



**Leveraging TSPSM/PSPSM
Metrics to drive Predictability
and Quality of product
releases**

An Intuit Perspective

- Sudhir P, Balasubrahmanyam P

Agenda

- What is TSP/PSP... and NOT???
- What does TSP/PSP provide?
- TSP/PSP Goals
- An Intuit Case Study
- Cumulative Analysis
- Benefits
- TSP/PSP sharable Best Practices
- Elements for TSP/PSP success
- Summary
- References & Acknowledgements
- Q & A

What is TSP/PSP... and NOT???

- It is NOT:

- A Silver Bullet solution
- A radically different approach to development
- A new programming language
- A way to invent more task time

- It is:

- A process that enables the team members to take firm control of the project schedule
- A framework that allows detailed planning and tracking of project status
- A process that helps in building high quality systems
- A team building approach

What does PSP provide?

■ Measurements

- Size
- Time
- Defects

■ Best Practices

- Task Breakdown - > Detailed planning
- Time allotted to design
- Size estimation methods
- Time allotted to Review/Inspection
- Time tracking
- Defect tracking
- Metric analysis
- Coding standards

What does TSP provide?

Mindset change enabled by..

- Project Management
 - Detailed planning and tracking
 - Prescribed weekly meetings to review metrics
 - Task hour monitoring
 - Earned value
- Team Building
 - Share leadership/Roles
 - Team coach (project management co-pilot)
 - TSP launch (communicate with share holders)

The development process is not fundamentally different...
the mindset (managing by data) is different

TSP/PSP Goals

- Improve Quality -> Higher Productivity
- Predictability
- Visibility
- Efficiency
- Continual Improvement
- Self-directed teams
- Mindset change

An Intuit Case Study

■ Single Project

- Project is an reengineering effort to replace an existing legacy platform framework
- Code is developed mostly C#/.Net with some COM/C++, XML/XSLT
- Team size is 8 (6 Developers + 2 QA Engineers)
- Project is a 8 Full Time Engineers (FTEs) effort

■ Cumulative Summary across 3 projects

- Including 2 other new project initiatives which implemented TSP/PSP

Analysis of single project

- Quality analysis
- Plan analysis

Quality Goals: What?

