Software Engineering Institute | Carnegie Mellon

Team Software Process

A Performance Framework for Software Development

Are your software projects often late and over budget?

TSP provides a proven method that helps you plan, evaluate, manage, and control your work.

What is Team Software Process?

Team Software Process (TSP) guides engineering teams that are developing software-intensive products. Using TSP helps organizations establish a mature and disciplined engineering practice that produces secure, reliable software in less time and at lower costs.

TSP has been applied in small and large organizations in a variety of domains with similar results on first use, including

- productivity improvements of 25% or more
- reductions in cost and schedule variance to less than +/- 10%
- testing costs and schedule reductions of up to 80%

Why is TSP needed?

Software underlies advances in many areas, including medicine, science, weapons systems, and commerce. Software technology has advanced at an incredible pace, but the ability to manage software development and the quality of the software produced has not kept pace.

The software industry is the only high-tech industry that relies on testing to address quality and reliability issues. This is a strategy that cannot work, as evidenced by increases in serious system failures and software security and safety issues.

TSP reduces the number of post-release defects by 80% or more, so systems produced with TSP are more reliable and less likely to have latent defects that lead to these serious concerns.

TSP is an operational, "how-to" implementation of the principles and best practices that the SEI has advocated since its inception. TSP also includes many concepts that have not been implemented in other engineering methods, including

- self-directed team management
- an integrated measurement framework
- a comprehensive, pre-test quality management system
- a coaching model
- a team-focused strategy for rapid deployment
- a complete, operational process supported by training and tools

Why focus on teams?

The performance of large software projects often hinges on the performance of a team or even an individual. If they deliver late or provide a defective part, it can delay the entire project and impact the quality of the larger system. In software engineering, where so many system components are delivered late or are of poor quality, this is a pervasive problem.

The best strategy for improvement is to focus on individual and team performance. This strategy produces direct and measurable results, a faster return on investment, and provides a more flexible, tactical approach to improvement.

This is the essence of TSP, to improve performance from the bottom up, by working with individual developers and project teams.

Team Software Process

A Performance Framework for Software Development

What impact is TSP having on software development?

The SEI has published three technical reports on the results of using TSP. The first, published in 2000[1], summarized results from 12 projects in 4 organizations: Teradyne, Boeing, Hill AFB, and AIS. The second, published in 2003[2], summarized results from over 30 projects in 13 organizations. The third report, published in 2009[3], summarized results from Mexico's National TSP Initiative.

With very few exceptions, these reports and independent reports published in STSC's *Crosstalk* and *IEEE Software* show a consistent pattern of real, dramatic, quantitative benefits from using TSP.

Improvements in cost and schedule

Delivering projects on time and within budget is critically important. Studies of TSP projects showed that variance in cost and schedule improved from overruns mostly in the +20% to +70% range to a more balanced range of -20% to + 20%. For most projects the cost and schedule variance was less than 10%.

Improvements in productivity

A minimum 25% to 30% improvement in productivity was achieved on nearly all projects, and several projects reported much higher results.

Improvements in quality

The most significant improvements reported with TSP were improvements related to software quality. Most software development projects still rely heavily on testing to find and fix defects, leading to two problems. First, the cost of removing defects in test is one to two orders of magnitude greater than the cost of removing defects in formal review or inspection. Second, the effectiveness or yield of most testing processes is only around 50% or less. This means when large numbers of defects are found during testing, a large number of defects are also likely to escape into the released products.

TSP uses reviews and inspections for all artifacts produced by the team, guided by historical data on defect type and density. This reduces rework and lowers test costs and overall development costs by 30% or more.

TSP also reduces the number of post-release defects by 80%. The number of post-release defects per thousand lines of new or modified code is typically below 0.25, and many projects have reported zero defects found after user acceptance testing.

How do you get started with TSP?

The SEI provides the training and tools organizations need to implement TSP, and an implementation strategy designed to make TSP deployment faster and easier.

Implementation on a project-by-project basis is recommended, and training is the first step. A top-down approach is usually most effective, starting with training for management followed by training for engineers and those in support functions.

Organizations usually begin with the *TSP Executive Strategy Seminar*, a one-day class for top management. Next, a three-day session for project managers or team leaders: *Leading Development Teams*. Finally, a five-day class for software developers, *PSP Fundamentals*, and a three-day class for other team members, *TSP Team Member Training* are taken.

Once these courses have been completed, the project teams are ready to use TSP. This can usually be accomplished in about a month.

Once the projects are underway, the training and launching of additional projects and teams can begin. SEI also provides advanced training courses, *TSP Coach Training* and *PSP Instructor Training*, for organizations that want to develop the internal capability required to train and implement TSP on their own.

Who is using TSP?

A growing number of industry organizations are using TSP, including Microsoft, Oracle, Intuit, Adobe, EDS, FujiFilm, Toshiba, HitachiSoft, Softtek, and IBM. Government software organizations using TSP include NAVAIR, NAVOCEANO, Hill AFB, and JSOC.

For More Information

To learn more, please contact
Customer Relations
Phone: 412-268-5800
FAX: 412-268-6257
customer-relations@sei.cmu.edu
www.sei.cmu.edu/tsp

For Course Registration

www.sei.cmu.edu/products/courses/

[1] *The Team Software Process: An Overview and Preliminary Results of Using Disciplined Practices*, CMU/SEI-2000-TR-015.

[2] *The Team Software Process in Practice: A Summary of Recent Results*, CMU/SEI-2003-TR-014.

[3] *Deploying TSP on a National Scale: An Experience Report from Pilot Projects in Mexico*, CMU/SEI-2009-TR-011.

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.