Extending the TSP to Systems Engineering: Early Results from Team Process Integration

Anita Carleton
SEI

Del Kellogg and Jeff Schwalb
NAVAIR

A collaboration between the SEI and NAVAIR—Team Process Integration (TPI SM)—is currently underway. The TPI effort leverages the PSP and TSP research and body of practice. This article discusses the progress and performance through a pilot project with the AV-8B Systems Engineering team as well as others within NAVAIR that have utilized TPI in non-software domains. This article will share lessons and experiences with other industry/government organizations interested in applying the TSP in a non-software setting. The early results suggest some encouraging trends.

Since the emergence of software engineering in the 1960s, the size, pervasiveness, and complexity of software-intensive systems have increased by several orders of magnitude. The size of aircraft software systems in the 1960s approximated 1,000 lines of code while aircraft systems built in 2000 contained more than six million lines of code. The pervasiveness of software within aircraft systems has increased from controlling less than 10 percent of the functions the pilot performed in the 1960s to 80 percent in 2000 (as shown in Figure 1 on the following page).

We know that increases in software and system size contribute to increased complexity which, in turn, has contributed to pushing delivery and costs well beyond targeted schedules and budgets [1].

In a recent workshop conducted by the National Defense Industrial Association, the top issues relative to the acquisition and deployment of software-intensive systems were identified. Among them are:

- The impact of system requirements upon software is not consistently quantified and managed in development or sustainment.
- Fundamental systems engineering decisions are made without full participation of software engineering.
- Software life-cycle planning and management by acquirers and suppliers is ineffective.

So the biggest challenge is creating the right foundation: estimation, planning, development, and management practices as well as team processes, training, coaching, and operational support that will assist in a migration from buggy products and unnecessary rework (resulting in inflating development costs) to a proactive approach that builds integrated, quality software-intensive systems from requirements to field deployment.

SM TPI is a service mark of Carnegie Mellon University.

Background

The SEI's TSP provides engineers with a structured framework for doing software engineering work. It includes scripts, forms, measures, standards, and tools that show software engineers how to use disciplined processes to plan, measure, and manage their work [2]. The principal motivator for the TSP is the conviction that engineering teams can do extraordinary work if they are properly formed, suitably trained, staffed with skilled members, and effectively coached and led.

> *"The principal motivator for the TSP is the conviction that engineering teams can do extraordinary work if they are properly formed, suitably trained, staffed with skilled members, and effectively coached and led."*

The TSP is already being used with great results on software teams [3]. A Microsoft study reported that by using the TSP, software teams cut schedule error from 10 to one percent. With its TSP teams, Intuit has increased by 50 percent the time that teams can spend in developing products during a typical year-long release cycle: Increased quality has dramatically cut the testing time required. An analysis of 20 projects in 13 organizations showed TSP teams averaged 0.06 defects per thousand lines of new or modified code. Approximately one-third of these projects were defect-free. Other studies show that TSP teams delivered their products an average of just six percent later than planned. This compares favorably with industry data showing that more than half of all software projects were more than 100 percent late—or were cancelled. These TSP teams also improved their productivity (size of developed code per hour of development time) by an average of 78 percent.

NAVAIR develops, acquires, and supports the aircraft and related weapons systems used by the U.S. Navy and Marine Corps. In recent years, interest in applying TSP to non-software domains has increased. The SEI TSP team has collaborated with NAVAIR to expand the TSP to teams that do other engineering along with software. These include areas such as systems engineering and integration, product integrity, CM/DM/QA (Configuration Management/Data Management/Quality Assurance), and process improvement itself.

