



# TSP at Adobe: Experiences of Three Teams in a Product Group

Barbara Spencer Sept 22, 2010 Adobe Systems, Inc.



## Topics

- Large product group and their three TSP teams
- Stakeholder feedback
- Program takeaways

## Large Product Family

- Great product
- Excellent, highly experienced team
- Tens of millions of LOC
- Several hundred people directly involved
- Dependencies on ~40 groups/third parties/companies
- Hundreds of millions of users

## Previous Process

- Modified waterfall
- Large database tracking features, enhancement requests, and tasks
  - Detailed estimates for entire release cycles
  - Developers assigned tasks for entire release cycles
  - Developers and Quality Engineers (QEs) working as separate teams
- Informal design and review processes
- Teams working hard to avoid cost overruns

## 2010 Quality Goals

- Product team adopted several of Adobe's recommended quality initiatives and set goals for their release, including:
  - Peer reviews of all code changes
  - Find 10% of all defects in code reviews
  - 25% fewer bugs in product compared to previous release

## New for Three Teams: TSP

- Three of product's teams began doing TSP in 2009
  - Two teams doing TSP-Multi for a large product area
  - Third team doing TSP as single team for large feature area
- Changes for teams:
  - Dev and QE organized in the same team
  - Detailed planning and tracking by iteration
  - Planned reviews and inspections
  - Team launches, relaunches, and retrospectives
  - Roles for team members

## TSP Multi-Teams: Two teams

- 15 developers, 8 quality engineers, and 2 team leaders
- One program manager, one quality manager
- Managers wanted to attend TSP meetings
- Weekly coordination meetings attended by team leaders, planning managers, and program manager

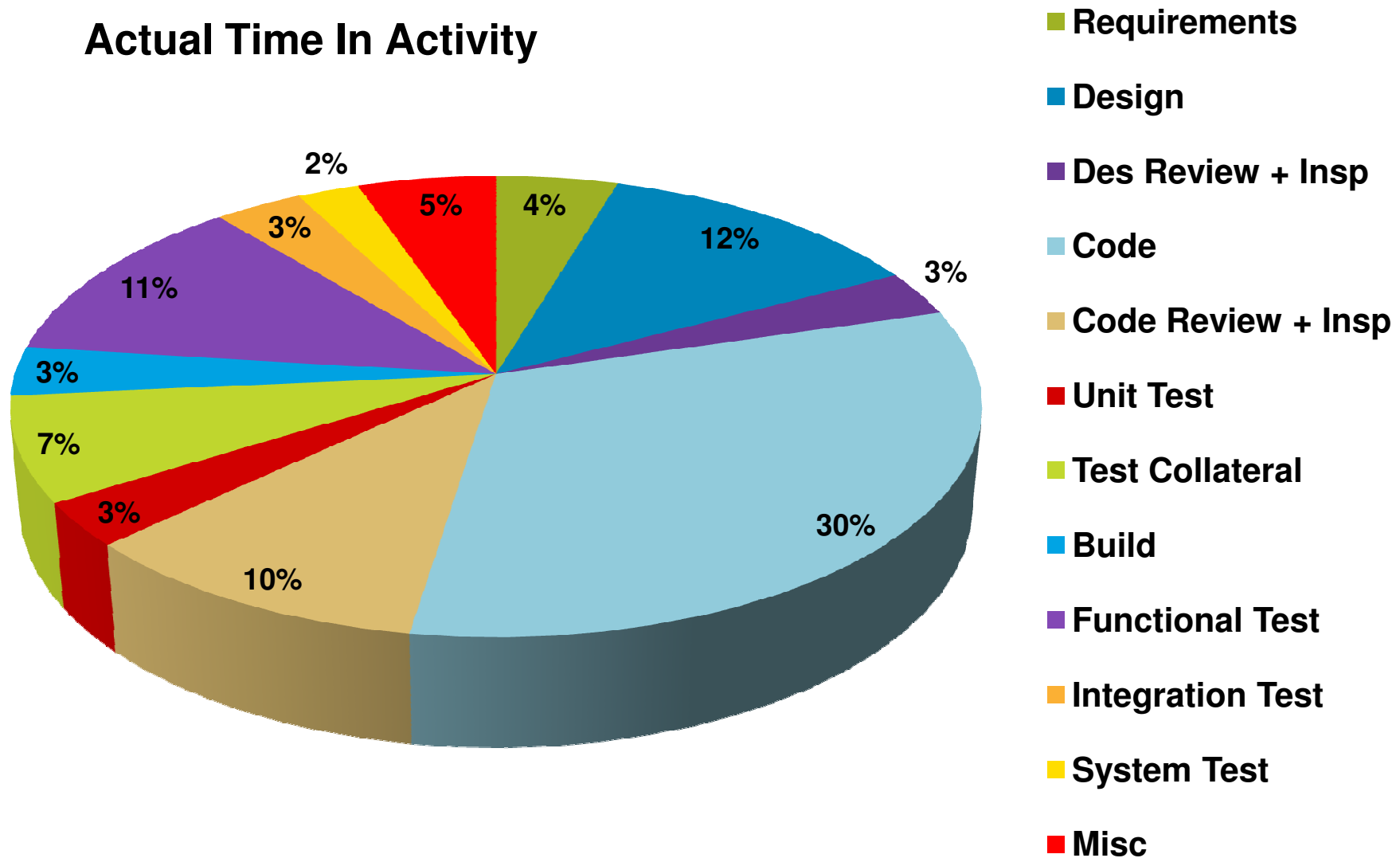
## Experiences of Two Teams Doing TSP-Multi

