Introduction to the Team Software Process

James Over
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Tutorial Objectives

This tutorial answers the following questions:

- What is the Team Software Process?
- What does the Team Software Process do?
- How does the Team Software Process work?
- How do the Team Software Process and CMMI relate?
- What is the experience with the Team Software Process ?
- How do you introduce the Team Software Process ?

Agenda

When	•Topics
9:00 – Break	•Case study: a project in trouble •Team Software Process and its implementation strategy •TSP concepts
Break – Lunch	•Why projects fail •Case study: launching the project
Lunch – Break	•Case study: launching the project (continued)
Break – 5:30	•Case study: team-working framework •Corporate experience with TSP •TSP and CMMI •Building internal support for TSP

The Project

Day 1

Management was under great pressure to put out a new version of their primary software product.

Marketing was demanding a release within 9 months.

The development staff thought this was impossible.

A previous project with similar scope and resources took two years to complete.

You’ve been asked to lead the project. What would you do?

Your Choices

What do you think of the schedule?

- Whose date is 9 months?
- How does this compare with prior projects?
- Do you agree?

What are your choices?

- Accept the 9 month schedule.
- Complain and then accept it.
- Say you will do it, but not in 9 months.
- Update your resume.

The way you answer will determine whether the project succeeds or fails.

High Profile, High Risk Projects

Discussion topic

- You’ve been asked to lead the project.
- What would you do?

10 minutes

The Project

Day 2

The proposed schedule for the project was impossible.

Tom, the newly appointed manager of the project, and Bob, a software architect, were frustrated by management’s

- unreasonable schedule demands.
- lack of concern for software quality.

They decided to investigate potential solutions, including the TSP.

Team Software Process (TSP)

TSP is a process that is specifically designed for software teams.

It’s purpose is to build high-performance teams and help them

- plan their work
- negotiate their commitments with management
- manage and track projects to a successful conclusion
- produce quality products in less time
- achieve their best performance without the “death march” ending

Reliable Estimates

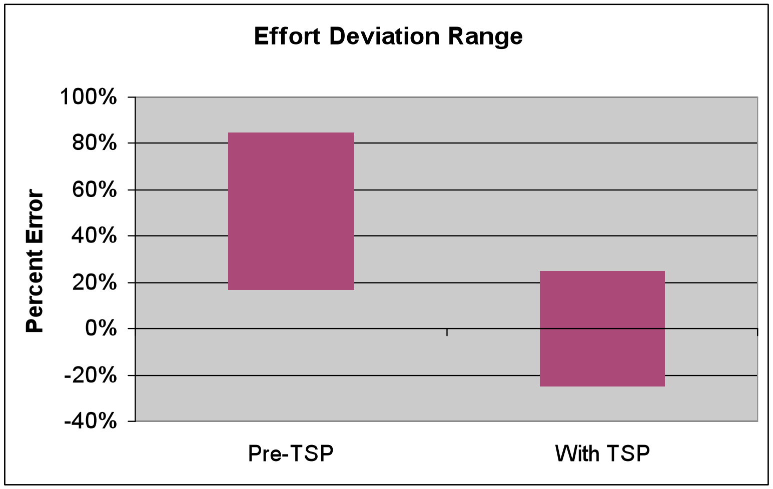
From a study of fifteen projects in four organizations at all maturity level except ML4.

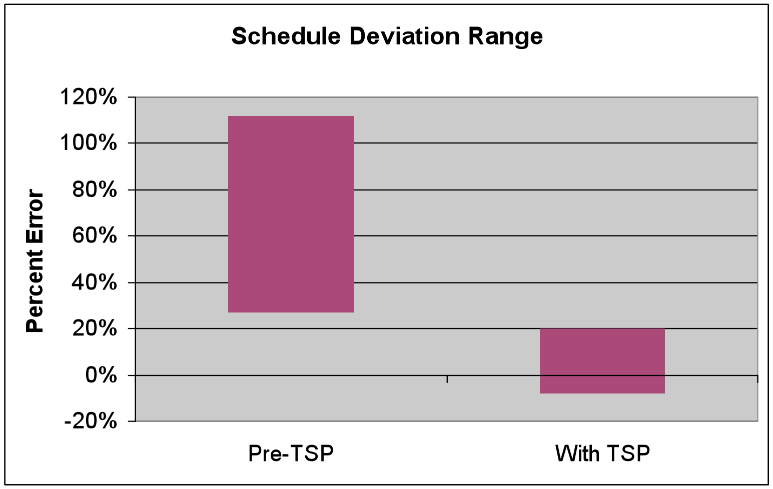
TSP improved effort and schedule predictability on every project.

Effort (Cost) Performance	
Study baseline	+17% to +85%
TSP	-25% to +25%

Schedule Performance	
Study baseline	+27% to +112%
TSP	-8% to +20%

Source: CMU/SEI-TR-2000-015





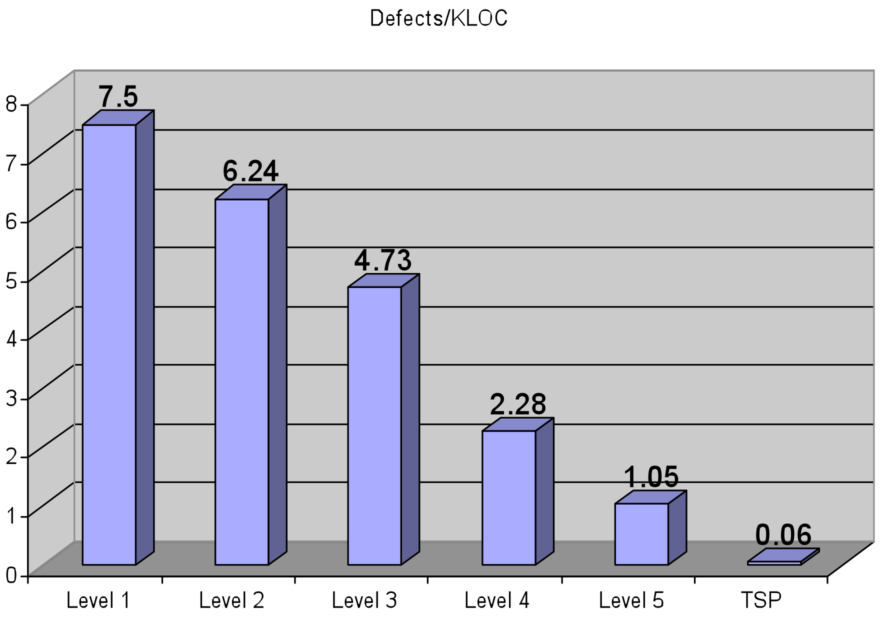
Reliable Products

From a study of 20 projects in 13 organizations at all maturity levels.

TSP teams averaged 0.06 defects per thousand lines of new or modified code.

Approximately 1/3 of these projects were defect-free.

These results are substantially better than those achieved in high maturity organizations.



Source: CMU/SEI-2003-TR-014

TSP Impact and Performance Results

Performance Category	Mean	Min.	Max.	# Data Points
Effort estimation error	5%	-24%	25%	21
Schedule estimation error	6%	-20%	27%	21
System test effort**	4%	2%	7%	21
Cost of quality	17%	4%	38%	21
Product quality*	0.06	0.0	0.2	21

*Post-release defects reported thousand new or modified lines of code

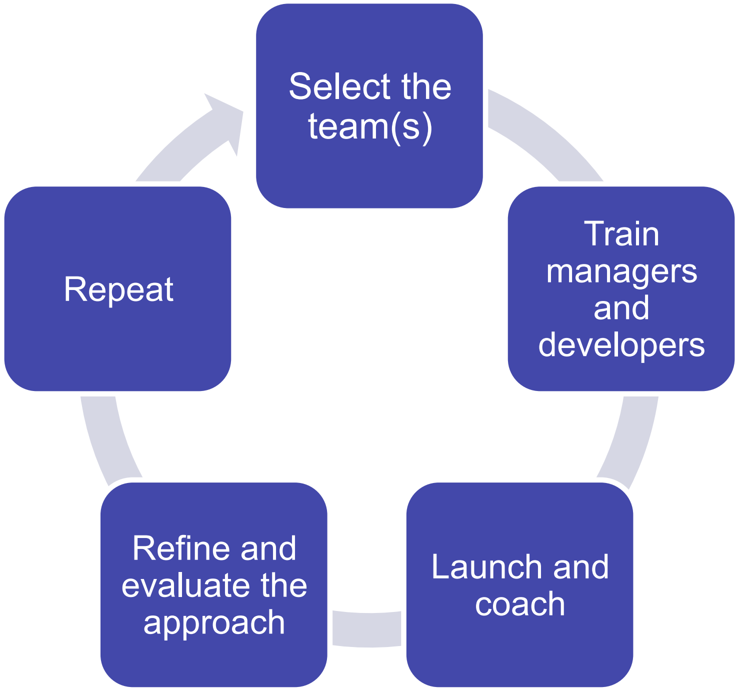
**System test effort as a percentage of total development effort

Source: Davis, N.; & Mullaney, J. The Team Software Process in Practice: A Summary of Results (CMU/SEI-2003-TR-014)

TSP Implementation Strategy

TSP is implemented project-by-project.

- Select two or three teams.
- Train top-down, starting with senior managers, then project managers, then team members.
- When the managers and team are trained, conduct a TSP Launch to kick-off each project.
- Evaluate and fine tune the approach.
- Repeat this cycle increasing scope at a sustainable pace.


Select the team(s)
Train managers and developers
Launch and coach
Refine and evaluate the approach
Repeat

TSP Product Suite: Process, Training, Tools

Process Notebook

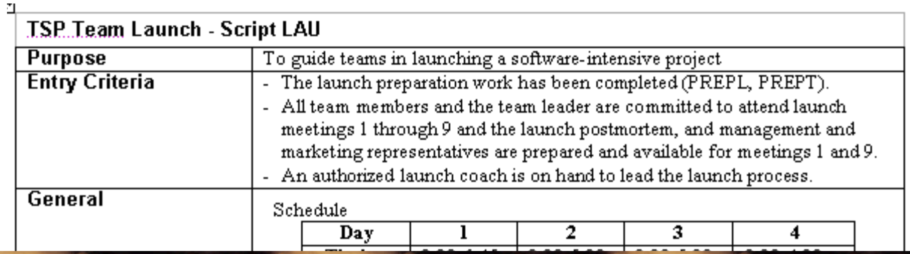
- Process scripts
- Forms
- Guidelines and standards
- Role descriptions

Training and Textbooks

- Executives
- Project Managers
- Engineering
- TSP Coach
- TSP Trainer

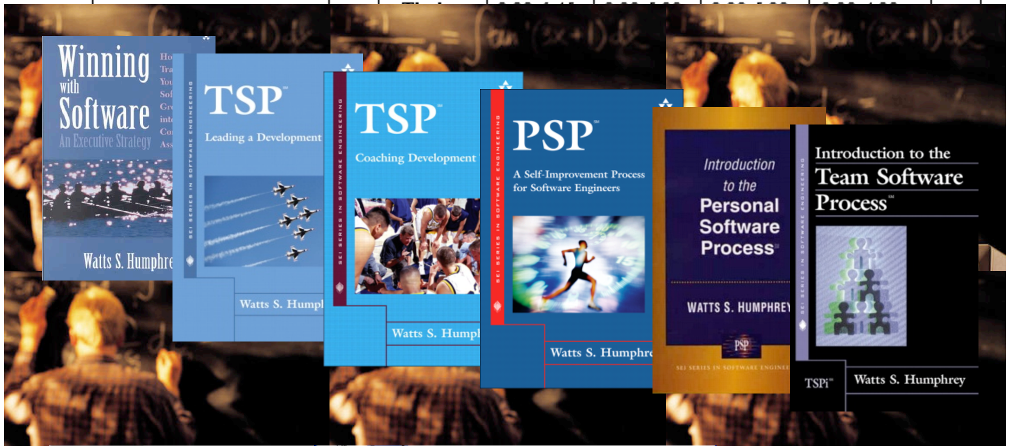
Tools

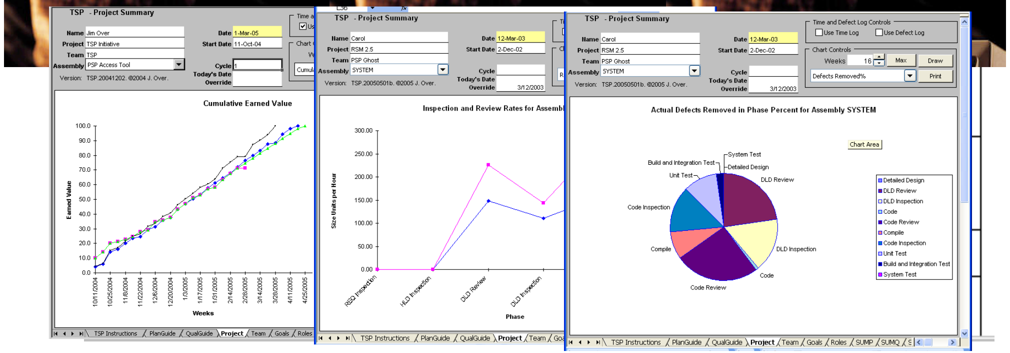
- TSP Workbook
- PSP Workbook
- Coach/Trainer Workbook



TSP Team Launch - Script LAU	
Purpose	To guide teams in launching a software-intensive project
Entry Criteria	- The launch preparation work has been completed (PREPL, PREPT). - All team members and the team leader are committed to attend launch meetings 1 through 9 and the launch postmortem, and management and marketing representatives are prepared and available for meetings 1 and 9. - An authorized launch coach is on hand to lead the launch process.
General	Schedule

Day	1	2	3	4





TSP - Project Summary

Name: Jim Over
Project: TSP Initiative
Team: TSP
Assembly: PSP Access Tool
Date: 1-Mar-05
Start Date: 11-Oct-04
Cycle: 1
Today's Date Override:
Version: TSP.20041202. ©2004 J. Over.

TSP - Project Summary

Name: Carol
Project: RSM 2.5
Team: PSP Ghost
Assembly: SYSTEM
Date: 12-Mar-03
Start Date: 2-Dec-02
Cycle:
Today's Date Override: 3/12/2003
Version: TSP.20050501b. ©2005 J. Over.

TSP - Project Summary

Name: Carol
Project: RSM 2.5
Team: PSP Ghost
Assembly: SYSTEM
Date: 12-Mar-03
Start Date: 2-Dec-02
Cycle:
Today's Date Override: 3/12/2003
Version: TSP.20050501b. ©2005 J. Over.

Time and Defect Log Controls
☐ Use Time Log ☐ Use Defect Log
Chart Controls
Weeks: 16
Defects Removed%

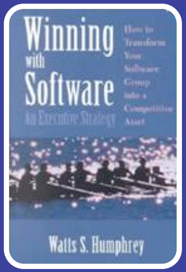
The Project Timeline Using TSP

Bob and Tom thought that the TSP's project-focused, rapid deployment strategy would be a perfect fit for the high-risk project.

They constructed this timeline and convinced the head of QA to fund the training and support.

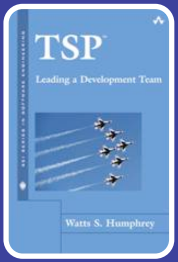
Task	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Training	➡									
TSP Executive Strategy Seminar	♦									
Leading Development Teams	♦									
PSP Fundamentals	♦									
Product Development		■ ■	▬	▬	▬	▬	▬	▬	▬	➤
Launches and Re-Launches		♦			♦			♦		
Postmortems				♦			♦			♦

TSP Training


Winning with Software
An Executive Strategy
How to Transform Your Software Group into a Competitive Asset
Watts S. Humphrey

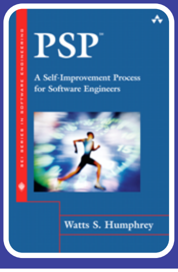
TSP Executive Strategy Seminar

- Building a “winning” organization
- Managing with facts and data
- One-day course



Leading a Development Team

- Building self-directed teams
- Motivating and leading self-directed teams
- Three-day course



PSP for Software Developers

- Using a defined and measured personal process
- Personal planning and tracking
- Personal quality management and design
- Five-day course

The Training Problem

The cost of training is known; the cost of not training is often ignored.

TSP changes the way managers and developers work, without proper training, managers and developers won’t understand TSP.

Without understanding they will continue to work as they always have with the same result.

TSP Concepts

Managing self-directed teams

Using processes and measures in engineering/creative work

Quality management

Management Styles

The principal management styles have been:

Frederick Taylor

Peter Drucker

Body Management

People as oxen that must be driven, directed, and motivated through fear.

Task Management

People as machines. Management knows the best way to get the work done. The workers follow.

Knowledge management

People as individuals. The knowledge worker knows the best way to get the work done. Management motivates, leads, and coaches.

Knowledge Work

“The key rule in managing knowledge work is this: managers can’t manage it, the workers must manage themselves.”

Software development is knowledge work.

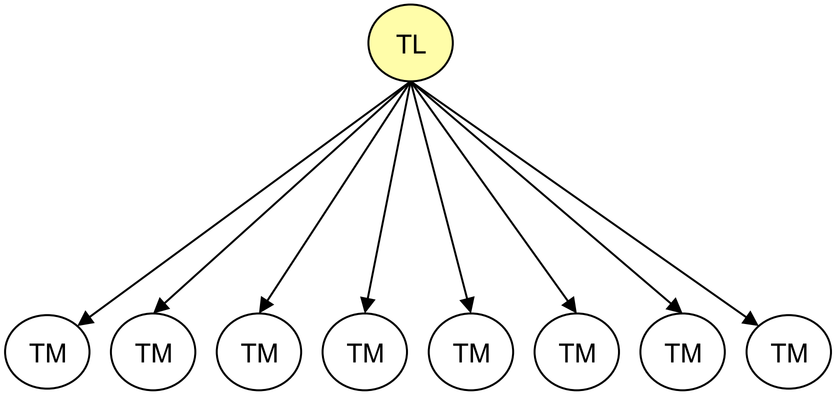
To manage software work, developers must

- be motivated
- make accurate plans
- negotiate commitments
- track their plans
- manage quality

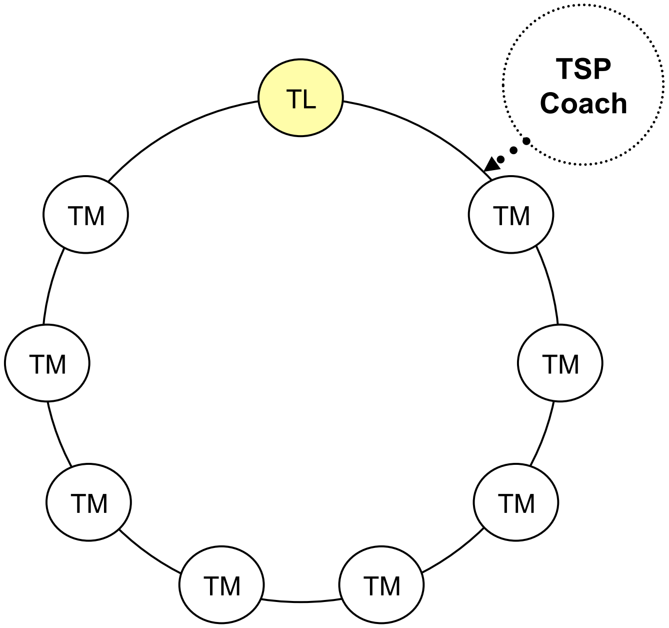
How is this accomplished?

Watts Humphrey,
creator of TSP

TSP Self-directed Team Management Style



Traditional team
The leader plans, directs, and tracks the team’s work.



Self-directed team
The team members participate in planning, managing, and tracking their own work.

The Project Manager or Team Leader’s Role

The team leader’s job on a TSP team is to

- guide and motivate the team in doing its work
- take the time to reach full consensus on all important issues
- ensure that the team establishes high standards for the work
- provide management support to the team
- support the team with management
- protect the team so that it can concentrate on the project

The TSP Coaching Role

The coach

- trains and facilitates the adoption of TSP
- works with the team leader to build the team
- observer that guides the team

Tiger Woods and his former coach, Hank Haney.

Team Leader vs. Coach

The team leader’s job is to use the team to build the product.

The coaches job is to use the project to build the team.

TSP Concepts

Managing self-directed teams

Using processes and measures in engineering/creative work

Quality management

Learning to Develop Software

In universities,

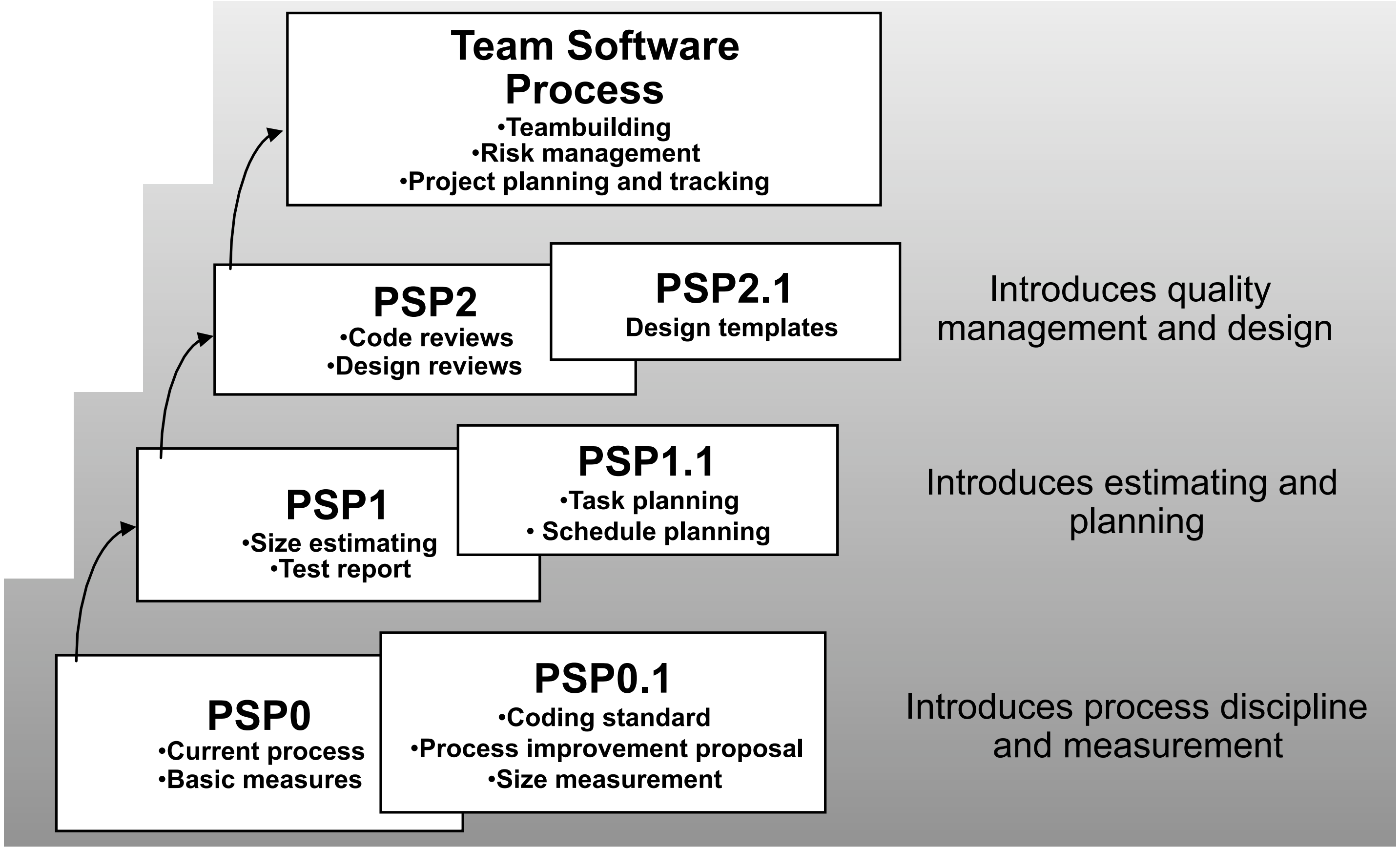
- the emphasis is on technical knowledge and individual performance.
- evaluation emphasizes code that runs, not how the student got there.
- the prevailing ethic is to code quickly and fix the problems in test.

In industry, team-working skills are also needed.

TSP uses the Personal Software Process to build these skills.

- planning and tracking the work
- measuring and managing quality
- anticipating and correcting problems

PSP Learning Stages



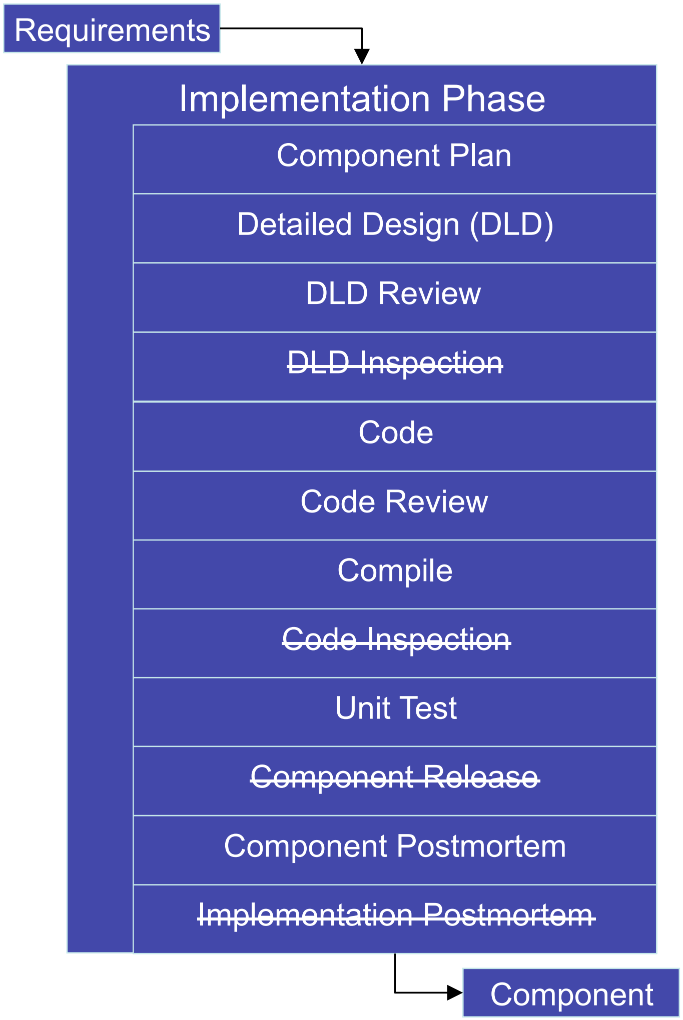
Developers write one or more programs at each PSP level

Using A Defined Process

The PSP process is like the TSP implementation phase, but without inspections, component release, and the implementation phase postmortem.

Developers learn the PSP by writing small programs and measuring the result.

They convince themselves of the benefits and also learn how to apply the concepts to their own work.



Requirements

Implementation Phase

Component Plan

Detailed Design (DLD)

DLD Review

~~DLD Inspection~~

Code

Code Review

Compile

~~Code Inspection~~

Unit Test

~~Component Release~~

Component Postmortem

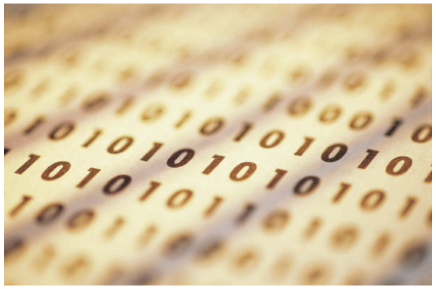
~~Implementation Postmortem~~

Component

The TSP/PSP Measurement Framework

Schedule

Effort



Size

Quality

Four direct measures apply to all processes and products.

- Estimates made during planning
- Directly measured by team members while working

The data are used to track project status and to analyze and improve performance.

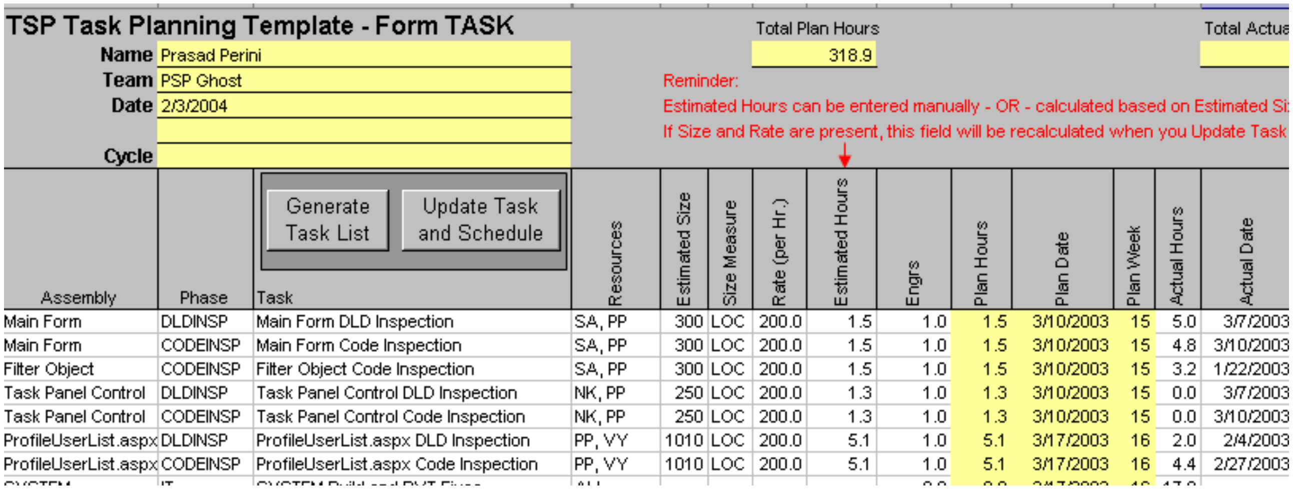
Benefit – direct measures, integrated into a measurement framework, provide flexibility.

Schedule

Schedule is the most commonly used project measure.

Schedule accuracy depends on granularity.

TSP schedule granularity is in hours, not days, weeks, or months.



