



Three Dimensions of Process Improvement

Part III: The Team Process

Watts S. Humphrey
Software Engineering Institute

Part I of this article (CROSSTALK, February 1998) described the Capability Maturity Model (CMM)[®], and Part II (CROSSTALK, March 1998) addressed the Personal Software Process (PSP)SM. The CMM provides an overall framework that has helped many organizations improve their performance and the PSP shows engineers how to use process principles in doing their personal work. Part III describes the Team Software Process, which shows integrated product teams how to use these processes to consistently produce quality products on aggressive schedules and for their planned costs.

Building a Supportive Team Environment: The Team Software Process

The Team Software Process (TSP) extends and refines the CMM and PSP methods to guide engineers in their work on development and maintenance teams. It shows them how to build a self-directed team and how to perform as an effective team member. It also shows management how to guide and support these teams and how to maintain an environment that fosters high team performance. The TSP has five objectives:

- Build self-directed teams that plan and track their work, establish goals, and own their processes and plans. These can be pure software teams or integrated product teams (IPT) of three to about 20 engineers.
- Show managers how to coach and motivate their teams and how to help them sustain peak performance.
- Accelerate software process improvement by making CMM Level 5 behavior normal and expected.
- Provide improvement guidance to high-maturity organizations.
- Facilitate university teaching of industrial-grade team skills.

The SEI's work is supported by the Department of Defense. Capability Maturity Model and CMM are registered with the U.S. Patent and Trademark Office. Personal Software Process, PSP, Team Software Process, and TSP are service marks of Carnegie Mellon University.

The principal benefit of the TSP is that it shows engineers how to produce quality products for planned costs and on aggressive schedules. It does this by showing engineers how to manage their work and by making them owners of their plans and processes.

Team-Building Strategies Are Not Obvious

Generally, when a group of engineers starts a project, they get little or no guidance on how to proceed. If they are lucky, their manager or one or two of the experienced engineers will have worked on well-run teams and have some ideas on how to proceed. In most cases, however, the teams have to muddle through a host of issues on their own. Following are some of the questions every software team must address.

Figure 1. TSP structure.

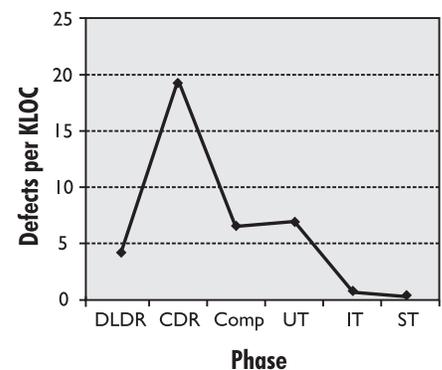
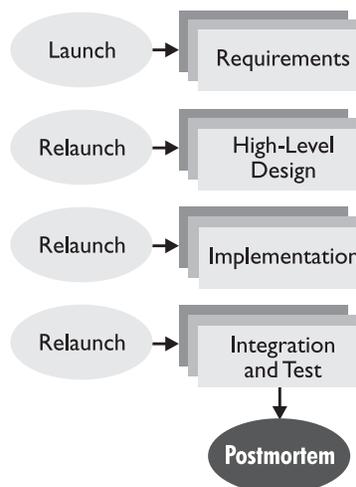


Figure 2. Defects per thousand lines of code (KLOC) removed by phase.

- What are our goals?
- What are the team roles and who will fill them?
- What are the responsibilities of these roles?
- How will the team make decisions and settle issues?
- What standards and procedures does the team need and how do we establish them?
- What are our quality objectives?
- How will we track quality performance, and what should we do if it falls short?
- What processes should we use to develop the product?
- What should be our development strategy?
- How should we produce the design?
- How should we integrate and test the product?
- How do we produce our development plan?

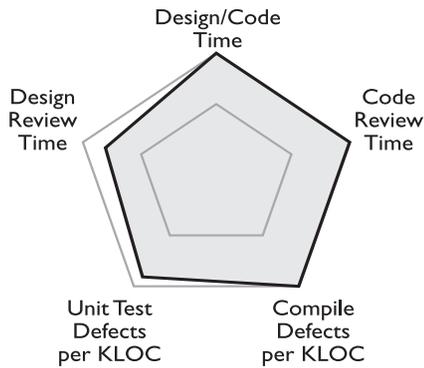


Figure 3. *Component 7 quality profile.*

- How can we minimize the development schedule?
- What do we do if our plan does not meet management's objectives?
- How do we assess, track, and manage project risks?
- How can we determine project status?
- How do we report status to management and the customer?

Most teams waste a great deal of time and creative energy struggling with these questions. This is unfortunate, since none of these questions is new and there are known and proven answers for every one.

The TSP Process

The TSP provides team projects with explicit guidance on how to accomplish their objectives. As shown in Figure 1, the TSP guides teams through the four typical phases of a project. These projects may start or end on any phase, or they can run from beginning to end. Before each phase, the team goes through a complete launch or relaunch, where they plan and organize their work. Generally, once team members are PSP trained, a three-day launch workshop provides enough guidance for the team to complete a full project phase. Teams then need a two-day relaunch workshop to kick off each of the second and each of the subsequent phases. These launches are not training; they are part of the project.