- Some Dev's felt did not need TSP to estimate or produce a quality product but agreed to try it
- Said launches take too long 'Too much overhead'
- Did not like tracking time but liked the projects data driven and self managed
- Most developers adopted TSP, did well, and preferred TSP to previous process
- Quality engineers like being on TSP team and like having more involvement in the projects
- Everyone: 'Quality much higher going into test'



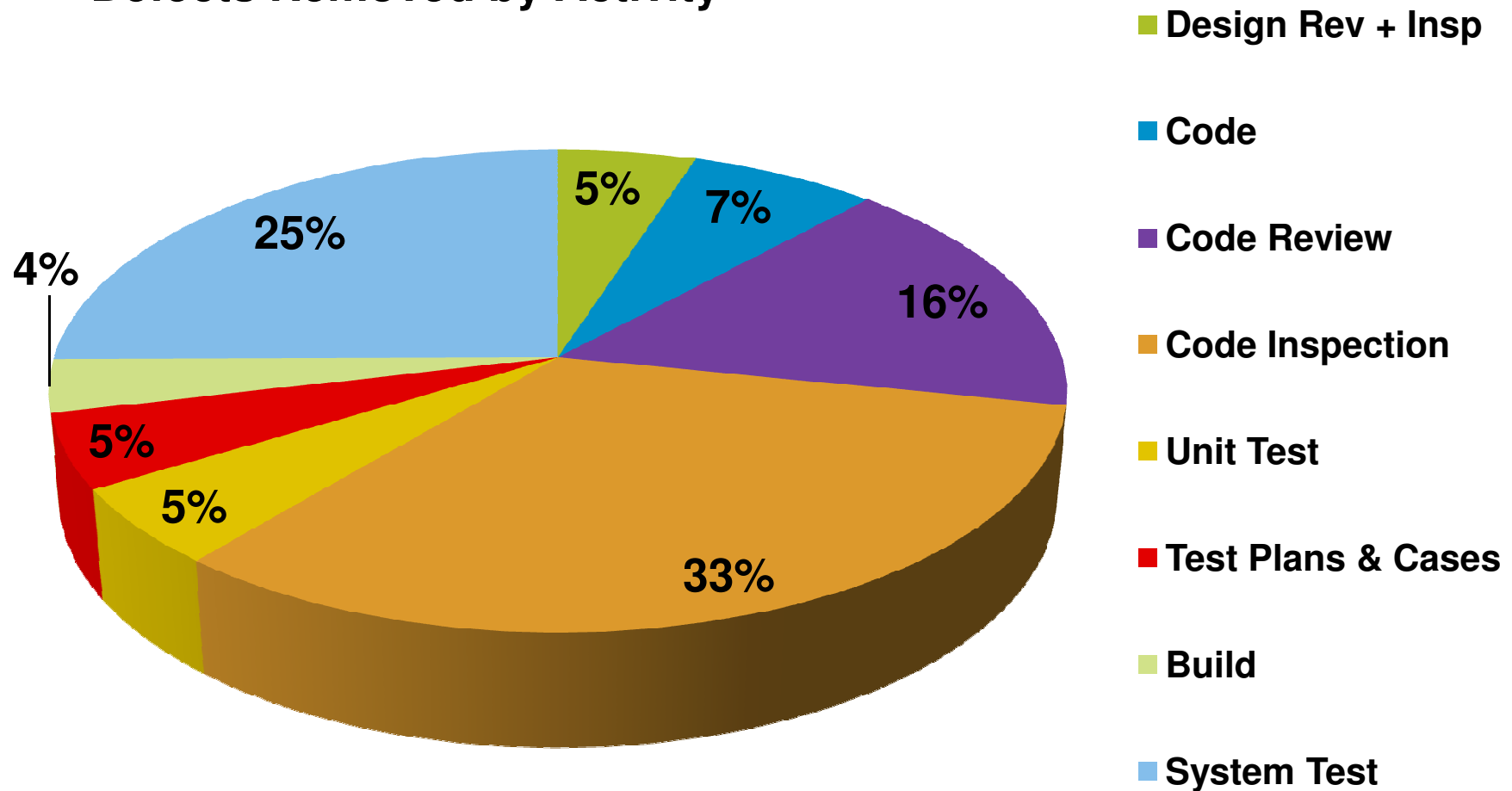
## To Date Results - Two Teams Doing TSP-Multi

### Actual Time In Activity

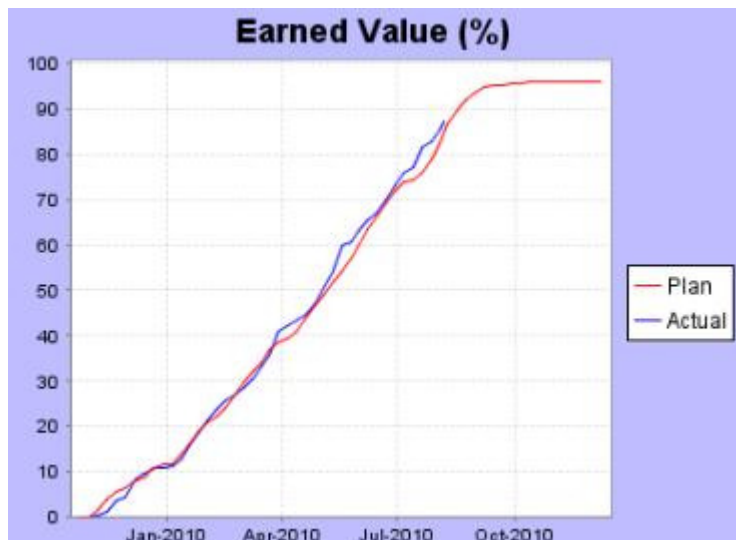
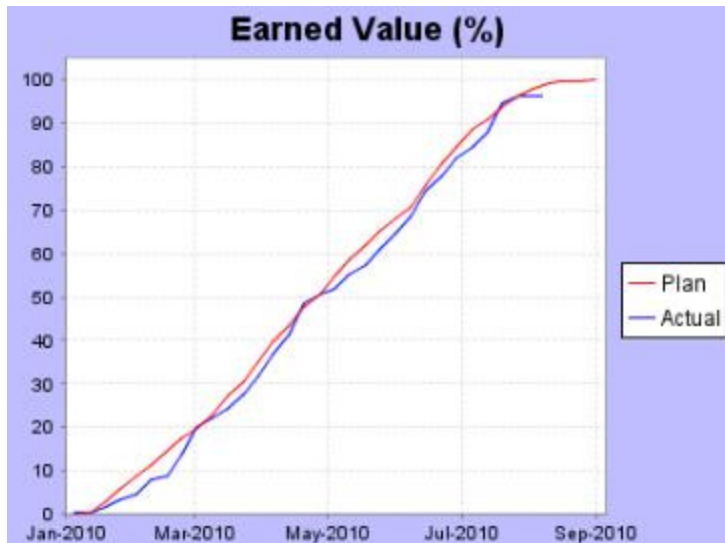