TSP Task Planning Template - Form TASK

Name	Prasad Perini
Team	PSP Ghost
Date	2/3/2004
Cycle	

Total Plan Hours: 318.9

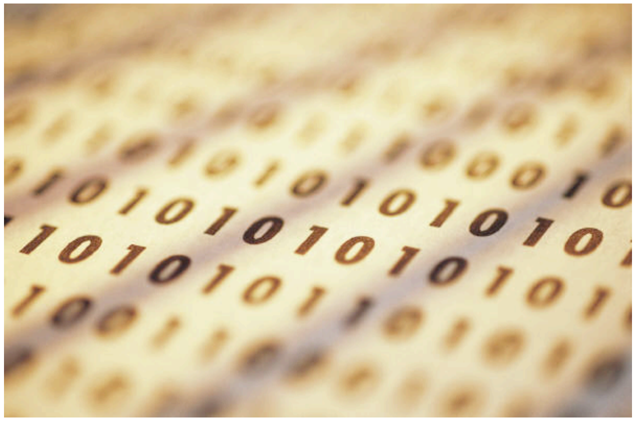
Total Actua

Reminder:
Estimated Hours can be entered manually - OR - calculated based on Estimated Si
If Size and Rate are present, this field will be recalculated when you Update Task

Assembly	Phase	Task	Resources	Estimated Size	Size Measure	Rate (per Hr.)	Estimated Hours	Engrs	Plan Hours	Plan Date	Plan Week	Actual Hours	Actual Date
Main Form	DLDINSP	Main Form DLD Inspection	SA, PP	300	LOC	200.0	1.5	1.0	1.5	3/10/2003	15	5.0	3/7/2003
Main Form	CODEINSP	Main Form Code Inspection	SA, PP	300	LOC	200.0	1.5	1.0	1.5	3/10/2003	15	4.8	3/10/2003
Filter Object	CODEINSP	Filter Object Code Inspection	SA, PP	300	LOC	200.0	1.5	1.0	1.5	3/10/2003	15	3.2	1/22/2003
Task Panel Control	DLDINSP	Task Panel Control DLD Inspection	NK, PP	250	LOC	200.0	1.3	1.0	1.3	3/10/2003	15	0.0	3/7/2003
Task Panel Control	CODEINSP	Task Panel Control Code Inspection	NK, PP	250	LOC	200.0	1.3	1.0	1.3	3/10/2003	15	0.0	3/10/2003
ProfileUserList.aspx	DLDINSP	ProfileUserList.aspx DLD Inspection	PP, VY	1010	LOC	200.0	5.1	1.0	5.1	3/17/2003	16	2.0	2/4/2003
ProfileUserList.aspx	CODEINSP	ProfileUserList.aspx Code Inspection	PP, VY	1010	LOC	200.0	5.1	1.0	5.1	3/17/2003	16	4.4	2/27/2003
SYSTEM	IT	SYSTEM Build and BVT Fixes	ALL					0.0	0.0	3/17/2003	16	17.0	

Size

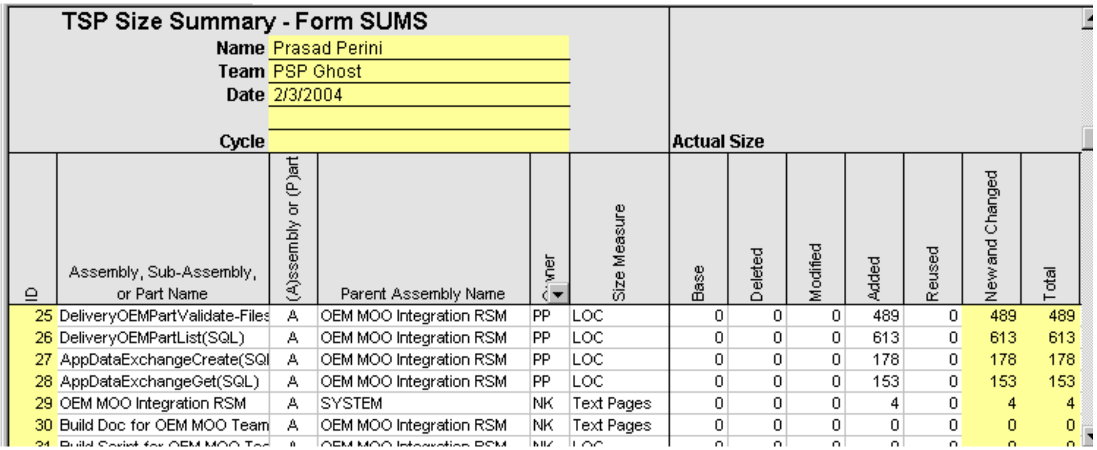
Size is a measure of the magnitude of the deliverable, e.g. lines of code or function points, pages.



TSP size measures are selected based on their correlation with time.

TSP also uses size data to

- normalize other measures
- track progress



TSP Size Summary - Form SUMS

Name	Prasad Perini
Team	PSP Ghost
Date	2/3/2004
Cycle	

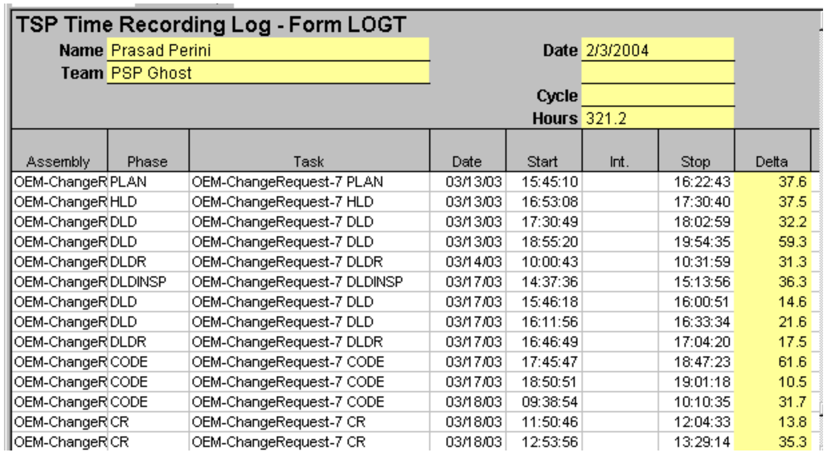
						Actual Size						
ID	Assembly, Sub-Assembly, or Part Name	(A)ssembly or (P)art	Parent Assembly Name	Owner	Size Measure	Base	Deleted	Modified	Added	Reused	Newand Changed	Total
25	DeliveryOEMPartValidate-Files	A	OEM MOO Integration RSM	PP	LOC	0	0	0	489	0	489	489
26	DeliveryOEMPartList(SQL)	A	OEM MOO Integration RSM	PP	LOC	0	0	0	613	0	613	613
27	AppDataExchangeCreate(SQI	A	OEM MOO Integration RSM	PP	LOC	0	0	0	178	0	178	178
28	AppDataExchangeGet(SQL)	A	OEM MOO Integration RSM	PP	LOC	0	0	0	153	0	153	153
29	OEM MOO Integration RSM	A	SYSTEM	NK	Text Pages	0	0	0	4	0	4	4
30	Build Doc for OEM MOO Team	A	OEM MOO Integration RSM	NK	Text Pages	0	0	0	0	0	0	0
31	Build Script for OEM MOO Tea	A	OEM MOO Integration RSM	NK	LOC	0	0	0	0	0	0	0

Time

Time is a measure of time on task.

The TSP time measure is task hours, i.e. the time spent on a project task, minus interruption time.

TSP team members record their time as they work, not at the end of the day, week, or month.



TSP Time Recording Log - Form LOGT

Name Prasad Perini

Team PSP Ghost

Date 2/3/2004

Cycle

Hours 321.2

Assembly	Phase	Task	Date	Start	Int.	Stop	Delta
OEM-ChangeR	PLAN	OEM-ChangeRequest-7 PLAN	03/13/03	15:45:10		16:22:43	37.6
OEM-ChangeR	HLD	OEM-ChangeRequest-7 HLD	03/13/03	16:53:08		17:30:40	37.5
OEM-ChangeR	DLD	OEM-ChangeRequest-7 DLD	03/13/03	17:30:49		18:02:59	32.2
OEM-ChangeR	DLD	OEM-ChangeRequest-7 DLD	03/13/03	18:55:20		19:54:35	59.3
OEM-ChangeR	DLDR	OEM-ChangeRequest-7 DLDR	03/14/03	10:00:43		10:31:59	31.3
OEM-ChangeR	DLDINSP	OEM-ChangeRequest-7 DLDINSP	03/17/03	14:37:36		15:13:56	36.3
OEM-ChangeR	DLD	OEM-ChangeRequest-7 DLD	03/17/03	15:46:18		16:00:51	14.6
OEM-ChangeR	DLD	OEM-ChangeRequest-7 DLD	03/17/03	16:11:56		16:33:34	21.6
OEM-ChangeR	DLDR	OEM-ChangeRequest-7 DLDR	03/17/03	16:46:49		17:04:20	17.5
OEM-ChangeR	CODE	OEM-ChangeRequest-7 CODE	03/17/03	17:45:47		18:47:23	61.6
OEM-ChangeR	CODE	OEM-ChangeRequest-7 CODE	03/17/03	18:50:51		19:01:18	10.5
OEM-ChangeR	CODE	OEM-ChangeRequest-7 CODE	03/18/03	09:38:54		10:10:35	31.7
OEM-ChangeR	CR	OEM-ChangeRequest-7 CR	03/18/03	11:50:46		12:04:33	13.8
OEM-ChangeR	CR	OEM-ChangeRequest-7 CR	03/18/03	12:53:56		13:29:14	35.3

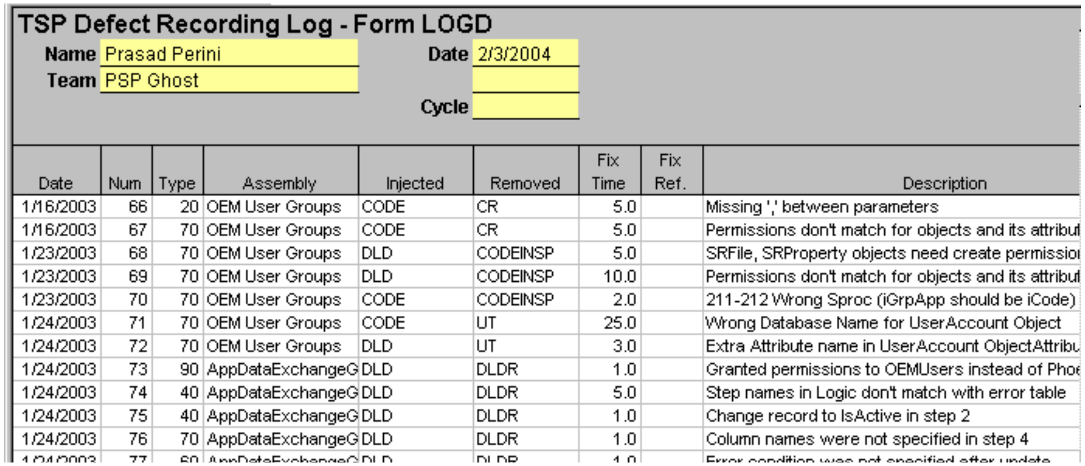
Defects

Defects are the measure of quality in the TSP.

Any change to an interim or final work product, made to ensure proper design, implementation, test, use, or maintenance, is a defect in the TSP.

Defects are logged as they are found and fixed.

Defect tracking takes place throughout the process.



TSP Defect Recording Log - Form LOGD

Name Prasad Perini
Team PSP Ghost
Date 2/3/2004
Cycle

Date	Num	Type	Assembly	Injected	Removed	Fix Time	Fix Ref.	Description
1/16/2003	66	20	OEM User Groups	CODE	CR	5.0		Missing ',' between parameters
1/16/2003	67	70	OEM User Groups	CODE	CR	5.0		Permissions don't match for objects and its attribut
1/23/2003	68	70	OEM User Groups	DLD	CODEINSP	5.0		SRFile, SRProperty objects need create permissio
1/23/2003	69	70	OEM User Groups	DLD	CODEINSP	10.0		Permissions don't match for objects and its attribut
1/23/2003	70	70	OEM User Groups	CODE	CODEINSP	2.0		211-212 Wrong Sproc (iGrpApp should be iCode)
1/24/2003	71	70	OEM User Groups	CODE	UT	25.0		Wrong Database Name for UserAccount Object
1/24/2003	72	70	OEM User Groups	DLD	UT	3.0		Extra Attribute name in UserAccount ObjectAttribu
1/24/2003	73	90	AppDataExchangeG	DLD	DLDR	1.0		Granted permissions to OEMUsers instead of Phoe
1/24/2003	74	40	AppDataExchangeG	DLD	DLDR	5.0		Step names in Logic don't match with error table
1/24/2003	75	40	AppDataExchangeG	DLD	DLDR	1.0		Change record to IsActive in step 2
1/24/2003	76	70	AppDataExchangeG	DLD	DLDR	1.0		Column names were not specified in step 4
1/24/2003	77	60	AppDataExchangeG	DLD	DLDR	1.0		Error condition was not specified after update

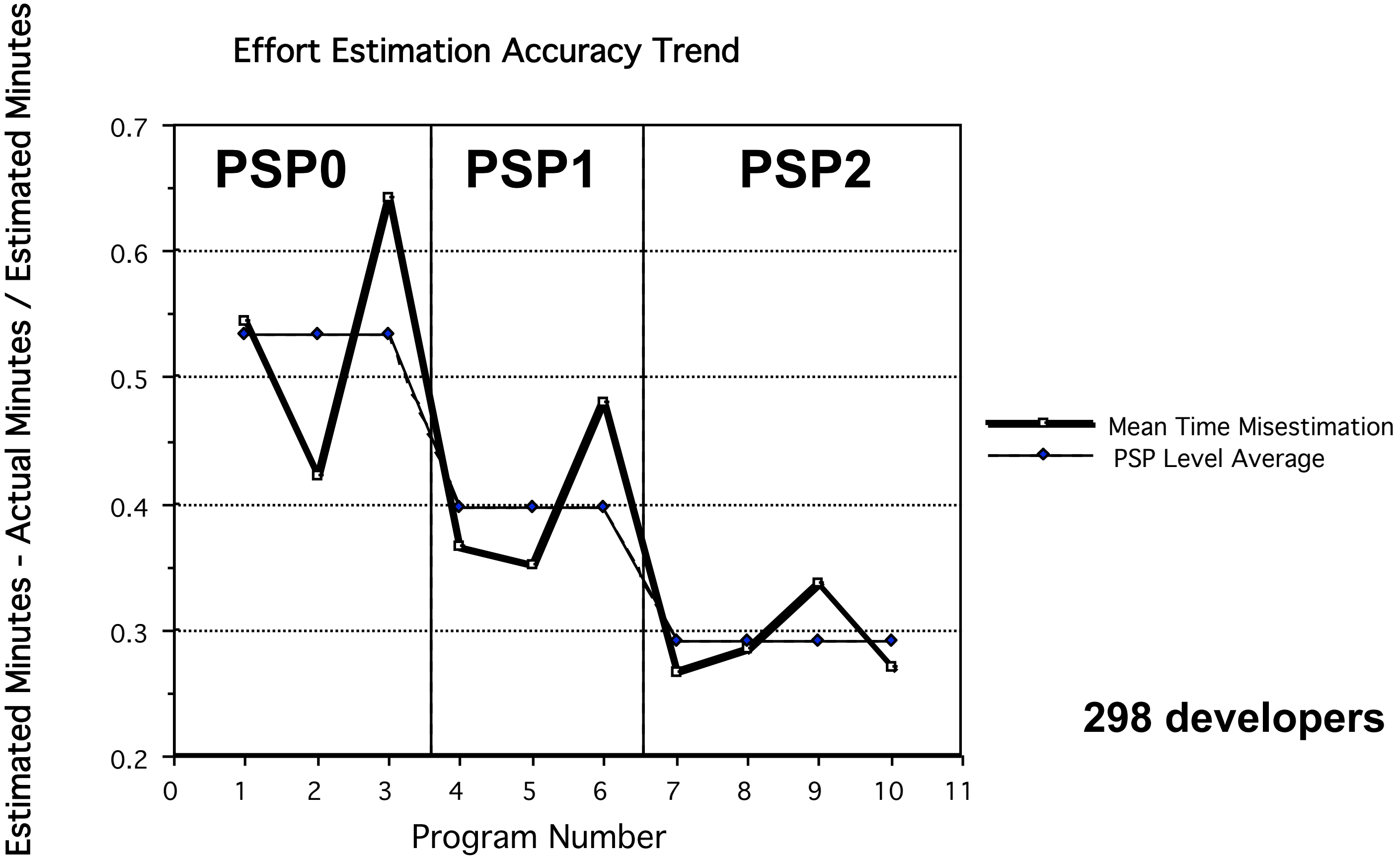
What the Direct Measures Provide

Management measures derived from the base measures are used by the team to manage the project and manage quality.

Project management measures: earned value, productivity , estimation accuracy, estimation size and effort prediction intervals, cost performance index, time in phase distributions, …

Quality management measures: defects injected and removed in each process phase, defect density, defect injection and removal rates, process yield, phase yield, review and inspection rates, cost of quality, percent defect free, quality profiles, quality profile index, …

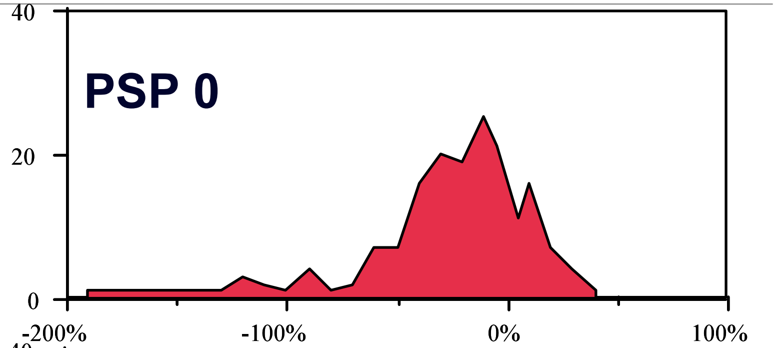
PSP Improves Estimating Accuracy -1



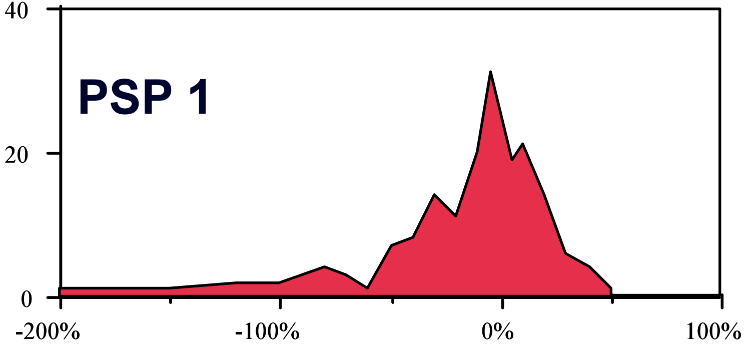
298 developers

PSP Improves Estimating Accuracy -2

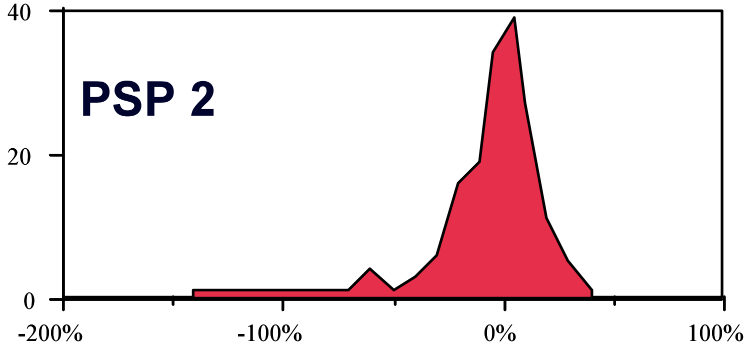
Majority are under-estimating



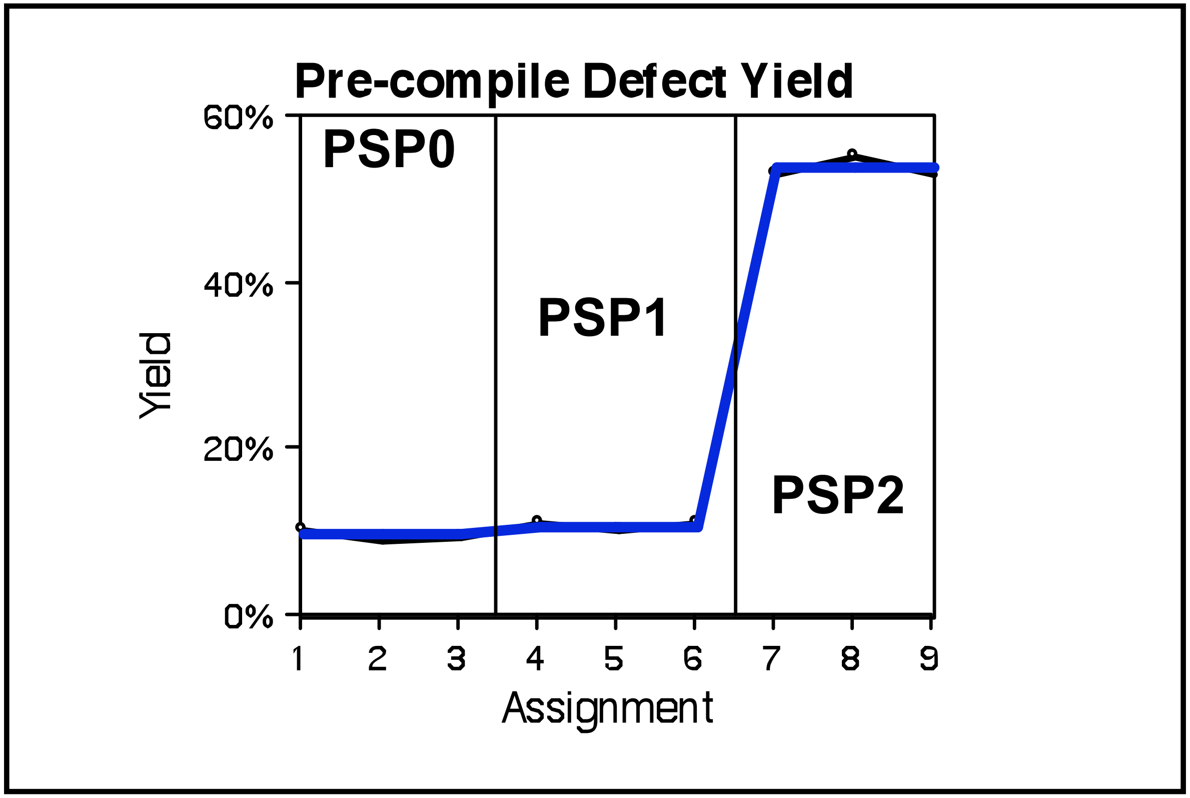
Balance of over- and under-estimates



Much tighter balance around zero



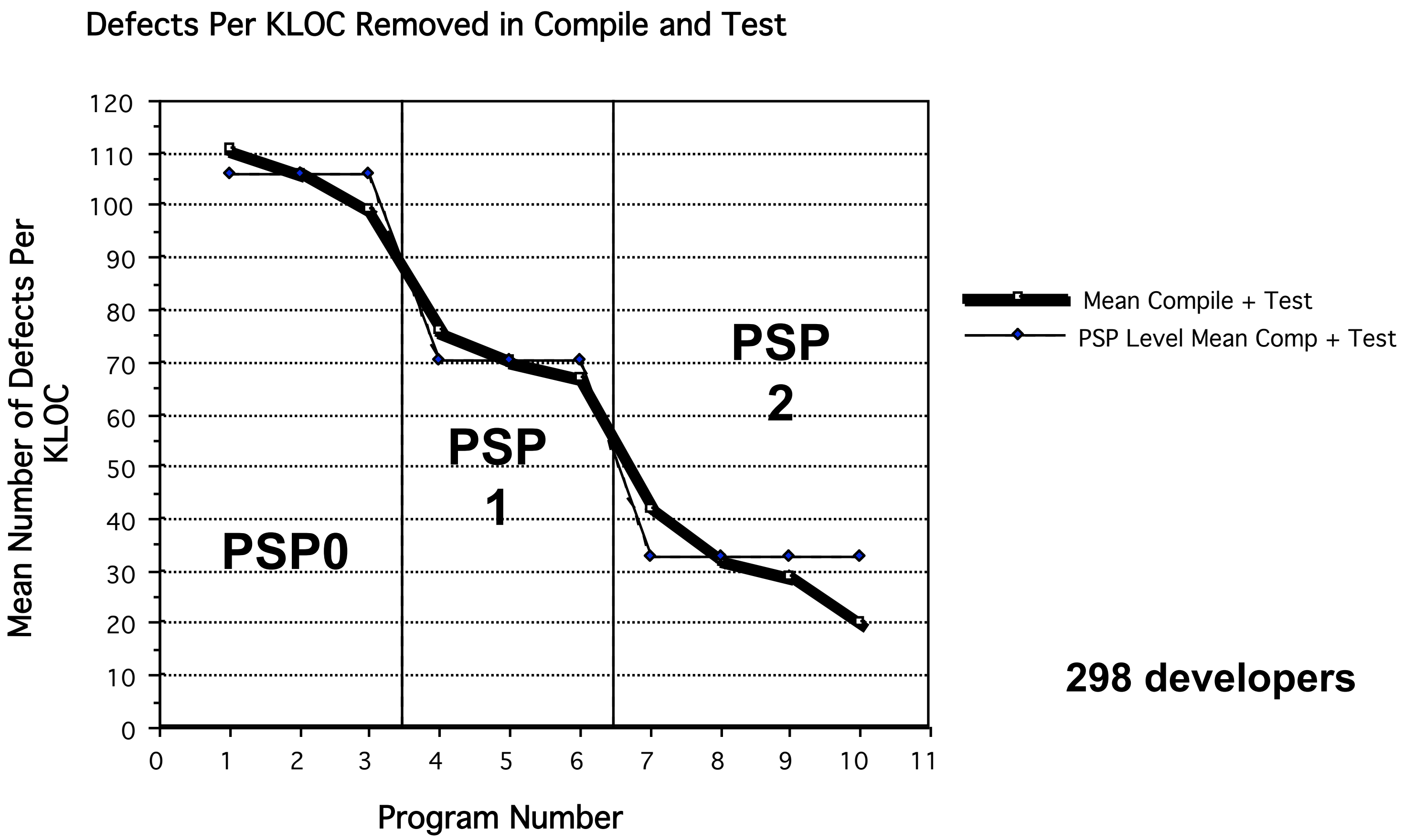
PSP Improves Process Yield



298 developers

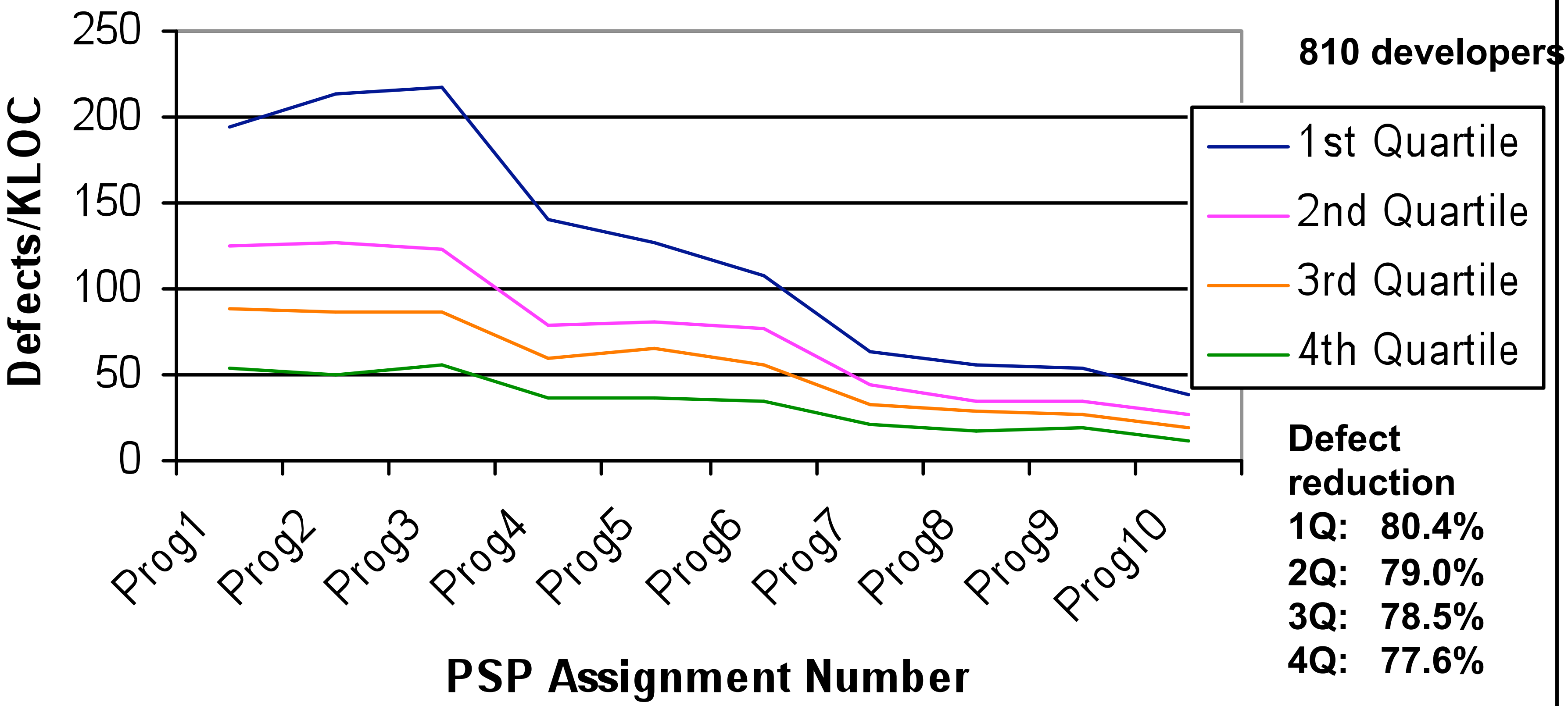
A higher-yield process will result in fewer defects in test.

PSP Quality Results



298 developers

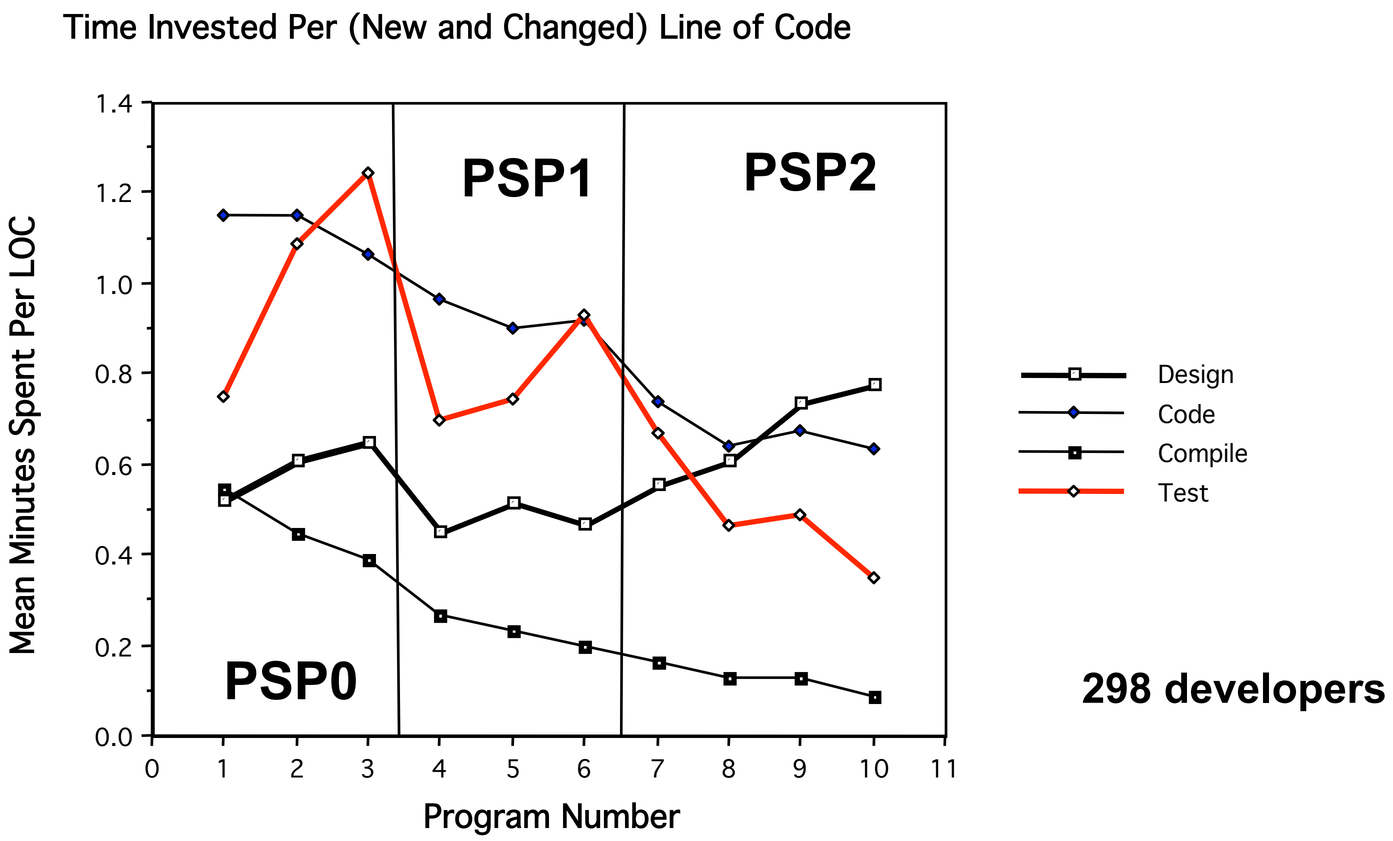
Test Defects - from PSP Training



810 developers

Defect reduction
1Q: 80.4%
2Q: 79.0%
3Q: 78.5%
4Q: 77.6%

PSP Design Time Results



TSP Concepts

Managing self-directed teams

Using processes and measures in engineering/creative work

Quality management

Software Industry Quality Strategy

The software industry is the only modern high-tech industry that ignores quality until test.

