
Applying the Team Software Process at Intuit

Jason Chancey, Intuit

Noopur Davis, Davis Systems

SD Best Practices 2005
Boston, MA

Topics

- **Background**
- **Overview of TSP**
- **Highlights of standard development processes in QuickBooks division of Intuit**
- **Integrating TSP/PSP with Intuit QuickBooks processes**
- **Adoption of PSP by individual engineers**
- **Key successes of the application of TSP**
- **Key challenges to integrating TSP**
- **Planned improvements to be adopted by the pilot team for their next project**

The Best Projects

- The best projects are an artful balance of conflicting forces.
- They must consider
 - business needs
 - technical capability
 - customer desires
- If any facet is slighted, product quality will suffer.
- To build superior products, teams must understand the complete context for their projects.

The Best Teams

- **The best teams**
 - define their own processes
 - produce their own plans
 - make their own commitments
 - direct their own projects
 - regularly track and report on their work
 - consistently use their selected methods
 - develop the best products

- **The Team Software Process (TSP)**
 - builds self-directed teams
 - guides these teams in producing superior products

Team Software Process - 1

- The Team Software Process (TSP) is an integrated set of practices for developing software.
- TSP is an process-based solution to common software engineering and management issues.
 - cost and schedule predictability
 - productivity and product quality
 - process improvement
- Unlike some methods, TSP
 - teams are self-directed.
 - emphasizes measurement and quality management.
 - provides immediate and measurable benefits.
 - accelerates CMMI-based improvement.

The Team Software Process -2

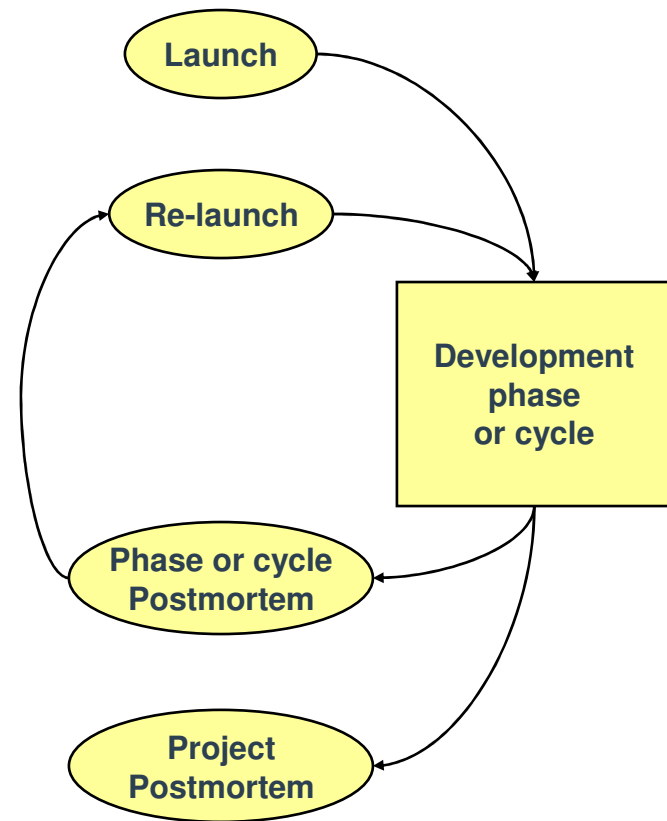
- **The objective of the TSP is to**
 - put software professionals in charge of their work
 - make them feel personally responsible for the quality of the products they produce
 - provide a team environment that supports individual excellence
 - put teams in charge of their processes and commitments
 - show teams how to produce processes and plans that best fit their needs

The Team Software Process -3

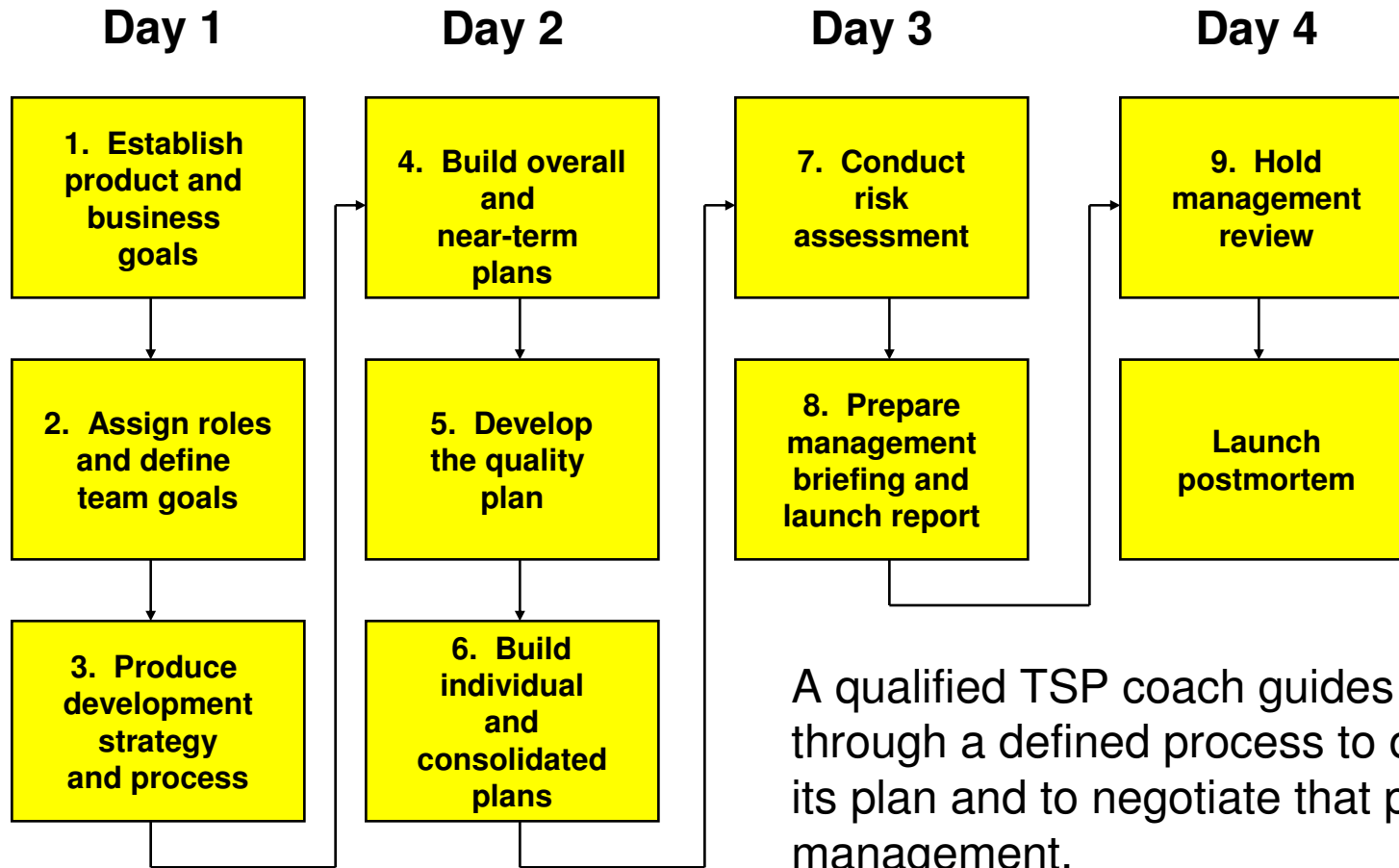
- **With TSP, software teams**
 - establish aggressive but realistic commitments
 - are always aware of their status and progress
 - keep management informed
 - manage their work to consistently meet their commitments