## To Date Results - Two Teams Doing TSP-Multi

### Defects Removed by Activity



## To Date Results: Two TSP-Multi teams



**The two TSP-Multi teams tracked well to their EV.**

**They cleared their development backlog of components by the end of the development portion of the project.**

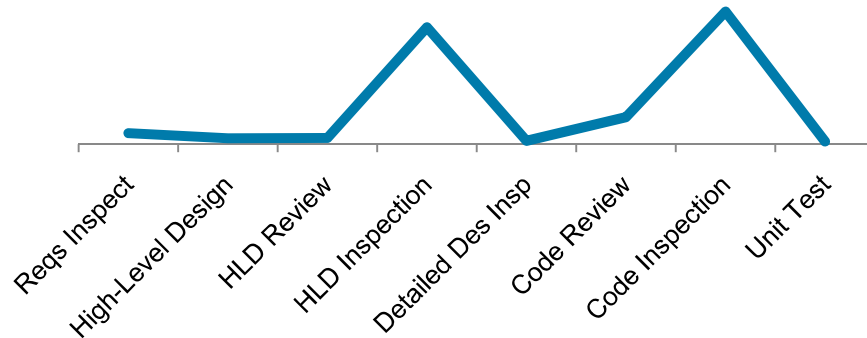
**The two teams exceeded expectations for functionality. Quality much higher than previous release.**

## Experiences of Third TSP Team

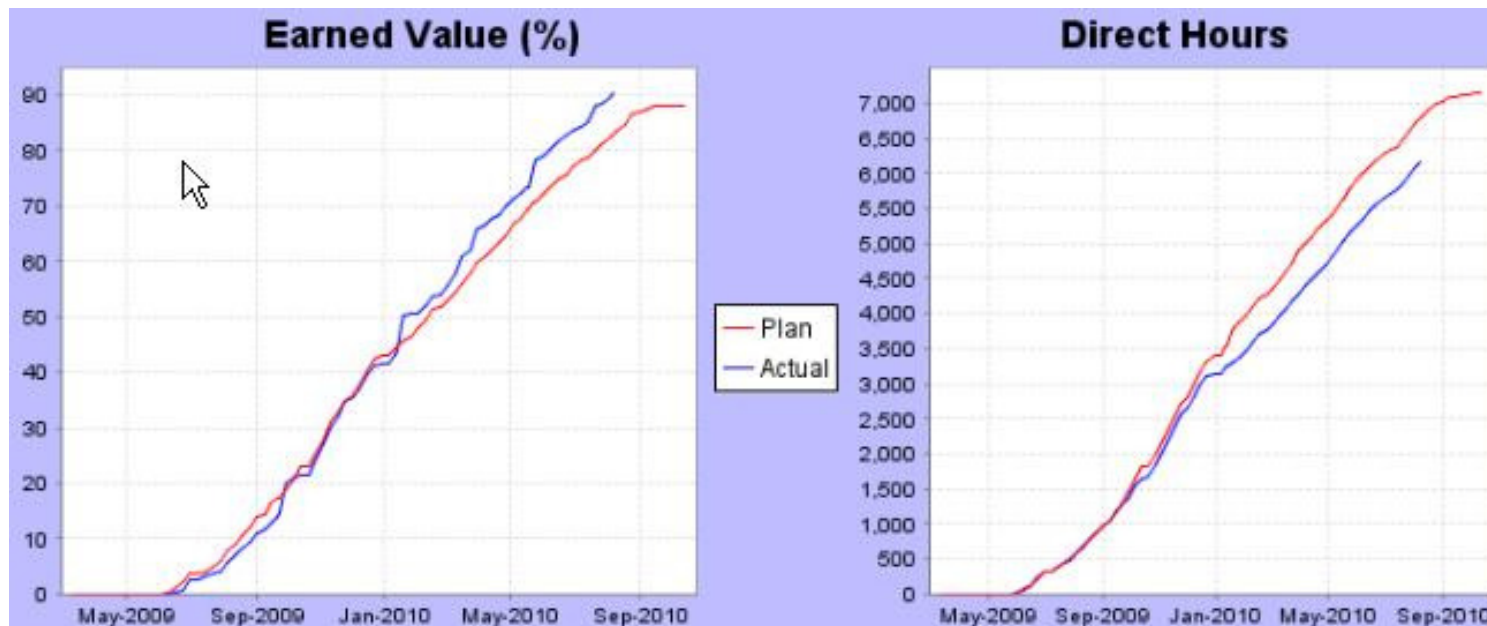
- Medium size team, strong Team Leader
- Team enthusiastic about TSP
- Followed overall team's 'Waterfall' process and used basic TSP
- Did up front prototyping and design
- Did inspections online using Code Collaborator
- Team Leader shortened relaunches to fit within one day
- Team Leader streamlined their TSP to reduce overhead