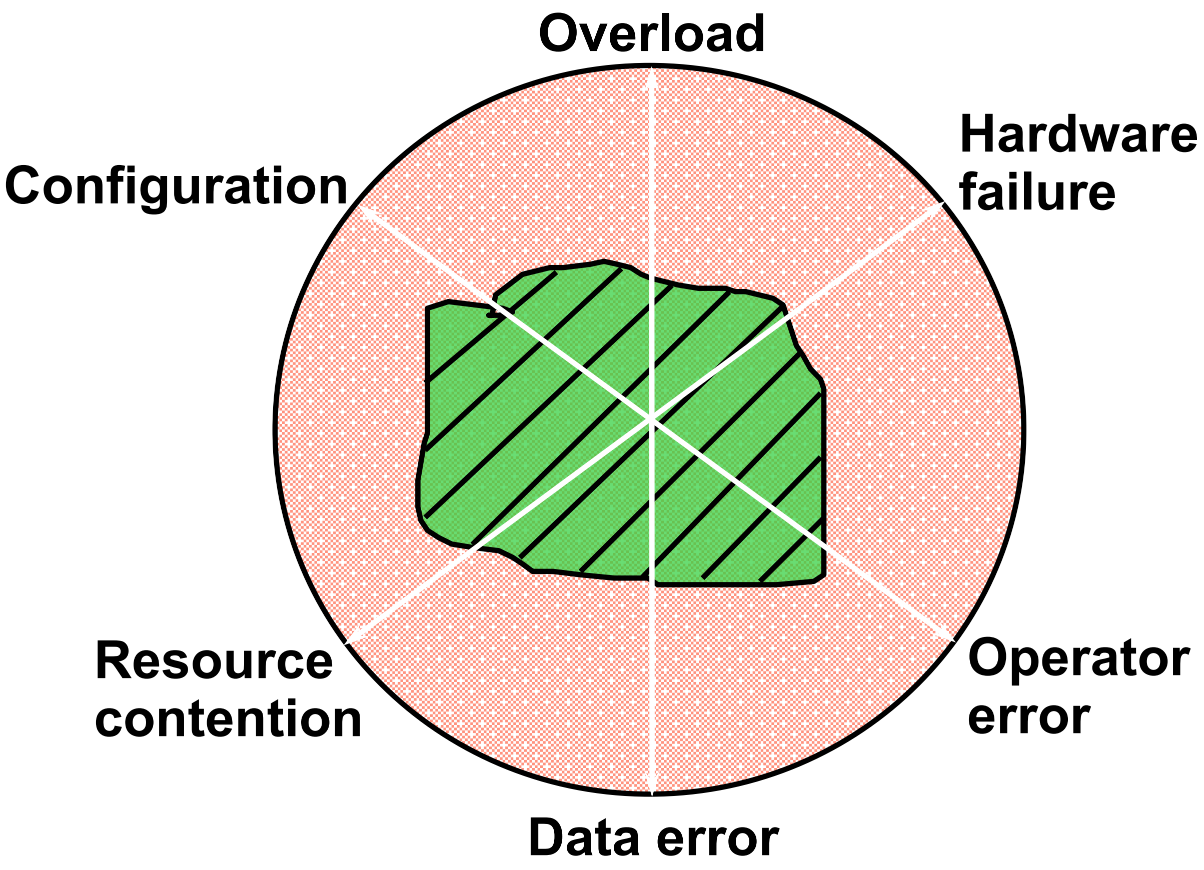
Most software defects are found in or after test when defect removal costs are the highest and the methods are the least effective.

This strategy results in defective products and unnecessary rework that inflates development costs by 30% to 40% or more.

This strategy is also a principal cause of unexpected delays, system failures, and software security vulnerabilities.

Linux crash on Airbus Entertainment System

Testing Coverage



Overload

Hardware failure

Operator error

Data error

Resource contention

Configuration

Safe and secure region = tested (shaded green)

Unsafe and insecure region = untested (shaded red)

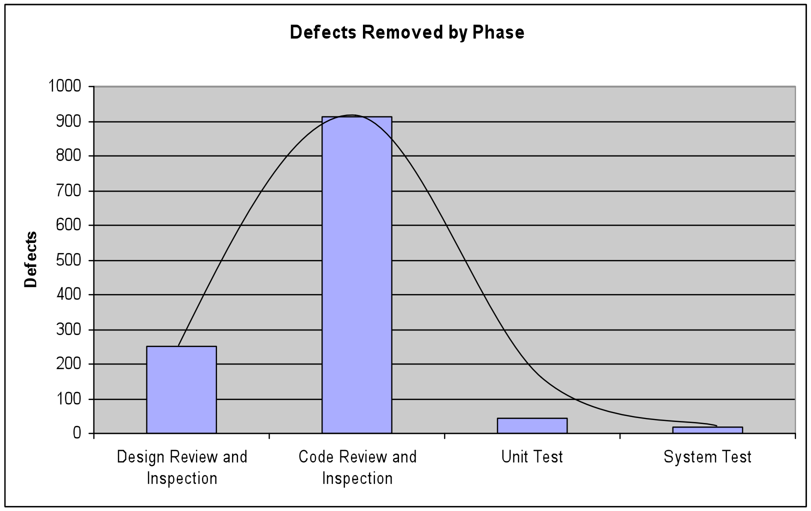
Put a Quality Product into Test

IBM's Dr. Harlan Mills asked: "*How do you know that you've found the last defect in system test?*"

"You never find the first one."

If you want a quality product out of test, you must put a quality product into test.

To put a quality product into test you must manage quality at every step.



TSP Quality Management Practices -1

Planning for quality

- TSP quality planning estimates the number of defects injected and removed at each phase based on historical injection rates and phase yields.
- Removal rates, review rates, phase time ratios, defect densities, and other quality indicators are then calculated by the tools.

Measuring and tracking quality

- Developers track every defect found and fixed.
- Quality is reviewed weekly by the quality manager and the team.

TSP Quality Management Practices -2

Defect removal filters

- Every activity that finds and removes defects can be thought of as a defect removal filter, e.g. reviews, inspections, compilers, static analyzers, etc.
- TSP has many such filters.

Capture/Recapture

- TSP uses capture/recapture to estimate the defects missed in inspections.

Defect prevention

- Every defect found in system test or later is analyzed to prevent future escapes.
- Every defective module is re-inspected.

Quality and the Team

High quality can only be achieved by the development team.

To manage quality they must

- have control of their process
- have the proper data to track quality
- be properly trained and motivated

The self-directed team management style empowers the team to manage quality.

The integrated measurement framework provides the data.

PSP provides the training, motivation, and commitment.

The Project Completes Training

Day 30

The training was completed in 30 days.

Bob and Tom were very happy with the results.

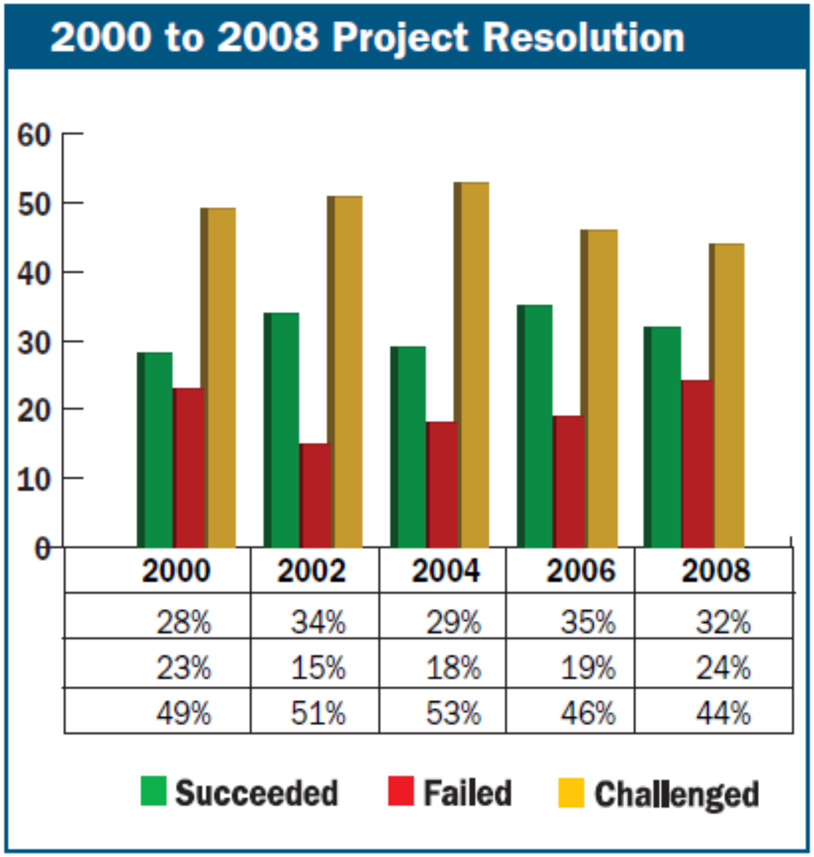
The team did not believe that management would change.

Management thought the team would not have the discipline to manage their work.

Agenda

When	•Topics
9:00 – Break	•Case study: a project in trouble •Team Software Process and its implementation strategy •TSP concepts
Break – Lunch	•Why projects fail •Case study: launching the project
Lunch – Break	•Case study: launching the project (continued)
Break – 5:30	•Case study: team-working framework •Corporate experience with TSP •TSP and CMMI •Building internal support for TSP

Failed Projects



2000 to 2008 Project Resolution

2000	2002	2004	2006	2008
28%	34%	29%	35%	32%
23%	15%	18%	19%	24%
49%	51%	53%	46%	44%

Successful projects delivered on time, on budget, with required features and functions.

Challenged projects were late, over budget, and/or failed to deliver all of the required features and functions.

Failed projects were cancelled prior to completion or delivered and never used.

Source: Standish group 2009 Chaos report.

Project Failure

Discussion topic: Why do teams fail?

10 minutes

What Makes Teams Fail?

There are lots of ways to make teams fail.

- Start late.
- Demand impossible schedules.
- Under-staff the project.
- Manage to the schedule.
- Fail to manage quality.
- Lack of teamwork.

Start Late

Many factors determine how long projects take.

- Staffing experience and staff size
- the size of the job
- knowledge about requirements
- job complexity
- degree of change

Nothing, however, can make up for a late start.

Demand an Impossible Schedule

To destroy a project, edict the schedule and don’t plan.

With an impossible schedule,

- you cannot make a plan to meet the date
- you must then work without a plan
- you cannot coordinate or track the work

This is when everyone is in the dark.

- You don’t know where you are.
- And neither does anyone else.

Understaff the Project

Pretend every project is staffed.

- Don’t set priorities.
- Expect part-time engineers to produce.
- Push for maximum overtime.

With understaffed projects, you

- feel the work is low priority
- are not personally committed
- make a lot of mistakes
- just try to get through test

Manage to Schedule

Schedule is all that matters.

- Quality is not measured.
- There is no time for training.
- There is no time for inspections.
- The top priority is getting into test.

This is when

- the job seems endless
- there is no sense of progress
- you just want to throw it over the wall
- products are late and defective

Failure to Manage Quality

When quality isn't managed,

- projects appear to be farther ahead than they really are.
- testing and rework account for half the schedule.
- testing is unpredictable; no one knows how long it will take to fix the open critical defects.

As schedule pressure increases, shortcuts are taken that make quality worse, and the schedule slips again.

Lack of Teamwork

Software development is like a team sport, the best results are achieved when the team members work together.

For software teams to produce high-quality products on aggressive schedules, they must

- be involved in the work
- be committed to its success
- share a common process and plan
- have a personal commitment to quality
- work cooperatively to meet the team goals

The Advantage of Self-directed Teams

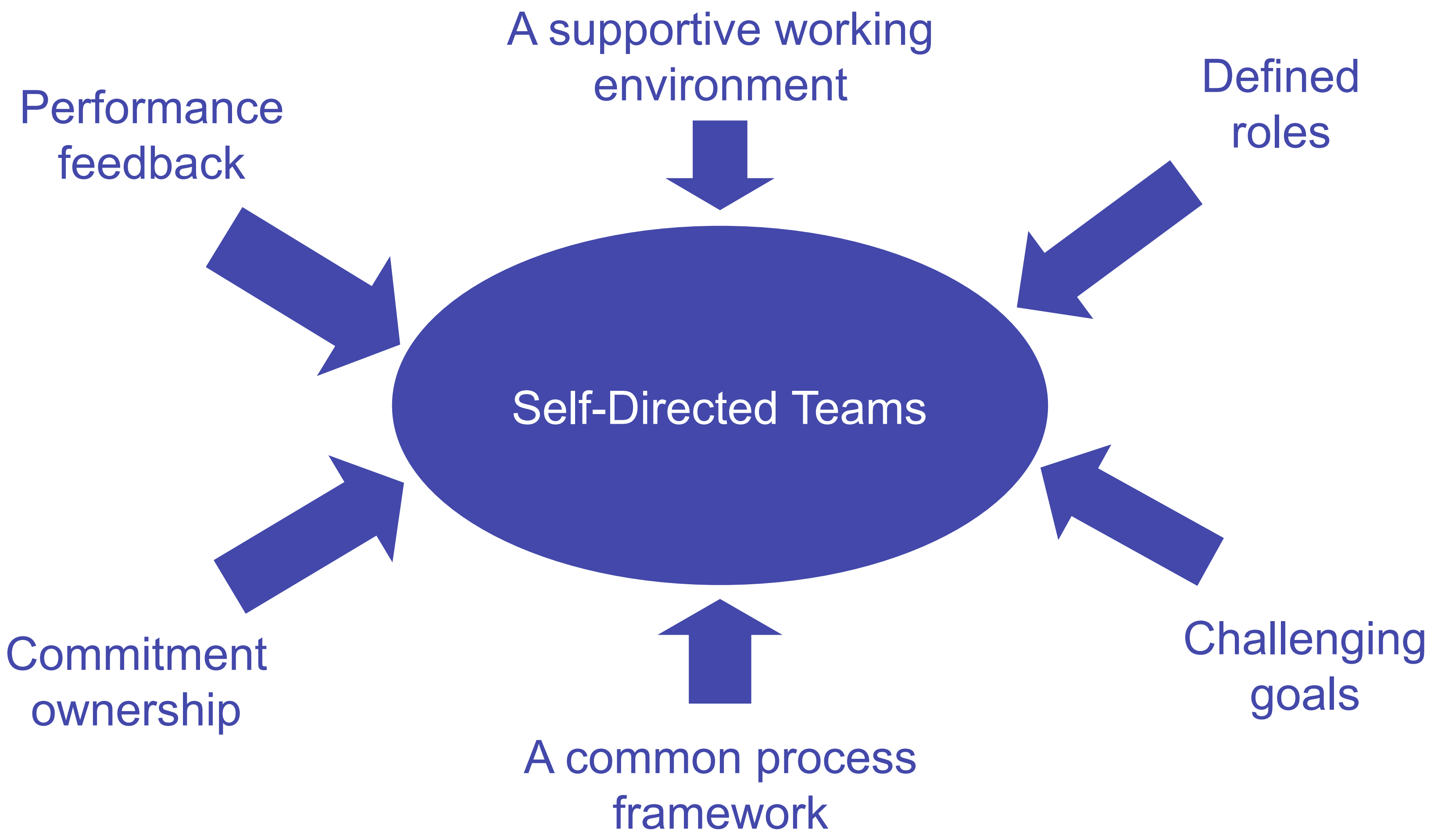
Self-directed teams

- develop their own plans
- negotiate commitments
- track their work
- keep management informed of project status and risks

Self-directed teams

- are empowered by their management
- are personally committed
- enjoy their work
- can resolve many team failure modes

Producing Self-Directed Teams



The Project and The Team

Day 31

The new product was *still* critically needed in 9 months.

The only requirements document was a list of the features in the competitor’s product that marketing had prepared.

The project team had

- 2 managers
- 9 software engineers
- 5 hardware engineers

The managers and the team were trained.

They were ready to launch.

TSP Process Structure

TSP projects are divided into cycles.

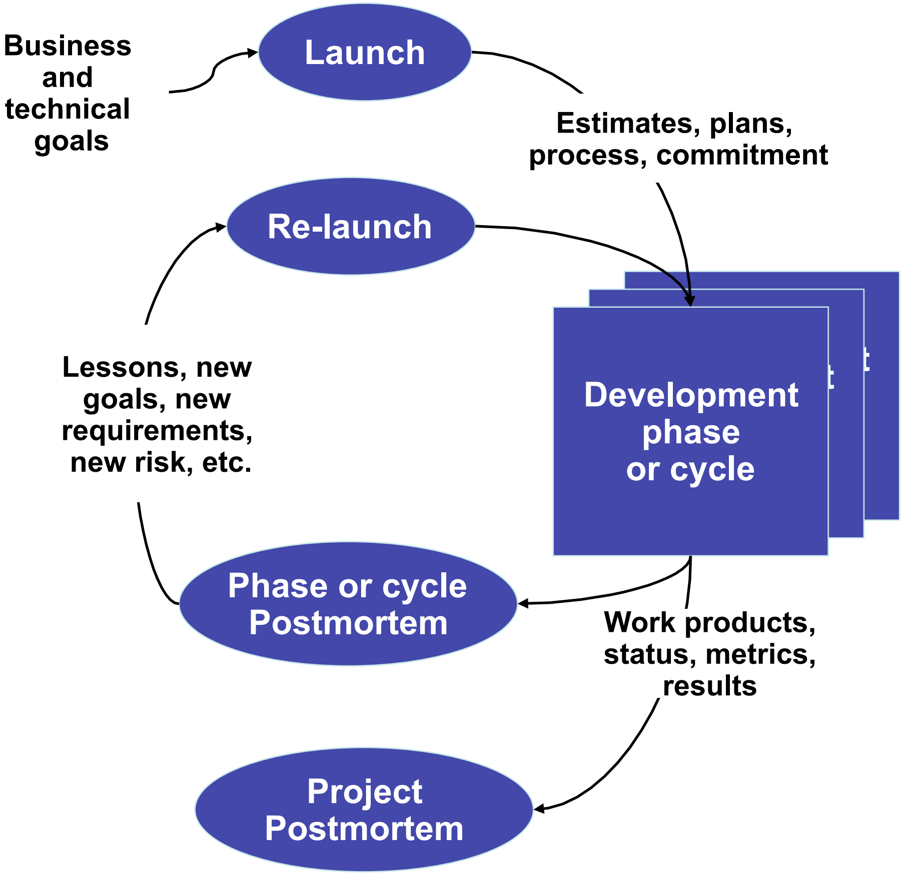
Each cycle starts with a launch or re-launch and ends with a postmortem.

Cycle content is determined by the team.

Any lifecycle model can be supported.

TSP projects can start on any lifecycle phase.

TSP supports whatever structure makes the most business and technical sense.


Business and technical goals
Launch
Estimates, plans, process, commitment
Re-launch
Lessons, new goals, new requirements, new risk, etc.
Development phase or cycle
Phase or cycle Postmortem
Work products, status, metrics, results
Project Postmortem

The TSP Launch Process

The launch process performs essential tasks.

- Without a launch process, these tasks are generally addressed only when needed.
- This is often too late to prevent problems.
- It often causes unanticipated project delays.

The launch process steps are performed quickly when the engineers follow these guidelines:

- do the work as quickly as practical.
- be thorough but don’t bother with formality.
- build on what has been done before.

Key Objectives of the TSP Launch

Put professionals in charge of their own personal work.

Provide a team environment that supports individual excellence.

Enable teams to produce processes and plans that best fit their needs.

Those that do the work, own the process, make the plans and make the commitments.

The TSP Launch is the first step in this commitment process.

The TSP Launch Products

In the launch and relaunch workshops, the team develops a standard suite of launch products.

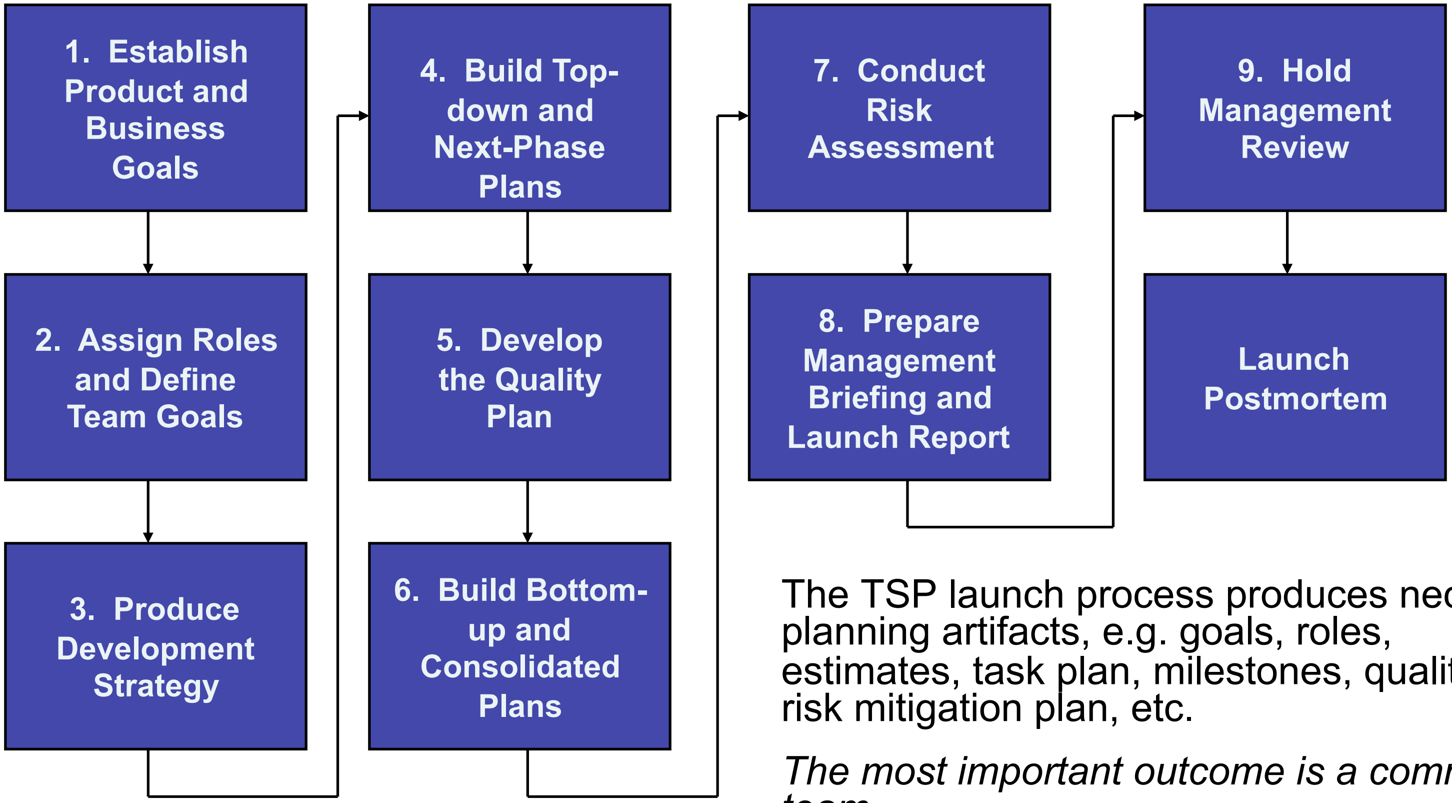
These launch products provide a solid foundation for the project plan.

They provide a sound basis for guiding and tracking the project.

Launch Products

- documented team goals
- team-member role assignments
- inventory of processes
- a measurable quality plan
- a facilities support plan
- an overall development strategy
- a detailed next-phase team plan
- individual plans for the next phase
- a consolidated team plan
- a project risk assessment

The TSP Launch Process



1. Establish Product and Business Goals
2. Assign Roles and Define Team Goals
3. Produce Development Strategy
4. Build Top-down and Next-Phase Plans
5. Develop the Quality Plan
6. Build Bottom-up and Consolidated Plans
7. Conduct Risk Assessment
8. Prepare Management Briefing and Launch Report
9. Hold Management Review
Launch Postmortem

The TSP launch process produces necessary planning artifacts, e.g. goals, roles, estimates, task plan, milestones, quality plan, risk mitigation plan, etc.

The most important outcome is a committed team.

Meeting 1 - Understand the Project Objectives

In meeting 1, the team meets with management to understand the project objectives.

- Management describes the business goals and objectives, e.g. business need, resources, schedule, success criteria.
- Marketing or the customer describes the product goals and objectives, e.g. market, customer needs, features, success criteria.
- The team asks questions.

Management said “Failure is not an option!”

Day 31

Management placed the team under enormous pressure at the start of the launch.

- schedule and functionality were non-negotiable
- resources were fixed
- failure was equated to “out of business”

The team was convinced the project was impossible.

The Project Team Responds

Day 31

No one felt comfortable making a plan and commitment.

- there were no requirements or designs
- the project was impossible anyway

Without a plan the team had three choices

- quit or wait to be fired
- agree to management's demands and be fired when the project is late
- make a plan

They decided to make a plan.

Launch Meeting 2

The purpose of launch meeting 2 is to guide the team in setting project goals and establishing team member roles.

Team members identify and select the roles of their choice.

A primary and an alternate are selected for each role.

Small teams may assign roles to groups or individuals outside of the team.

Team Goals

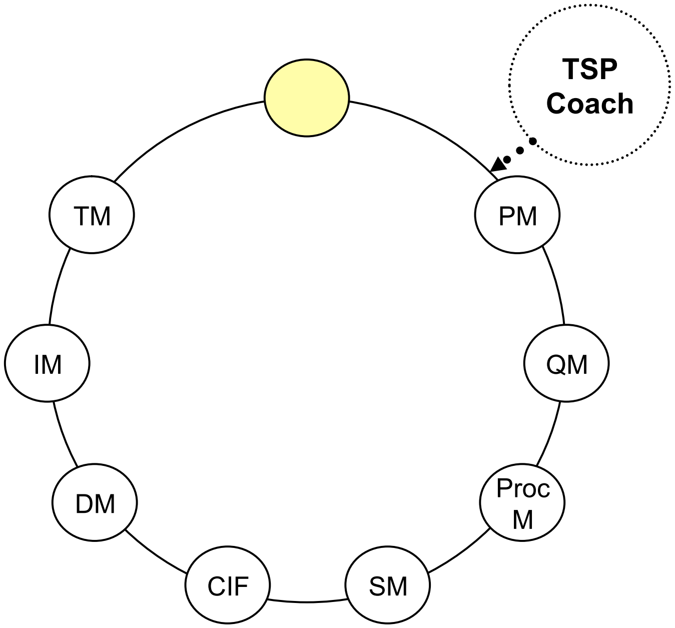
The team develops a business needs statement to guide the project.

- Why does management want this project?
- What will the project contribute to the business?

Measurable goals are established for each project stakeholder.

- customer
- user
- management
- team
- team member roles

Assign Team Roles



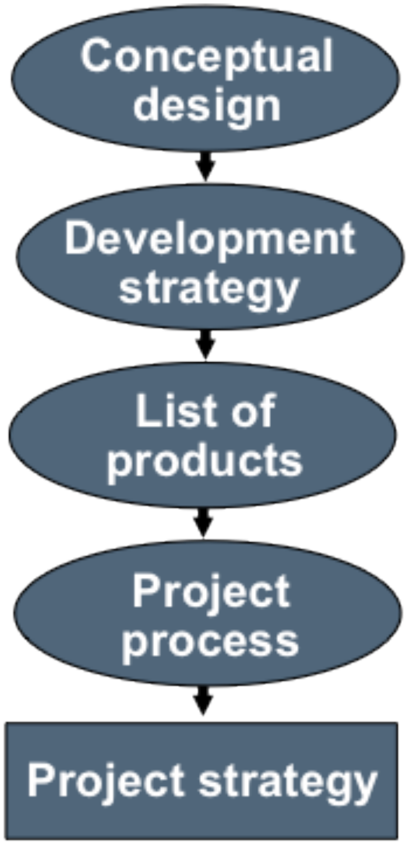
Self-directed team roles

Eight pre-defined roles distribute traditional project management responsibilities across the team.

All team members have traditional roles, e.g. developer, tester, etc.

Project Management Roles
Planning manager – responsible for tracking the plan.
Quality manager – responsible for tracking the quality plan.
Process manager – responsible for ensuring process discipline and for process improvement.
Support manager – responsible for ensuring that support needs are met and for configuration management.
Technical Roles
Customer interface manager – responsible for the interface to the customer or customer representative.
Design manager – responsible for the design practices and quality.
Implementation manager – responsible for implementation practices and quality.
Test manager – responsible for test practices and quality.

Meeting 3 - Define the Work and the Approach



In meeting 3, the team accomplishes three important prerequisites to building the team plan.

1. identifies all of the **work** the team needs to do
2. identifies the **build strategy** the team will use to develop the software
3. identifies or defines the **processes** the team will follow to do the work

Product Conceptual Design

The product conceptual design is the “big picture” view of the product, it is not a high-level design.

It includes the major parts of the products, i.e. what needs to be built to meet management’s minimum requirements

The conceptual design answers the question, “If I had these parts I could build this system.”

Development Strategy

The development strategy is the “big picture” view of the development effort.

- development increments and builds
- general version content and freeze points
- prototypes needed
- integration and test strategy

Initial Estimates

Rough size and effort estimates are needed to define the development strategy.

