects and in planning and managing their work. The most important part is that TSP shows managers how to guide and coach their software teams to consistently perform at their best.

8. Overhead, overhead, overhead!!!!

“O yeah! You actually get your money back after 1200 LOC.” [Humphrey 96].

Launches and re-launches	5.0%
Post-mortems	1.5%
Weekly team meetings	2.5%
Time logging	2.5%
Defect logging	1.0%
Role manager tasks	5.0%
Total	17.5%

Table 3

Table of overheads introduced by TSP in the project. [8] It is true that TSP induces overheads in the project, but all those overheads are justified. Organizations create entire staffs to plan or to evaluate effectiveness, or to track time and defects, or to perform many of the other functions questioned above. While having such *overhead* positions may indeed be necessary in a particular organization, the TSP overhead model features the people who are doing the work also doing that little bit extra that actually helps them do the work more effectively and efficiently. TSP overhead performs necessary project functions regardless of the size of the organization, and can also help the people responsible for planning, evaluating, and tracking do their own jobs more effectively by freeing them to deal with the cross-project and other organizational issues that they are intended to address.

A project team managing and measuring its own work against its own commitments, and getting results like those previously cited, seems to have no need to apologize for its practices or to justify what percentage of the time they take to execute.

9.Risks Associated

Not all TSP implementations are successful. Unfortunately, the SEI has not received data from a team that has failed. They have data from projects that were cancelled midstream or where teams were told to stop using the TSP when management sponsorship was lost. The comments in this section show that the biggest risks with the TSP are the same risks associated with technology transition in general: lack of management sponsorship, introduction costs, and resistance to change.

10.Does TSP Really Works?

“One project increased its delivered quality by 10 times and reduced its effort by 20 percent compared to a previous project.”- Hill Air Force Base quality management. The data summarized in this section come from all TSP presentations developed for the

- Software Engineering Process Group (SEPG) conferences (<http://www.sei.cmu.edu/sepg>) and the SEI Software Engineering Symposiums for the years 2001 through 2003 [Ciurczak 02, Davis 01, Janiszewski 01, Narayanan 02, Pracchia 03, Riall 02, Serrano 03, Schwalb 03, Sheshagiri 02, Webb 02]. The data presented here represent thirteen

organizations and over twenty projects from these organizations. Some organizations presented summary data from more than one project without specifying the number of projects, so the exact number of projects could not be determined.

- ABB, Inc.
- Advanced Information Services
- Bechtel
- Cognizant Technology Solutions
- Electronic Brokering Services (EBS) Dealing Resources, Inc.
- Hill Air Force Base
- Honeywell
- Microsoft Corporation
- Naval Air Warfare Center
- Quarksoft, S.C.
- SDRC
- United Defense, LP
- Xerox

The first published results of TSP indicate that it does (Using the TSP on the TaskView Project, D. Webb and W. S., CrossTalk, February 1999, pp. 3-10). [6] TaskView is a product that automates portions of the Air Force's flight planning process. TaskView was recently rehosted with added functionality by a team of engineers at the Ogden Air Logistics Center, Hill Air Force Base, Utah. Almost 27,000 lines of new or changed code were produced in a project time of eight months. Quantitatively, the TaskView project completed one-month ahead of its original committed date with 55% added functionality without cost overruns (engineering effort was 2.5% under what had been planned). The PSP/TSP focus on quality produced a very high quality product, which resulted in a 20% schedule saving during CPT&E and system testing. [4].

The first organization to report their result after using both TSP and PSP was Teradyne. Their return of investment, ROI, analysis indicated that by using TSP and PSP on 2 projects totaling 112 KLOC, they have benefited by approximately \$5.3 million to this date. [5]

The report of TSP results published by Noopur Davis and Julia Mullaney in September 2003 for the Software Engineering Process Management Program contains a summary of data from 20 TSP projects in 13 organizations, and stories from people who have used the TSP. These TSP teams delivered their products an average of 6% later than they had planned. The schedule error for these teams ranged from 20% earlier than planned to 27% later than planned. This compares favorably with industry data that show over half of all software projects were more than 100% late or were cancelled. These TSP teams also improved their productivity by an average of 78%. The teams met their schedules while producing products that had 10 to 100 times fewer defects than typical software products. They delivered software products with average quality levels of 5.2 sigma, or 60 defects per million parts (lines of code). In several instances, the products delivered were defect free. [5]