NAVAIR already has a proven track record with the TSP and has demonstrated return on investment on their software projects [4, 5]. Table 1 (on the following page) shows TSP results from two NAVAIR programs: the AV-8B's Joint Mission Planning System (JMPS) program and the P-3C program. This result, due to the reduction in defect density, is a gross savings of $3,225,606 ($3,782,153 less the investment of $556,547). In turn, the ROI is derived from the cost savings compared to the cost of initially putting the TSP in place; in this case, it was a ratio of better than 7 to 1. Further, these organizations each reached CMM Level 4 in less

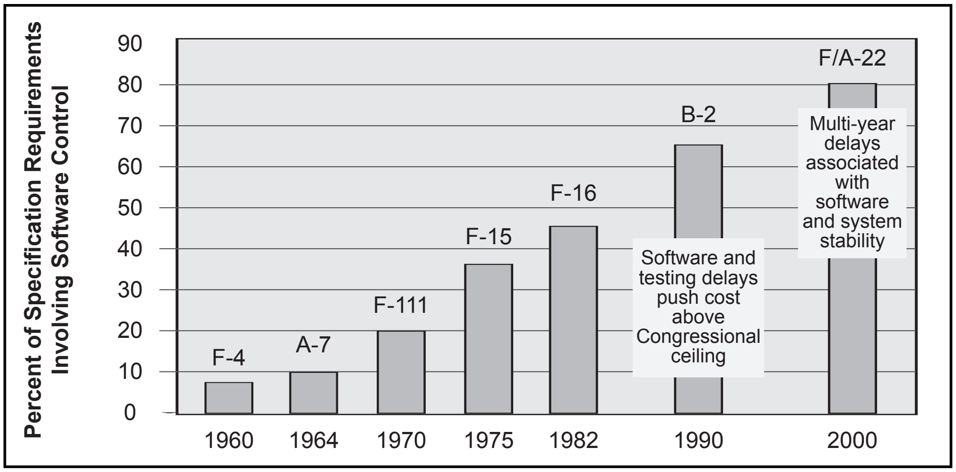

Figure 1: *Increasing Capabilities and Challenges of Software in DoD Systems* [1]

than 30 months—instead of the typical six years.

Very similar results occurred with other programs at that time, like with the E-2C aircraft program, also achieving CMM Level 4 in less than 30 months with their development teams using the TSP at the same time. Most recently (Jan. 2010), the H1 aircraft program worked less than 20 months to obtain a CMMI Level 3 rating while their development team used TSP to maintain aircraft software for the fleet.

The organizations referenced have standardized the TSP for all of their software development and maintenance work. These early adopters of the TSP are meeting their mission of producing higher quality products while maintaining significant cost savings. Their development teams now like using the TSP, saying of their staffs, "Once they have adopted it, they can't imagine working any other way." In all presented cases, the initial investment was returned in their first project and has then gone forward time and again to benefit the organizations for many years.

Results from these examples continue to inspire other NAVAIR System Support Activities (SSAs) to use the TSP. There are more than 20 additional NAVAIR SSAs now pursuing software process improvement activities. NAVAIR is seeing recurring savings and can now direct cost savings to the procurement of additional aircraft and weapons. In addition, NAVAIR used the TSP to accelerate CMMI improvement.

Starting TPI Efforts

Based on the demonstrated, measured success of software projects using the TSP in NAVAIR, other teams asked if they could apply the same processes to systems engineering and software/systems acquisition projects. As a result, NAVAIR has teamed with the SEI to expand the TSP framework to a technology called TPI. The SEI is also receiving additional requests to apply the TSP to non-software settings since it is becoming increasingly difficult to solve software problems without addressing systems engineering issues.

The NAVAIR/SEI collaboration entails testing the hypothesis that we can achieve the same kind of performance improvements applying TPI to systems engineering as we did applying the TSP to software projects, thereby improving management and communications in software-intensive systems and acquisitions. Our approach will entail conducting a series of pilot projects to determine if extending TSP practices to systems engineering results in measurable improvement. We will then use the results of this work to establish common processes for both systems and software engineering across the NAVAIR teams. Initially, the AV-8B Joint SSAs (developing the Harrier Aircraft) was selected as the systems engineering pilot program.

In kicking off these efforts, we realized that there were a number of research challenges that specifically had to be addressed. We extended the TSP practices to systems engineering by:

- Determining the baseline performance for systems engineering work at NAVAIR.
- Developing prototype processes/process definitions/scripts for systems engineering.
- Formulating relevant measures, especially size and quality measures pertinent to systems engineering.
- Building conviction and discipline in our leadership and team member training materials for teams that don't necessarily write software programs.
- Developing an extensible tool that allows for outlining any process, for collecting data unobtrusively, and for defining a measurement framework pertinent to any engineering domain.