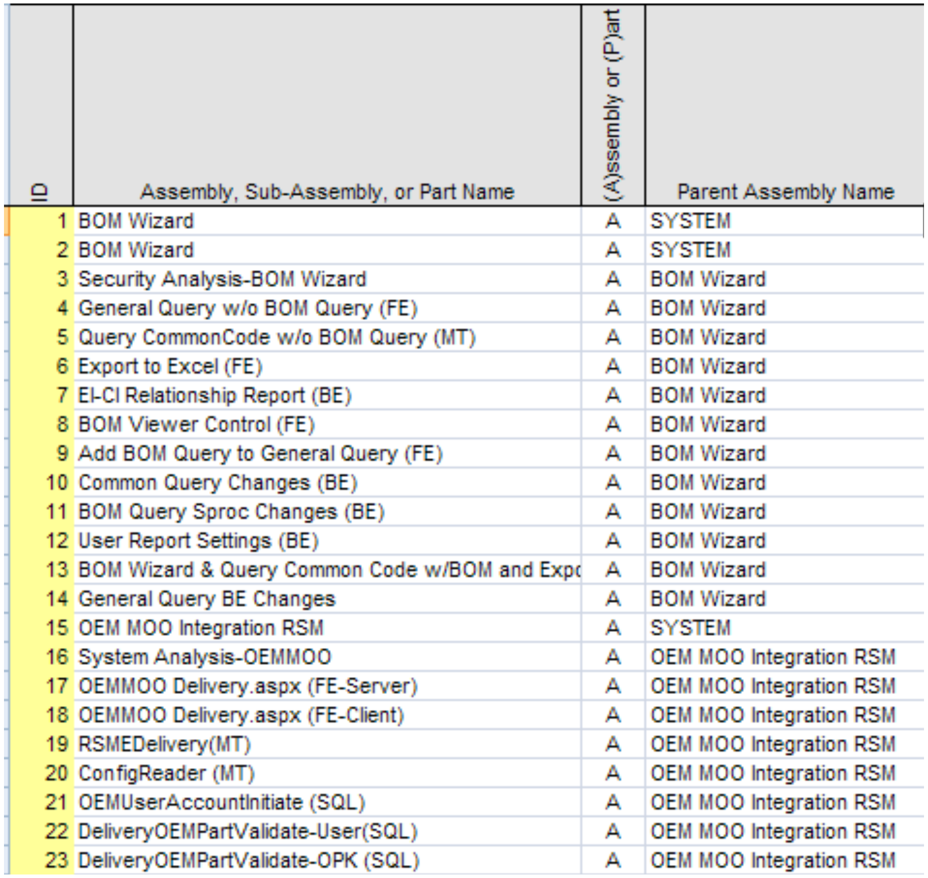
- size estimates of the individual parts in the product conceptual design
- effort estimates for each increment

These estimates are based on available historical data, the engineer’s PSP data, and/or best guess.

List of Products and Features

The list of products and features includes everything that will be produced.

- requirements
- specifications
- designs
- software
- test cases
- documentation
- installation procedures
- …



ID	Assembly, Sub-Assembly, or Part Name	(A)ssembly or (P)art	Parent Assembly Name
1	BOM Wizard	A	SYSTEM
2	BOM Wizard	A	SYSTEM
3	Security Analysis-BOM Wizard	A	BOM Wizard
4	General Query w/o BOM Query (FE)	A	BOM Wizard
5	Query CommonCode w/o BOM Query (MT)	A	BOM Wizard
6	Export to Excel (FE)	A	BOM Wizard
7	EI-CI Relationship Report (BE)	A	BOM Wizard
8	BOM Viewer Control (FE)	A	BOM Wizard
9	Add BOM Query to General Query (FE)	A	BOM Wizard
10	Common Query Changes (BE)	A	BOM Wizard
11	BOM Query Sproc Changes (BE)	A	BOM Wizard
12	User Report Settings (BE)	A	BOM Wizard
13	BOM Wizard & Query Common Code w/BOM and Expo	A	BOM Wizard
14	General Query BE Changes	A	BOM Wizard
15	OEM MOO Integration RSM	A	SYSTEM
16	System Analysis-OEMMOO	A	OEM MOO Integration RSM
17	OEMMOO Delivery.aspx (FE-Server)	A	OEM MOO Integration RSM
18	OEMMOO Delivery.aspx (FE-Client)	A	OEM MOO Integration RSM
19	RSMEDelivery(MT)	A	OEM MOO Integration RSM
20	ConfigReader (MT)	A	OEM MOO Integration RSM
21	OEMUserAccountInitiate (SQL)	A	OEM MOO Integration RSM
22	DeliveryOEMPartValidate-User(SQL)	A	OEM MOO Integration RSM
23	DeliveryOEMPartValidate-OPK (SQL)	A	OEM MOO Integration RSM

Development Process

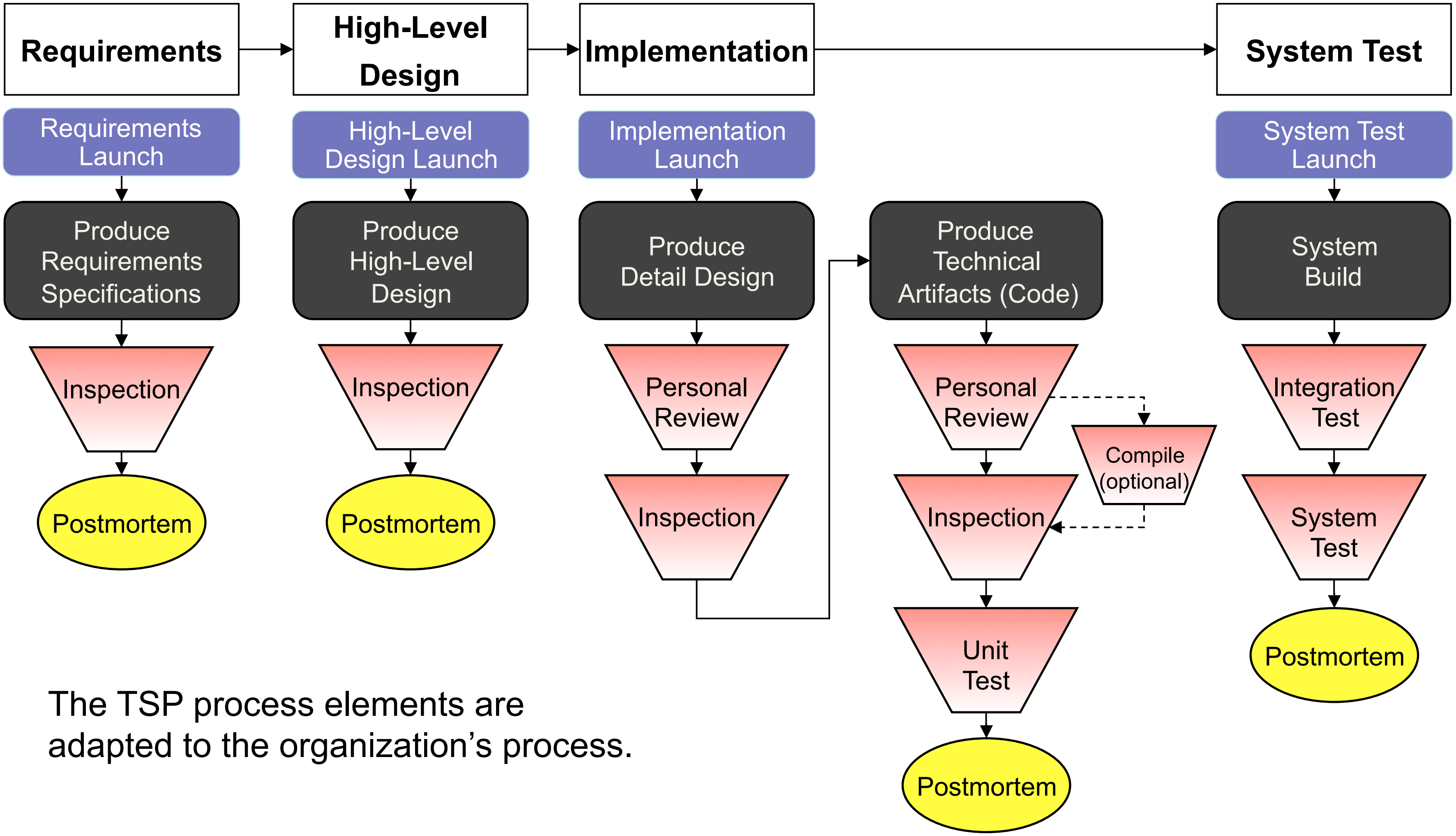
The team next plans their development process.

Having defined what to build the team is prepared to define how to build it.

The development process is based on the organization’s standard process and the TSP.

This step produces a process plan.

TSP Development Process



The TSP process elements are adapted to the organization’s process.

The Process Plan

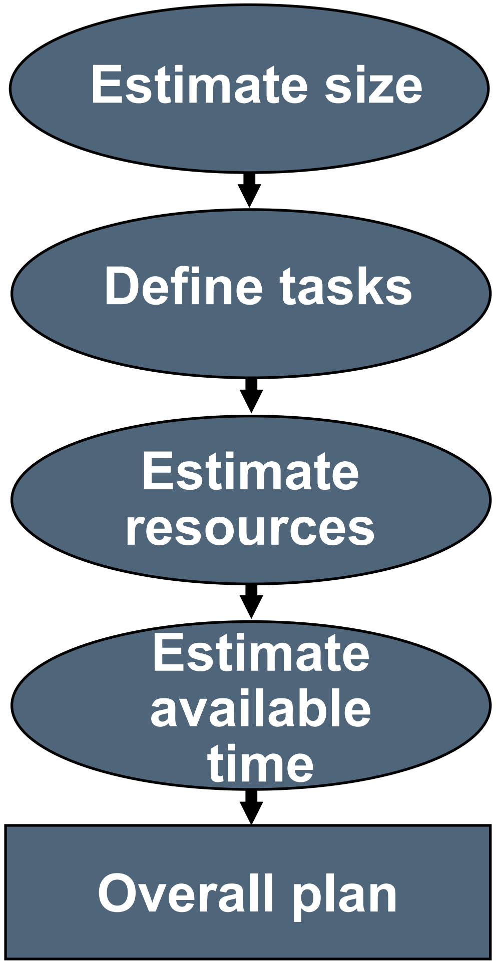
The process plan is

- an inventory of process elements that will be used by the team
- a plan for building any missing elements.

Size and time estimates are made for producing any missing elements.

Responsibility for producing or acquiring these elements is then typically assigned to the process manager.

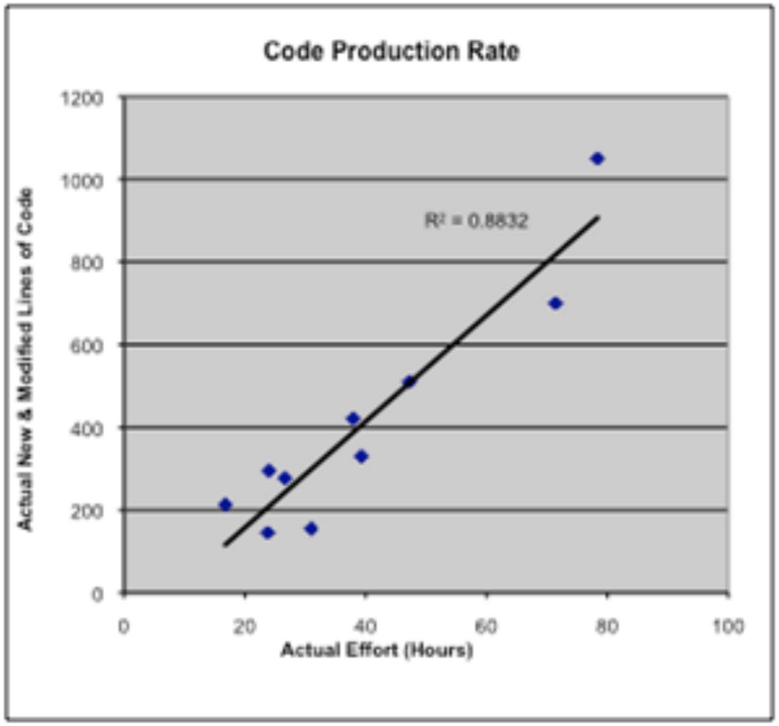
Meeting 4 - Build the Overall Plan



In launch meeting 4, the team creates the overall plan by establishing

- the estimated size of each work product (how big is the job)
- the tasks needed to complete the work (with effort estimates)
  - next-phase tasks, detailed to the work-step level
  - later phases at a high level
- the estimated team hours available each week for the work
- an initial schedule for the project

Size and Effort

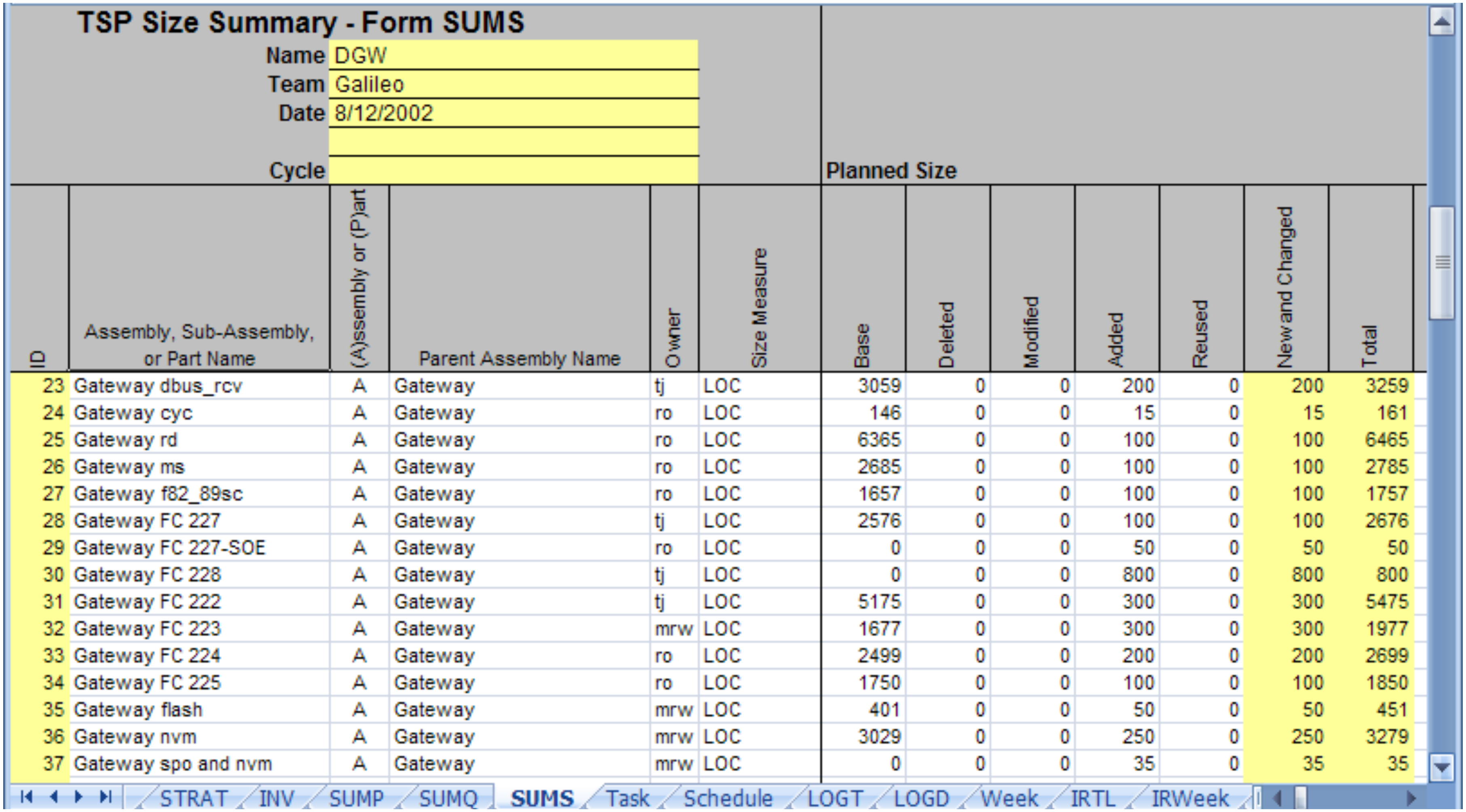


The size measure is selected based on its correlation to effort.

Detailed component size estimates reduce estimation risk.

Total effort can then be predicted for each component.

Estimated Size



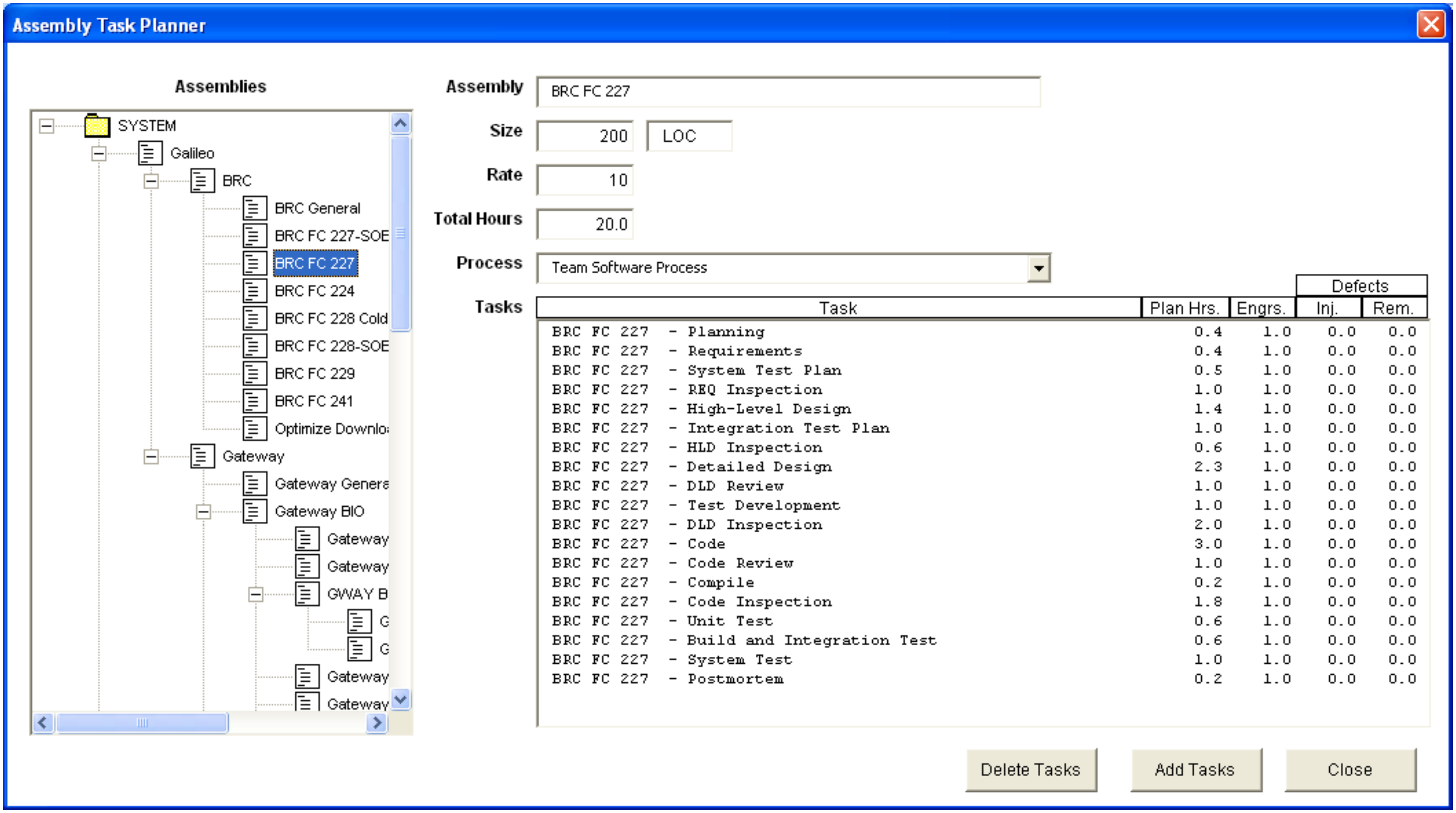
TSP Size Summary - Form SUMS

Name	DGW
Team	Galileo
Date	8/12/2002
Cycle	

Planned Size

ID	Assembly, Sub-Assembly, or Part Name	(A)ssembly or (P)art	Parent Assembly Name	Owner	Size Measure	Base	Deleted	Modified	Added	Reused	Newand Changed	Total
23	Gateway dbus_rcv	A	Gateway	tj	LOC	3059	0	0	200	0	200	3259
24	Gateway cyc	A	Gateway	ro	LOC	146	0	0	15	0	15	161
25	Gateway rd	A	Gateway	ro	LOC	6365	0	0	100	0	100	6465
26	Gateway ms	A	Gateway	ro	LOC	2685	0	0	100	0	100	2785
27	Gateway f82_89sc	A	Gateway	ro	LOC	1657	0	0	100	0	100	1757
28	Gateway FC 227	A	Gateway	tj	LOC	2576	0	0	100	0	100	2676
29	Gateway FC 227-SOE	A	Gateway	ro	LOC	0	0	0	50	0	50	50
30	Gateway FC 228	A	Gateway	tj	LOC	0	0	0	800	0	800	800
31	Gateway FC 222	A	Gateway	tj	LOC	5175	0	0	300	0	300	5475
32	Gateway FC 223	A	Gateway	mrw	LOC	1677	0	0	300	0	300	1977
33	Gateway FC 224	A	Gateway	ro	LOC	2499	0	0	200	0	200	2699
34	Gateway FC 225	A	Gateway	ro	LOC	1750	0	0	100	0	100	1850
35	Gateway flash	A	Gateway	mrw	LOC	401	0	0	50	0	50	451
36	Gateway nvm	A	Gateway	mrw	LOC	3029	0	0	250	0	250	3279
37	Gateway spo and nvm	A	Gateway	mrw	LOC	0	0	0	35	0	35	35

Task Planning



Assembly: BRC FC 227

Size: 200 LOC

Rate: 10

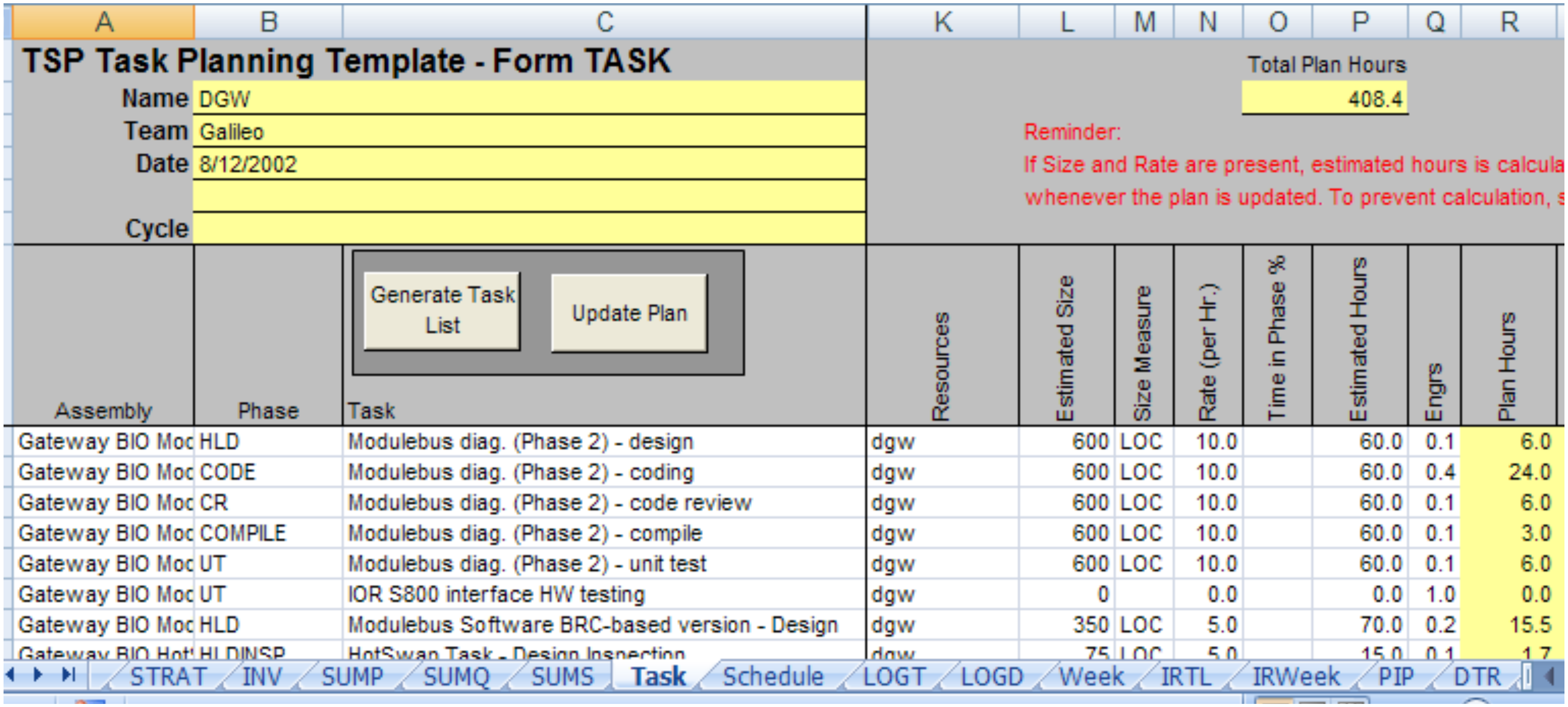
Total Hours: 20.0

Process: Team Software Process

Tasks

Task	Plan Hrs.	Engrs.	Defects Inj.	Defects Rem.
BRC FC 227 - Planning	0.4	1.0	0.0	0.0
BRC FC 227 - Requirements	0.4	1.0	0.0	0.0
BRC FC 227 - System Test Plan	0.5	1.0	0.0	0.0
BRC FC 227 - REQ Inspection	1.0	1.0	0.0	0.0
BRC FC 227 - High-Level Design	1.4	1.0	0.0	0.0
BRC FC 227 - Integration Test Plan	1.0	1.0	0.0	0.0
BRC FC 227 - HLD Inspection	0.6	1.0	0.0	0.0
BRC FC 227 - Detailed Design	2.3	1.0	0.0	0.0
BRC FC 227 - DLD Review	1.0	1.0	0.0	0.0
BRC FC 227 - Test Development	1.0	1.0	0.0	0.0
BRC FC 227 - DLD Inspection	2.0	1.0	0.0	0.0
BRC FC 227 - Code	3.0	1.0	0.0	0.0
BRC FC 227 - Code Review	1.0	1.0	0.0	0.0
BRC FC 227 - Compile	0.2	1.0	0.0	0.0
BRC FC 227 - Code Inspection	1.8	1.0	0.0	0.0
BRC FC 227 - Unit Test	0.6	1.0	0.0	0.0
BRC FC 227 - Build and Integration Test	0.6	1.0	0.0	0.0
BRC FC 227 - System Test	1.0	1.0	0.0	0.0
BRC FC 227 - Postmortem	0.2	1.0	0.0	0.0

Fine Tuning the Task Plan



TSP Task Planning Template - Form TASK

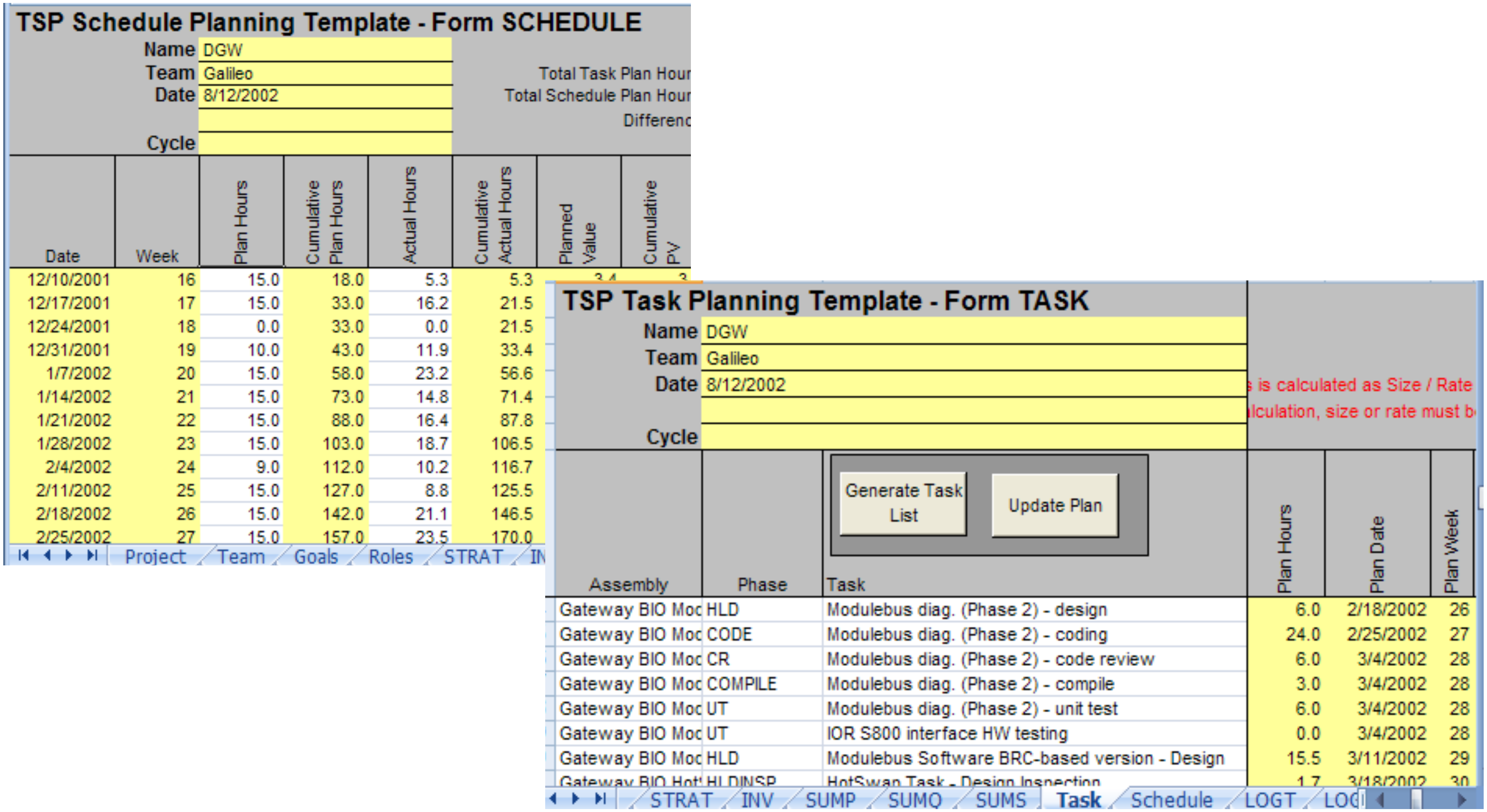
Name	DGW
Team	Galileo
Date	8/12/2002
Cycle	