Measure	TSP Projects Results 2000 ⁸ <i>Average</i> <i>Range</i>	TSP Projects Results 2003 ⁹ <i>Average</i> <i>Range</i>
Effort error	-4% -25% to +25%	26% 5% to 65%
Schedule error	5% -8% to +20%	6% -20% to 27%
System test defects (defects/KLOC)	NA 0 to 0.9	0.4 0 to 0.9
Acceptance test/released defects (defects/KLOC)	NA 0 to 0.35	0.06 0 to 0.2
Duration for system test (days/KLOC)	NA 0.1 to 1.1	0.5 0.2 to 0.8

Table 4: Results Comparison Between 2000 and 2003 [5]

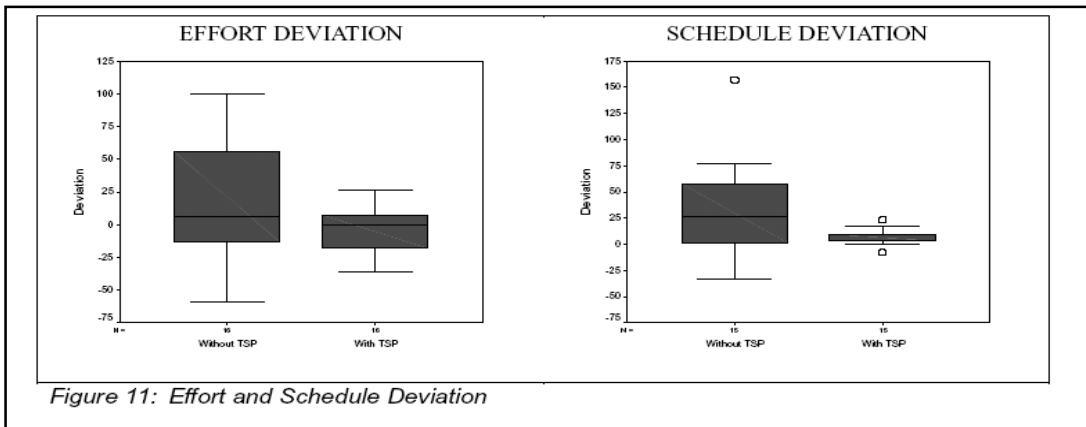
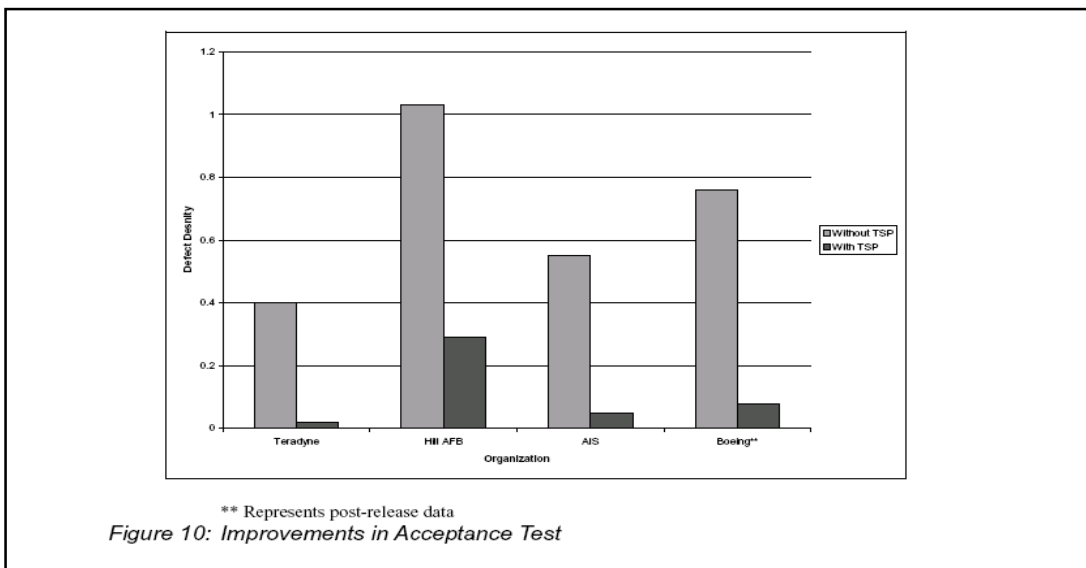
The most recently published example of the benefits of using the TSP is from a Honeywell presentation at the 2002 Software Engineering Process Group conference [5]. Pavlik and Riall claim a better-than-70-percent software productivity increase from one release of an avionics control system to the next, with a total of 22 percent savings in total systems and software effort.

