TSP Can Be the Building Blocks for CMMI

Alan S. Koch
ASK Process, Inc.

Your organization has a mandate to achieve Capability Maturity Model® Integration (CMMI®) Level 3. Why would you even consider adding the Team Software Process^{SM} (TSP^{SM}) to your plate when it is already overflowing? In this article, I will discuss how TSP – far from adding work to a CMMI initiative – can potentially reduce the time and effort that will be required to achieve your goals. Simultaneously, TSP will engage your engineers in disciplined processes, giving them an appreciation for good processes along with the desire to adopt improved processes in every area of the organization.

Your organization has a mandate to achieve Capability Maturity Model® (CMM®) Integration (CMMI®) Level 3. It is going to be a long road. There are processes to define, documents to write, people to train, and evidence to collect. With so much to do and so many people involved, why would you even consider the Team Software Process^{SM} (TSP^{SM})? Why add something else to your plate when it is already overflowing with work? Most organizations adopt the TSP to achieve significant improvements in product quality, to reduce development time, and to get more accurate project estimates. Too many organizations believe they must choose between those benefits and the recognized organizational maturity level or capability level rating provided by the CMMI.

In this article, I will discuss how the TSP can potentially reduce the time and effort required to achieve your CMMI goals, eliminating the need to choose between two sets of laudable objectives.

If you are just starting out on a CMMI initiative, the TSP can help you to bootstrap your process definition activities. TSP does this by providing starting points for many of your new processes, as well as training your engineers to be capable and productive at defining the processes they use. Naturally, the more progress you have already made in advancing the CMMI, the less this bootstrapping will help. But even if you are well on your way to your CMMI goals, the TSP can make the road smoother in these ways, for example:

Figure 1: *TSP Versus CMM [1]*

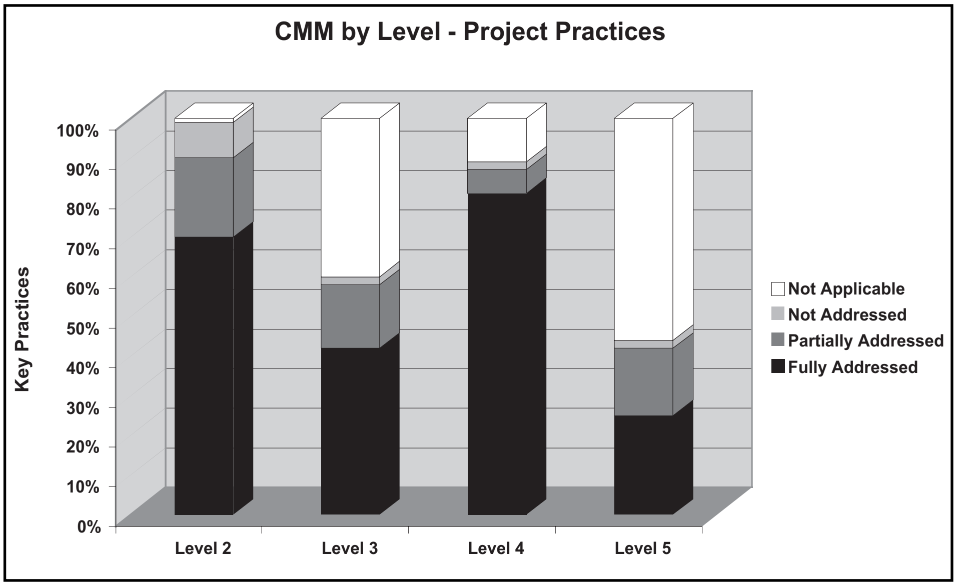

Figure Legend
Not Applicable: The content of the key practice does not apply in the scope of the TSP (key practices that deal with organizational activities are not applicable to the TSP).
Not Addressed: The TSP does not address any of the content of the key practice.
Partially Addressed: The TSP covers part of the content of the key practice.
Fully Addressed: The TSP fully covers the content of the key practice.