Total Plan Hours 408.4

Reminder:
If Size and Rate are present, estimated hours is calcula
whenever the plan is updated. To prevent calculation, s

Assembly	Phase	Task	Resources	Estimated Size	Size Measure	Rate (per Hr.)	Time in Phase %	Estimated Hours	Engrs	Plan Hours
Gateway BIO Moc	HLD	Modulebus diag. (Phase 2) - design	dgw	600	LOC	10.0		60.0	0.1	6.0
Gateway BIO Moc	CODE	Modulebus diag. (Phase 2) - coding	dgw	600	LOC	10.0		60.0	0.4	24.0
Gateway BIO Moc	CR	Modulebus diag. (Phase 2) - code review	dgw	600	LOC	10.0		60.0	0.1	6.0
Gateway BIO Moc	COMPILE	Modulebus diag. (Phase 2) - compile	dgw	600	LOC	10.0		60.0	0.1	3.0
Gateway BIO Moc	UT	Modulebus diag. (Phase 2) - unit test	dgw	600	LOC	10.0		60.0	0.1	6.0
Gateway BIO Moc	UT	IOR S800 interface HW testing	dgw	0		0.0		0.0	1.0	0.0
Gateway BIO Moc	HLD	Modulebus Software BRC-based version - Design	dgw	350	LOC	5.0		70.0	0.2	15.5
Gateway BIO Hot'	HLDINSP	HotSwan Task - Design Inspection	dgw	75	LOC	5.0		15.0	0.1	1.7

Creating the Schedule



TSP Schedule Planning Template - Form SCHEDULE

Name DGW
Team Galileo
Date 8/12/2002
Cycle

Total Task Plan Hour
Total Schedule Plan Hour
Differenc

Date	Week	Plan Hours	Cumulative Plan Hours	Actual Hours	Cumulative Actual Hours	Planned Value	Cumulative PV
12/10/2001	16	15.0	18.0	5.3	5.3	3.4	3
12/17/2001	17	15.0	33.0	16.2	21.5		
12/24/2001	18	0.0	33.0	0.0	21.5		
12/31/2001	19	10.0	43.0	11.9	33.4		
1/7/2002	20	15.0	58.0	23.2	56.6		
1/14/2002	21	15.0	73.0	14.8	71.4		
1/21/2002	22	15.0	88.0	16.4	87.8		
1/28/2002	23	15.0	103.0	18.7	106.5		
2/4/2002	24	9.0	112.0	10.2	116.7		
2/11/2002	25	15.0	127.0	8.8	125.5		
2/18/2002	26	15.0	142.0	21.1	146.5		
2/25/2002	27	15.0	157.0	23.5	170.0		

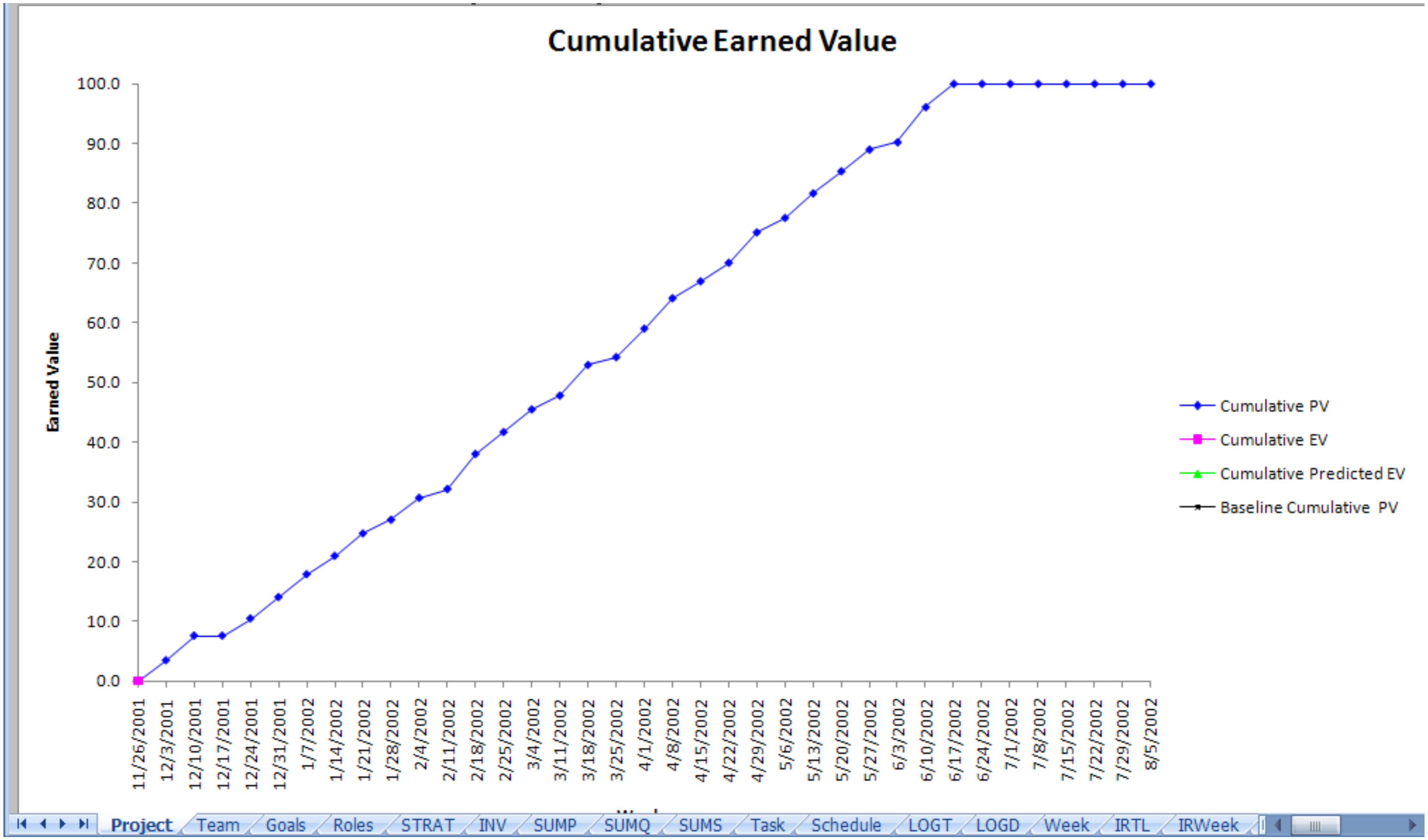
TSP Task Planning Template - Form TASK

Name DGW
Team Galileo
Date 8/12/2002
Cycle

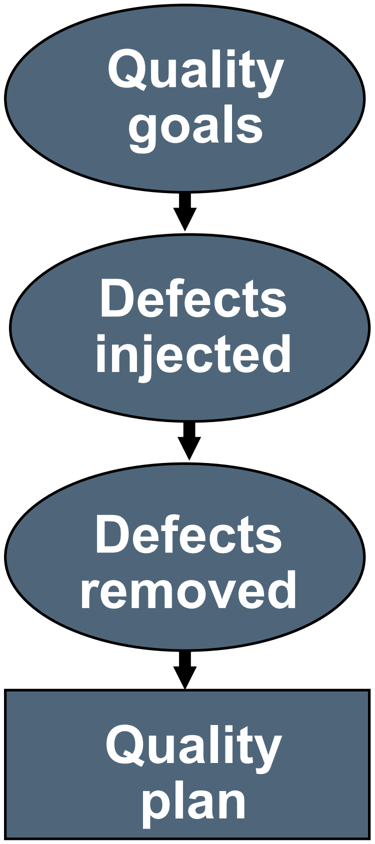
s is calculated as Size / Rate
lculation, size or rate must b

Assembly	Phase	Task	Plan Hours	Plan Date	Plan Week
Gateway BIO Moc	HLD	Modulebus diag. (Phase 2) - design	6.0	2/18/2002	26
Gateway BIO Moc	CODE	Modulebus diag. (Phase 2) - coding	24.0	2/25/2002	27
Gateway BIO Moc	CR	Modulebus diag. (Phase 2) - code review	6.0	3/4/2002	28
Gateway BIO Moc	COMPILE	Modulebus diag. (Phase 2) - compile	3.0	3/4/2002	28
Gateway BIO Moc	UT	Modulebus diag. (Phase 2) - unit test	6.0	3/4/2002	28
Gateway BIO Moc	UT	IOR S800 interface HW testing	0.0	3/4/2002	28
Gateway BIO Moc	HLD	Modulebus Software BRC-based version - Design	15.5	3/11/2002	29
Gateway BIO Hot	HLDINSP	HotSwap Task - Design Inspection	1.7	3/18/2002	30

The Earned Value Plan



Meeting 5 - Build the Quality Plan



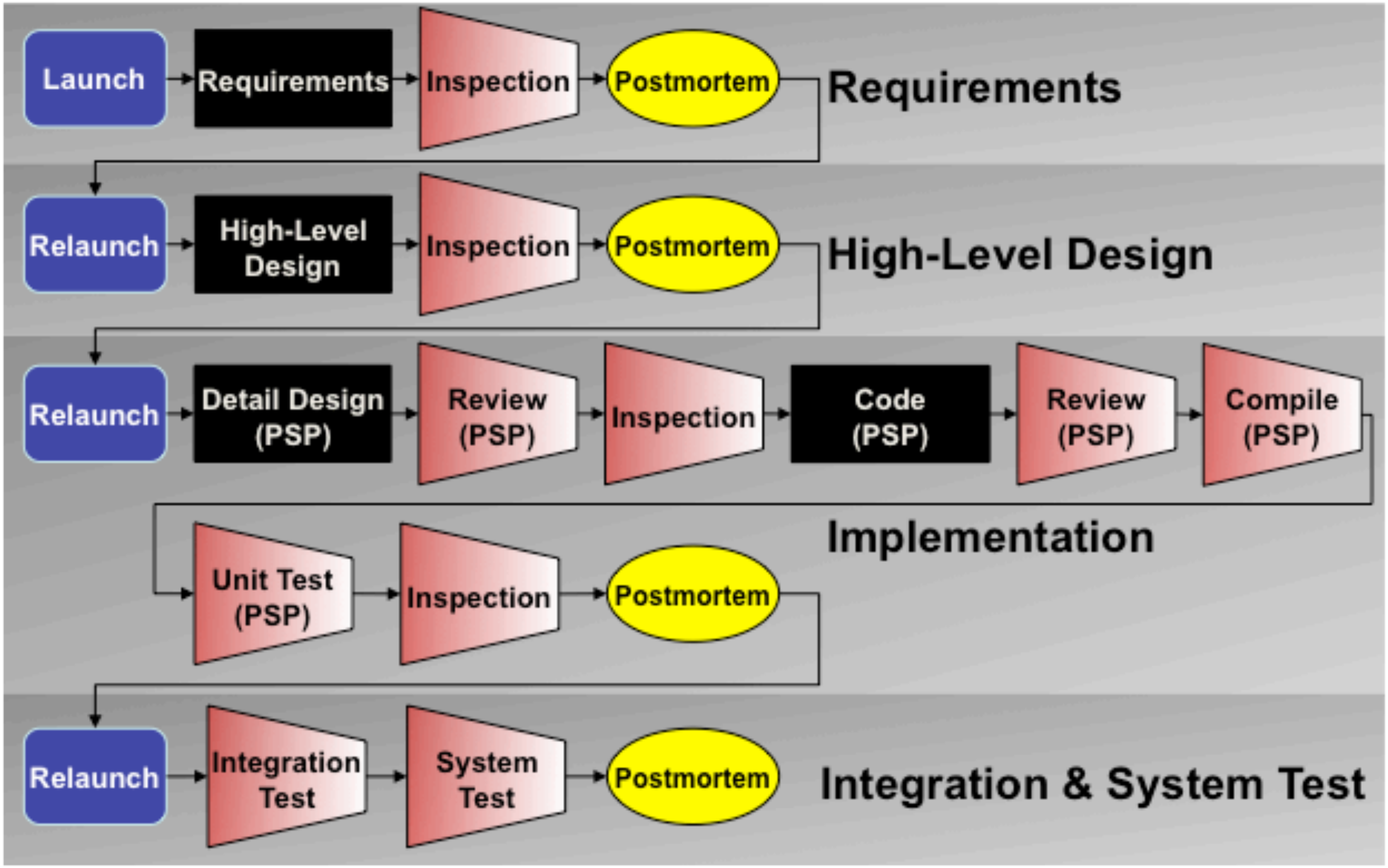
In launch meeting 5, the team builds a quality plan that estimates the

- number of defects that will be injected in each phase
- number of defects that will be removed in each phase
- quality (defect density) of the final product
- quality (process quality index) of the development process

The team ensures that the plan meets the quality goals.

Defect Removal Filters

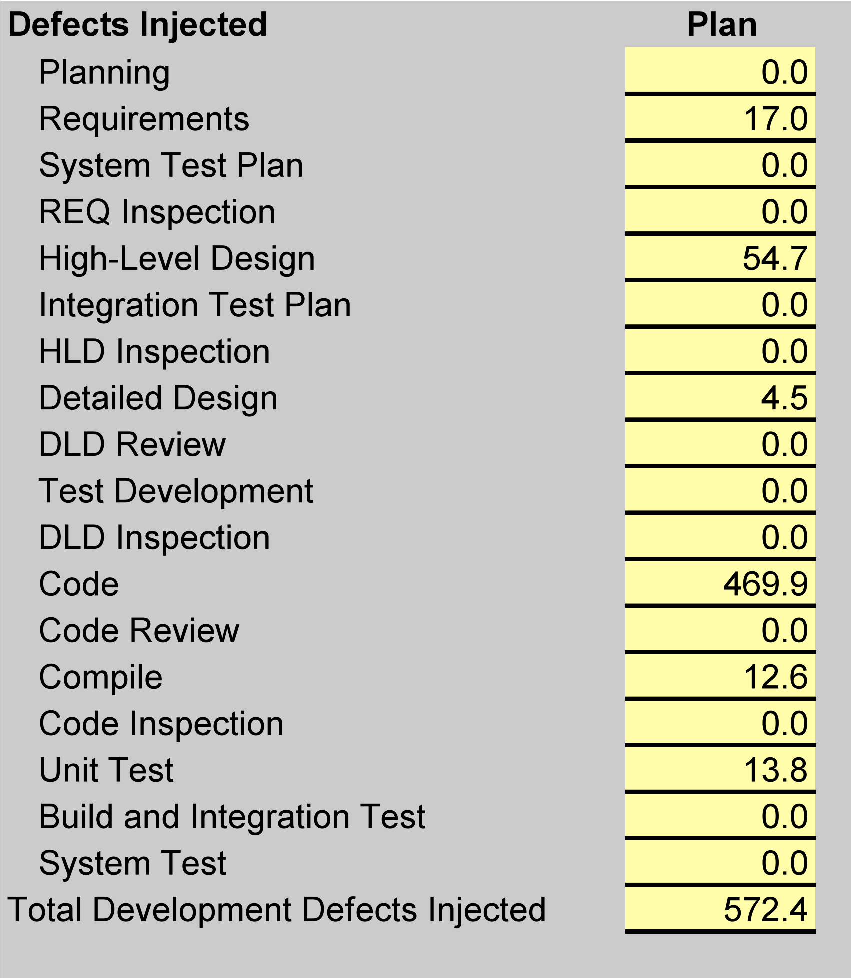
🟦 Planning
⬛ Development
▷ Defect Filter
🟡 Postmortem



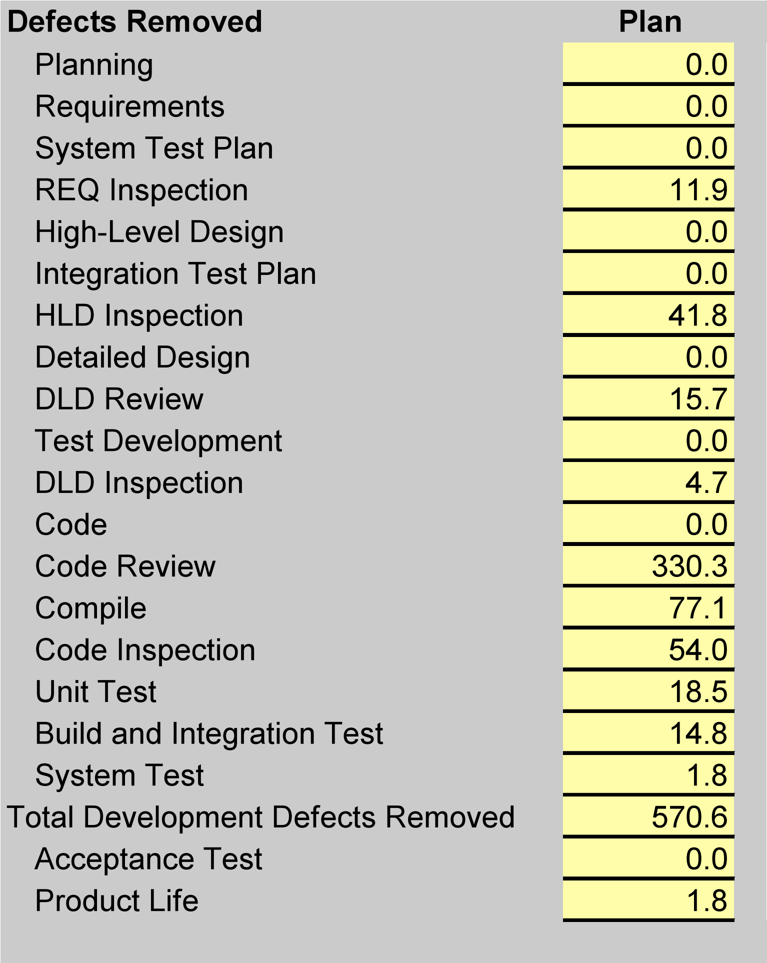
Economics of Quality

	Efficiency	Effectiveness	Predictability
	Avg. removal rate (defects/hr)	Phase yields (% of defects removed)	Estimated effort
Design Review	1.5	50% to 70%	Low variability - based on product size
Design Inspection	0.5	50% to 70%	
Code Review	4	50% to 70%	
Code Inspection	1	50% to 70%	
Unit Test	0.2	35% to 50%	High variability - based on time to find & fix defects
Integration Test	0.1	35% to 50%	
System Test	0.05	35% to 50%	

TSP Defect Injection/Removal Plan



Defects Injected	Plan
Planning	0.0
Requirements	17.0
System Test Plan	0.0
REQ Inspection	0.0
High-Level Design	54.7
Integration Test Plan	0.0
HLD Inspection	0.0
Detailed Design	4.5
DLD Review	0.0
Test Development	0.0
DLD Inspection	0.0
Code	469.9
Code Review	0.0
Compile	12.6
Code Inspection	0.0
Unit Test	13.8
Build and Integration Test	0.0
System Test	0.0
Total Development Defects Injected	572.4



Defects Removed	Plan
Planning	0.0
Requirements	0.0
System Test Plan	0.0
REQ Inspection	11.9
High-Level Design	0.0
Integration Test Plan	0.0
HLD Inspection	41.8
Detailed Design	0.0
DLD Review	15.7
Test Development	0.0
DLD Inspection	4.7
Code	0.0
Code Review	330.3
Compile	77.1
Code Inspection	54.0
Unit Test	18.5
Build and Integration Test	14.8
System Test	1.8
Total Development Defects Removed	570.6
Acceptance Test	0.0
Product Life	1.8

Quality Plan Contents

The quality plan goals are used like control limits, to support early identification of quality problems.

The quality plan includes these key derived measures

- percent defect free
- yield by phase
- inspection and review rates
- defect density by phase
- development time ratios
- defect ratios

Assessing the Plan for Quality

High quality is not achieved by accident.

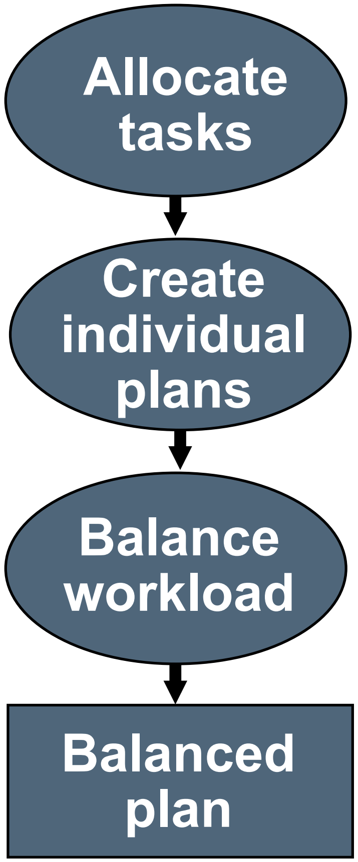
The team's plan should include defect removal and defect prevention steps before testing.

Adequate time should be planned for

- requirements and design
- personal reviews
- inspections by peers

The planned product defect density should meet the team's quality goal.

Meeting 6 - Build Individual Work Plans



In launch meeting 6, each team member builds a plan to which he or she can commit.

In building their plans, the team members

- allocate tasks to individuals
- refine size and effort estimates using their own data and processes
- break tasks to the granularity of around 10 hours or less per task
- estimate their own available task hours for each week
- create an earned-value plan
- balance workloads across all team members.

The Need for Detailed Individual Work Plans

With detailed plans the engineers

- know what tasks to do next
- get data for future detailed planning
- have plans that are easier to compare with actual results

With detailed plans, engineers can also

- track progress in detail
- know where they are on the plan
- get regular performance feedback

Personal Plan Review

Each developer presents his or her plan.

Team members consider whether

- the plan is complete
- the plan is sufficiently detailed
- the tasks are consistent with the team’s overall plan
- the plans should be adjusted
- the plan seems reasonable and achievable

The developer makes any needed adjustments.

Plan Consolidation

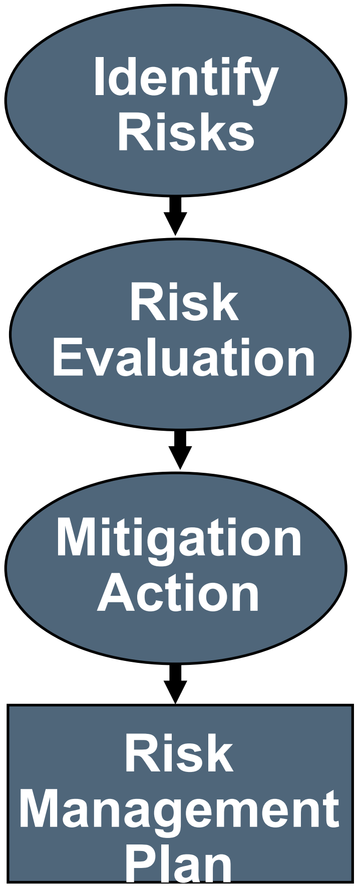
The planning manager then leads the team in producing a composite team plan.

The product is based on a roll-up of the individual plans using the TSP support tool.

If the rolled-up plan does not match the top-down plan, adjustments are made

- balance workload or add resources
- increase schedule
- reduce requirements

Meeting 7 - Risk Management

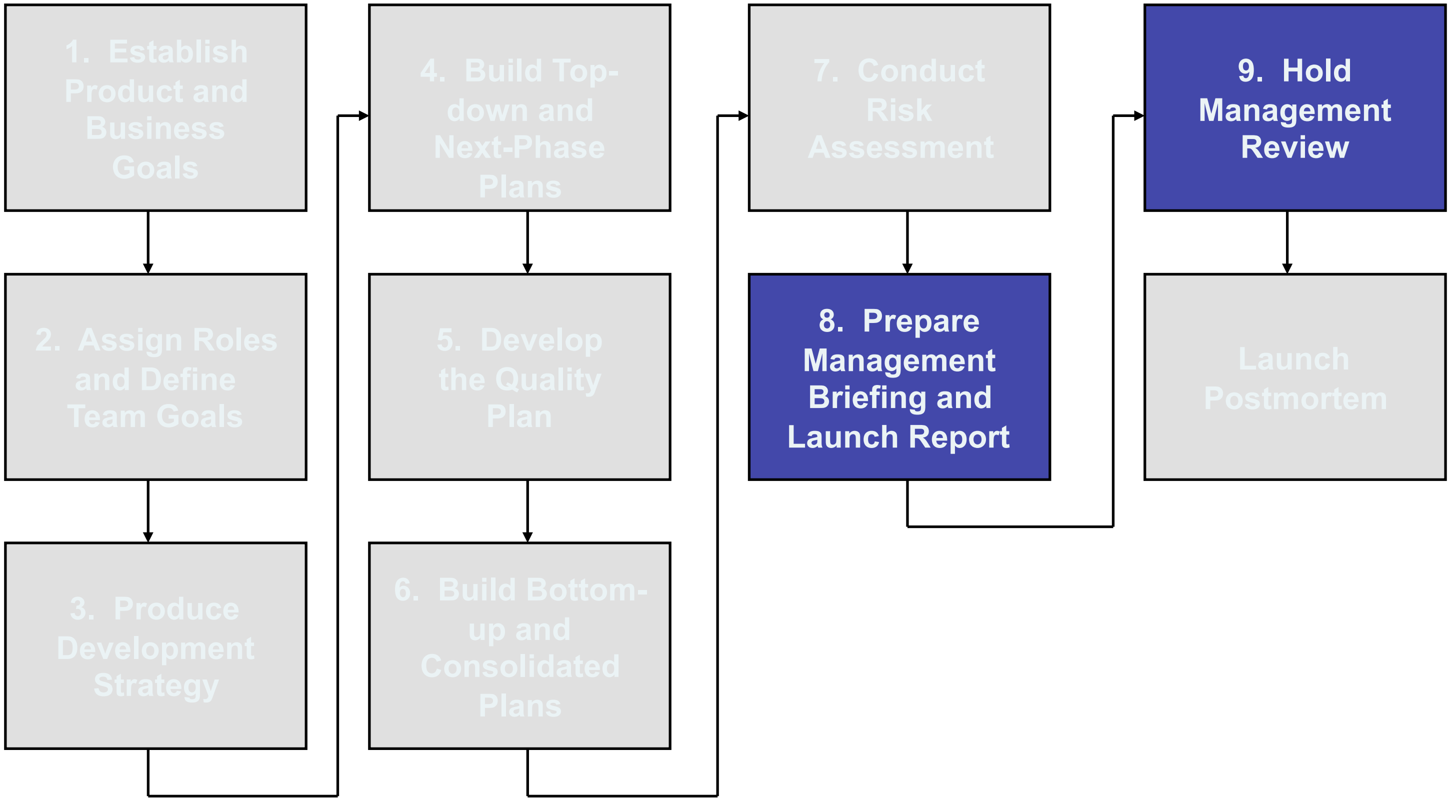


In launch meeting 7, the team develops a risk management plan.

In building the risk management plan the team members

- identify project risks
- evaluate each risk for high, medium, or low impact and likelihood
- assign each high or medium risk to a team member and define an action date and mitigation strategy

Launch Meetings 8 and 9


1. Establish Product and Business Goals
2. Assign Roles and Define Team Goals
3. Produce Development Strategy
4. Build Top-down and Next-Phase Plans
5. Develop the Quality Plan
6. Build Bottom-up and Consolidated Plans
7. Conduct Risk Assessment
8. Prepare Management Briefing and Launch Report
9. Hold Management Review
Launch Postmortem

The Project Plan Summary

The project produced a very detailed plan but it had at least one problem, it did not meet management’s goal for release.

Size and Effort	Plan
Size (New and Changed LOC)	110,000
Effort (Task Hours)	16,000
Schedule Months	18

Product Quality (Defects/KLOC)	Plan
Integration Test	1.0
System Test	0.1
Field Trial	0.0
Operation	***0.0***

Meeting 9 - Making a Commitment

Day 34

The team leader briefed management on the plan.

Under strong management pressure, the team

- explained their approach.
- justified the effort required.

Management reluctantly accepted the plan and the team began development.