While their delivery was on time, even more significant is that the quality of their delivered product was 10 times better than the previous release, while the delivered functionality was three times what was originally planned. I would say that a 17.5 percent process overhead for the TSP was more than worth it to Honeywell.

At the same conference, a presentation by John Ciurczak of EBS Dealing Resources claimed a "37.5 percent reduction in execution stage cycle"

Boeing, AIS and Hill Air Force Base have reported receiving significant benefits from TSP and PSP use. The following charts are compilation of result data from Teradyne, Boeing, AIS and Hill Air Force Base.

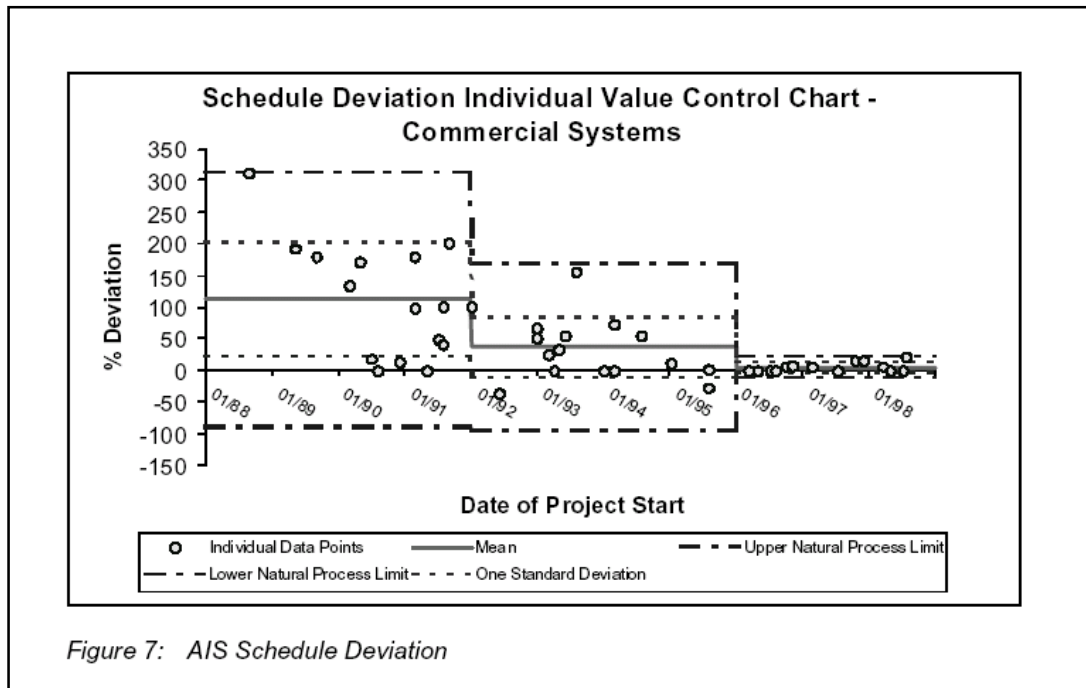
Figure source [5]



Besides these organizations, some organizations like AIS have successfully developed and implemented their own TSP-like processes. AIS's work with disciplined engineering methods pre-dates the availability of the TSP in its current form, so AIS developed its

own team process that uses the PSP and is consistent with the TSP. All work currently done by AIS software engineers uses these disciplined methods.

Figure source [5]



THE AIS began implementing CMM in their organization in 1992. Although they saw some improvements, they were still not very good at achieving their business objectives of being able to deliver high quality software on time and within budget consistently. [Jagdish04]

They began exploring PSP when it was made available commercially in 1995 by institutionalizing PSP and incorporating it into their organization's standard CMM process in 1996. As a result, they started some significant improvements in achieving their business objectives.

However, in using CMM and PSP together, they realized that there were some gaps between the two. Hence, they plugged these gaps through their own process improvement ideas.

These gaps are what you see today in the form of TSP. When TSP was formally made available to commercial organizations, they integrated the formal TSP into our standard process. [Jagdish04]

After the implementation of the PSP approach, two things enabled AIS to improve schedule predictability even further. Jagadish Kamatkar of AIS reports from personal experience that individuals' schedules for their components are more accurate, and there is more commitment to those schedules. The quality of work is reportedly better, and the corresponding reduction in rework helps engineers meet planned schedules.