TSP Overview

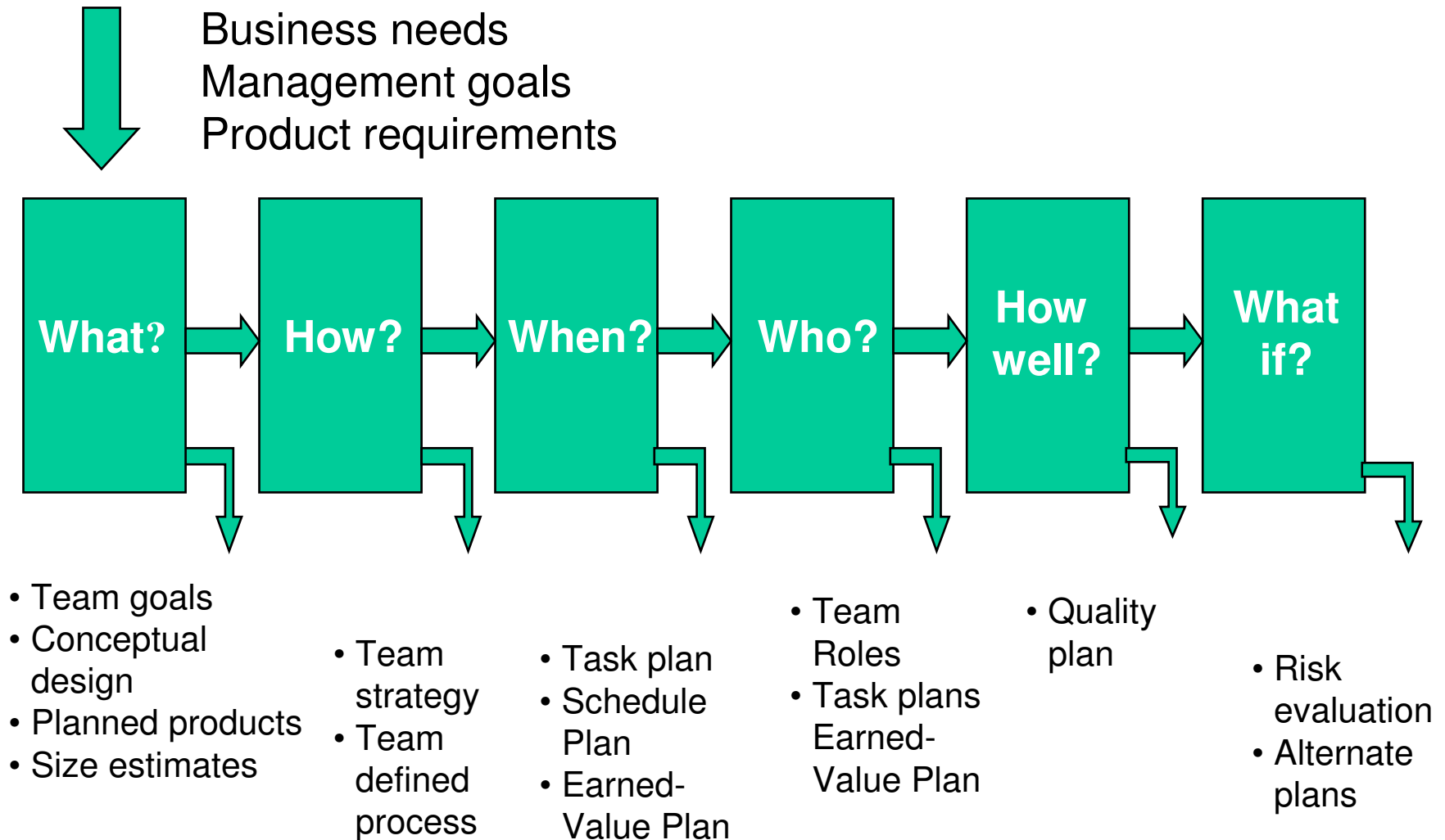
- The TSP is a framework and a process structure for building and guiding self-directed teams.
- The TSP addresses
 - team-building
 - team-working
- Each phase or cycle of a TSP project starts with a launch or re-launch.
- The standard strategy is to
 - develop in increments
 - use multiple cycles
 - work-ahead



The TSP Launch Process



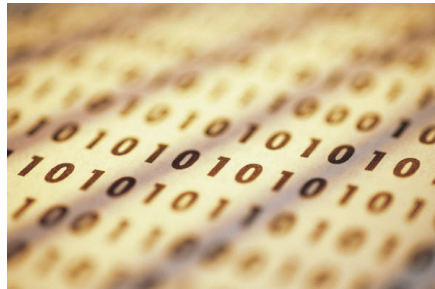
The TSP Launch Products



The Team Plan

- TSP team members produce a conceptual design.
- They identify needed tasks.
- They estimate their personal available task hours for each week.
- They produce the overall team schedule.
- With personal and TSP defect data, they estimate the yields and defects for each process step.
- They make and balance personal plans for the next few weeks.
- They produce a project risk assessment.
- They negotiate the team plan with management.

TSP Core Measures



Size



Effort



Quality



Schedule

- Source: CMU/SEI-92-TR-019

TSP Measurement Framework -1

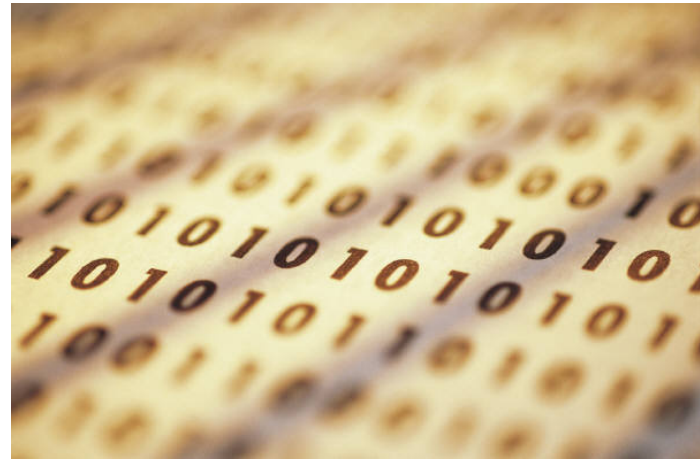
- **“Quality without numbers is just talk.”**
- **TSP uses four base measures.**
 - size
 - effort
 - defects injected and removed
 - schedule
- **Both plan and actual data are gathered and recorded.**
- **Actual data are used for tracking to the plan.**
- **The personal and team data are archived to provide a repository of historical data for future use.**

TSP Measurement Framework -2

- The TSP measurement framework consolidates individual data into a team perspective.
- Other views of the data are available e.g. by product or part, by phase, by task, by week, etc.
- Many other measures are derived from this base.