The current TSP version uses 23 scripts, 14 forms, and three standards. The TSP scripts define 173 launch and development steps. None of the steps is complex, but each is defined in enough

detail so the engineers can see how to do what they have to do. These scripts guide the teams through the steps of launching and running their projects.

The TSP Launch Process

To start a TSP project, the launch process script leads teams through the following steps.

- Review project objectives with management and agree on and document team goals.
- Establish team roles.
- Define the team's development process.
- Make a quality plan and set quality targets.
- Plan for the needed support facilities.
- Produce an overall development strategy.
- Make a development plan for the entire project.
- Make detailed plans for each engineer for the next phase.
- Merge the individual plans into a team plan.
- Rebalance team workload to achieve a minimum overall schedule.
- Assess project risks and assign tracking responsibility for each key risk.

In the final launch step, the team reviews their plans and the project's key risks with management. Once the project starts, the team conducts weekly team meetings and periodically reports their status to management and to the customer.

In the three-day launch workshop, TSP teams produce

- Written team goals.
- Defined team roles.
- A process development plan.
- The team quality plan.
- The project's support plan.
- An overall development plan and schedule.
- Detailed next-phase plans for each engineer.
- A project risk assessment.
- A project status report.

Early TSP Results

While the TSP is still in development and has only been used with 10 industrial and three student teams, the early

results are encouraging. 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.

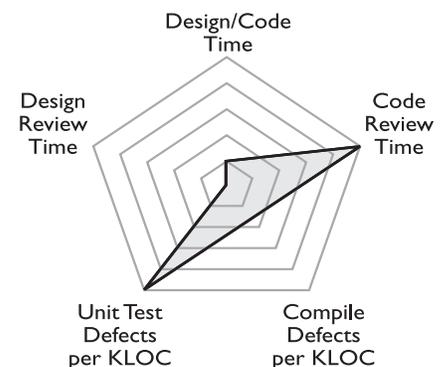
TSP teams also gather the data they need to analyze component quality before integration and system testing. This is done with the aid of the component quality profile, which shows five quality parameters in a bullseye format. With a profile that nearly fills the entire bullseye, as in Figure 3, quality is judged to be good. A profile like that in Figure 4, however, indicates likely problems. The five profile dimensions are shown in Table 1 and explained below.

The data for Figures 3 and 4 came from two of the ERAU team's components. Component 7 had no integration or system test defects, and Component 9 had one integration defect. As can be seen from Figure 4, the development work for Component 9 had inadequate design time, no design review time, and high compile defects. The only surprise is that this component had only one defect in integration test and none in system test.

The component quality profile is based on the following criteria.

- Design time is greater than 50 percent of coding time.
- Design review time is greater than 50 percent of design time.
- Code review time is greater than 50 percent of coding time.
- Compile defects are less than 10 per KLOC.
- Unit test defects are under five per KLOC.

Figure 4. *Component 9 quality profile.*



Dimension	Meaning
Design/Code Time	The ratio of detailed design time to coding time.
Code Review Time	The ratio of code review time to coding time.
Compile Defects/KLOC	The defects per KLOC found in compile.
Unit Test Defects/KLOC	The defects per KLOC found in unit test.
Design Review Time	The ratio of detailed design review time to detailed design time.

Table 1. Component quality profile dimensions.

When a factor meets or exceeds these criteria, that profile dimension is at one or on the edge of the bullseye. When the criteria are not met, say 25 percent design review time instead of 50 percent, that dimension value would be one-half or halfway to the center of the bullseye. Once teams gather enough of their own data, they should establish the profile criteria that work best for them.

With TSP data, engineers can determine which components are most likely to have defects before they start integration and system testing. By reworking these defect-prone components before test entry, they can save a substantial amount of test time and produce higher-quality products.

How the TSP Helps Teams Behave Professionally

Perhaps the most powerful consequence of the TSP is the way it helps teams manage their working environment. The most common problem product teams face is unreasonable schedule pressure. Although this is normal, it can also be destructive. When teams are forced to work to unreasonable schedules, they are unable to make useful plans. Every plan they produce misses the edicted schedule and is therefore unacceptable. As a result, they must work without the guidance of an orderly plan and will generally take much longer to complete the project than they otherwise would.

The TSP team's responsibility is to plan and produce a quality product as rapidly and effectively as they can. Conversely, it is management's responsibility to start projects in time to finish when needed. When similar projects have taken 18 months and management demands a nine-month schedule, this is clearly unrealistic. Where was management nine months ago when the project should have started? Although the business need may be real, the team's schedule is only part of the problem. Under these conditions, it is essential that management and the team work together to rationally determine the most effective course of action. This will often involve added resources, periodic replanning, or early attention to high-risk components.