- The Personal Software Process^{SM} (PSP^{SM}) training is a potent tool for overcoming the natural resistance that engineers often display toward process changes, and the TSP launch and weekly process can diffuse any remaining reservations they might have. By the time the TSP team is working together, most resistance has been worked through, so they become proponents of process change rather than resistors. Instead of having to *push* process improvements on these software engineers, you may find that they *pull* eagerly for them.
- Your TSP projects will yield cost savings, productivity improvements, and quality advances that will more than pay for the costs of introduction. In fact, even with good CMMI processes in place, your TSP projects will accelerate the rate of return on your entire process improvement initiative, allowing you to either accelerate your process work or realize the returns on it earlier.

TSP and the CMMI

The TSP has the same roots as the CMMI, being based on the Software Engineering Institute's (SEI^{SM}) early research that produced the CMM for Software (SW-CMM). As such, it aligns well with the CMMI and partly or fully satisfies the CMMI's goals. Figure 1, which was published by the SEI in 2002, shows the degree to which the TSP addresses the key practices of the SW-CMM.

Figure 1 shows that the majority of the key practices in the CMM are at least partially addressed by the TSP. However, how are they addressed in the CMMI? The SEI is expected to publish a similar analysis of the TSP versus the CMMI soon, and because of the similarities between the models, we should expect similar results. In a presentation at the 2003 Software Engineering Process

® Capability Maturity Model, CMM, and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
SM CMM Integration, Team Software Process, Personal Software Process, TSP, PSP, and SEI are service marks of Carnegie Mellon University.

Group conference [2], the SEI reported these preliminary findings about the specific practices (SP) and generic practices (GP) of the CMMI:

- **Project Management SPs:** Most fully or largely implemented.
- **Process Management SPs:** Majority partially or largely implemented.
- **Engineering SPs:** Majority fully or largely implemented.
- **Support SPs:** No consistent pattern as yet.
- **Generic Practices:** No policies in the TSP, but most other GPs at all capability levels are either taught in PSP training or practiced by TSP teams, or both.

What Is the PSP and TSP?

How do PSP and TSP address these CMMI goals? They do so through rigorous training followed by establishing a team environment that encourages the trained engineers to use in their day-to-day work what they learned in the training.

PSP

The first step in a TSP initiative is to train all team members who can write programs in PSP. The PSP is more than just training; it is a boot camp consisting of about 40 hours of classroom instruction, 10 programming assignments, and three data-analysis exercises, requiring a total of about 150 hours for the average programmer to complete.

The result of the PSP boot camp is that the programmers do not just learn *about* good processes, they actually improve their own processes, measure the effects of those process changes, quantify the benefits they have experienced, and set goals for further improvements. The PSP achieves these results by leading students through three steps.

1. In PSP0, they lay a simple foundation for the learning to come:
   - Following simple process scripts.
   - Collecting three basic measures of their work (time spent, size of products produced, and defects corrected).
   - Performing a simple post-project analysis.
2. In PSP1, they begin to build the capability to plan and manage their own work:
   - Following a defined project planning process.
   - Using their own prior data to make increasingly more accurate estimates for each programming assignment.
   - Planning their work at a level of detail that allows them to track and manage their progress.
3. In PSP2, they focus on achieving significant quality improvements:
   - Using their prior data to plan for incremental improvements in the quality of their programs.
   - Removing defects early using personal review techniques guided by their own prior defect performance.
   - Identifying and capitalizing on defect prevention opportunities in their program design and implementation methods.

Those who complete the PSP boot camp emerge with the knowledge and skills to make accurate plans, work to those plans, and produce superior quality products.

> *"The TSP has the same roots as the CMMI, being based on the Software Engineering Institute's early research that produced the CMM for Software. As such, it aligns well with the CMMI and partly or fully satisfies the CMMI's goals."*

TSP

The TSP then provides the project framework in which programmers can carry these classroom skills back to the workplace and use them to transform their team's performance. Although the TSP is not primarily about training, it *does* include training for certain players:

- The *TSP Executive Strategy Session* (which is usually held before the engineers receive PSP training) provides senior managers with the opportunity to identify key software development-related issues, determine how the PSP/TSP will address those issues, and outline a strategy for making significant progress on those issues in a timely and cost-effective way.
- *Managing TSP Teams* training provides lower-level managers with the tools and methods for making the most of their TSP teams' ability to estimate accurately, plan appropriately, and self-manage.
- *Introduction to Personal Processes* allows the non-programmers on the TSP teams to learn about the same topics that the programmers learned in the PSP boot camp, though without the intensive work.