Estimated and Actual Size

- Size is a measure of the magnitude of the software deliverable, e.g. lines of code or function points.
- Size is estimated and reported for each component.
- The actual size is measured and reported for each component.
- Size data are used to
 - estimate effort
 - normalize other measures
 - track progress



Estimated and Actual Effort

- Effort is a measure of time “on task.”
- The recommended effort measure is a *task hour*.
- Task hours should be measured and reported by
 - component
 - process phase
 - project week



- How many task hours are there in a 40 hour week?
- *About 15 to 20.*

Estimated and Actual Schedule

- Schedule shows the work completed or planned to be completed in a given period.
- Schedule accuracy depends on granularity of tasks.
- Recommended task granularity
 - Less than 10 hours for near-term tasks
 - From one to four weeks for remaining tasks
- Estimated and actual task-completion dates are reported.



Estimated and Actual Quality

- Defects are a measure of quality.
- Definition: when a software product must be changed to ensure proper design, implementation, test, use, or maintenance, a defect has been identified.
- Estimated and actual defects injected and removed are reported.

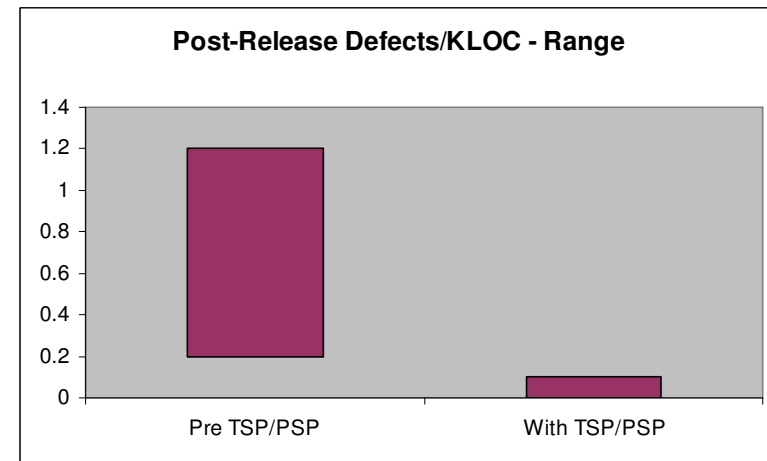
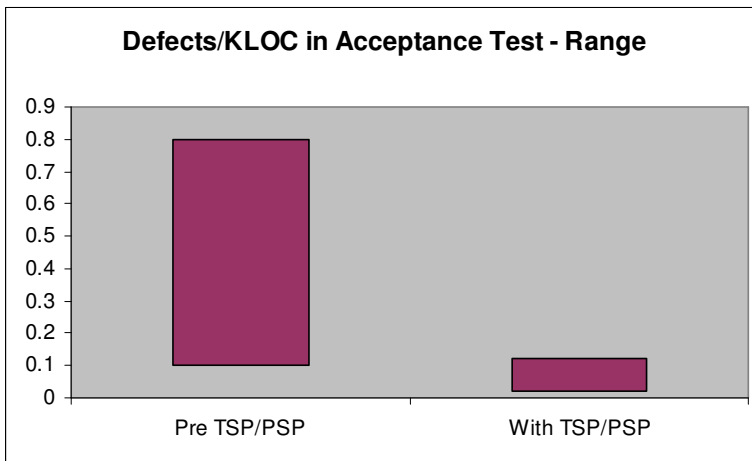
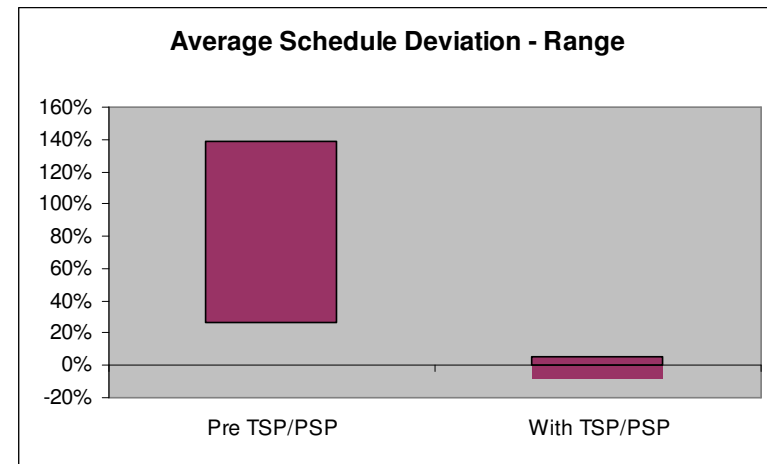
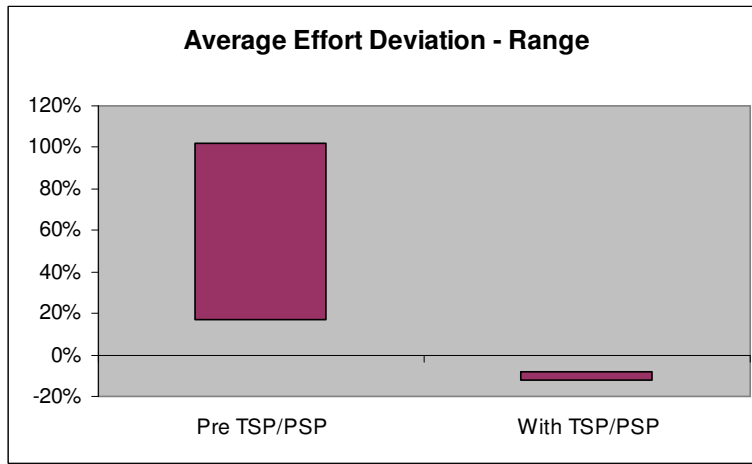


Derived Measures

- The core measures support many other derived measures that are used to evaluate plans and track program status.

- Examples include
 - planned value, earned value, and predicted earned value
 - estimation accuracy for size, time, and quality
 - time in phase distribution
 - defect density
 - defect removal rate
 - review rates
 - process and phase yield
 - cost of quality
 - percent defect-free
 - defect removal profiles
 - quality profile and process quality index

TSP Improves Performance



Source: CMU/SEI-2000-TR-015

TSP Performance Comparison

Performance Category	Typical Industry Performance	TSP Impact Study (2003)
Schedule Error	180% avg.	6% avg.
Effort Error	130% avg.	26% avg.
System Test defects per KLOC	5 to 25 approx.	0.4 avg.
Released defects per KLOC	1 to 7	0.06 avg.

Source: TSP projects from CMU/SEI-2003-TR-014, a study of 20 projects in 13 organizations conducted in 2003.

Adoption

- Some of the organizations using, piloting, or preparing to pilot the TSP.