Agenda

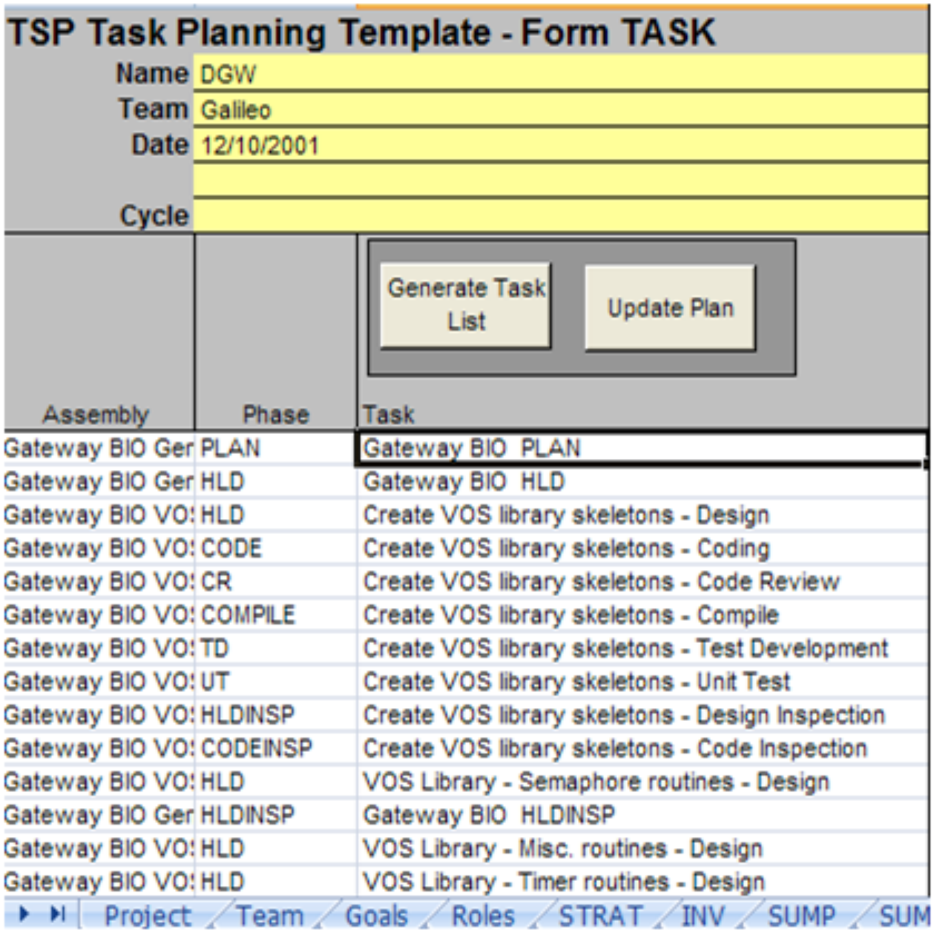
When	•Topics
9:00 – Break	•Case study: a project in trouble •Team Software Process and its implementation strategy •TSP concepts
Break – Lunch	•Why projects fail •Case study: launching the project
Lunch – Break	•Case study: launching the project (continued)
Break – 5:30	•Case study: team-working framework •Corporate experience with TSP •TSP and CMMI •Building internal support for TSP

Work Begins

Day 37

Weekly planning

- Priorities for the week
- Task order
- Time Tracking
- Defect Tracking
- Task Completion

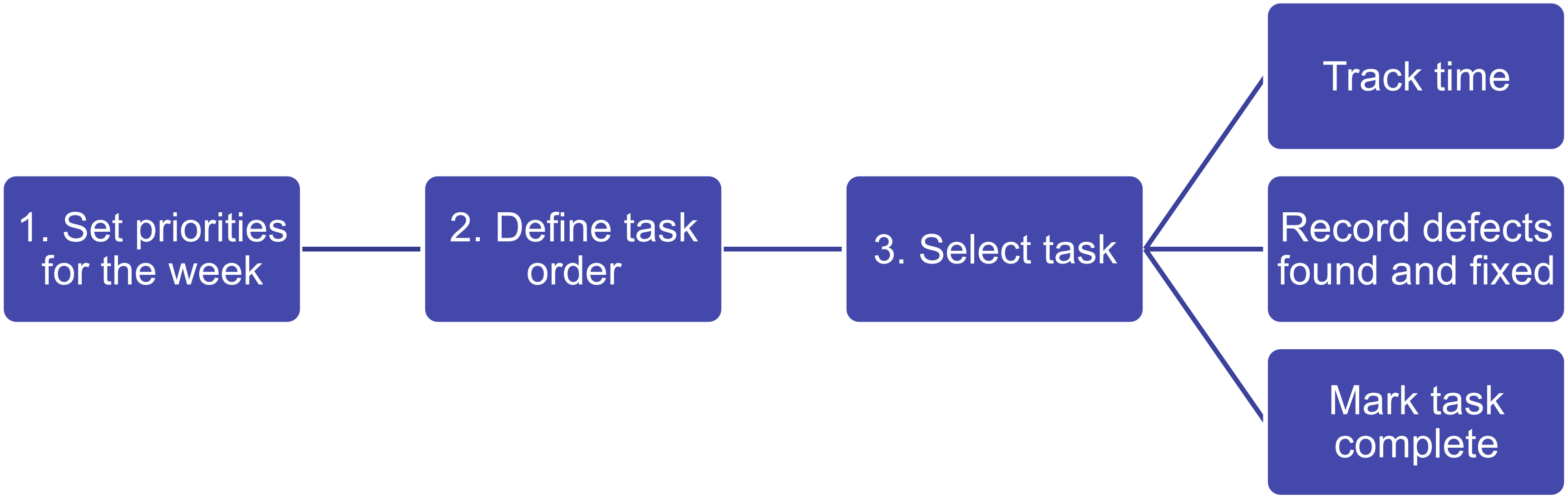


TSP Task Planning Template - Form TASK

Name	DGW
Team	Galileo
Date	12/10/2001
Cycle	

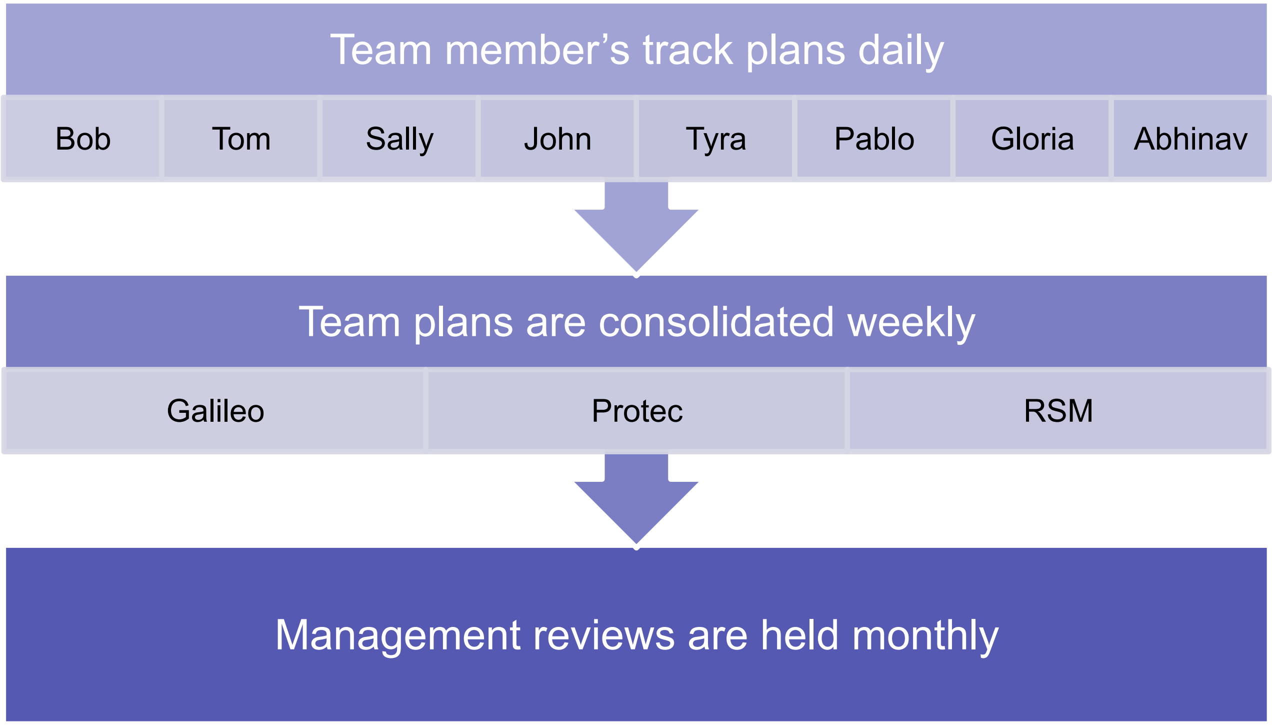
Assembly	Phase	Task
Gateway BIO Ger	PLAN	Gateway BIO PLAN
Gateway BIO Ger	HLD	Gateway BIO HLD
Gateway BIO VO:	HLD	Create VOS library skeletons - Design
Gateway BIO VO:	CODE	Create VOS library skeletons - Coding
Gateway BIO VO:	CR	Create VOS library skeletons - Code Review
Gateway BIO VO:	COMPILE	Create VOS library skeletons - Compile
Gateway BIO VO:	TD	Create VOS library skeletons - Test Development
Gateway BIO VO:	UT	Create VOS library skeletons - Unit Test
Gateway BIO VO:	HLDINSP	Create VOS library skeletons - Design Inspection
Gateway BIO VO:	CODEINSP	Create VOS library skeletons - Code Inspection
Gateway BIO VO:	HLD	VOS Library - Semaphore routines - Design
Gateway BIO Ger	HLDINSP	Gateway BIO HLDINSP
Gateway BIO VO:	HLD	VOS Library - Misc. routines - Design
Gateway BIO VO:	HLD	VOS Library - Timer routines - Design

Personal Tracking



The Team-Working Framework

The TSP team-working framework helps the project move forward.



Team member's track plans daily

Bob | Tom | Sally | John | Tyra | Pablo | Gloria | Abhinav

Team plans are consolidated weekly

Galileo | Protec | RSM

Management reviews are held monthly

Weekly Status

Team members meet each week to assess progress.

- Role managers present evaluation of the plan and data
- Goal owners present status on product and business objectives
- Risk owners present status on risk mitigation plans and new risks
- Team members present status on their plans

Plan deviations are addressed each week.

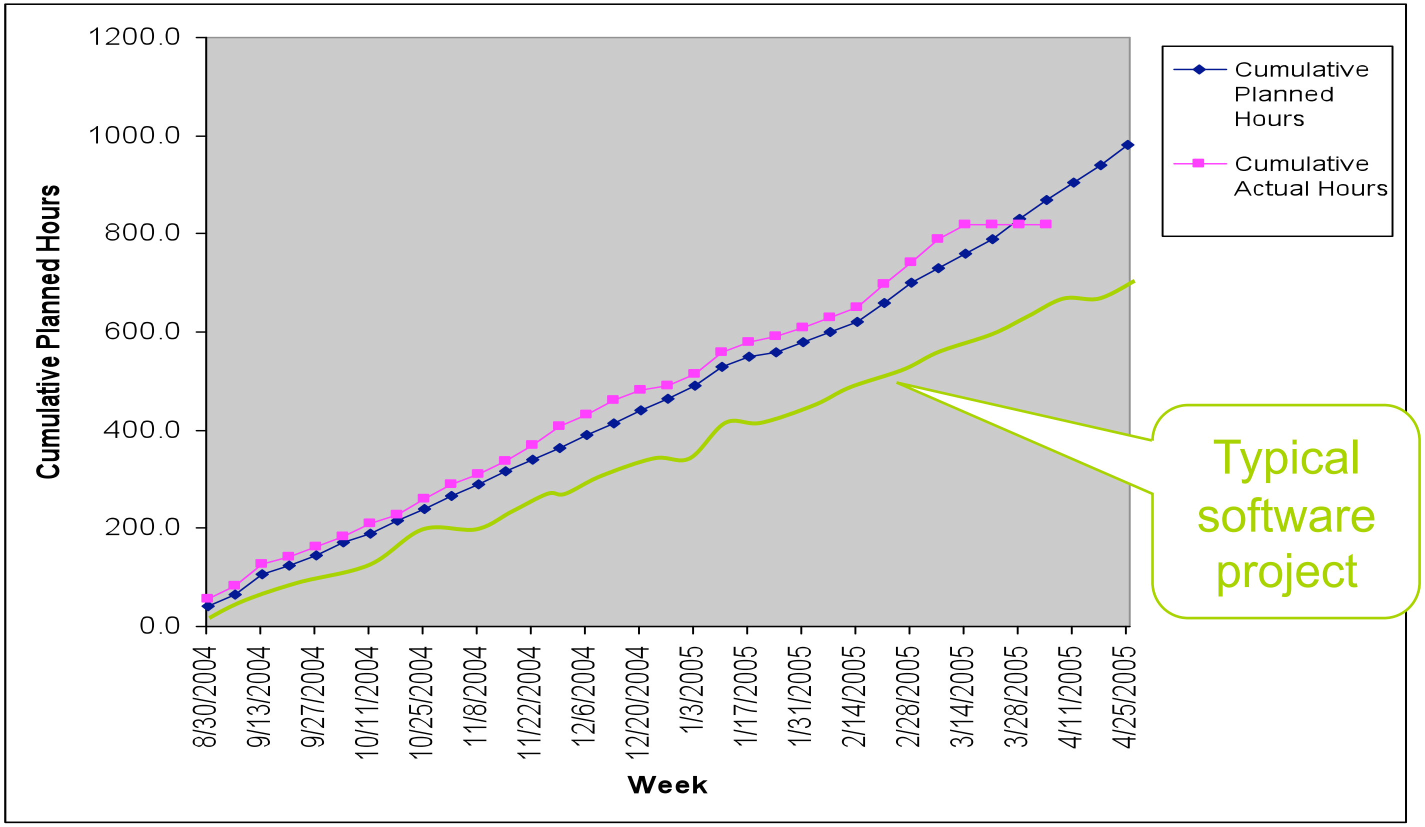
Significant deviations, e.g. new requirements, trigger a replan.

Performance Data Reviewed

- Baseline Plan Value
- Plan Value
- Earned Value
- Predicted Earned Value
- Earned Value Trend
- Plan Task Hours
- Actual Task Hours
- Tasks/Milestones completed
- Tasks/Milestones past due
- Tasks/Milestones next 2 weeks
- Effort against incomplete tasks
- Estimation Accuracy
- Review and Inspection Rates
- Injection Rates
- Removal Rates
- Time in Phase Ratios
- Phase and Process Yield
- Defect Density
- Quality Profile (QP)
- QP Index
- Percent Defect Free
- Defect Removal Profile
- Plan to Actual Defects Injected/Removed

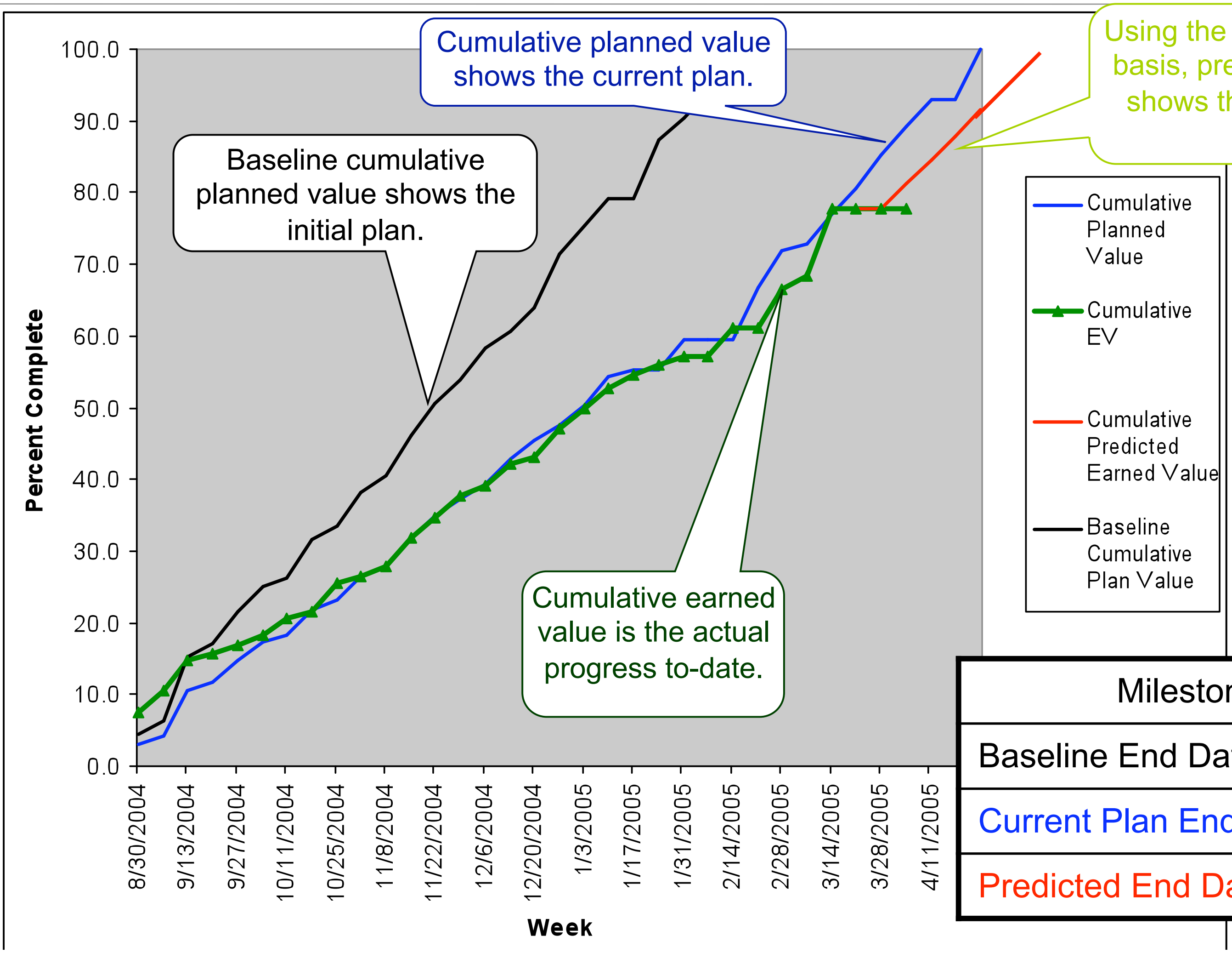
Resource Tracking

Cumulative plan and actual resource hours shows resource burn rate and potential source of slip



Typical software project

Earned Value Tracking



Cumulative planned value shows the current plan.

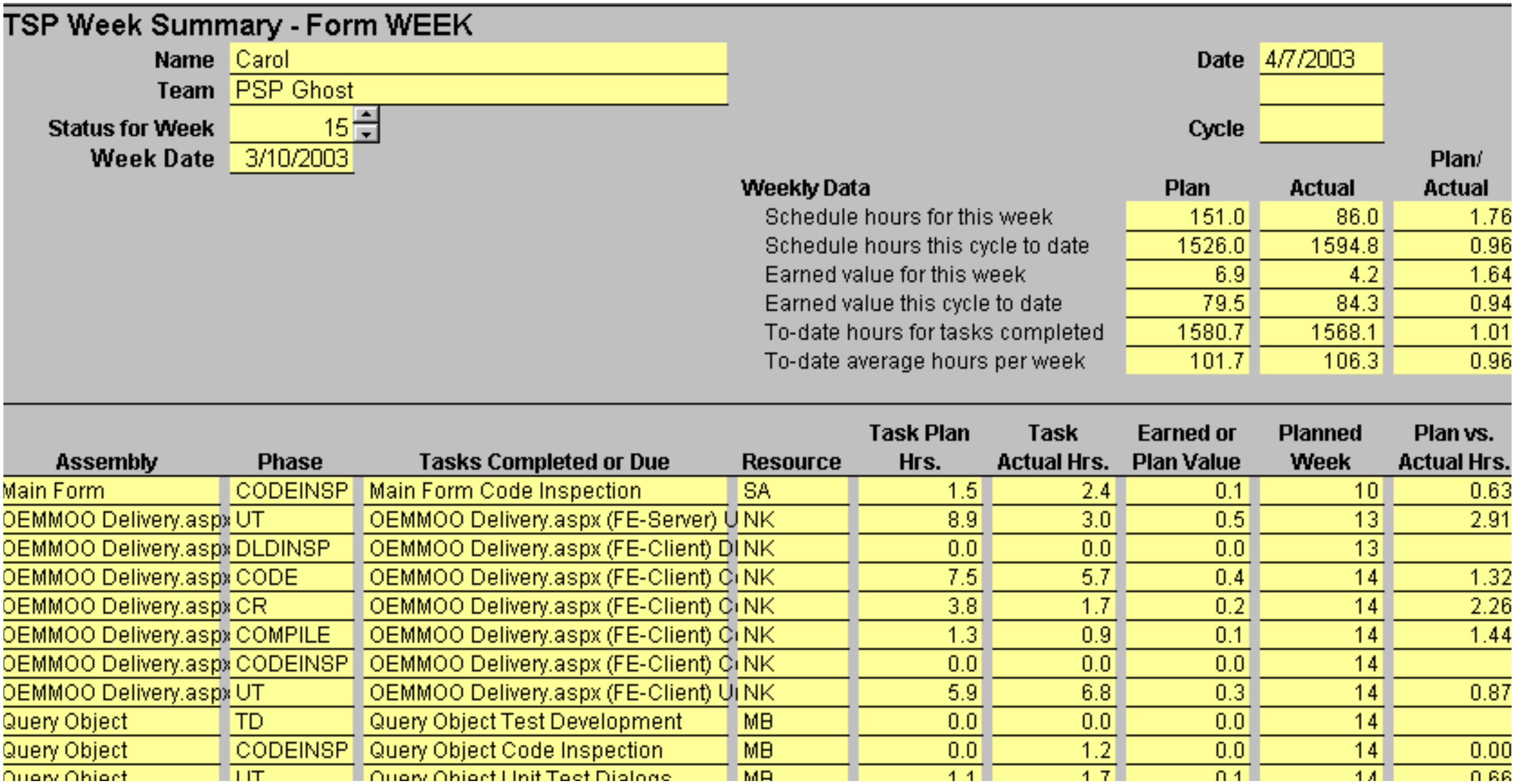
Baseline cumulative planned value shows the initial plan.

Cumulative earned value is the actual progress to-date.

Using the rate of progress as a basis, predicted earned value shows the likely completion date.

Milestone	Date
Baseline End Date	2/14
Current Plan End Date	4/25
Predicted End Date	5/16

TSP Weekly Status Report



TSP Week Summary - Form WEEK

Name: Carol

Team: PSP Ghost

Status for Week: 15

Week Date: 3/10/2003

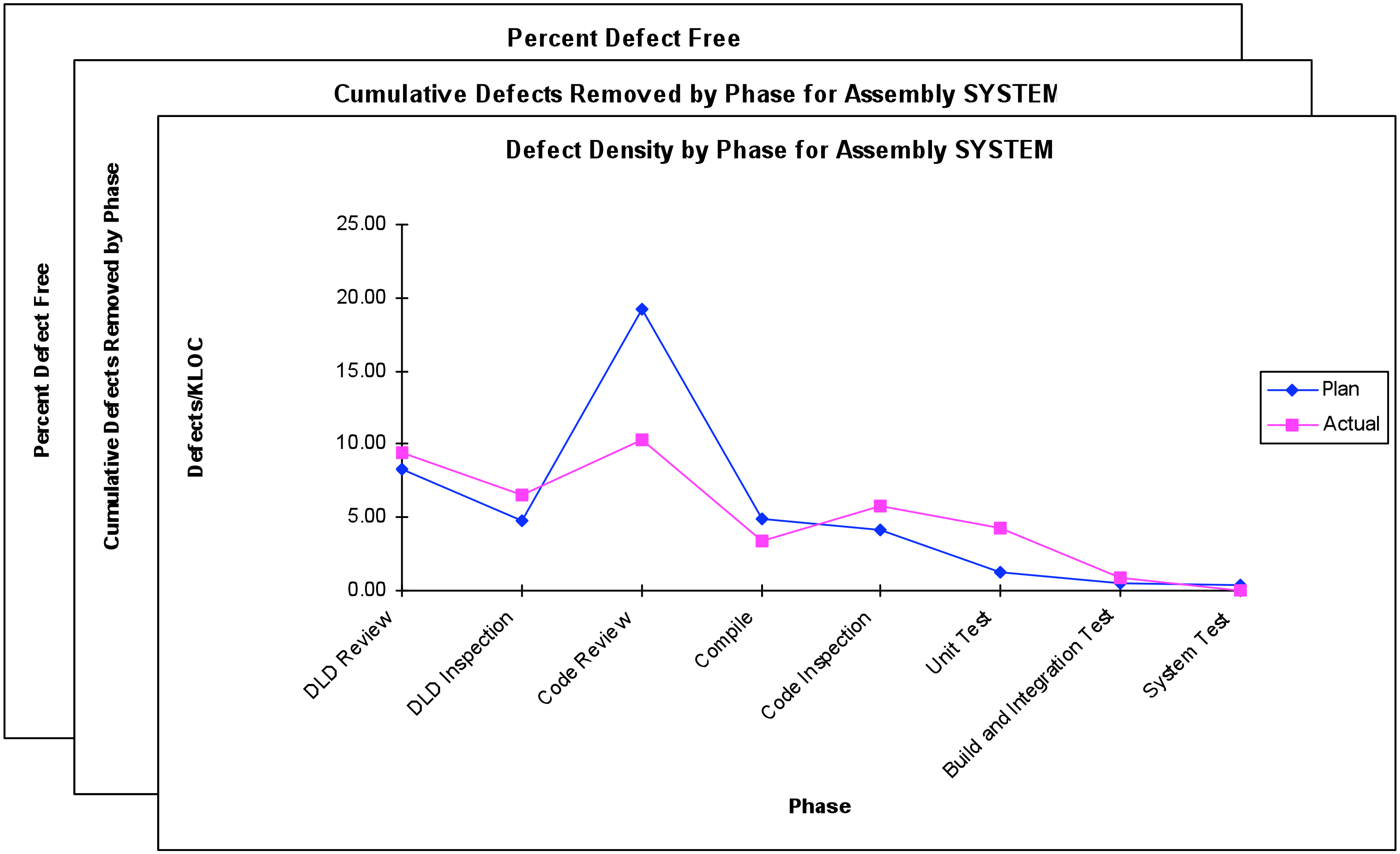
Date: 4/7/2003

Cycle:

Weekly Data	Plan	Actual	Plan/ Actual
Schedule hours for this week	151.0	86.0	1.76
Schedule hours this cycle to date	1526.0	1594.8	0.96
Earned value for this week	6.9	4.2	1.64
Earned value this cycle to date	79.5	84.3	0.94
To-date hours for tasks completed	1580.7	1568.1	1.01
To-date average hours per week	101.7	106.3	0.96

Assembly	Phase	Tasks Completed or Due	Resource	Task Plan Hrs.	Task Actual Hrs.	Earned or Plan Value	Planned Week	Plan vs. Actual Hrs.
Main Form	CODEINSP	Main Form Code Inspection	SA	1.5	2.4	0.1	10	0.63
OEMMOO Delivery.aspx	UT	OEMMOO Delivery.aspx (FE-Server) U	NK	8.9	3.0	0.5	13	2.91
OEMMOO Delivery.aspx	DLDINSP	OEMMOO Delivery.aspx (FE-Client) DI	NK	0.0	0.0	0.0	13	
OEMMOO Delivery.aspx	CODE	OEMMOO Delivery.aspx (FE-Client) C	NK	7.5	5.7	0.4	14	1.32
OEMMOO Delivery.aspx	CR	OEMMOO Delivery.aspx (FE-Client) C	NK	3.8	1.7	0.2	14	2.26
OEMMOO Delivery.aspx	COMPILE	OEMMOO Delivery.aspx (FE-Client) C	NK	1.3	0.9	0.1	14	1.44
OEMMOO Delivery.aspx	CODEINSP	OEMMOO Delivery.aspx (FE-Client) C	NK	0.0	0.0	0.0	14	
OEMMOO Delivery.aspx	UT	OEMMOO Delivery.aspx (FE-Client) U	NK	5.9	6.8	0.3	14	0.87
Query Object	TD	Query Object Test Development	MB	0.0	0.0	0.0	14	
Query Object	CODEINSP	Query Object Code Inspection	MB	0.0	1.2	0.0	14	0.00
Query Object	UT	Query Object Unit Test Dialogs	MB	1.1	1.7	0.1	14	0.66

Quality Tracking



Quality Profile

The TSP Quality Profile is a quality early warning indicator.

It examines criteria that are effective predictors of system test and post-release quality, and produces a graph of the result.

It supports drill down to any level for further analysis, e.g. in software:

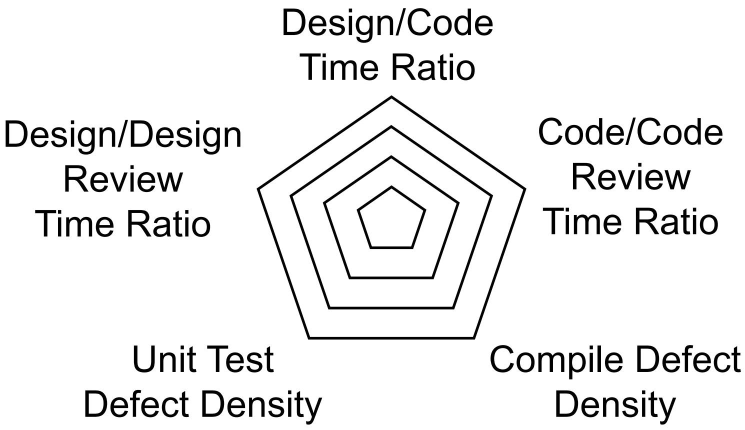
system → component → module → class.

Quality Profile Criteria

1. Design time = coding time
2. Design review time = ½ design time
3. Code review time = ½ coding time
4. Compile defects < 10 per KLOC
5. Unit test defects < 5 per KLOC

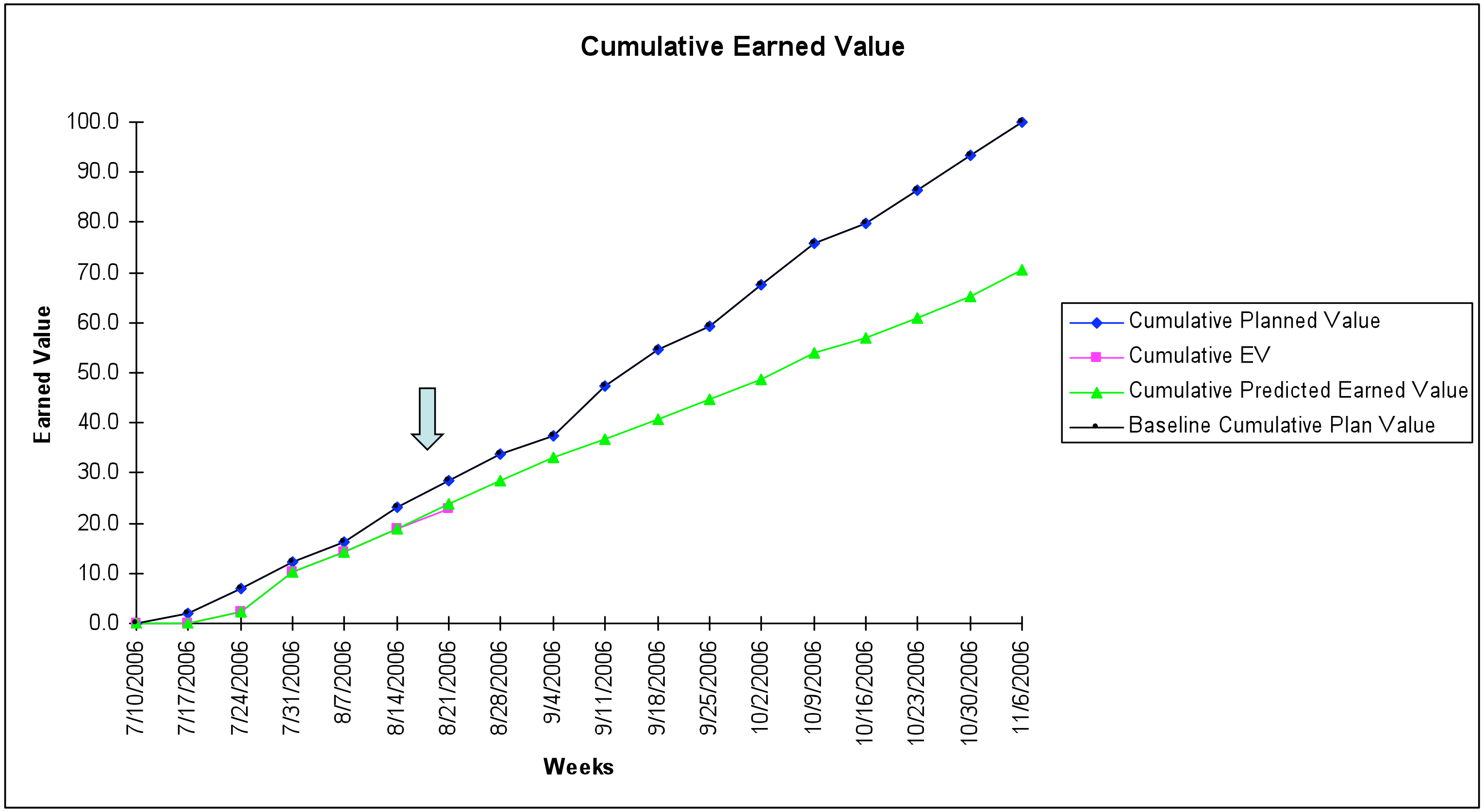
If satisfied, a criterion has a value of 1, and is drawn along the outer edge of the chart.

