

Microsoft's Pilot of TSP Yields Dramatic Results

Kelly Kimberland

In today's fast-paced software industry, organizations struggle to keep up with the demand for quickly produced software. Faced with reductions in budget and staffing, companies must make sacrifices to save time and money. Software quality suffers most in this schedule-driven market. For more than 20 years, the SEI has created and promoted methods that help organizations efficiently develop high-quality software. Microsoft, one of the world's largest software vendors, recently piloted a project using the SEI's Personal Software ProcessSM (PSPSM) and Team Software ProcessSM (TSPSM) with a team of software developers to change behaviors, improve processes, and deliver better software.

SEI Fellow Watts Humphrey has stated that the problem with defective software is not the programmers; it's how they are managed and trained. "The entire software culture continues to follow the expensive and ineffective test-based software quality strategy," Humphrey says. "Changing this mindset is difficult and time consuming. Microsoft's willingness to pilot the TSP demonstrated senior management's commitment to new quality and team-management strategies."

Microsoft Program Manager Carol Grojean was the team leader of the original pilot, and since then has become a TSP coach and mentor. She and Noopur Davis, a member of the SEI's technical staff, recently presented Microsoft's results at this year's Software Engineering Process Group (SEPG) Conference in Orlando, Florida.

TSP Builds Teams

After the original pilot, Grojean launched another Microsoft team using TSP. Prior to using the TSP, this team had just finished a 2.5 year project with an unsuccessful ending. The team had ten developers who were frustrated with the project, their jobs, and each other. The team was operating as individuals, not as a team. Now, after two weeks of intensive TSP training, they were being asked to spend a week in a room together making decisions as a team, not as individuals.

While a normal workday during a TSP launch is usually around nine hours, the first day of this TSP launch lasted more than 14 hours. The team members were reluctant to make goals and commitments, and they were concerned about management seeing their goals. When the second day of the launch lasted 15 hours, it was clear that the team members needed time to overcome the barriers built in the past few years. "They didn't know how to behave any differently," Grojean says. "They were reluctant to commit to clear, measurable goals for fear that those goals would be used against them if they failed. It is not easy for a team to realize that failing at a goal was better than not having a goal at all."

By the third day, the team was exhausted from the long days. Then, an amazing thing began to happen: the team members realized that they would accomplish nothing if they continued to

argue, and they began to work as a team. Though individuals were still defensive and hesitant about the process, they became more engaged. They knew they needed to come together to make the project work.

By the end of the four-day launch, the group was truly a team. “It is such an amazing transformation—we started with ten individual contributors and ended up with a well-organized, cooperative team,” Grojean says. She added that the transformation from individual contributors to one team is now her favorite part of the launch process.

The TSP accomplishes these remarkable results by combining proven project and quality management principles with well-known team building and coaching methods.

TSP Reduces Defects, Improves Quality

Watts Humphrey has said that even experienced and capable software developers inject one defect for every nine lines of code they write. For example, today’s common software-development practices result in 110,000 defects injected into a typical million line-of-code system, most of which must be found and fixed before the software can be released. “On average, poor quality software at customer delivery has about five defects per thousand lines of code (KLOC),” says Humphrey.

During TSP training, Microsoft developers reduced unit-test defects dramatically from more than 25 defects/KLOC to about 7 defects/KLOC. “Unit-test defects were reduced because these developers were removing most defects before unit test,” says the SEI’s Noopur Davis. She adds that “the goal of the TSP is to change how developers view testing; instead of thinking of testing as a defect-removal activity, developers start thinking of testing as validation of not just product functionality, but also as validation of the quality of the software. Developers strive to remove as many defects as possible from the software before testing, when defects are the easiest to find and fix.”

People are sometimes skeptical about training results, and want to see data from real projects to be convinced. Microsoft’s pilot project continued to show the quality improvements seen in training. The project received a high score on the process quality index (PQI), a process quality measure that TSP teams use as an indicator of product component quality. A PQI of 0.4 or greater on a scale of 0.0 to 1.0 is considered to be an indicator of a high-quality component. Microsoft’s pilot project had a PQI of 0.52.

Microsoft teams, like other software development teams, spend 40 – 60 % of their total development in test because they use that time to find and fix defects within the product. However, because Microsoft’s TSP pilot team spent time in earlier removal activities like personal reviews and team inspections, they spent only 11.5% of the total project effort in test. “The focus on spending more time up front is an important behavior change; the team is using the process to plan their schedule rather than letting the schedule drive their plan,” Davis says.

Ultimately, the pilot team ended up delivering its products to test on schedule and with high quality. “Microsoft results compared favorably with other TSP teams, which have delivered products with an average defect density of just 0.06 defects/KLOC,” says Davis. Additionally, since the code was truly complete at the end of the coding phase, the developers were able to advance to the next development cycle of the product rather than having to stick around during the testing to fix defects. This resulted in a 35% project cost savings. While the team delivered to test on the original committed date, the sharp reduction in test time made the team members available for other work earlier than planned.

For more information, contact—

Customer Relations

Phone
412-268-5800

Email
customer-relations@sei.cmu.edu
World Wide Web
<http://www.sei.cmu.edu/tsp/>