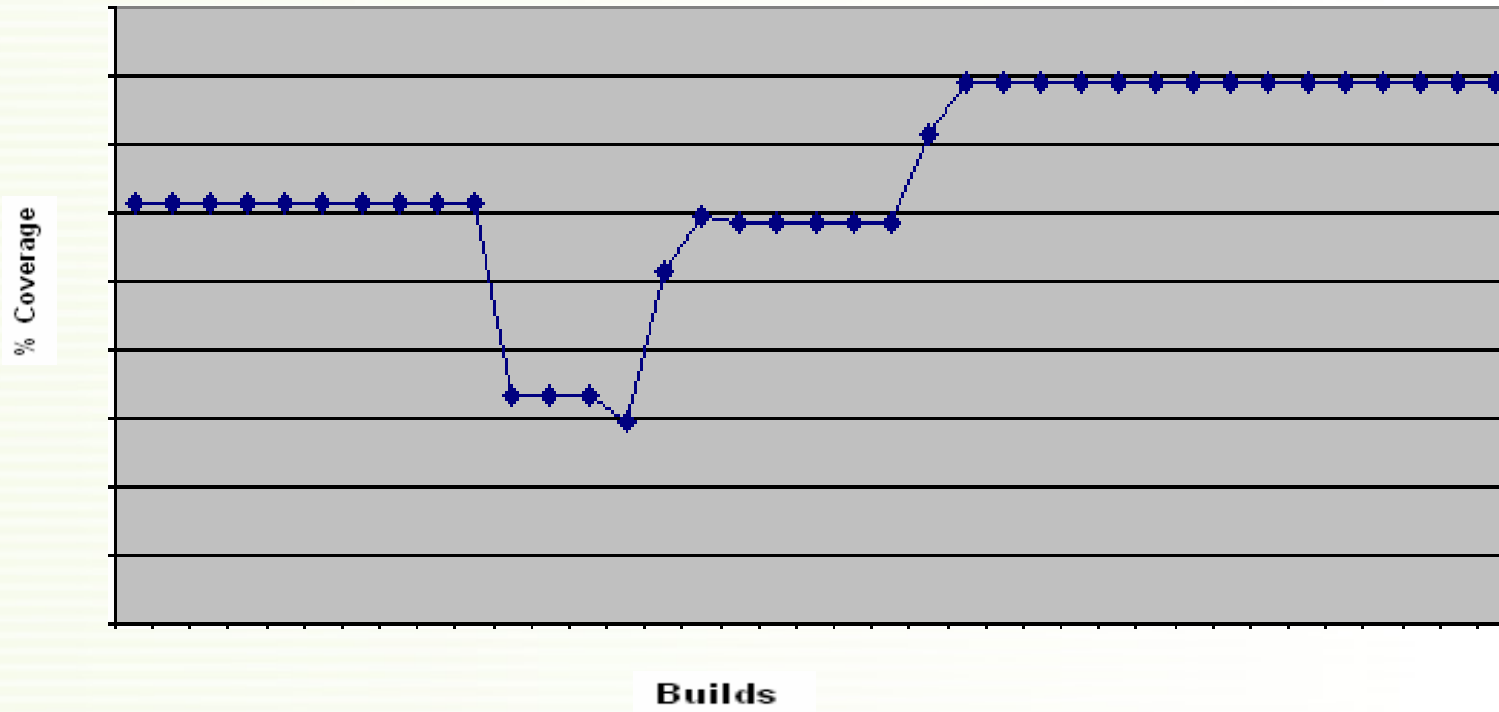
- 90% of code will be reviewed
- 90% of all feature changes will be design reviewed
- All test plans/Quality plans will be reviewed
- Unit tests to cover >70% of code path
- Post “code complete” defects at X% or less compared to previous release
- Defect density at System test 50% less than company average
- 95% or more build success on group branch (one level above team branch)

Quality Goals: How?

- C# and .Net Code coverage test execution
- Performance tests using windows Powershell
- Tools used: NUnit, NCover, NCoverExplorer, Microsoft FxCop
- Design reviews, code reviews, Inspections

Quality Analysis

- Could sustain high build quality through
 - Unit tests
 - And covering more scenarios