Quality Profile

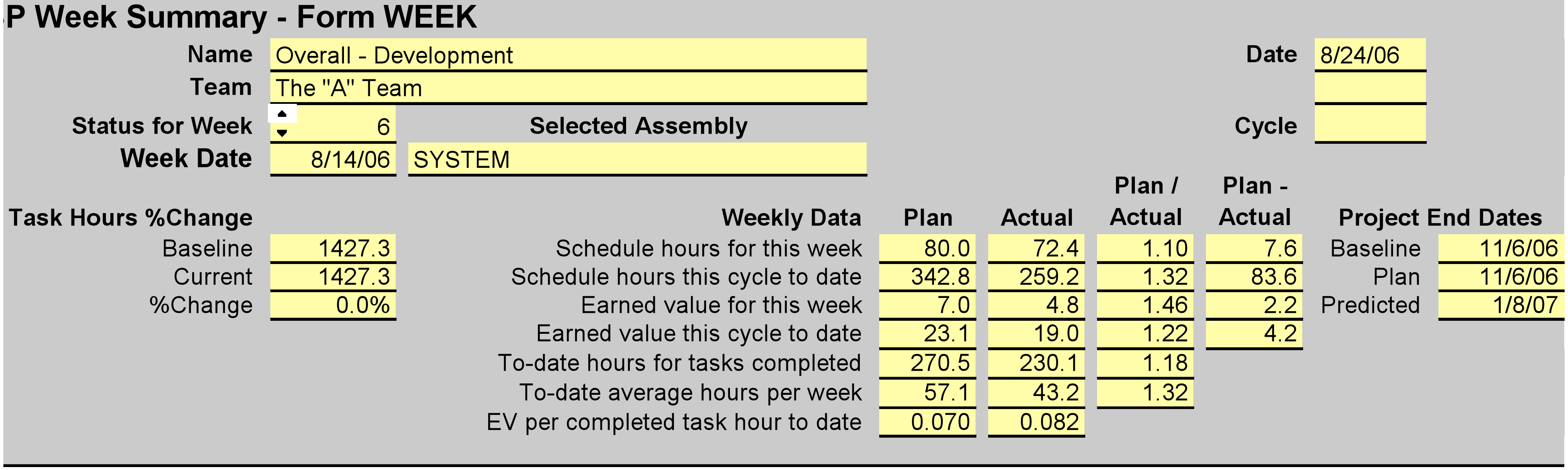


The Project in Week 6

Day 79



Why Are We Behind?



P Week Summary - Form WEEK

Name	Overall - Development		Date	8/24/06
Team	The "A" Team			
Status for Week	6	Selected Assembly	Cycle	
Week Date	8/14/06	SYSTEM		

Task Hours %Change	
Baseline	1427.3
Current	1427.3
%Change	0.0%

Weekly Data	Plan	Actual	Plan / Actual	Plan - Actual
Schedule hours for this week	80.0	72.4	1.10	7.6
Schedule hours this cycle to date	342.8	259.2	1.32	83.6
Earned value for this week	7.0	4.8	1.46	2.2
Earned value this cycle to date	23.1	19.0	1.22	4.2
To-date hours for tasks completed	270.5	230.1	1.18	
To-date average hours per week	57.1	43.2	1.32	
EV per completed task hour to date	0.070	0.082		

Project End Dates	
Baseline	11/6/06
Plan	11/6/06
Predicted	1/8/07

- Earned Value is Behind by 22%
- 18% over-estimated for work completed thus far
- **32% over-estimated “on-task” hours**
- If we do nothing different it is likely we will finish 2 months behind

Plan Dynamics

Teams need detailed plans to make realistic commitments and to coordinate and track their work.

Detailed plans are accurate only for brief periods of time.

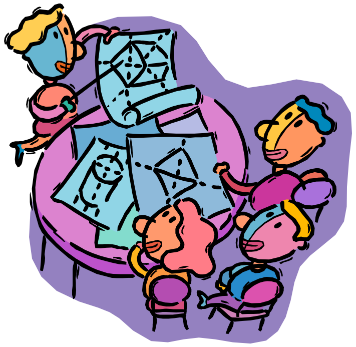
- As engineers work, they learn more about the job.
- The work among individuals becomes unbalanced.
- Organizations and teams are dynamic.

Consequently, TSP teams maintain their plans dynamically.

Communicating Commitment Changes

Whenever changes are made to the plan, the team must make sure they

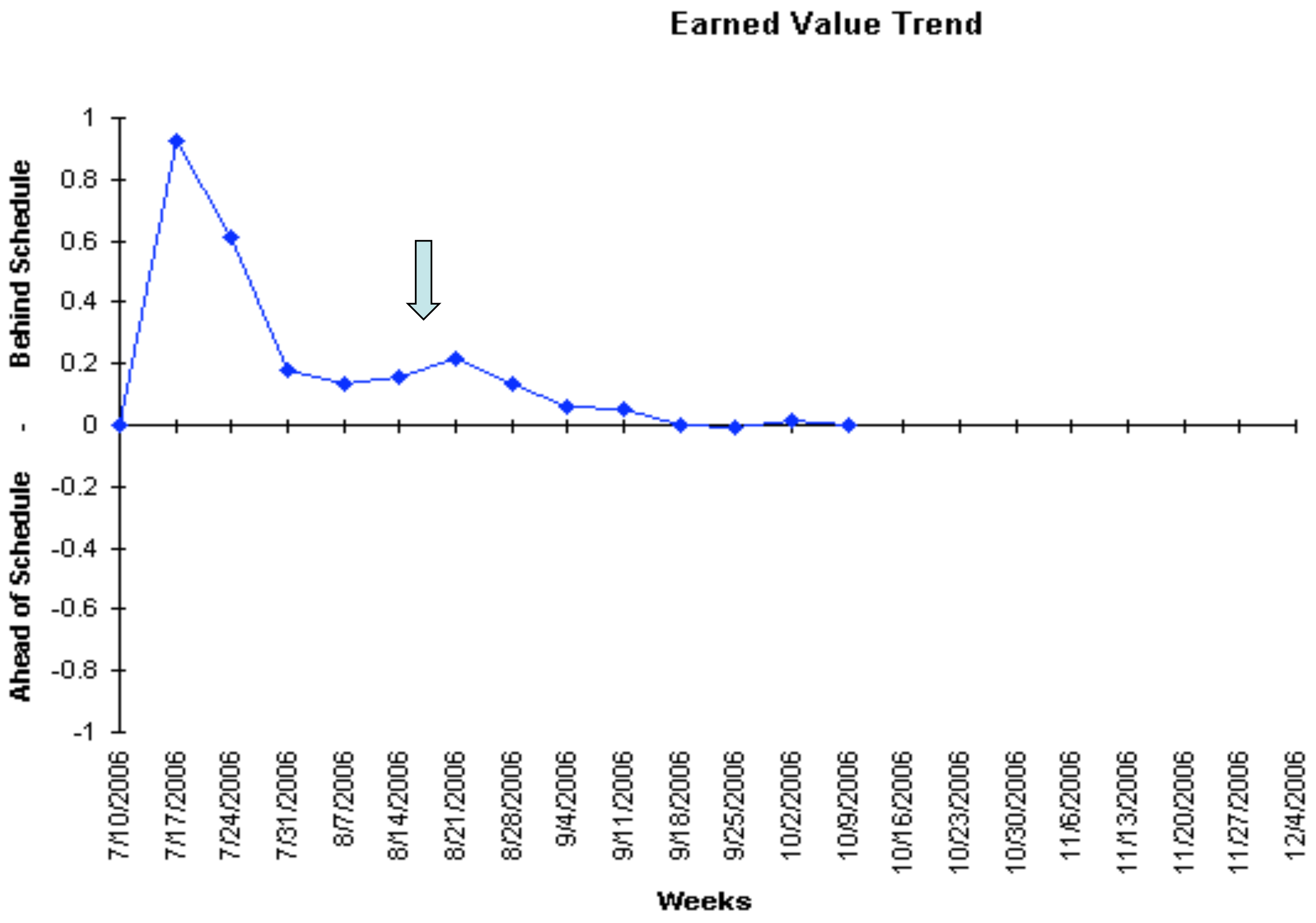
- review plan changes
- verify they understand why the changes were made
- review impact to commitments



If the change to the plan impacts commitments

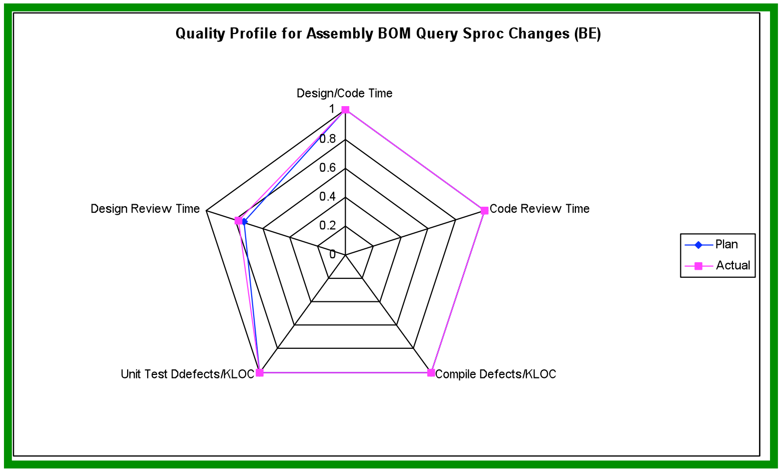
- consider alternatives
- offer choices and recommendations to stakeholders

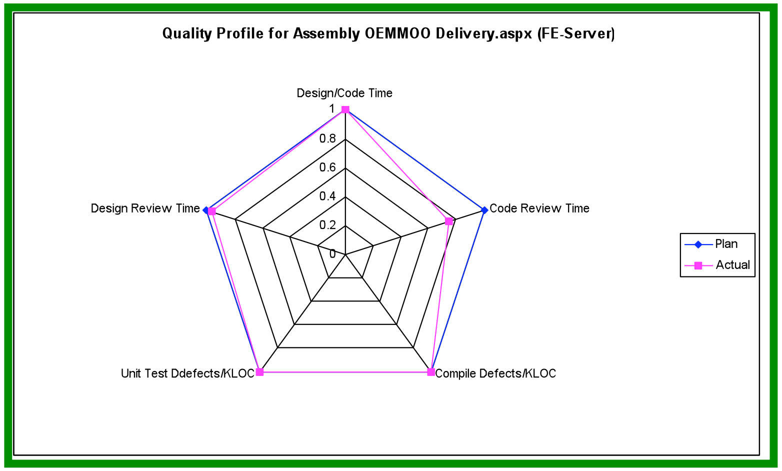
Focus Hours Works!

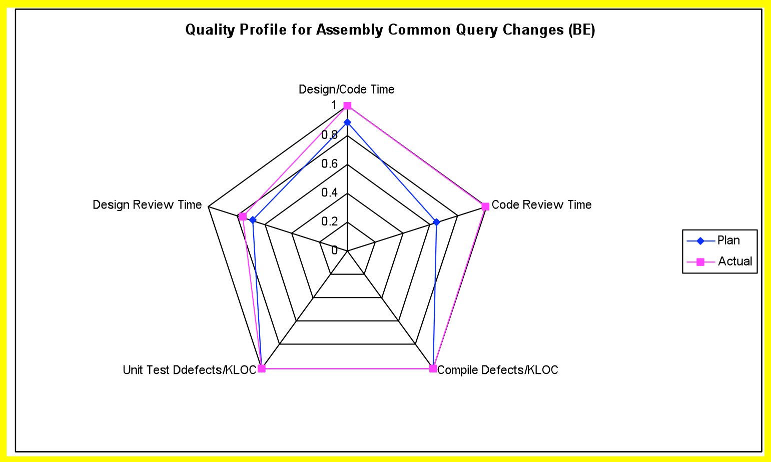


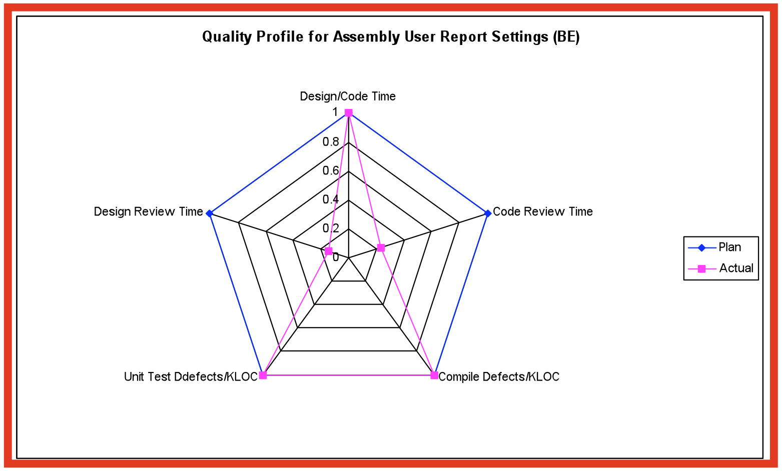
A Project Quality Problem

Month 15


Quality Profile for Assembly BOM Query Sproc Changes (BE)


Quality Profile for Assembly OEMMOO Delivery.aspx (FE-Server)


Quality Profile for Assembly Common Query Changes (BE)


Quality Profile for Assembly User Report Settings (BE)

Teamwork: Results

Month 17

The project was completed 17 months later with these results.

- Quality levels improved 20 times over prior projects.
- Actual effort and schedule were within 8% of plan (early).

The product worked so well that the customer ended their relationship with the competitor.

Size and Effort	Plan	Actual
Size (New and Changed LOC)	110,000	89,900
Effort (Task Hours)	16,000	14,711
Schedule Months	18	17

Product Quality (Defects/KLOC)	Plan	Actual
Integration Test	1.0	0.2
System Test	0.1	0.4
Field Trial	0.0	0.02
Operation	***0.0***	***0.0***

Agenda

When	•Topics
9:00 – Break	•Case study: a project in trouble •Team Software Process and its implementation strategy •TSP concepts
Break – Lunch	•Why projects fail •Case study: launching the project
Lunch – Break	•Case study: launching the project (continued)
Break – 5:30	•Case study: team-working framework •Corporate experience with TSP •TSP and CMMI •Building internal support for TSP

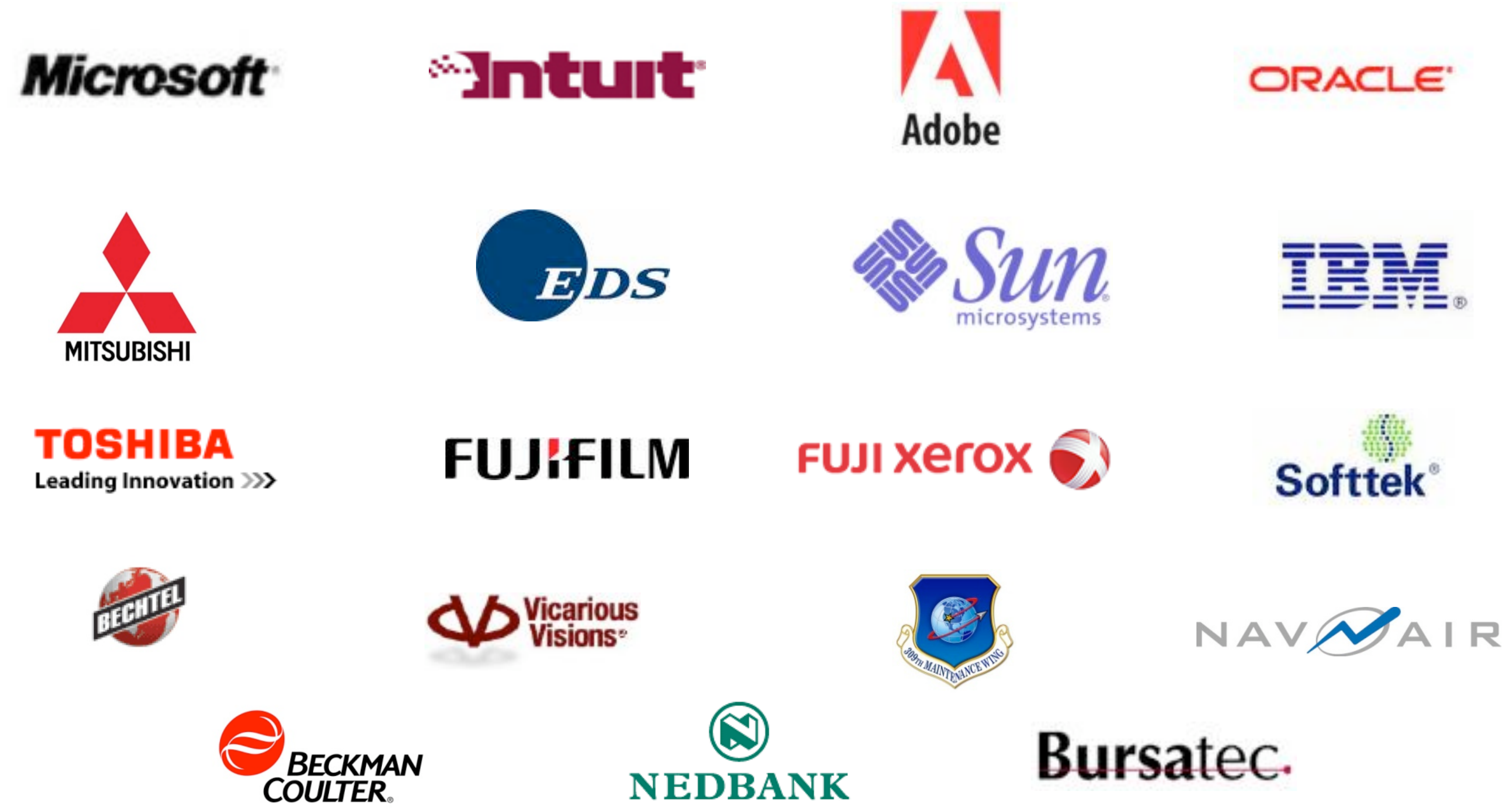
User Experience and the Business Case for TSP

The principal costs of introducing TSP are training costs and lost opportunity cost resulting from time spent in training.

The principal benefits are

- lower development costs and shorter schedules
- more functionality per release and improved productivity
- lower defect density in both system test and in the delivered product
- improved work-life balance for the developers
- improved customer satisfaction
- fast track to higher performance and higher maturity

TSP Adoption



Schedule Management

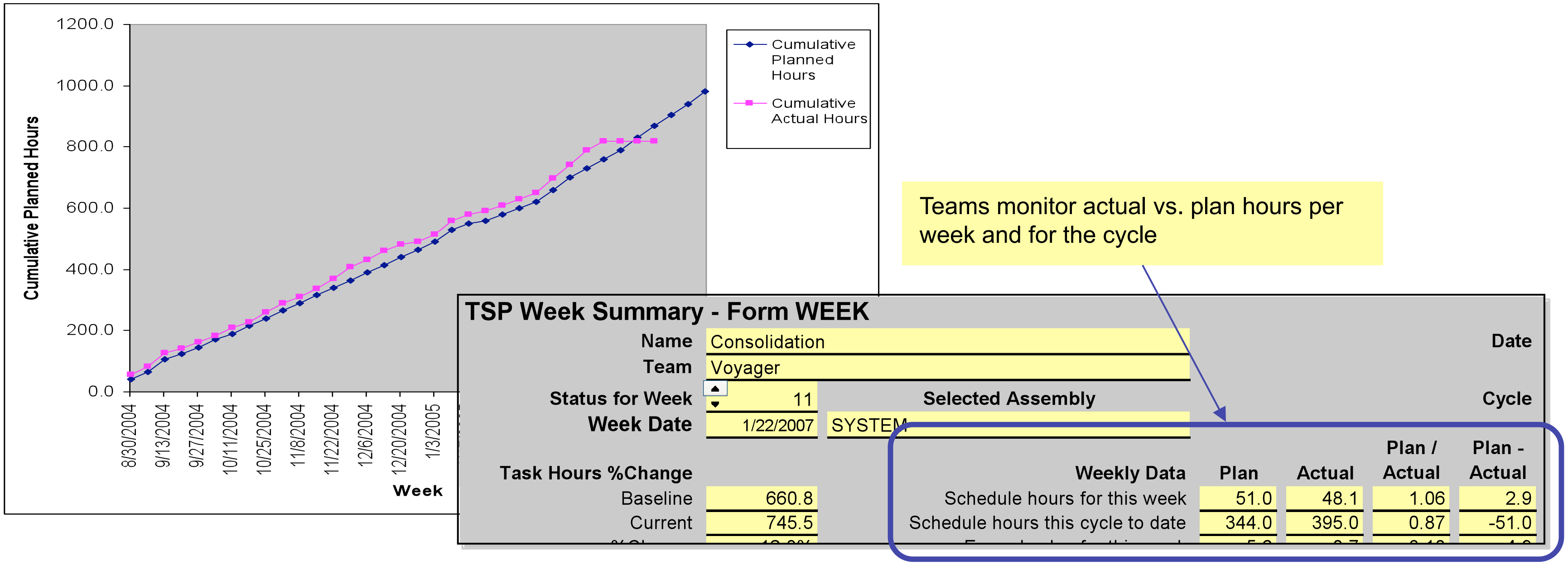
First-time TSP projects at Microsoft had a 10 times better mean schedule error than non-TSP projects at Microsoft as reflected in the following table.

Microsoft Schedule Results	Non-TSP Projects	TSP Projects
Released on Time	42%	66%
Average Days Late	25	6
Mean Schedule Error	10%	1%
Sample Size	80	15

Managing Task Hours

Task hours are the hours that teams spend on planned tasks and do not include unplanned but necessary tasks like meetings, courses, coordination, handling mail, etc.

When measured, tracked, and managed, the team can usually improve task hours, but management can't. ***Why?***



Teams monitor actual vs. plan hours per week and for the cycle

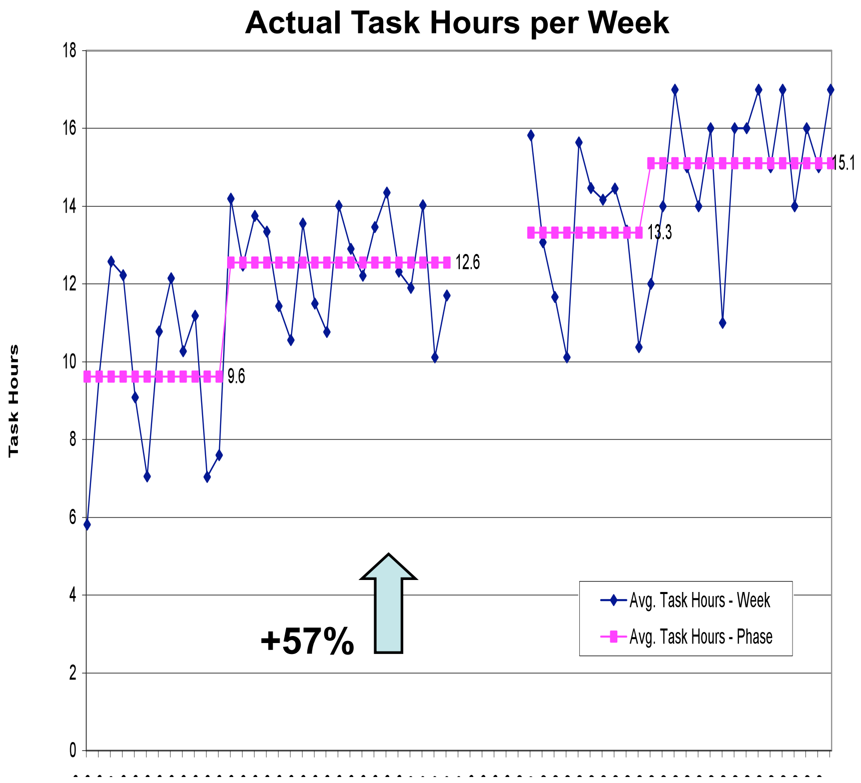
TSP Week Summary - Form WEEK

Name	Consolidation						**Date**
Team	Voyager						
Status for Week	11	**Selected Assembly**					**Cycle**
Week Date	1/22/2007	SYSTEM					
						Plan /	**Plan -**
Task Hours %Change			**Weekly Data**	**Plan**	**Actual**	**Actual**	**Actual**
Baseline	660.8		Schedule hours for this week	51.0	48.1	1.06	2.9
Current	745.5		Schedule hours this cycle to date	344.0	395.0	0.87	-51.0

Improving Task Hours

At Allied Signal average task hours per developer per week were improved from 9.6 hours to 15.1 hours through quiet time, process documentation, more efficient meetings, etc.

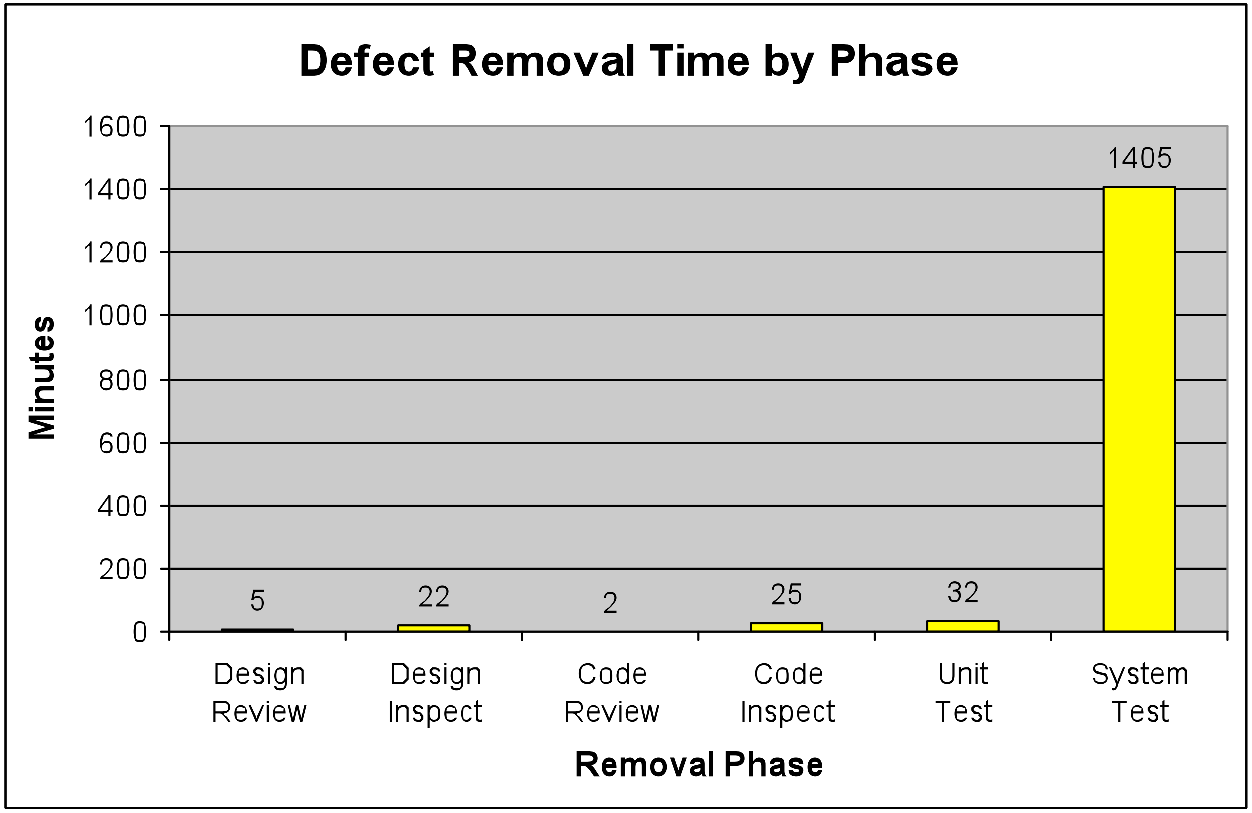
This is equivalent to a 57% increase in productivity.



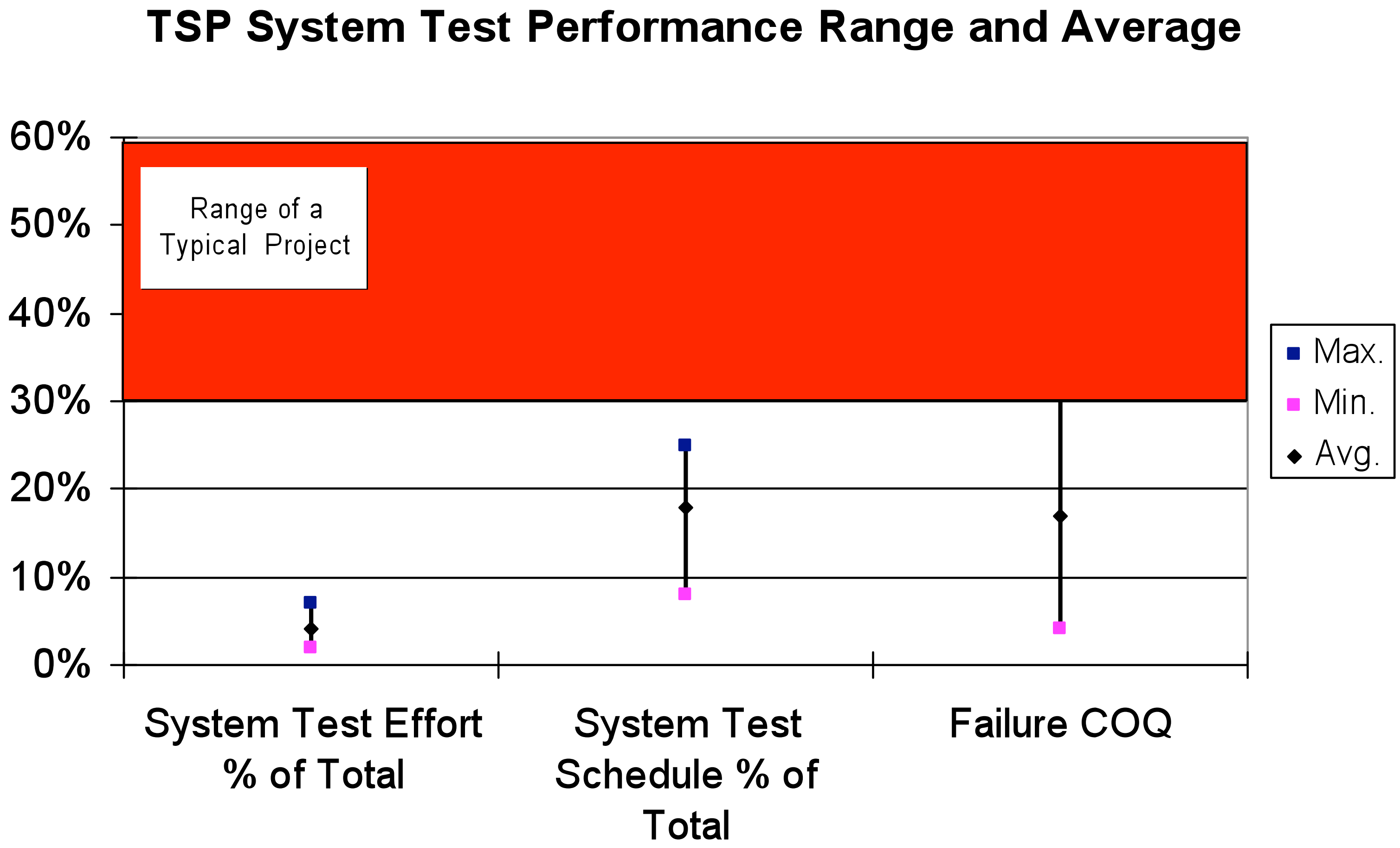
Source: Allied Signal

Reviews and Inspections Save Time

Xerox found that TSP quality management practices reduced the cost of poor quality by finding and removing defects earlier when costs are lower.