Early results of applying TPI show some encouraging trends. The AV-8B Systems Engineering pilot project team is changing the way they do their work and is beginning to see some results similar to those realized by TSP teams. The AV-8B team is practicing more disciplined methods for planning and executing their work. They are meeting their missions and beginning to see some cost savings. In addition, the pilot team is inspiring other NAVAIR 4.0 SSAs to pursue process improvement [6].

Benefits

Through the pilot effort, we are seeing some of the following benefits:

Establishment of a Systems Engineering Baseline

We are beginning to establish a baseline for systems engineering performance at NAVAIR that can be used for estimating, planning, and tracking projects and programs:

- The requirements productivity rate varies between three and nine require-

Table 1: *TSP Results at NAVAIR*

Project	Defect Density Before TSP	Defect Density After TSP	Total Defects Before TSP	Total Defects After TSP	Average Cost to Fix	Product Size (KSLOC)	Cost Savings From Reduced Defects
AV-JMPS	1.13	0.59	501	261	$8,330	443.0	$1,992,663
P-3C	4.60	0.60	176	23	$8,432	38.3	$1,789,490
						Total Savings:	$3,782,153

ment statements per hour, depending on the complexity of the project[2].

- By just tracking requirements size growth, the team was able to decrease the rate of project size growth from 23.6 percent in the initial development cycle to 11.5 percent in the subsequent development cycle.
- By collecting the planned and actual requirements size and growth for the various components and the team productivity rate, the team builds up historical data that can be used on future projects.
- To quote one team leader: "Prior to TPI, we made estimates in a bubble. Now we are establishing and maintaining baselines for all of our releases, which allow us to make better estimates and more realistic plans and schedules."

Establishment of Planning Practices

Planning at the program and team level is now accomplished by holding multi-team launches that involve all of the teams implementing either the TSP or TPI. At first, they plan for no more than four months of work at a time so that their tasks can be detailed enough with fairly stable component sets. The component sets start to vary for a longer development duration so their plan would be less stable. This process is used by the AV-8B program to understand requirements from management, assemble plans, allocate work, and achieve commitment to plans from management and team members. The overall plan for the year and the next-phase plan are developed by the teams, work is allocated by the team, and the schedule is determined and committed to by team members.

Establishing Tracking Practices

For tracking purposes, work is broken down into small chunks that can easily be tracked (tasks are tracked at a granularity of less than 10 hours). Tracking only the task hours per week (planning for around 20) allows two or three tasks to be completed each week. Work is tracked daily by team members and discussed weekly in team meetings: Every team member knows how they are performing to their individual plan and the team plan. Monthly status reports are derived from the consolidated weekly reports by the team leader and presented to the integrated product team leads.

Twelve team members were able to achieve (on average) between 18 and 22 on-project task hours per week. The team performed well above the planned task hours: 15 per week in the first cycle.

The engineers embraced project planning and tracking. Each individual is able to track personal commitments to the team, enabling the team to better monitor commitments to the program. Tracking the work helped the team members with staying on-task, commenting that: "I need to stop doing *X* to get back on track. It is very easy to see the impact daily and weekly of not working to the plan."

Developing Standard Processes, Measures, and Tools

Standard processes, measures, terminology, and tools were developed and used by the AV-8B Program:

- The PSP-derived Excel spreadsheet and a process support technology Access-based tool were used for estimating, planning, and tracking work for team members and team leads.
- Team members identified, defined, and documented all systems engineering standard life-cycle processes in the tool. The team defined and developed an 18-step overall systems engineering process and a 482-step detailed systems engineering process.
- Through the defined processes, NAVAIR was able to maintain the consistency of processes across projects/programs. The defined processes also offered the ability to cross-train individuals. One integrated product team lead said: "We have a team concept across our program with all of the sub-teams (systems engineering, product integrity, software, test, lab, etc.). We also have a common set of processes and metrics to help all of the teams better communicate and address dependencies across the teams."

Performance Trends

With no historical data to go by, the team's initial plan identified a guess at a goal of less than 5 percent schedule slip and measured performance against the goal. The actual performance had an overrun of less than 10 percent. Now with some historical data, the team can set more realistic goals and try to continually improve on them. As far as cost and quality performance, size and effort estimates were within ±10 percent of what was planned, and there were no high-priority problem reports coming out of test.