11. What Does TSP Have For Our Company?

We have to recognize that controlling software processes significantly affects our ability to be successful in business. We seem to know what we want our teams to be doing, but we struggle with how to do it. The Team Software Process (TSP) is designed to provide both a strategy and a set of operational procedures for using disciplined software process methods at the individual and team levels.

Table source [5]

Table 5: Summary Results of Using the TSP

Category	Without TSP	With TSP
Effort deviation (% average, range)	16% -60% to +100%	-4% -36% to +26%
Schedule deviation (% average, range)	33% -33% to +157%	6% -8% to 23%
System test defect density (de- fect/KLOC)	1 to 8	0.1 to 1.1
Acceptance test and release de- fect density (defects/KLOC)	0.2 to 1.9	0 to 0.35
Duration of system test (days/KLOC)	1 to 7.7	0.1 to 1.1

Ours is a developing company. People like doing excellent work and TSP is providing them with a chance to do so. TSP would introduce the concept of synergy in the organization where the team would be there to support the individuals throughout the project. The success and failures would be collectively owned and yet, there would scope for individuals to set up their goals, achieve them and excel, thus providing them a constant motivation for performing quality work. Each individual of the team would be collecting data of his own performance and thus, would be competing against his own set of guidelines and performance standards.

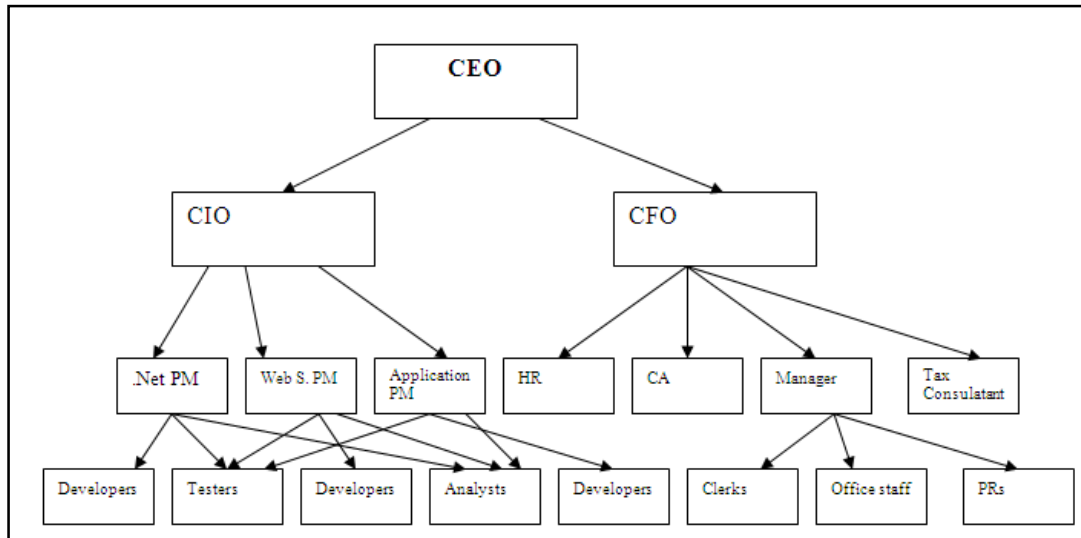
The TSP teams would be better able to manage changing customer needs. When a customer asks for additional functionality, the team would be able to assess immediately and accurately how this change affects the detailed plan. The team can make adjustments to its plans, if necessary, to accommodate the customer request. The team can also judge whether the change indicates a change in cost or schedule, so that the

Change can be negotiated with the customer. Without detailed plans, teams are often forced to take on additional work, without relief on either cost or schedule.

Above all, TSP would help our organization produce high quality software in scheduled time thus excelling in the field of software development and making a mark as a company.

12. Company Assumptions

The company Infobeans Inc. is a small software development company mostly handling the overseas projects, which are outsourced. The company structure is as follows :-



Since Infobeans' differentiator is quality software development, the company decided to base its operations on the Capability Maturity Model for Software and chose the Personal Software Process as the means for implementing the Capability Maturity Model. Infobeans staff is organized into a management team that runs the company, a group of software engineers that develop the software, and a small administrative staff that attends to administrative and operational issues. The total number of employees for this company is 50. The company works in the latest technologies like .Net and web services and it has 15 PSP certified professionals. It is on CMM level 2 and is aspiring to achieve CMM level 3. The annual turn over of the company would be in range of 500,000 dollars.

I have assumed that I work as a software analyst in the company and I am making the proposal for adoption of TSP to the CEO and CIO of the company.