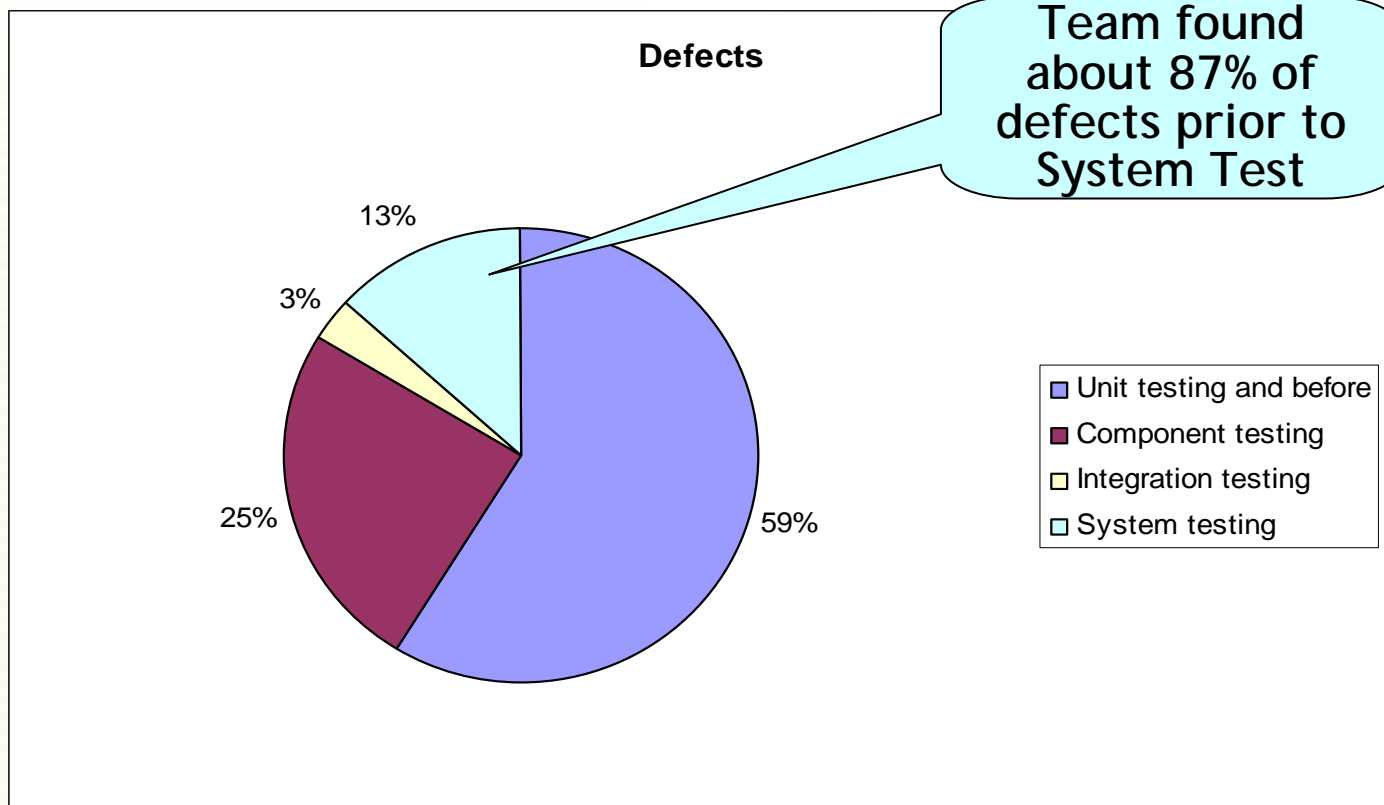


Quality Analysis

- Final Product Performance
 - Business requirement was to improve the existing infrastructure's performance by 4X
 - The team actually delivered 13X improvement