With all these pieces in place, the team is ready for the key activity of the TSP: the project launch. This is a four-day workshop in which the TSP team members estimate and plan their project, utilizing goals, objectives, requirements and constraints from senior management and clients, along with their data on their own performance on prior projects. The launch culminates in a presentation by the TSP team of the project plans to senior management and client, and, often, negotiation of goals and requirements in light of constraints and expected performance. The final result is a project plan that is aggressive yet achievable, and is agreed to by all stakeholders.

After the launch, the TSP also provides the team with processes and tools for regular progress monitoring, identifying and acting on corrective actions, and reporting status up the management chain. For projects that take longer than a few months, TSP relaunches provide a basis for incremental project planning and regular realignment of plans with project progress.

PSP and TSP Costs

As can be seen from the descriptions of PSP and TSP, the largest cost component of introducing TSP in an organization is the time the engineers spend in training and project launches. Each engineer spends three to four weeks in PSP training (usually spread over two to three months), and all project participants spend nearly a week in each project launch.

These costs are embraced with the expectation that the time invested will be returned at the end of the first project as the system testing time shrinks due to the improved quality of the system produced by the TSP-trained team.

TSP Need Not Be Additional Work

How can I say that the *TSP need not be additional work* when I just discussed the amount of time and effort required to introduce it? That is work that an organization certainly would not be engaged in if it did the CMMI alone! This is true. But the point of this discussion is that by investing in the TSP, you can make substantial *reductions* in the time and effort for the overall CMMI effort.

This is suggested in reports published in CrossTalk about CMM implementation at the AV-8B Joint System Support Activity (JSSA), at China Lake, Calif. In September 2002 [3], JSSA reported achieving CMM Level 2 in a relatively quick 14 months by adding the TSP to its strategy. Then, in January 2004, JSSA made a follow-up report about moving from CMM Level 2 to Level 4 in only 16 months (as opposed to the normal 50 months), crediting the TSP with their almost unheard-of pace. JSSA's experience shows that you can capitalize on the TSP's proven framework for process improvement to speed your CMMI initiative along.

How can this be? As discussed in the earlier analysis, the TSP addresses many of the same subjects as the CMMI, but it does so from the opposite perspective. Where the CMMI takes an organizational perspective, the TSP comes at these topics from the perspective of the individual engineers and the teams in which they operate. This perspective has two main advantages.

First, the TSP builds understanding and acceptance of process discipline from the grass roots up. The PSP (as the first step of a TSP initiative) builds understanding of the important principles in the individual engineers. The PSP boot camp for engineers not only teaches these important principles, but it provides the setting in which each engineer can try them out and prove their worth based on personal experience and data. The TSP then shows the engineers how to apply those same principles on real projects in a team setting.

The second main advantage is that the TSP works in the small. Anyone who has been involved in organizational change knows of the difficulty in changing the way that people work. Even if they are not hostile to the change, most people will be reticent to abandon the tried-and-true for the unknown. By focusing on one team at a time (and teams of PSP-trained individuals at that), the TSP mitigates many of the difficulties inherent in organizational change.

Taken together, these two advantages give the TSP the potential to accelerate the organization's process improvement effort not by initiating yet another effort, but by including the TSP as a strategic part of the larger CMMI improvement initiative.

Get Engineers Involved at the Beginning

The CMMI by its nature tends to not affect many of the engineers in the early stages of an improvement effort. This is because the first process areas (PA) the organization works on at Level 2 are generally focused on management rather than engineering. The Project Planning, Project Monitoring and Control, Supplier Agreement Management, and Measurement and Analysis PAs tend to deal with topics that only marginally affect engineers. Even the Requirements Management PA (as opposed to the Requirements Development PA) is more about management issues than technical ones.