# To-Date Results for Third TSP Team

Defects Removed by Activity



**Summary: The three teams made great progress in their quality journey.**



## Stakeholder Feedback

- Team Leader
- QE Manager
- Program Manager
- Second Level Manager

## Team Leader Feedback: Successes



## Team Leader Feedback: Learnings/Challenges

- Initial perception - TSP requires too much overhead. Having gone through it, the team feels their time was well spent. The coach helped us remove/avoid overhead. The transition happened around the 2<sup>nd</sup> and 3<sup>rd</sup> iteration.
- Integration with bug database a must have.
- TSP tool could be improved. One of the significant improvements would be to support multiple people to add/delete tasks at the same time.
- LOC tracking: Tracking for mixed environments was one of the challenges. Tracking lines of code when multiple people worked in a single branch was the other one. The team decided to not use LOC-based estimation for this release.
- Adopt TSP fully in overall product team: Having to work with TSP in addition to other processes was challenging for the team. Team leader and coach should shield the team as needed.



## QE Manager Feedback: Successes

- Concept of team. While it sounds trivial and routine, TSP really brought QE & Dev closer together.
- QE work becomes part of completing development. The workflow demands QE involvement early on to complete work. A shift from focusing solely on Development completion.
- Individual QE involvement comes much earlier, in design activity. QE can contribute before coding by participating in design inspections which lowers cost of bug fixing later.
- Works best when requirements are clear.
- Clear sense of prioritization and work definition enabled better decision making by individuals.
- Early Quality Indicators were positive.

## QE Manager Feedback: Challenges

- Complete work Fully in an iteration. TSP works well when teams fully complete work in an iteration – from design to testing.
- Defect logging. Lots of discussion on this process resulted in some confusion.
- Overhead definition. An early team definition would have eliminated debates later on. What is overhead? What overhead work will be planned? What activities make up overhead, etc.
- Dealing with external dependencies. Are difficult to track and can often be very interruptive to planned time.
  - A solution would be to include other teams' activities in the TSP.
- As much as possible bring work components to full completion in an iteration and avoid delaying functional QE testing to future iterations.
- More training for QE. QE training was 2 days; QE could benefit from more time and some hands-on training.