Employee Work/Life Balance

TPI helped improve employee work/life balance. In order to get their job done before implementing TPI, employees routinely worked overtime. With TPI (and in order to get their 18-22 task hours per week), they did not have to work as much overtime. Overtime was decreased from being standard practice—sometimes 25 percent or more—to occasional overtime hours (less than 10 percent).

Customer Responsiveness

Customer responsiveness has improved to the fleet, the naval aviators, and to the internal program managers. The systems engineering team is able to more easily adapt to program and personnel changes. The pilots are beginning to provide input early on in the project—during the launch process—before the work has commenced (instead of providing feedback during the test phases). Program management feels that the TSP/TPI efforts are a success because the teams understand their work and the dependencies among all of the teams. The systems engineering team can also plan for a percentage of unplanned tasks to use their data to negotiate impact and trade-offs of unplanned work to planned work.

More Teams Doing TPI

We have since launched more non-software teams using the TPI approach. One of these is a mixed engineering team at Joint Munitions Effectiveness Matrix Weaponeering Systems that is applying the TPI to their non-software work as well as to their software team. This team has been using TPI for more than a year, has gone through four launch/relaunches, and has seen the types of benefits that the AV-8B team has seen. They also are seeing steady progress in making more accurate and precise estimates of their work, and have refined the triggers that would initiate an adjustment of their behavior so they stay on schedule.

Another example is the Precision Attack Weapon System Tactical Program Office, demonstrating the effectiveness of the TPI approach for one of their systems engineering teams. Their team has been using the TPI approach for more than a year and saw immediate benefits. During the initial launch, they developed a never-before-seen detailed plan that gave senior management the needed data to get additional project funding without having to *arm wrestle* the Program Manager, Air (PMA).

Then there is the P-3 lab team at the Patuxent River, Maryland Naval Air Station, who has been applying this approach to the many configurations of the lab setup they must provide. The P-3 team started applying TPI as an approach to the implementation phase of their Black Belt DMAIC (or Define, Measure, Analyze, Improve, and Control) project. Since starting about three years ago, the team has since provided two annual cycles of lab services and is halfway through their third. The P-3 lab team supports their customers by providing more than a dozen lab configurations across the PMA. This breaks into two basic types of support: usage in terms of running tests, and support in terms of configuring labs for those tests. Aggregate lab usage data shows a deviation of 12 percent less than planned while aggregate lab support data shows a deviation of .5 percent more than planned. While performance is impressive, deviation was at times greater when examined at the individual lab-customer level. As expected, this aggregate deviation demonstrates the advantage of estimating in smaller increments.

At the time of writing this article, several other process improvement efforts at NAVAIR are getting started with plans of applying TSP to their software teams and TPI to their non-software teams.

Summary

All engineering efforts must start with integrated teams. These teams must plan their work—and work to those plans—while collecting basic measures. They must then apply analyses to this data and derive metrics to determine their status on current work and, eventually, as a source for improving their planning capability on future work. From this approach, we have seen quality products and services delivered over and over with the potential for further improvement.

To make this happen, we have seen the need to put in place the TPI foundation of estimation and planning processes, team processes, development and management practices, effective and timely training, as well as launch, coaching, and operational support.

Projects that have adopted these methods have shown a dramatic increase in product quality and fidelity of schedule and effort estimates. The methods are supported by a doctrine that trains and sustains performance and quality improvement in an organization.

This article has shown what is possible when teams use TPI to establish this foundation to meet critical business needs. The end result is the delivery of high quality systems, on cost, and with improved productivity.◆