> *"The metrics practices the engineers learn in the PSP and TSP have the potential to form the basis for the organization's MA [measurement and analysis] processes and procedures."*

The unfortunate result of this management focus at Level 2 is that the engineers tend not to be engaged in the effort, and either feel that they are being left out or that they are lucky to avoid involvement. Of course, this situation changes dramatically when the organization begins focusing on Level 3 and its engineering-specific PAs such as Requirements Development, Technical Solution, Product Integration, Verification, and Validation.

By making the TSP an integral part of your CMMI improvement effort, you assure your software engineers are facile with topics that are traditionally the territory of managers, and the focus at CMMI Level 2 (measurement and analysis, project planning, and project monitoring and control), while building grassroots acceptance of process discipline and establishing good processes in each team.

In the following three sections, I will look specifically at the PAs that are the primary focus of the TSP. They are measurement and analysis (MA), project planning (PP), project monitoring and control (PMC) at Level 2, and verification (Ver) at Level 3. Because of the TSP's primary focus on these PAs, TSP teams can play a critical role in working out and pilot-testing processes for them.

Measurement and Analysis

PSP training starts out by showing the engineer how to collect three primary measures: time, size, and defects. It focuses on these three (as the engineers soon learn) because they are critical to achieving the PSP's goals of accurate planning and quality management. As the engineers go through the training, they are encouraged to analyze their individual data to understand how it illuminates their performance, and to use it as the basis for any process improvements.

The TSP builds on this basic understanding of a few measures by prompting each team to identify the metrics they will need based on the goals of the project, then to collect and analyze those metrics regularly. So the TSP implements MA at the individual project level, starting with goals and objectives, identifying metrics to support them, and collecting and using them throughout the project.

The metrics practices the engineers learn in the PSP and TSP have the potential to form the basis for the organization's MA [measurement and analysis] processes and procedures. In addition, as these are developed, TSP teams will provide a perfect infrastructure in which to pilot test them. Since the TSP team members will understand metrics and how to collect and use them, they will provide the feedback you require to evaluate the effectiveness of those candidate processes and procedures.

If you have already established your organization's MA processes and procedures, then your TSP teams will use them just as any other project team does. The only difference is likely to be the enthusiasm with which they embrace metrics and the constructive criticism they will provide to help you with improving your MA procedures and standards.

Project Planning and Control

After instituting the collection of basic metrics, the PSP teaches engineers to use their own data to plan their individual work. They analyze their assignments in light of the projects they have undertaken to date, and use their own data to make reasoned and achievable plans for completing it. They then learn to compare their actual performance against those plans so they can improve their future plans, in addition to seeking ways to improve their processes.

The TSP harnesses these skills to lead the engineers through estimating and planning the work for the entire team. After each team member is assigned specific work items, he or she uses personal data to produce a detailed individual work plan, and those plans are rolled together and balanced to produce the team's final project plan. Then every week, the team collects and analyzes the data on its actual performance and if deviations from their plan call for it, they identify, implement, and track corrective actions.

The practices your engineers will learn in the PSP and TSP can form a solid foundation for your organization's PP and PMC processes and procedures. Your engineers will be able to tell you how well those practices work in your organization as well as providing ideas for ways to fine-tune them to their specific needs. Then, as you document your organizational PP and PMC processes and procedures, they will easily be able to incorporate them into their ways of working and give you feedback on how well they work. With the PSP and TSP as a foundation, establishing your organization's PP and PMC processes should be relatively easy and fast.

If you have already established your organization's PP and PMC processes and procedures, then your TSP-trained engineers will use them as the basis for their own planning and tracking activities. But because of the team's deepening understanding of the mechanisms for planning and tracking projects, they will be a regular source of suggestions for improving your organizational PP and PMC standards to better fit the organization's needs.

Verification

The other major focus of the PSP and TSP is on quality management. The PSP teaches the engineers how to track their defects, and then use that information along with good review processes to do an effective job of personal design and code reviews. The TSP adds peer reviews, giving the team members the tools and methods they need to remove the vast majority of defects from their programs *before* they begin testing, thus improving product quality *and* reducing system test time.