13. Foot note

1. ¹Webb David R., Managing Risk with TSP. CrossTalk
2. ²Wilson Kay. Class Notes
3. ³ Watts S. Humphrey, “Three Dimensions of Process Improvement Part III: The Team Process,” CrossTalk,
4. ⁴ A Review of Personal Software Process (PSP)SM and Team Software
5. ⁵ Davis Noopur and Mullaney Julia Team Software Process (TSP) in Practice: A Summary of Recent Results
6. ⁶ Webb David, Humphrey Watts S., “Using the TSP on the TaskView Project,” CrossTalk
7. ⁸ McHale Jim, TSP: Process Costs and Benefits.,” CrossTalk
 - a. Process (TSP)SM <http://www.stsc.hill.af.mil/>.

14.Reference

1. Webb David R., Managing Risk with TSP. CrossTalk, Software Technology Support Center, Hill Air Force Base, UT, April 1998, pp.14-17.
<http://www.stsc.hill.af.mil/crosstalk/2000/06/webb.html>

2. Wilson Kay. Class Notes.

The collection of the PowerPoint slides and notes provided by Kay Wilson, which covers the concepts of CMM, CMMI and also list some good resources for TSP and PSP understanding.

3. Watts S. Humphrey, "Three Dimensions of Process Improvement Part III: The Team Process," CrossTalk, Software Technology Support Center, Hill Air Force Base, UT, April 1998, pp.14-17.
<http://www.stsc.hill.af.mil/crosstalk/1998/apr/dimensions.html>

This three-part series describes SEI's software process improvement initiative and the common themes of CMM, PSP, and TSP. Part I describes the CMM, part II the PSP, and part III the TSP. This series is notable because of its high level overview of all three technologies and how they can be combined to meet an organization's needs.

4. A Review of Personal Software Process (PSP)SM and Team Software Process (TSP)SM www.stsc.hill.af.mil.

The paper first defines disciplined development. It then reviews what practices are implied by individual discipline and reviews a technology, Personal Software Process (PSP) that supports individual discipline. The report then discusses the issues involved in forming and maintaining development teams and the potential of teams made up of disciplined individuals. The Team Software Process (TSP), a technology that supports teams of PSP trained engineers, is then reviewed.

5. Team Software Process (TSP) in Practice: A Summary of Recent Results
Noopur Davis and Julia Mullaney

Technical Report CMU/SEI-2003-TR-014, September 2003

Abstract: Most software organizations are facing critical business needs for better cost and schedule management, effective quality management, and cycle-time reduction. The Team Software Process addresses these critical business needs. This report provides results and implementation data from projects and

individuals that have adopted the TSP. The results show that TSP teams are delivering essentially defect-free software on schedule, while improving productivity. These data can be used for benchmarking and planning, motivation, lessons learned, and other guidance to those currently using the TSP or considering its use. The report also illustrates adoption experiences of practitioners in the field, including TSP team members, their managers, and their coaches and instructors.

6. Webb David, Watts S. Humphrey, "Using the TSP on the TaskView Project," CrossTalk, Software Technology Support Center, Hill Air Force Base, UT, February 1999, pp. 3-10.

This paper reports the first results of using the Team Software Process (TSP) on a software-intensive system project. The TaskView team at Hill Air Force Base used the TSP to deliver the product a month ahead of its originally committed date for almost exactly the planned costs. Because the engineers' productivity was 123% higher than on their prior project, they included substantially more function than originally committed. Testing was completed in one-eighth the normal time and, to date, the customer has reported no acceptance test defects.

7. "CHAOS: A Recipe for Success. Project Resolution: The 5-Year View." The Standish Group International, Inc., 1999.
8. TSP: Process Costs and McHale Jim, Benefits.," CrossTalk, Software Technology Support Center, Hill Air Force Base, UT Sep 2002.

<http://www.stsc.hill.af.mil/crosstalk/2002/09/mchale.html>

The article analyses the overheads due to application of TSP in a software project. It also attempts to clarify the confusion between the necessary project tasks and overheads.

[Humphrey 96] Humphrey, W. The Personal Software ProcessSM (PSPSM) (CMU/SEI-1996-TR-022, ADA387268). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.

<http://www.sei.cmu.edu/publications/documents/00.reports/00tr022.html>

9. Jagdish Kamatkar, Advanced Information Systems, personal communication and e-mails.
10. Dr. Galen Crow, Extended University, Personal Communication.