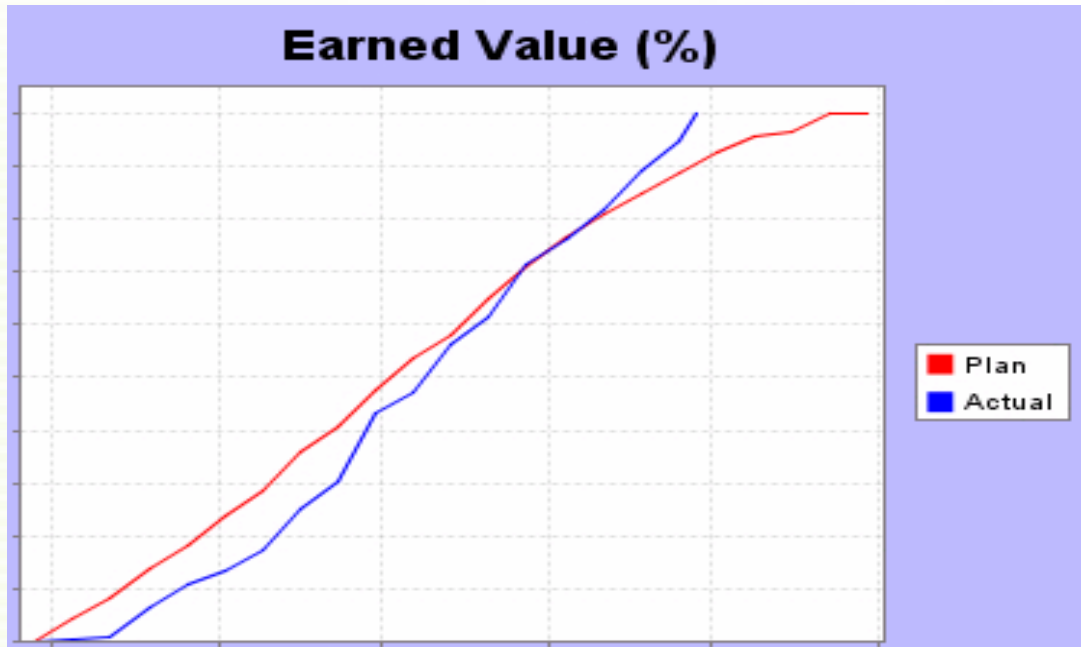
Quality Analysis

- Defect data



Plan Analysis

- **Schedule Predictability - How?**
 - Granular level of task estimation
 - Good understanding of inter-dependent tasks



- **Observations**
 - Tasks got completed ahead of time
 - Team spent less task hours

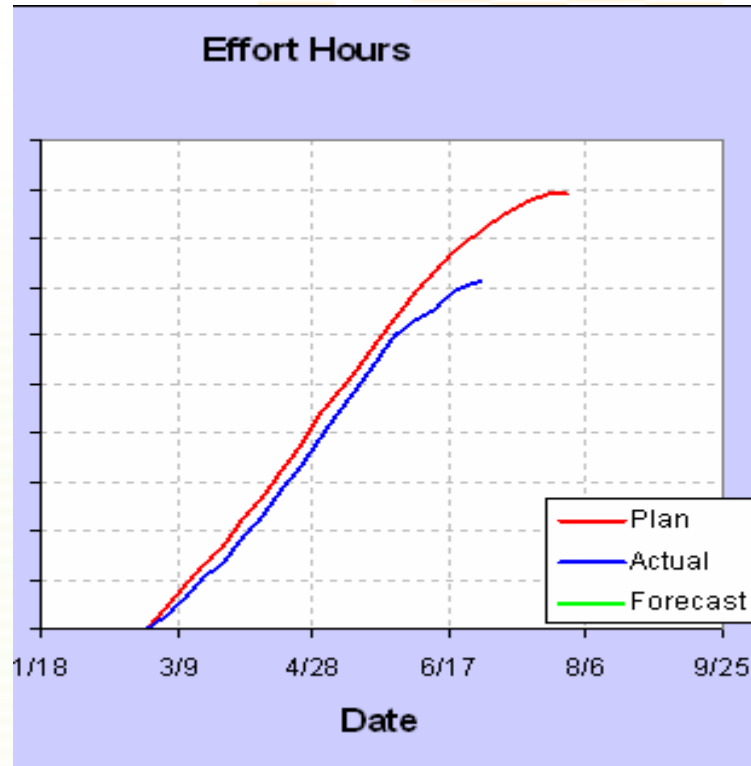
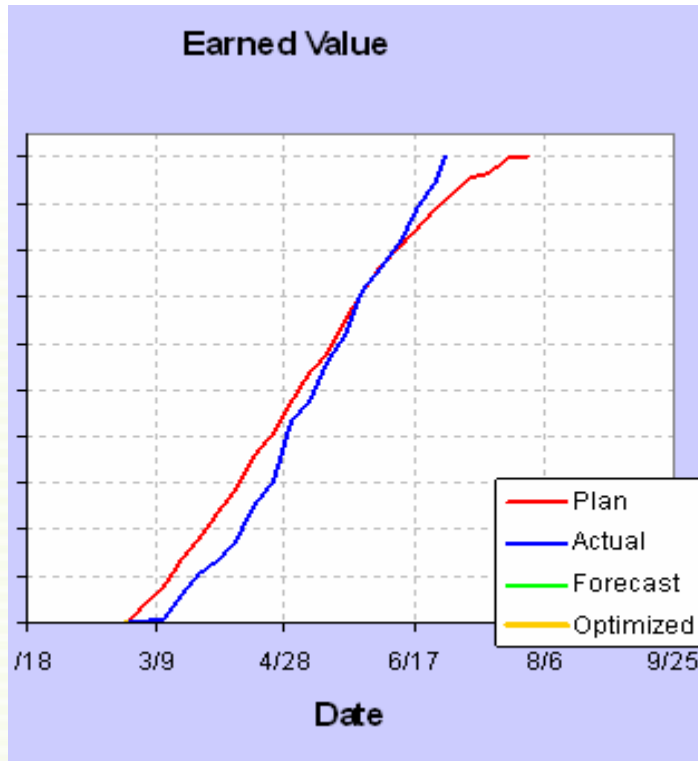
Cumulative Analysis