ABB
ABC Informatica
Advanced Information Services
Advanced Maturity Services, Inc.
Alan S. Koch Consultants
Ambient Consulting
AMCOM*
Boeing
Centre De Investigacion En
Matematicas
Census Bureau*
Carnegie Mellon University*
CQG, Inc.
CRSIP / STSC / DRAPER
Davis Systems
DOE / Los Alamos
DOE / Naval Reactors
DPC Cirrus
Dynamics Research Corp.
EDS
Hallex Associates
Heath Solutions, Inc.

Helsana*
Honeywell
IBM
Intuit*
Iomega
I.Q. Inc.
KPMG
L. G. Electronics
Lockheed Martin
LogiCare
Los Alamos National Laboratory
M/A-Com Private Radio Systems, Inc.
Microsoft*
Motiva
NASA Langley
NAVAIR
Naval Reactors*
NAVOCEANO*
NUWC
NCR/Teradata*
NCS Pearson
Northern Horizons

Northrop Grumman
NSWC / Keyport
Prodigia S.A. de C.V.
PS&J Consulting - Software Six Sigma
QuarkSoft
Respironics
Rockwell Collins
SAIC
Samsung SDS
Siberlink
STPP, Inc.
STSC
Trilogy
TYBRIN Corporation - Air Logistics
University of Alabama / Huntsville
University of Queensland
Vicarious Visions*
Xerox

***Organizations we are currently
working with**

Intuit and TSP

The Team Software Process (TSP) promises

- radical improvements in quality
- superior project status visibility
- predictability
- efficiency
- a framework for continual improvement

Questions

- How does TSP fit into existing culture and processes?
- Can TSP promises be fulfilled when working with a complex code base that has evolved over more than 10 years?

Intuit QuickBooks Process Highlights

- Requirements development
- User Interface design and specification
- Technical designs
- Release Commit
- Implementation
- Code Complete
- Functional test complete/UI freeze
- System test complete
- Beta ready
- Shutdown begins
- Manufacturing Release

Note: Phases overlap as needed. Phases shown here apply to software developers, not to systems testers or other functions in the organization.

Integrating TSP with Intuit QuickBooks Processes

Feature	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	
Feature 1	Implement part 1				Implement part 2				Imp. part 3	Implement part 4				
Feature 2	Requirements							Implement feature 2 framework			Implement feature 2			
Feature 3	Implement part 1			Implement part 2						Imp. part 3	Implement part 4			

Launch ↑

Relaunch ↑

Keys to success:

- Immediate PD start
- Extreme parallelism
- Incremental delivery
- Radically high quality (TSP/PSP)
- Aggressive tracking (TSP)

Integrating PSP with Intuit QuickBooks Processes

Feature	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	
Feature 1	Implement part 1				Implement part 2				Imp. part 3	Implement part 4				
Feature 2	Requirements							Implement feature 2 framework			Implement feature 2			
Feature 3	Implement part 1			Implement part 2						Imp. part 3	Implement part 4			

PSP applied during implementation

- Design, personal design review, design peer review
- Code, personal code review, code peer review
- Unit test

Adoption of PSP by Individual Engineers

- PSP was adopted to varying degrees
- All engineers kept detailed time logs.
- All engineers recorded defects, especially defects detected in inspection and test.
- All engineers kept their task plans up to date.
- All engineers provided weekly status to the team.
- Some engineers embraced the principles of the PSP, while others remained lukewarm.

Key Successes of the Application of TSP

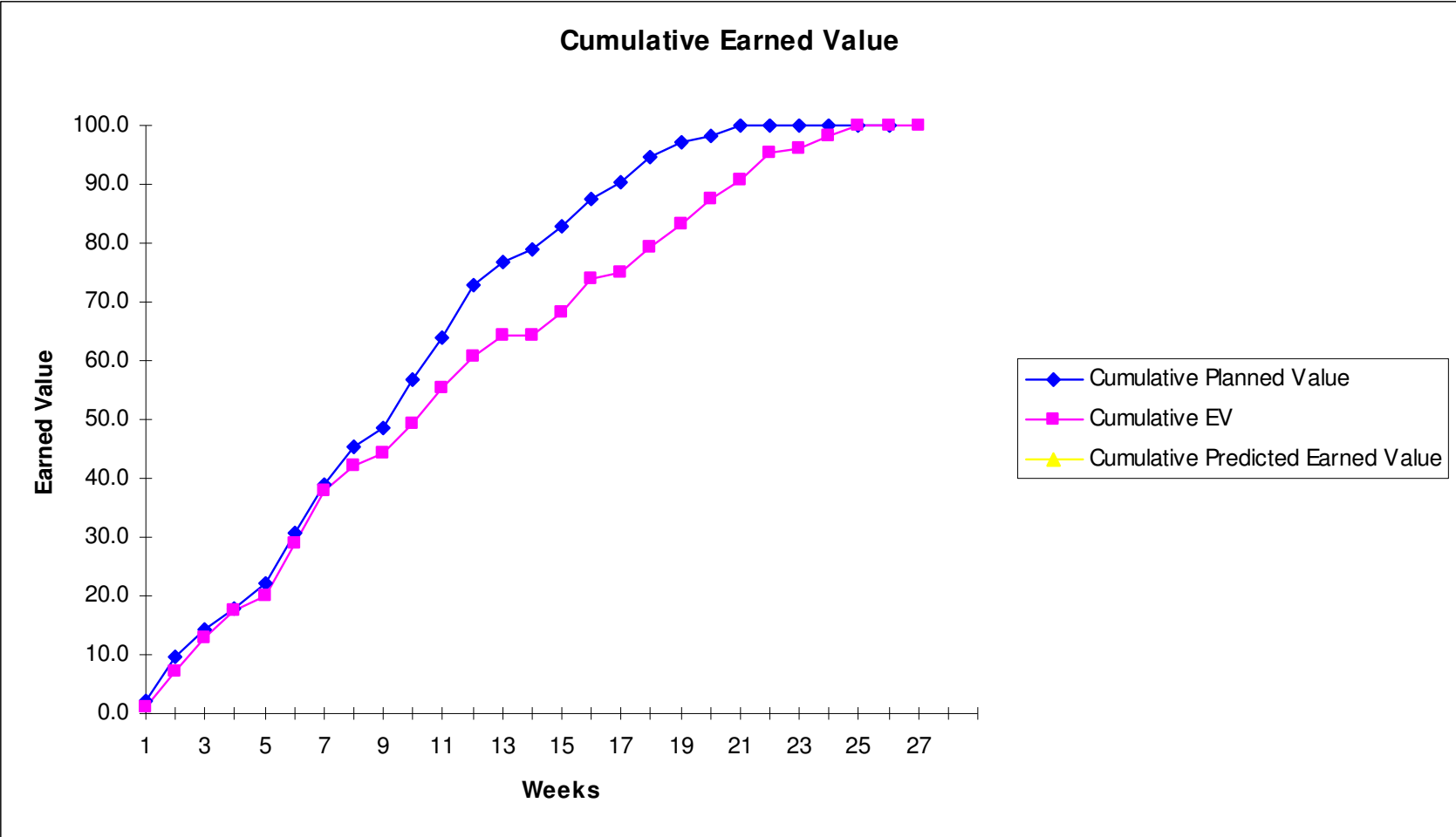
- Increased visibility into project status
- Improved quality
- Longer development cycle
- Team involvement