## Program Manager Feedback: Team Successes

- Multiple Iterations: We had 6-7 week iterations followed by a week of retrospective, planning and executive presentations/demos. Dev/QE could focus on specific tasks/priorities in an iteration with re-planning at the end.
- Task Tracking: Dev and QE tracked their own progress and communicated the progress each week in a structured TSP meeting using the TSP tool.
- Teamwork: Everybody on the team had team roles (Planning, Process, Quality, etc.) and was on a single team which worked well. Everyone feels part of the process.
- Project Backlog: Estimation was roughly calculated for Backlog features and re-evaluated each iteration. Features that didn't fit were moved to the "NiceToHave" list.
- Coordination meetings – Teams had lots of dependencies on other teams and we had weekly meetings for handling them.
- Deliverables: End of iteration showed progress and gave all teams a forum to demo their features, ask questions, and give feedback.

## Program Manager Feedback: Learnings/Challenges

- External Dependencies: Documentation, Installers, Project-wide Milestones, Shared Resources – TSP is more optimized for code development.
- Distributed Team: Being part of a larger project where the rest of the teams have a different process. Other processes add overhead in addition to TSP.
- Estimation challenges
  - Mixed coding environments / Size of work was hard to estimate / Scope changes / work complexity.
- Integrated Bug Tracking: Integration with bug database is a must have. Dev Defects have been tracked in TSP, all other Defects should be tracked in bug database. Defect categories don't map well to bug database. More defects were removed prior to test than were reported.
- TSP Executive Reporting: The TSP Tool worked better than feature database, but did not generate the kinds of reports our management was looking for. Need better reporting for management.

## Second Level Manager: Feedback All Three Teams

- Focus on Quality
  - Much higher quality of product going into test
  - > 25% Defects removed prior to QE Test (To Date; project not yet complete)
- Accuracy of Estimation
  - Several thousand tasks completed on schedule
- Communication within Product Areas
  - Involved QE input and testing early
  - Features more complete than previous releases

## Second Level Manager: Feedback All Three Teams

- Motivation
  - TSP teams added lots of cohesion between groups/people
- Increased Efficiency
  - Several thousand task hours logged; high level of Earned Value

**TSP beneficial but has some challenges**

## Program Takeaways

- Takeaways from Adobe's resident coach/TSP program manager
  - Challenges
  - Recommendations
  - In Closing

## Program Takeaways - Challenges

- TSP program may not be as envisioned by coaches or senior management
  - Only a portion of a large team may be doing TSP
  - Management teams' expectations high
  - Estimates may not be as accurate as expected; results may not be as expected
- Not everyone may embrace TSP, some critics
  - Not all senior developers interested in change
  - Some prefer not to track time or defects
  - QEs notice that TSP focuses mostly on developers
  - Can be difficult to apply TSP to QE activities
  - Some Team Leaders may customize TSP too much



## Program Takeaways – Recommendations for TSP

- Simplify, streamline, and adapt TSP to needs of teams
  - One day for a typical relaunch
  - Don't be too strict. Example: difficult to count LOC
- Improve TSP Tools
  - Full database backend needed for:
    - Full browser-based usage from any location
    - Coordination with other teams plans and data
    - Mapping of defects to bug database
  - Automated reporting of teams' progress to Management
- Include managers who are directly involved
  - Some of your strongest supporters are first-line managers
  - Find way to involve managers - recruit their help with reporting and get their input on how to improve program

## Program Takeaways – Silver Lining

- TSP highly beneficial in spite of challenges
  - Even if not fully deployed, TSP beneficial if there is at least one strong supporter in management team and have strong team leaders
  - Teams can benefit from using basic TSP
- Even if TSP not ‘textbook’ its practices are often adopted by many and teams benefit
  - Organizations adopting many TSP best practices
  - Curmudgeons and critics extolling TSP’s best practices
    - Peer reviews, early defect removal
    - Detailed planning and tracking
- TSP acting as beacon of best practices for company’s quality programs
- Combining quality goals with TSP is a great way to improve

# For More Information, contact...

Barbara Spencer  
Adobe Systems  
[bspencer@adobe.com](mailto:bspencer@adobe.com)  
<http://www.adobe.com>



**Adobe**