- Quality analysis
- Plan analysis

Quality Analysis: Results

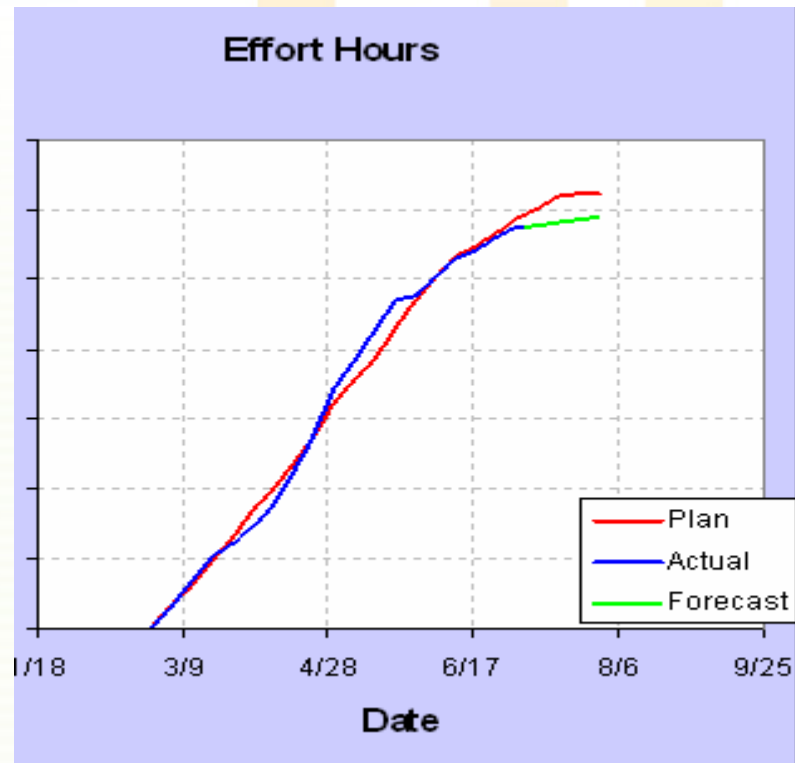
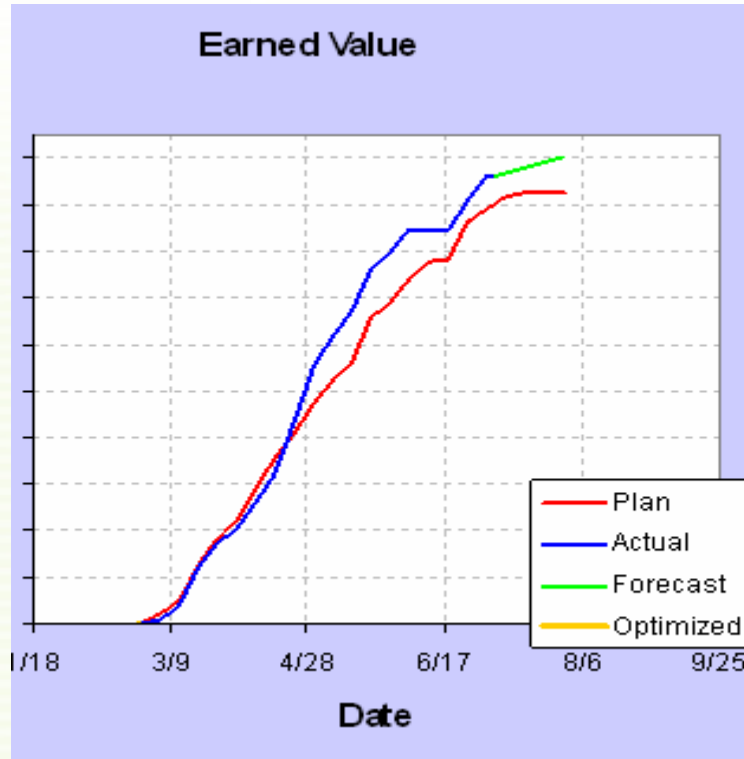
Quality Parameters	Results
System Test Defect Density	70% of Plan
Build success on group branch	100% of plan
Code review	90% of plan
Design review for feature changes	90% of plan
Unit Test Code coverage	99% of plan