Increased Visibility Into Project Status

Each team member, as well as the team as a whole, has detailed insight into project status

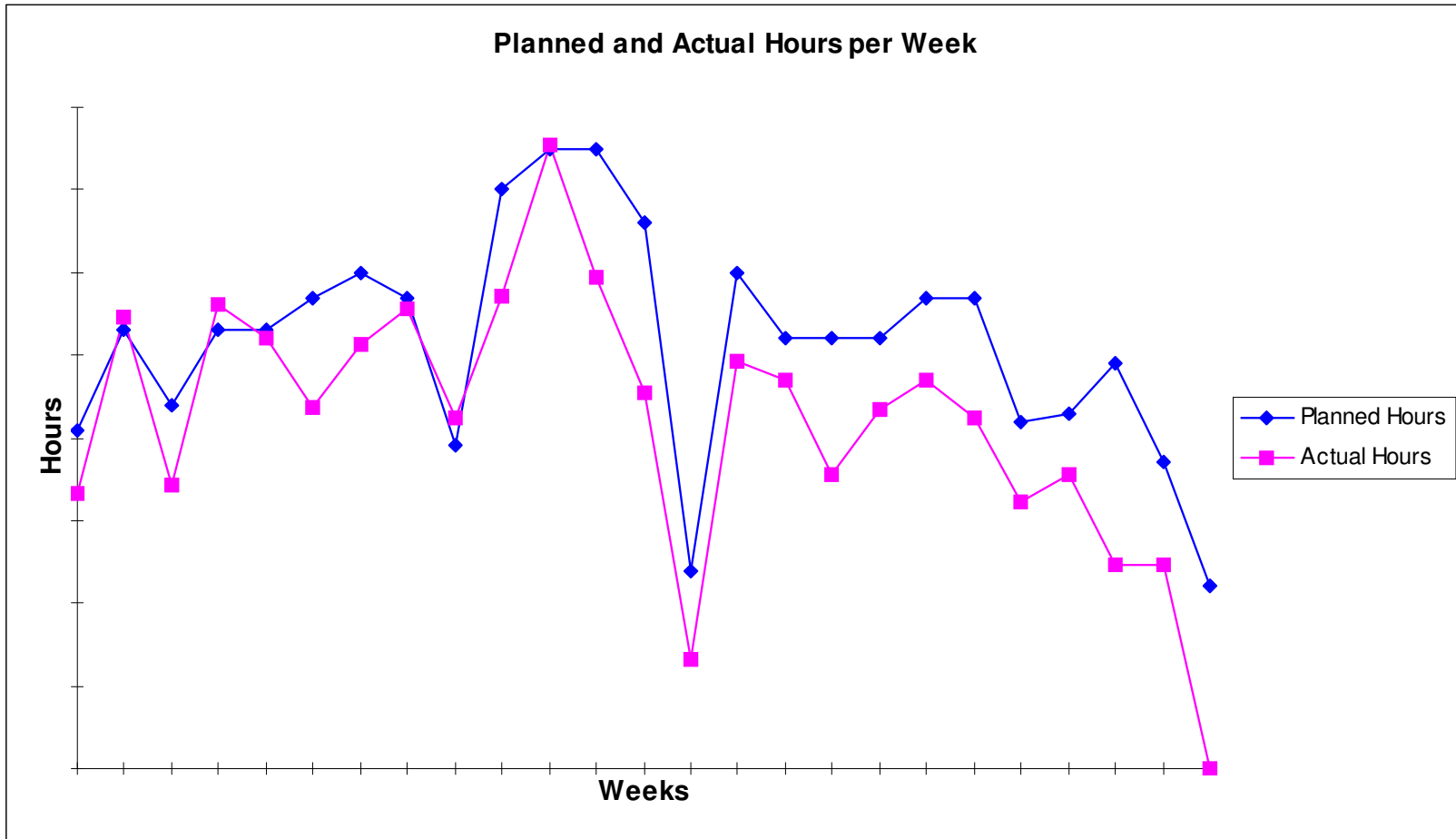
- Earned value
- Quality information from early phases
- Task hours
- Tasks completed
- Tasks remaining