References

1. Walker, Ellen. "Tech Views – Challenges Dominate Our Future." *DACS Software Tech News*. Oct. 2007 <www.softwaretechnews.com/stn_view.php?stn_id=43&article_id=86>.
2. Humphrey, Watts S. *TSP: Leading a Development Team*. Upper Saddle River, NJ: Addison-Wesley Publishers, 2006.
3. Davis, Noopur, and Julia Mullaney. *The Team Software Process (TSP) in Practice: A Summary of Recent Results*. SEI, Carnegie Mellon University. Technical Report CMU/SEI-2003-TR-014. Sept. 2003 <www.sei.cmu.edu/reports/03tr014. pdf>.
4. Wall, Daniel S., James McHale, and Marsha Pomeroy-Huff. *Case Study: Accelerating Process Improvement by Integrating the TSP and CMMI*. SEI, Carnegie Mellon University. Special Report CMU/SEI-2005-SR-012. June 2007 <www.sei.cmu.edu/reports/07tr013.pdf>.
5. Saint-Amand, David. *Process Improvement at NAVAIR Using TSP and CMM*. Proc. of the 1st Annual TSP Symposium. San Diego: Sept. 2006.
6. Carleton, Anita, et al. *Extending Team Software Process (TSP) to Systems Engineering: A NAVAIR Experience Report*. SEI, Carnegie Mellon University. Technical Report CMU/SEI-2010-TR-008. Mar. 2010 <www.sei.cmu.edu/reports/10tr008.pdf>.

Notes

1. This graphic was created based on a table called "System Functionality Requiring Software," but the original creator of the table is debated: either PM Magazine or a U.S. Air Force "Bold Strike" Executive Software Course from 1992. To view the table, see: Ferguson, Jack. "Crouching Dragon, Hidden Software: Software in DoD Weapon Systems." IEEE Software July/Aug. (2001): 105-107.
2. For example, AV-8B uses Telelogic DOORS Objects to identify the number of requirement statements and, hence, the size of the requirement set. Any organization/program product can be viewed as a comparable proxy.

Software Defense Application

Software defense organizations will benefit by learning about Team Process Integration, the continuing collaboration between the SEI and NAVAIR. As detailed in the article, results from current projects utilizing TPI show a gross savings of more than $3.7 million and a net savings of more than $3.2 million, with a return seven times the original investment. Quality improvement on two examined projects was a reduction in defect density from 1.1 to 0.59 defects per thousand LOC on one and 4.6 to 0.6 defects per thousand LOC on the other. TPI lowers costs, helps projects meet schedules, and improves productivity.

About the Authors

Anita Carleton is a senior member of the technical staff at the SEI, Carnegie Mellon University, where she has worked for more than 20 years on software process improvement, process measurement, and the TSP. She is the author of "Measuring the Software Process: Statistical Process Control for Software Process Improvement." Carleton has a degree in applied mathematics from Carnegie Mellon University and is a member of the IEEE Computer Society and the National Defense Industrial Association.

SEI
4500 Fifth AVE
Pittsburgh, PA 15213-2612
Phone (412) 268-7718
Fax: (412) 268-5758
E-mail: adc@sei.cmu.edu

Del Kellogg is a PSP Certified Developer, PSP Certified Instructor, and TSP Authorized Coach for NAVAIR-China Lake. He has spent most of his 30 years at NAVAIR working on development of embedded software for the A-7E, AV-8B, and the AH-1W aircraft. He has applied the PSP and TSP for the last nine years within multiple NAVAIR teams. He is currently working in the Process Resource Team at NAVAIR. Kellogg's background is in computer science, physics, and math, and received his bachelor's degree in computer science from the University of Idaho.

NAVAIR Systems/Software Support Center
1900 N Knox RD
BLDG 1494 (MS 6308)
China Lake, CA 93555
Phone: (760) 939-5494
Fax: (760) 939-0150
E-mail: delwyn.kellogg@navy.mil

Jeff Schwalb is employed by NAVAIR at China Lake, California, where he has been since 1984. He currently leads a NAVAIR enterprise team that helps provide continuous process improvement support across NAVAIR. Schwalb first became involved with process improvement in the 1990s using the SW-CMM, then becoming a certified PSP instructor and TSP coach. He has taught each of the TSP/PSP courses and has been involved in the TSP launch of several projects across NAVAIR. He is now working with the SEI to extend TSP practices into other domains. He received his bachelor's degree in computer science from California State University, Chico.

NAVAIR Systems/Software Support Center
1900 N Knox RD
Building 1494 (MS 6308)
China Lake, CA 93555-6106
Phone: (760) 939-6226
Fax: (760) 939-0150
E-mail: jeff.schwalb@navy.mil