Earned Value and Task Hours – Project A



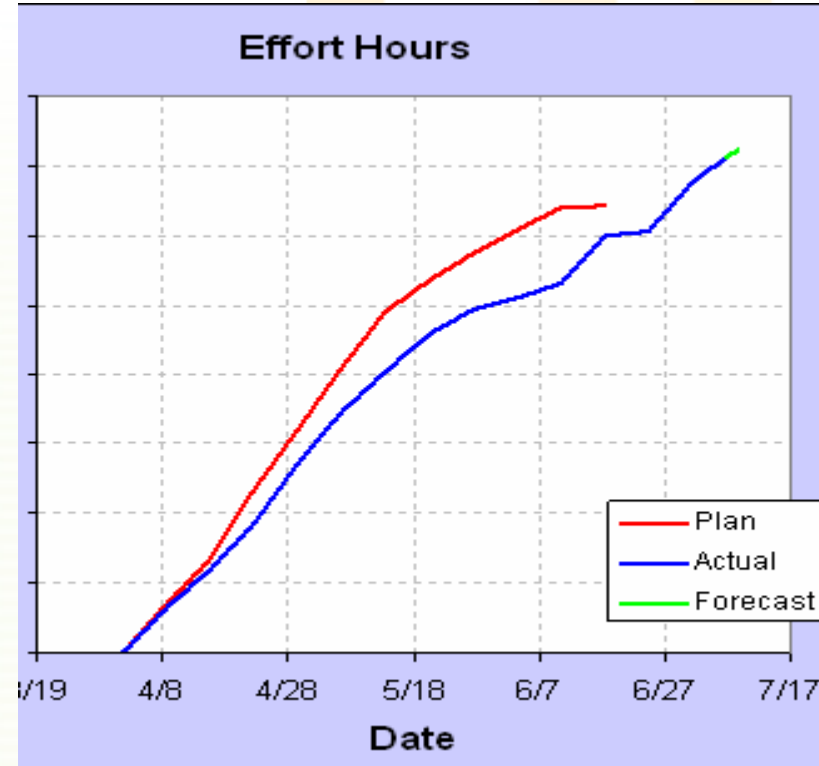
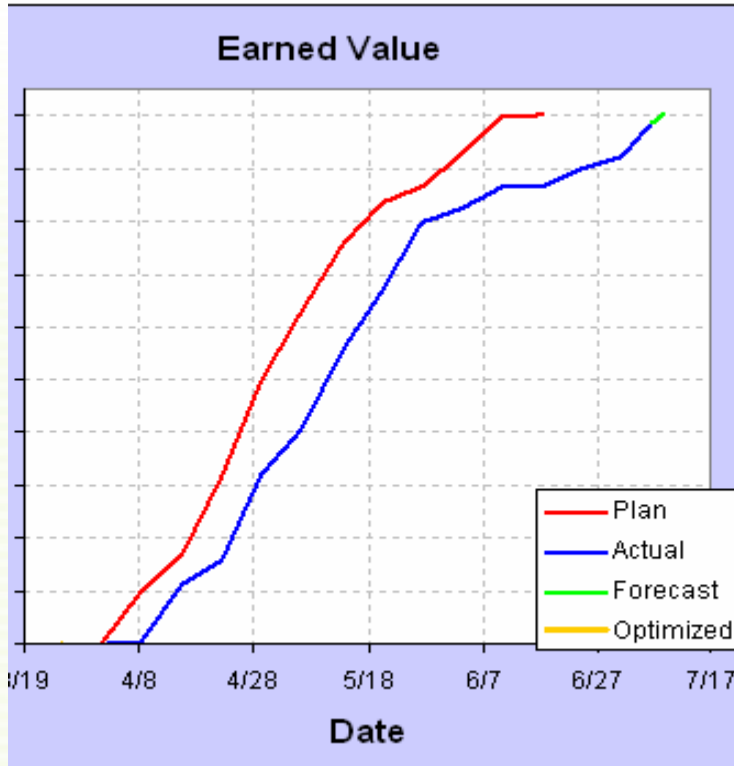
This team has done a good job of managing EV and Task Hours