Earned Value At Project Completion

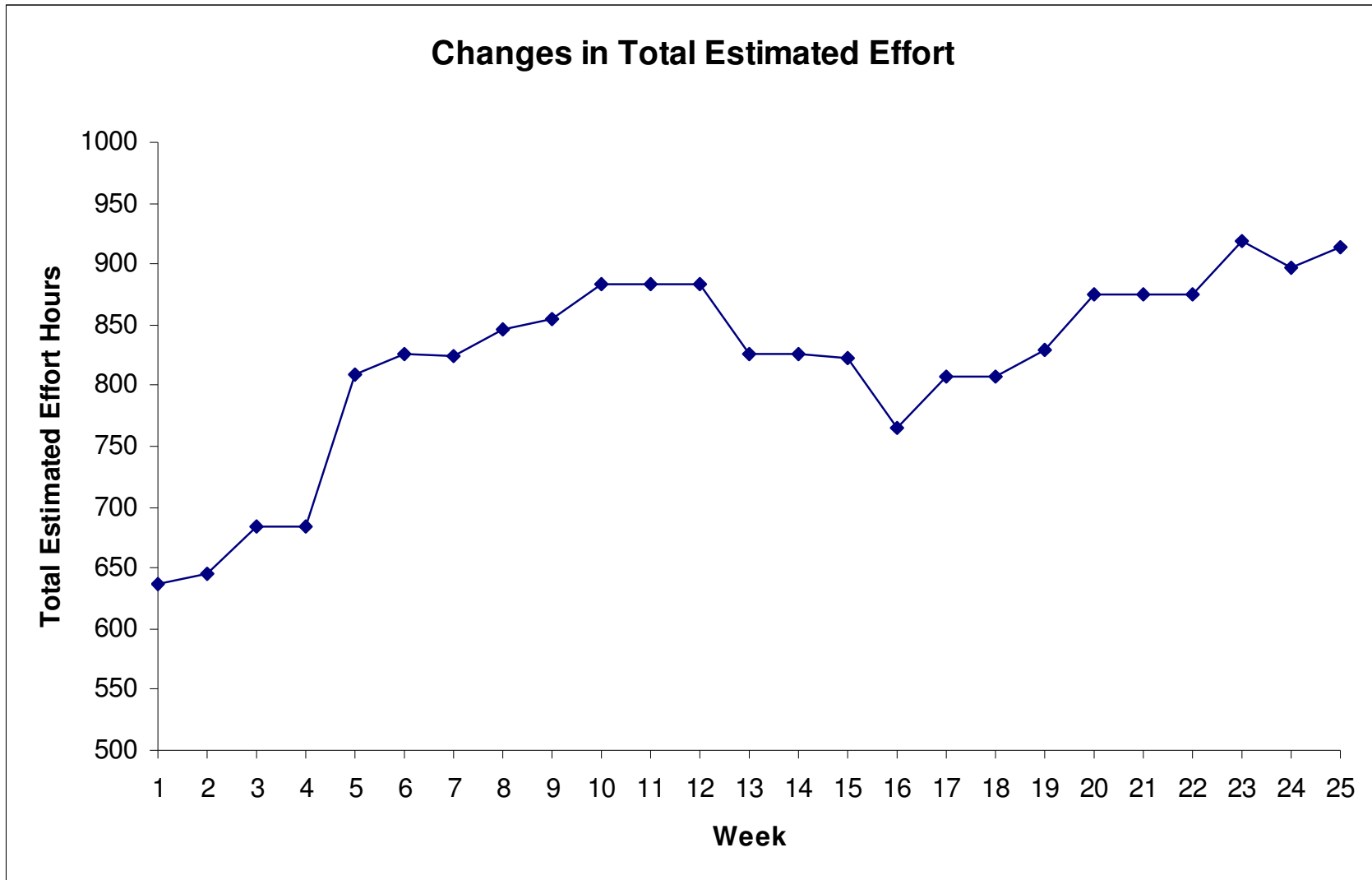


Task Hours

Mid-way through the project, people started rolling off.



Importance Of Re-Planning



Weekly Status -1

Some weeks were better...

Weekly Data	Plan	Actual	Plan / Actual
Schedule hours for this week	60.0	51.3	1.17
Schedule hours this cycle to date	361.0	325.0	1.11
Earned value for this week	8.1	8.8	0.92
Earned value this cycle to date	38.8	37.7	1.03
To-date hours for tasks completed	344.4	326.5	1.06
To-date average hours per week	51.6	46.4	1.11
EV per completed task hour to date	0.113	0.116	

Weekly Status -2

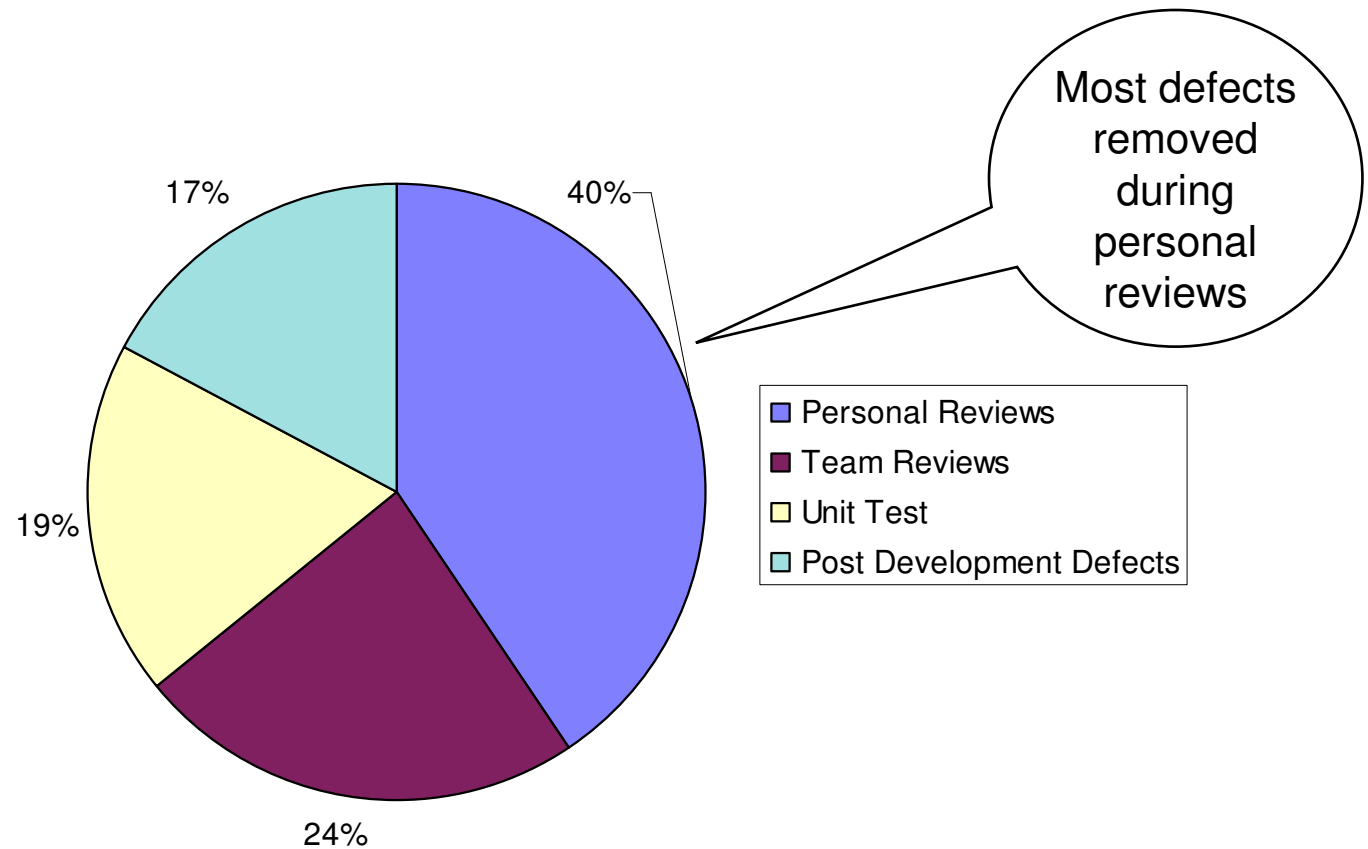
...than other weeks!

Weekly Data	Plan	Actual	Plan / Actual
Schedule hours for this week	70.0	57.1	1.23
Schedule hours this cycle to date	527.0	480.2	1.10
Earned value for this week	8.1	4.8	1.69
Earned value this cycle to date	56.6	49.2	1.15
To-date hours for tasks completed	449.2	463.3	0.97
To-date average hours per week	52.7	48.0	1.10
EV per completed task hour to date	0.126	0.106	

Plan vs. Actual

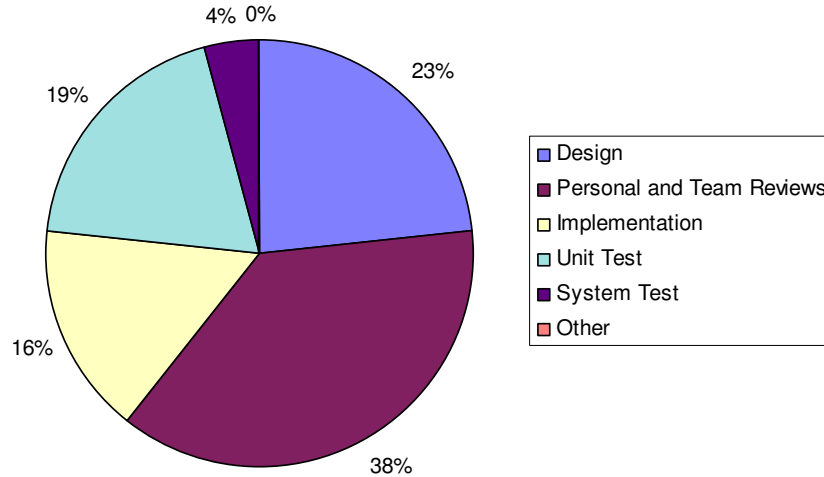
	Actual/Plan (Final/Re-launch)
Size (New and Changed LOC)	1.58
Effort (hours)	1.27
Schedule	1.22
Productivity (New and Changed LOC/Hr)	1.24