While TSP teams must consider every rational means for accelerating their work, in the last analysis, they must defend their plan and resist edicts that they cannot devise a plan to meet. If management wants to change job scope, add resources, or suggest alternate approaches, the team will gladly develop a new plan. In the end, however, if the team cannot produce a plan to meet the desired schedule, they must not agree to the date. So far, most TSP teams have been able to do this. Teams have found that the TSP provides them con-

vincing data to demonstrate that their plans are aggressive but achievable.

The TSP Manager-Coach

Perhaps the most serious problem with complex and challenging work is maintaining the discipline to consistently perform at your best. In sports and the performing arts, for example, we have long recognized the need for skilled trainers, conductors, and directors. Their job is to motivate and guide the performers and also to insist that everyone meet high personal standards. Although skilled players are essential, it is the coaches who consistently produce winning teams. There are many differences between software and athletic or artistic groups, but they all share a common need for sustained high performance. This requires coaching and support.

Software managers have not traditionally acted as coaches, but this is their role in the TSP. The manager's job is to provide the resources, interface to higher management, and resolve issues. But most important, the manager must motivate the team and maintain a relentless focus on quality and excellence.

Table 2. PSP and TSP coverage of CMM key process areas.

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

This requires daily interaction with the team and an absolute requirement that the process be followed, the data gathered, and the results analyzed. With these data, the manager and the team meet regularly to review their performance and to ensure their work meets their standards of excellence.

Conclusion

The CMM, PSP, and TSP provide an integrated three-dimensional framework for process improvement. As shown in Table 2, the CMM has 18 key process areas, and the PSP and TSP guide engineers in addressing almost all of them. These methods not only help engineers be more effective but also provide the in-depth understanding needed to accelerate organizational process improvement.

The CMM was originally developed to help the Department of Defense (DoD) identify competent software contractors. It has provided a useful framework for organizational assessment and a powerful stimulus for process improvement even beyond the DoD. The experiences of many organizations show that the CMM is effective in helping software organizations improve their performance.

Once groups have started process improvement and are on their way toward CMM Level 2, the PSP shows engineers how to address their tasks in a professional way. Although relatively new, the PSP has already shown its potential to improve engineers' ability to plan and track their work and to produce quality products.

Once engineering teams are PSP trained, they generally need help in applying advanced process methods to their projects. The TSP guides these teams in launching their projects and in planning and managing their work. Perhaps most important, the TSP shows managers how to guide and coach their software teams to consistently perform at their best. ♦

Acknowledgments

I thank all the many people who have participated in this work. I especially thank Jim Over for his invaluable support and assistance with both the PSP and TSP work and this series of articles. I also thank Linda Parker Gates, Tom Hilburn, Alan Koch, Mike Konrad, Mark Paulk, Bill Peterson, and Dave Zubrow for their many helpful comments and suggestions.

About the Author



Watts S. Humphrey is a fellow at the Software Engineering Institute (SEI) of Carnegie Mellon University, which he joined in 1986. At the SEI, he established the Process Program, led initial development of the CMM, introduced the concepts of Software Process Assessment and Software Capability Evaluation, and most recently, the PSP and TSP. Prior to joining the SEI, he spent 27 years with IBM in various technical executive positions, including management of all IBM commercial software development and director of programming quality and process.

He has master's degrees in physics from the Illinois Institute of Technology and in business administration from the University of Chicago. He is the 1993 recipient of the American Institute of Aeronautics and Astronautics Software Engineering Award. His most recent books include *Managing the Software Process* (1989), *A Discipline for Software Engineering* (1995), *Managing Technical People* (1996), and *Introduction to the Personal Software Process* (1997).

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Voice: 412-268-6379
E-mail: watts@sei.cmu.edu.

Telos® Corporation Achieves CMM Level 4

Telos Corporation recently achieved Software Engineering Institute (SEI) Capability Maturity Model (CMM) Level 4 rating for software engineering process maturity in support of the U.S. Army Software Engineering Center (SEC) in Lawton, Okla. Their Level 4 rating places Telos in the top 1.5 percent of the more than 600 CMM-appraised organizations in the software industry.

This achievement is the culmination of a six-year effort on the part of Telos and the Communications Electronics Command Software Engineering Center. The appraisal was conducted by a team consisting of personnel from Telos, SEC Fire Support Software Engineering, the SEI, and Lockheed Martin.

Telos has actively followed SEI guidelines since 1990 because they contribute to increased product quality, improved software performance, more predictable development schedules, and reduced system lifecycle costs. As part

of its software process improvement, Telos implemented a comprehensive development environment with a standardized software design method and formed integrated project teams comprised of software engineers, system analysts, programmers, and test and training personnel.

Telos Corporation specializes in network-based solutions for governments and industry worldwide. The company's Fort Sill, Okla. office has fielded nearly 100 major fire support system software versions and now maintains nearly 9 million lines of tactical and support systems code and thousands of pages of documentation. Telos also provides data integration, network security, network and systems integration, and unique products including wireless networks, training and simulation systems, and message handling systems. Telos headquarters are in northern Virginia's Netplex area.

Internet: <http://www.telos.com>