Earned Value and Task Hours – Project B



Team has completed almost all the tasks except some project management tasks

Earned Value and Task Hours – Project C



Team has completed almost all the tasks except some project management tasks

Plan Analysis: Summary

Effort Summary	Project A (% Plan Vs Actual)	Project B (% Plan Vs Actual)	Project C (% Plan Vs Actual)
Weeks until Code Complete	0	8	43
Effort (task hours)	-10	7	16
Developer Productivity (LOC/Hr)	25	7.3	22
Size (LOC)	1	53	40

■ Analysis:

- Teams took 17% extra week-effort for task completion
- 4% extra task hours were consumed
- Developer productivity increased by 18% than planned
- “Code expected to be written” increased by 30%

Benefits

- Helps engineers build time in their plan for design reviews, code reviews and inspections
- Reduced over-commitments by software engineers and the team
- Earlier the defects found, less time it takes to fix them and also saves System Test time/effort
- Improved productivity since engineers are in better control of their own work since they estimate the task hours
- PSP gets software engineers involved in process improvement and boosts organization's process improvement effort

TSP/PSP sharable Best Practices

- **Project Launch**
 - Enhanced communication with stakeholders
 - Team building
- **Detailed project planning**
 - Task level (5-15 hours per task)
 - Inspections (participants, conference room, dates)
 - Specific dependencies noted
 - Rolling integration drops and load balancing
- **Defect tracking**
 - “In process” and system test and production
- **Time tracking**
 - Where is development time spent?
- **Size tracking**
 - Easily measurable and correlated to effort
 - LOC is a best fit for this measurement

Elements for TSP/PSP success

- Focused and willing team
- Some experience with process or willingness to experiment
- Capable and committed project manager
- Committed and protective senior management
- Experienced and enthusiastic Coach
- Tools in place
- Training at all levels

Summary

- TSP/PSP provides a better implementation approach and make organization's journey faster to higher maturity level
- TSP/PSP provides additional support and strength required for high maturity organization to sustain process initiatives

References

- <http://www.sei.cmu.edu/tsp/>
- <http://processdash.sourceforge.net/>
- www.sstc-online.org/proceedings/2003/PDFFiles/pres1502.pdf
- PSPSM A self-Improvement Process for software Engineers by Watt S. Humphrey
- TSP/PSP at Intuit - Noopur Davis, SEI and Eileen Fagan, Intuit
- Critical Success Factors for TSP/PSP Adoption in a Consumer Software Company - Jim Sartain, Intuit

Acknowledgements

- The following are service marks of Carnegie Mellon University.

- Personal Software ProcessSM
- PSPSM
- Team Software ProcessSM
- TSPSM
- CMM IntegrationSM
- CMMISM

- The following are registered trademarks of Carnegie Mellon University

- Capability Maturity Model[®]
- CMM[®]

Q & A

Thanking You..

- **Sudhir P**
 - Senior Engineering Manager-QA
 - Sudhir_patnaik@intuit.com

- **Balasubrahmanyam P**
 - Senior Software QA Engineer
 - bpillalamarri@intuit.com