Percent Defects Removed By Activity



Component Analysis

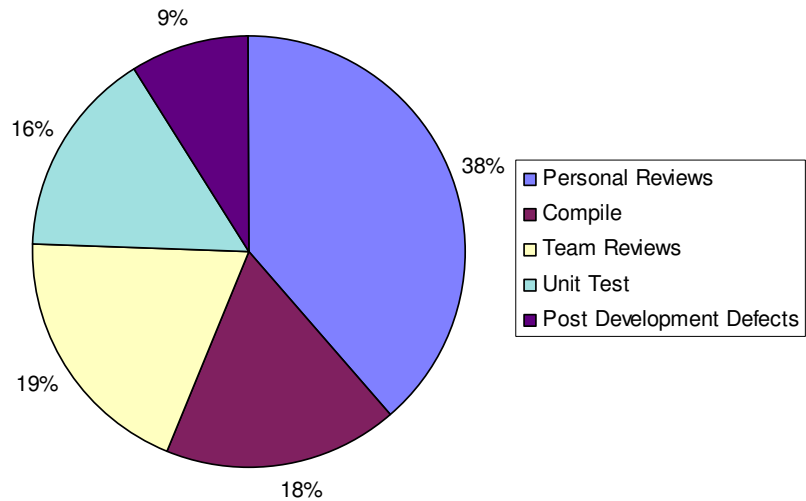
Percent Effort by Activity



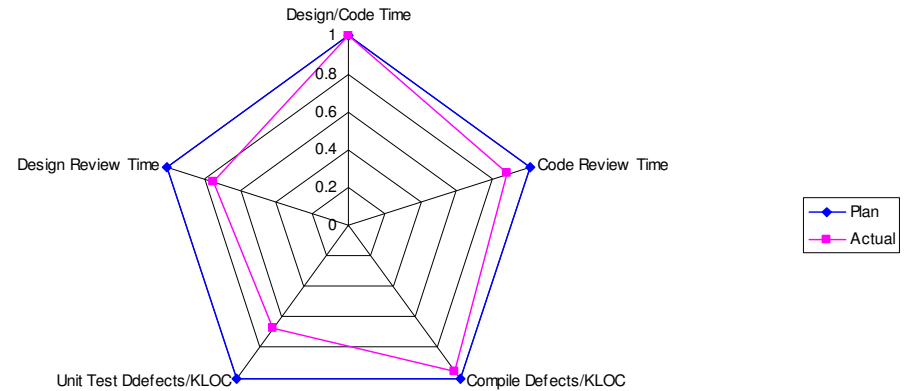
Program Size

	Plan	Actual
Total Requirements Pages (SRS)	0	0
Total HLD Pages (SDS)		
Total Detailed Design Lines		
Base LOC (B)	0	0
Deleted LOC (D)	0	2
Modified LOC (M)	0	24
Added LOC (A)	862	892
Reused LOC (R)	0	0
New and Changed LOC (N)	862	916
Total LOC (T)	862	890
Total New Reuse LOC	0	92