All of the basic processes and procedures that you will need for the Ver PA are embodied in the TSP's review methods. And, as with the PAs discussed so far, your TSP teams will be ready to pilot test your organization's Ver processes and procedures after they have been defined. And again, if you have already defined your organizational Ver processes and procedures, then your TSP teams will embrace them and provide improvement ideas for them.

Defining Your Other Processes

In addition to the four PAs discussed, your TSP teams will be indispensable as you develop the processes and procedures for all of the other PAs. Because the PSP and TSP address most of the CMMI PAs to at least *some* extent, your TSP teams will have at least some experience with most of the PAs that you will need to address.

As you begin working on any particular PA, your first step should be to discuss it with your TSP team members. As you find out how they address that particular PA, you will find that at least some of the procedures you will need already exist. Then, as you explore more deeply to expand those processes to fully satisfy the goals of the CMMI, your TSP team members will be likely to have good ideas about how to add any missing practices, or how to tune those that in some way fall short.

Because of their experience with using and improving disciplined processes, they will be strong members of any process action team that you establish. They will contribute practical ideas to meet the CMMI goals, and they will be able to evaluate alternatives from the basis of practical experience. TSP team members will help to make your process action teams effective at defining and documenting processes and procedures that will work in your organization.

Summary

Adding the TSP to a CMMI initiative does *not* mean adding more work to an already overworked group. Rather, it can be an effective way to accelerate that initiative by laying a solid foundation of process discipline, engaging the engineers from the very beginning, and providing processes that already address a significant number of CMMI practices. The costs of incorporating the TSP into your CMMI initiative should be more than returned as you achieve your CMMI goals more quickly and with less organizational pain.

In addition, by adopting the TSP, you can convert your engineers into allies in the process improvement initiative – people who will lobby for better processes, help you to realize them, and embrace them in their day-to-day work.◆

References

1. Davis, Noopur, and Jim McHale. "Relating the Team Software Process^{SM} (TSP^{SM}) to the Capability Maturity Model® for Software (SW-CMM®)." Pittsburgh, PA: Software Engineering Institute, Mar. 2003 <ftp://ftp.sei.cmu.edu/pub/documents/02.reports/pdf/02tr008.pdf>.
2. McHale, Jim. "The Case for Using TSP^{SM} With CMM®/CMMI®." Software Engineering Process Group (SEPG) 2003, Boston, MA <www.sei.cmu.edu/cmmi/presentations/sepg03.presentations/tsp.pdf>.
3. Hefley, Bill, Jeff Schwalb, and Lisa Pracchia. "AV-8B's Experience Using the TSP to Accelerate SW-CMM Adoption." CrossTalk Sept. 2002 <www.stsc.hill.af.mil/crosstalk/2002/09/hefley.html>.

Additional Reading

1. Pracchia, Lisa. "The AV-8B Team Learns Synergy of EVM and TSP Accelerates Software Process Improvement." CrossTalk Jan. 2004 <www.stsc.hill.af.mil/crosstalk/2004/01/0401pracchia.html>.

About the Author

Alan S. Koch is president of ASK Process, Inc., a training and consulting company that helps organizations to improve the return on their software investment by focusing on the quality of their software products and the processes they use to develop them. Koch was with the Software Engineering Institute (SEI^{SM}) for 13 years where he became familiar with the Capability Maturity Model®, earned the authorization to teach the Personal Software Process^{SM} (PSP^{SM}), and worked with Watts Humphrey in pilot testing the Team Software Process^{SM} (TSP^{SM}). He also worked in software development for 28 years. Koch is an SEI-authorized PSP Instructor and TSP Coach candidate, an SEI Transition Partner for the PSP and TSP, and a certified Project Management Professional. He is a member of the Project Management Institute and author of "Agile Software Development: Evaluating the Methods for your Organization."

ASK Process, Inc.
4378 Ridge RD
Natrona Heights, PA 15066
Phone: (412) 849-0421
E-mail: alan.s.koch
@askprocess.com