Percent Defects Removed by Activity

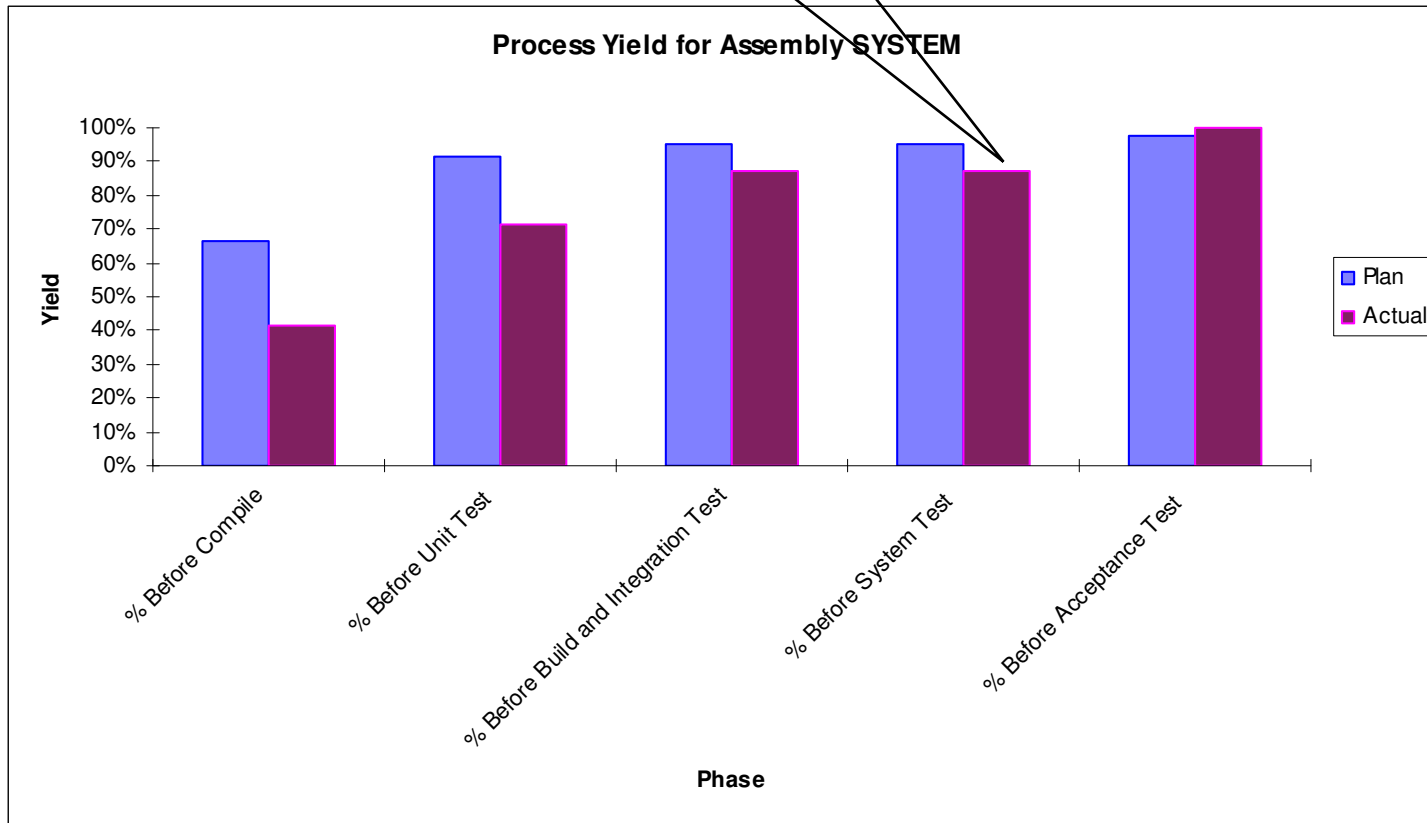


Quality Profile for Assembly JobCostsByVendor Reports



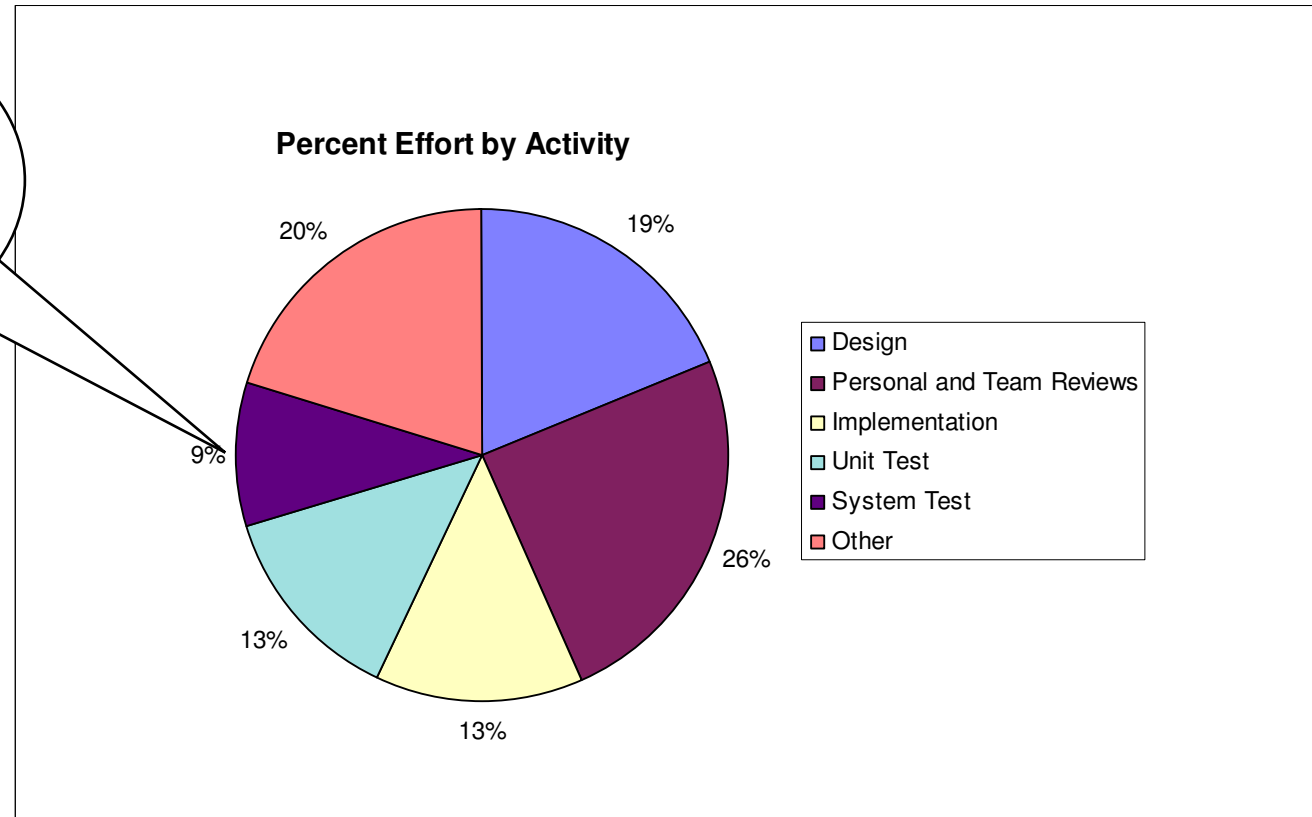
Process Yields

87% of known defects removed before system test



Longer Development Cycle

Compare to non-TSP teams who typically spend 50% supporting system test!¹



¹ Source: *The Team Software Process in Practice: A Summary of Recent Results*, Davis and Mullaney, SEI Technical Report CMU/SEI-2003-TR-014, <http://www.sei.cmu.edu/publications/documents/03.reports/03tr014.html>

Team Member Involvement -1

- Team member comments during the project postmortem
 - *“Beginning to like the process. Makes interaction with people more efficient. You know what other team members are doing.”*
 - *“Liked clear definition of what people are responsible for. Promotes ownership of tasks.”*
 - *“Lots of things I liked. The power it gives us at getting better at estimating and planning. All the fun data it gives us to see how we can improve. There is a shift in the mental sense to accept the fact that there are defects, and where we can improve is what to do about the defects.”*

Team Member Involvement -2

- Team member comments (continued)
 - *“It protects us from ourselves. The task plan includes the things that we always say we will do...and it helps us feel good about them when we do them.”*
 - *“Wish requirements were better expressed. Very little guidance exists for requirements (in the TSP).”*
 - *“Logging defects early gives an indication of remaining defects.”*

Team Member Involvement -3

- Team member comments (continued)
 - *“The tool is not flexible enough.”*
 - *“The tool was my main complaint.”*
 - *“The TSP creates a lot of interdependencies, but the tool does not help you track them.”*
 - *“Logging every little change I made as a defect was difficult.”*
 - *“Almost an overbearing importance on system test defects. Some system test defects were not very important at all.”*

Key Challenges to Integrating TSP

- **The TSP tool could improve for**
 - managing dependencies
 - managing milestones
- **PSP training**
- **Communication**
 - with non-TSP teams
 - with Release Management
- **Launching using industry data rather than your own**
- **Balancing roles**
 - Manager/Team Lead/Coach/Planning Manager
 - Team Roles (Planning Manager, Quality Manager...)

Planned Improvements -1

- **Apply TSP to requirements phase.**
 - include personal review
 - include team inspection
 - develop specific checklists
 - log time spent and defects found
- **Include architects in all design inspections.**
- **Include code champions in code inspections.**
- **Separate our high-level and detailed designs, with personal reviews and inspections for both.**
- **Develop list of QuickBooks-specific assumed behaviors. Use this checklist to help review and inspect designs.**

Planned Improvements -2

- During initial launch, focus on getting detail for requirements and plan for requirements activities.
- Investigate conceptual design before the launch. Let architects review conceptual design during the launch.
- Manage expectations so organization understands that re-planning will occur.
- Full cross-functional participation in the launch.

Conclusion

- **What worked well**
 - Team commitment to trying the processes
 - Earned value tracking focused us on our task plans, and protected our quality assessment activities
- **What did not work well**
 - Should have had Product Manager more involved during launch
 - Need to separate our high-level and detailed designs
 - Want to apply to requirements phase to reduce downstream defects

Contact Information

Jason Chancey

- jason_chancey@intuit.com

Noopur Davis

- ndavis@davissys.com

Visit the SEI TSP web site at

- <http://www.sei.cmu.edu/tsp>