Reduced Rework

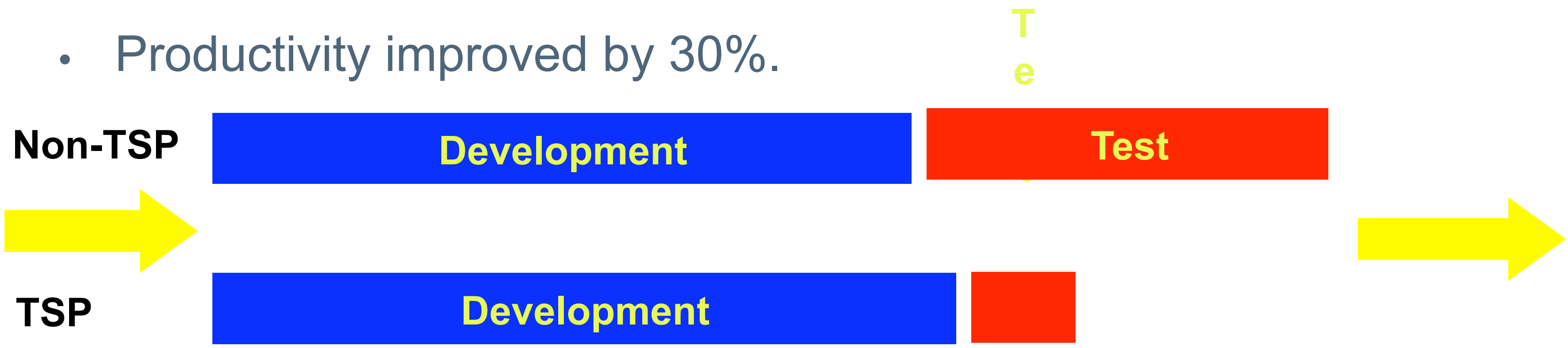


Source: CMU/SEI-TR-2003-014

Productivity Improvement

From data on over 40 TSP teams, Intuit has found that

- post code-complete effort is 8% instead of 33% of the project
- for TSP projects, standard test times were cut from 4 months to 1 month or less.
- Productivity improved by 30%.



Non-TSP

Development

T
e

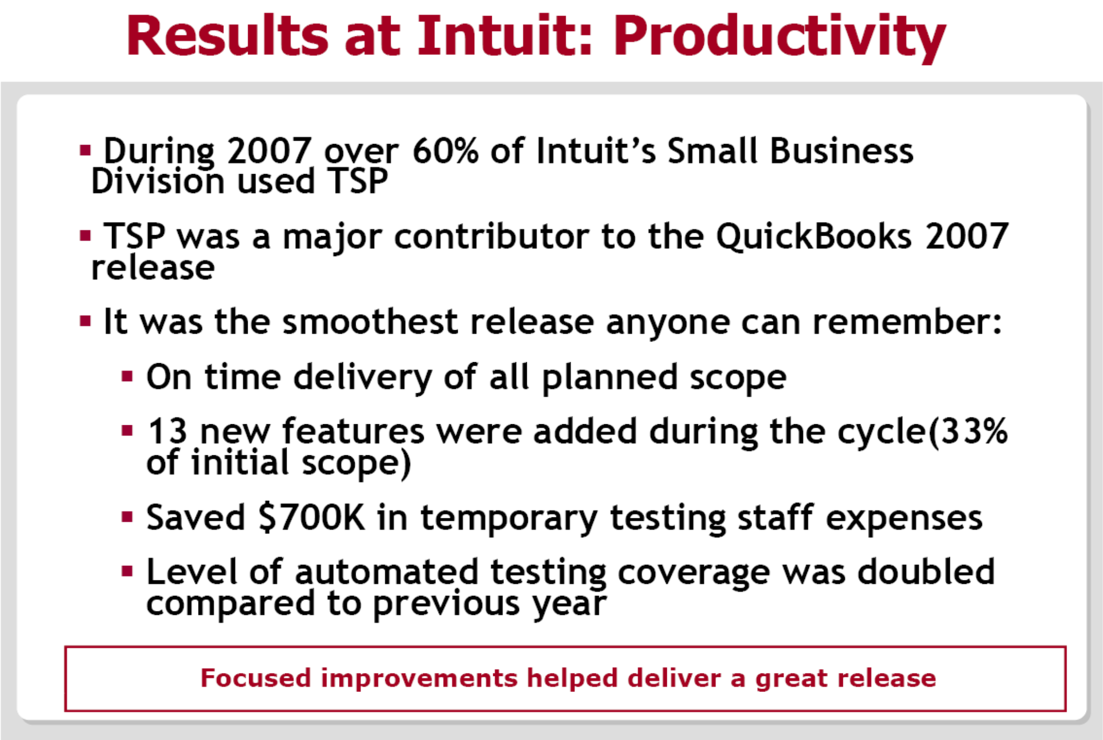
Test

TSP

Development

Intuit Productivity Improvement

By putting a quality product into system test Intuit improved productivity and reduced cost while delivering 33% more functionality than planned.



Results at Intuit: Productivity

- During 2007 over 60% of Intuit's Small Business Division used TSP
- TSP was a major contributor to the QuickBooks 2007 release
- It was the smoothest release anyone can remember:
  - On time delivery of all planned scope
  - 13 new features were added during the cycle(33% of initial scope)
  - Saved $700K in temporary testing staff expenses
  - Level of automated testing coverage was doubled compared to previous year

Focused improvements helped deliver a great release

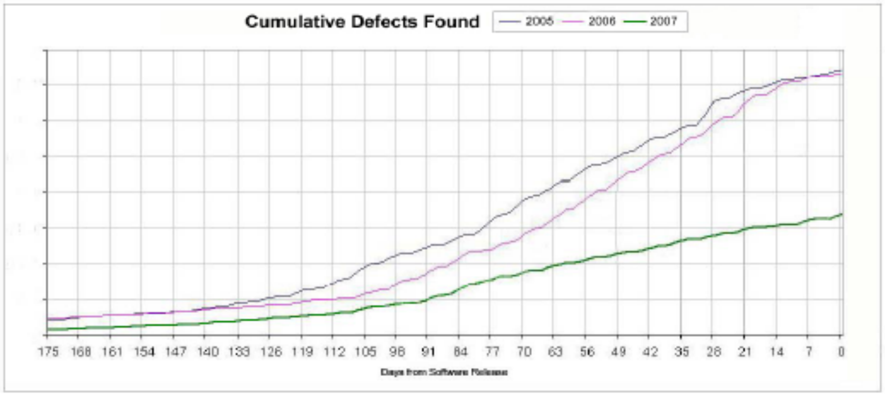
Source: Intuit

Intuit Quality Improvement

TSP reduced defects found in system test by 60% over the previous two releases of QuickBooks 2007 release.

Intuit has also recently reported a savings of $20M from a reduction in customer support calls on QuickBooks 2007.

Results at Intuit: Improved Quality



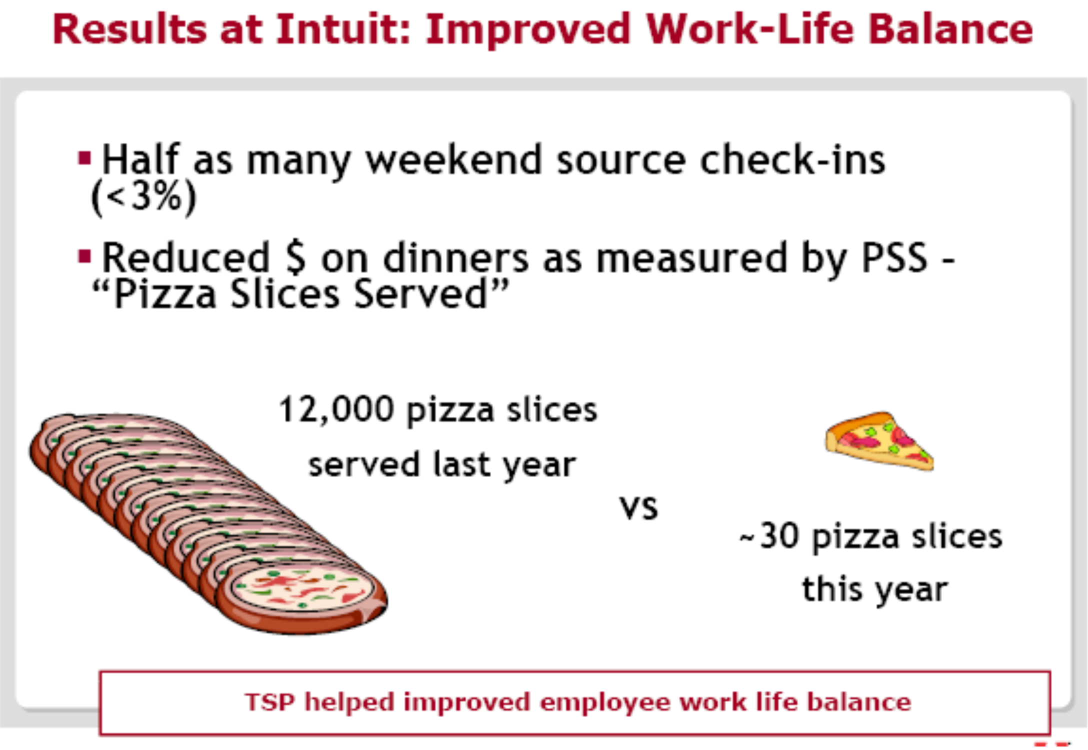
In 2007 ~60% fewer defects were found
in System Test than the previous two releases

Source: Intuit

Work-Life Balance

Finding and retaining good people is critical to long-term success.

Intuit found that TSP improved work-life balance, a key factor in job satisfaction.



Results at Intuit: Improved Work-Life Balance

- Half as many weekend source check-ins (<3%)
- Reduced $ on dinners as measured by PSS - “Pizza Slices Served”

12,000 pizza slices served last year

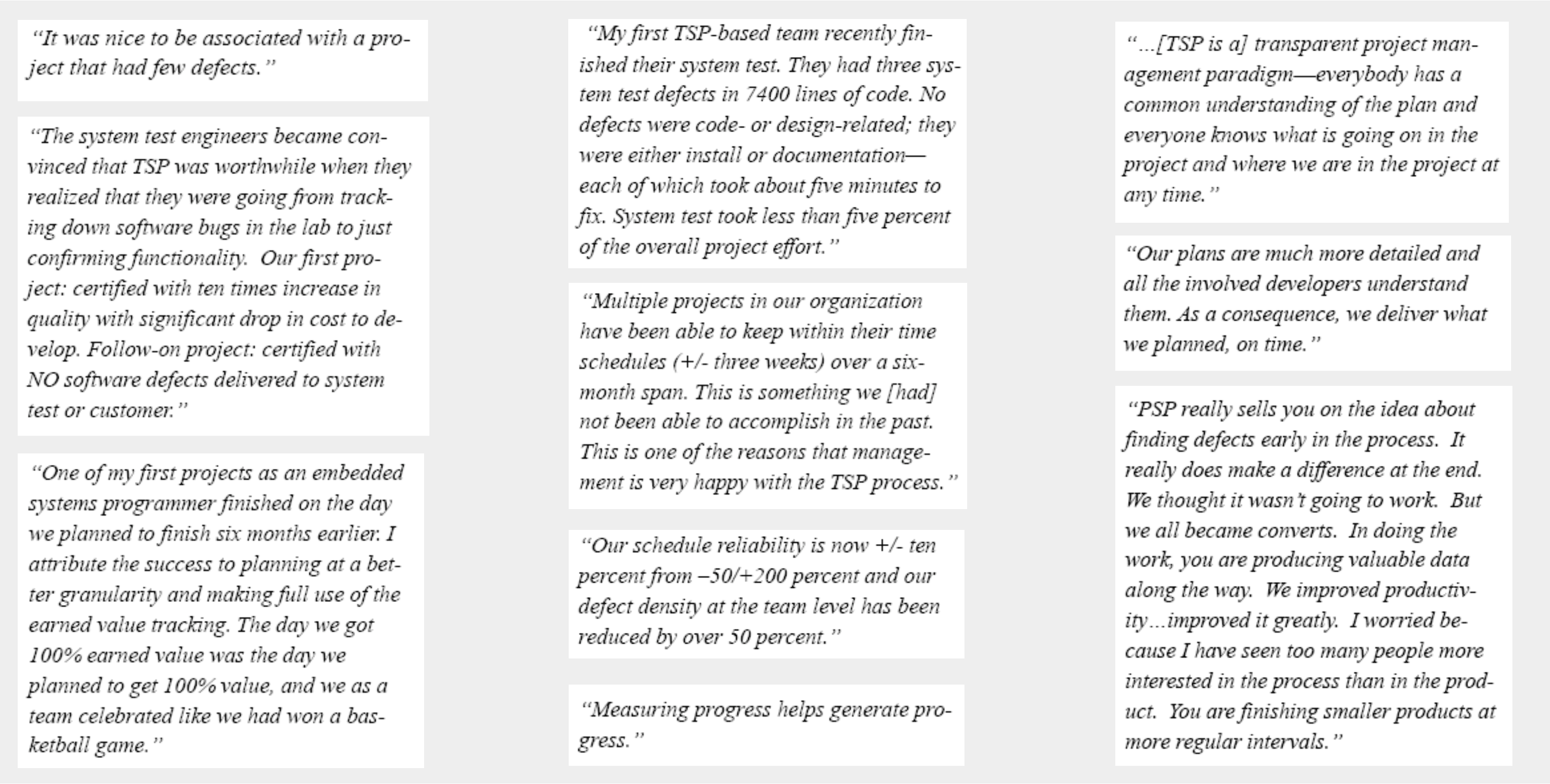
vs

~30 pizza slices this year

TSP helped improved employee work life balance

Source: Intuit

A Process for Managers and Developers



"It was nice to be associated with a project that had few defects."

"The system test engineers became convinced that TSP was worthwhile when they realized that they were going from tracking down software bugs in the lab to just confirming functionality. Our first project: certified with ten times increase in quality with significant drop in cost to develop. Follow-on project: certified with NO software defects delivered to system test or customer."

"One of my first projects as an embedded systems programmer finished on the day we planned to finish six months earlier. I attribute the success to planning at a better granularity and making full use of the earned value tracking. The day we got 100% earned value was the day we planned to get 100% value, and we as a team celebrated like we had won a basketball game."

"My first TSP-based team recently finished their system test. They had three system test defects in 7400 lines of code. No defects were code- or design-related; they were either install or documentation—each of which took about five minutes to fix. System test took less than five percent of the overall project effort."

"Multiple projects in our organization have been able to keep within their time schedules (+/- three weeks) over a six-month span. This is something we [had] not been able to accomplish in the past. This is one of the reasons that management is very happy with the TSP process."

"Our schedule reliability is now +/- ten percent from –50/+200 percent and our defect density at the team level has been reduced by over 50 percent."

"Measuring progress helps generate progress."

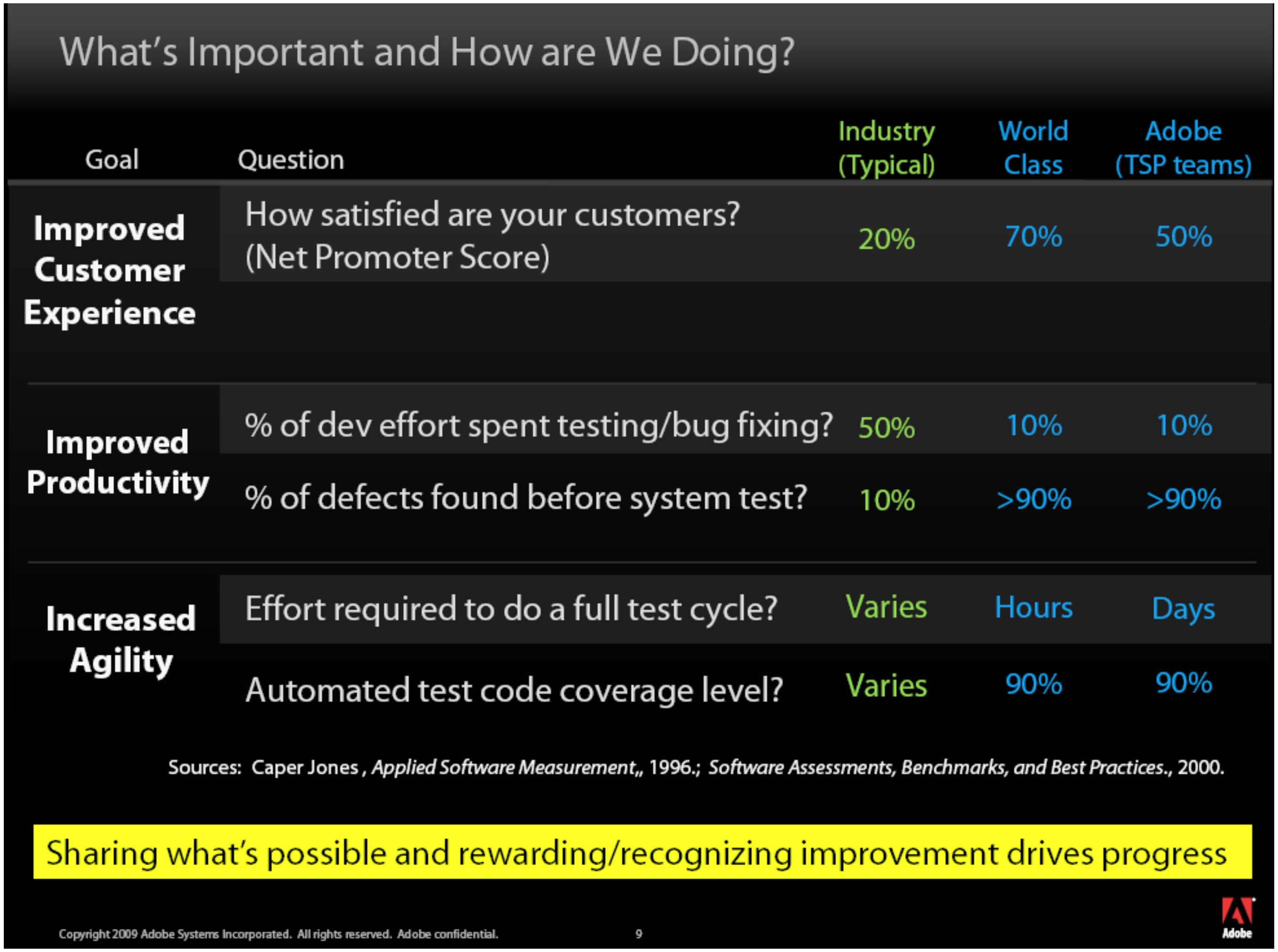
"...[TSP is a] transparent project management paradigm—everybody has a common understanding of the plan and everyone knows what is going on in the project and where we are in the project at any time."

"Our plans are much more detailed and all the involved developers understand them. As a consequence, we deliver what we planned, on time."

"PSP really sells you on the idea about finding defects early in the process. It really does make a difference at the end. We thought it wasn't going to work. But we all became converts. In doing the work, you are producing valuable data along the way. We improved productivity...improved it greatly. I worried because I have seen too many people more interested in the process than in the product. You are finishing smaller products at more regular intervals."

Source: CMU/SEI-TR-2003-014

Impact of TSP at Adobe



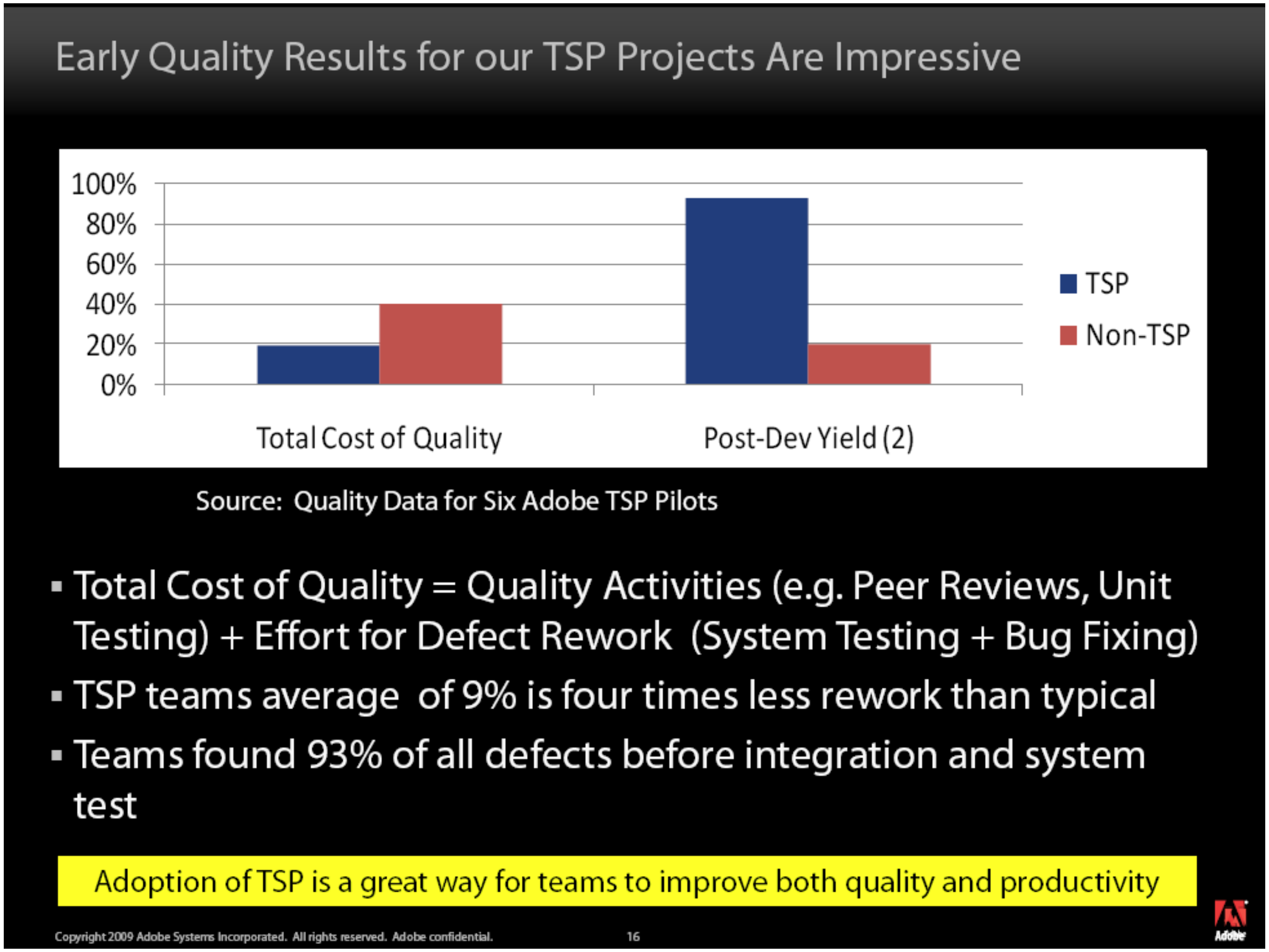
What's Important and How are We Doing?

Goal	Question	Industry (Typical)	World Class	Adobe (TSP teams)
Improved Customer Experience	How satisfied are your customers? (Net Promoter Score)	20%	70%	50%
Improved Productivity	% of dev effort spent testing/bug fixing?	50%	10%	10%
	% of defects found before system test?	10%	>90%	>90%
Increased Agility	Effort required to do a full test cycle?	Varies	Hours	Days
	Automated test code coverage level?	Varies	90%	90%

Sources: Caper Jones , *Applied Software Measurement,*, 1996.; *Software Assessments, Benchmarks, and Best Practices.*, 2000.

Sharing what's possible and rewarding/recognizing improvement drives progress

TSP Quality Improvements at Adobe



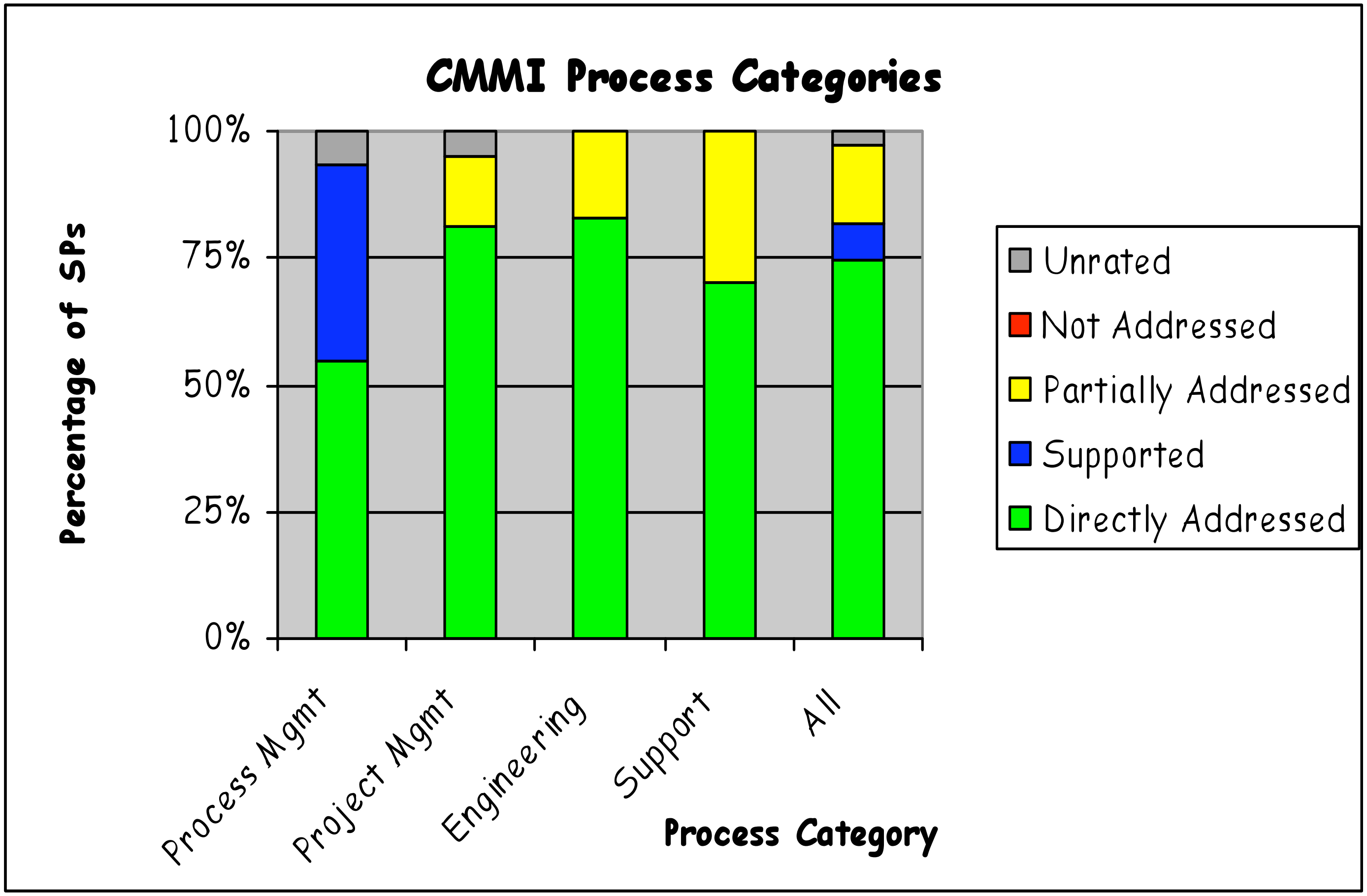
Early Quality Results for our TSP Projects Are Impressive

Source: Quality Data for Six Adobe TSP Pilots

- Total Cost of Quality = Quality Activities (e.g. Peer Reviews, Unit Testing) + Effort for Defect Rework (System Testing + Bug Fixing)
- TSP teams average of 9% is four times less rework than typical
- Teams found 93% of all defects before integration and system test

Adoption of TSP is a great way for teams to improve both quality and productivity

TSP Implements CMMI -1



Unrated - out of scope for TSP.

Not addressed - project practice that TSP does not cover.

Partially addressed - project practices that TSP addresses with some weakness of omission

Supported - organizational practices that TSP supports.

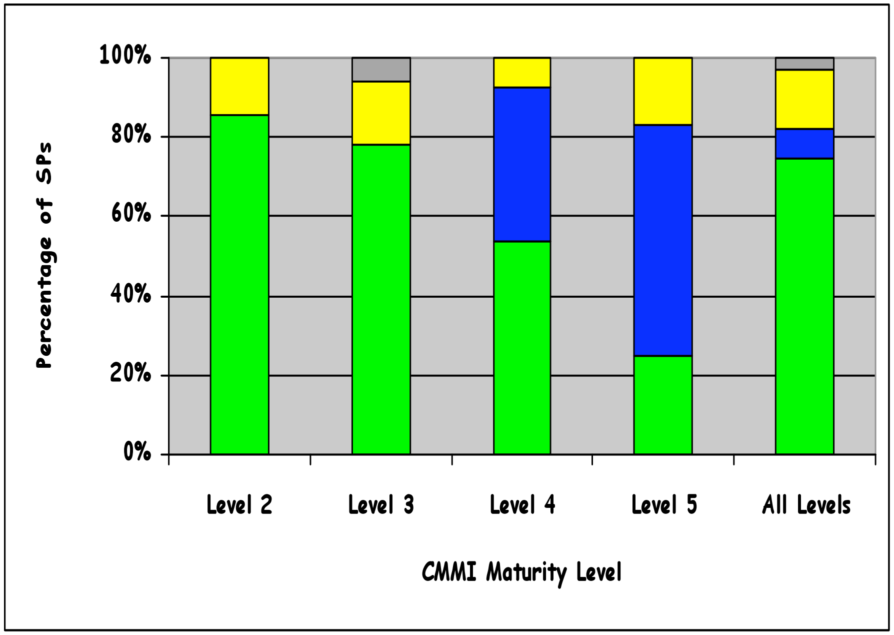
Directly Addressed - TSP practices meet the intent of the CMMI specific practice (SP) without significant reservations.

Based on a SCAMPI C of the latest version of TSP

TSP Implements CMMI -2

An organization using TSP has directly addressed or implemented most specific practices (SP).

- 85% of SPs at ML2
- 78% of SPs at ML3
- 54% of SPs at ML4
- 25% of SPs at ML5
- 80% of ML2 and ML3 SPs
- 75% of SPs through ML5



Most generic practices are also addressed.

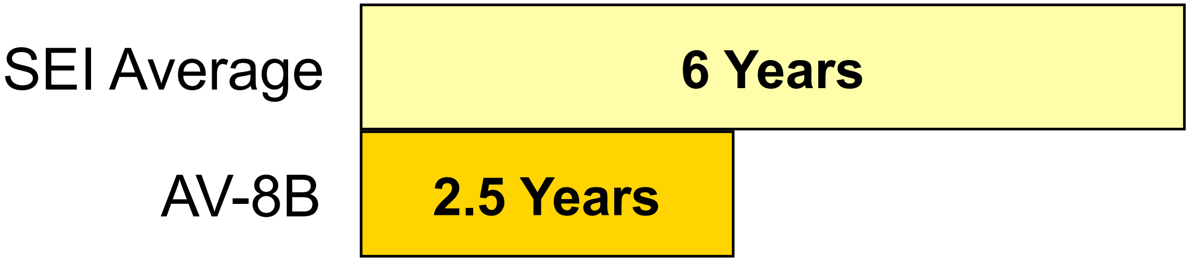
Based on a SCAMPI C of the latest version of TSP

NAVAIR AV-8B TSP/CMMI Experience

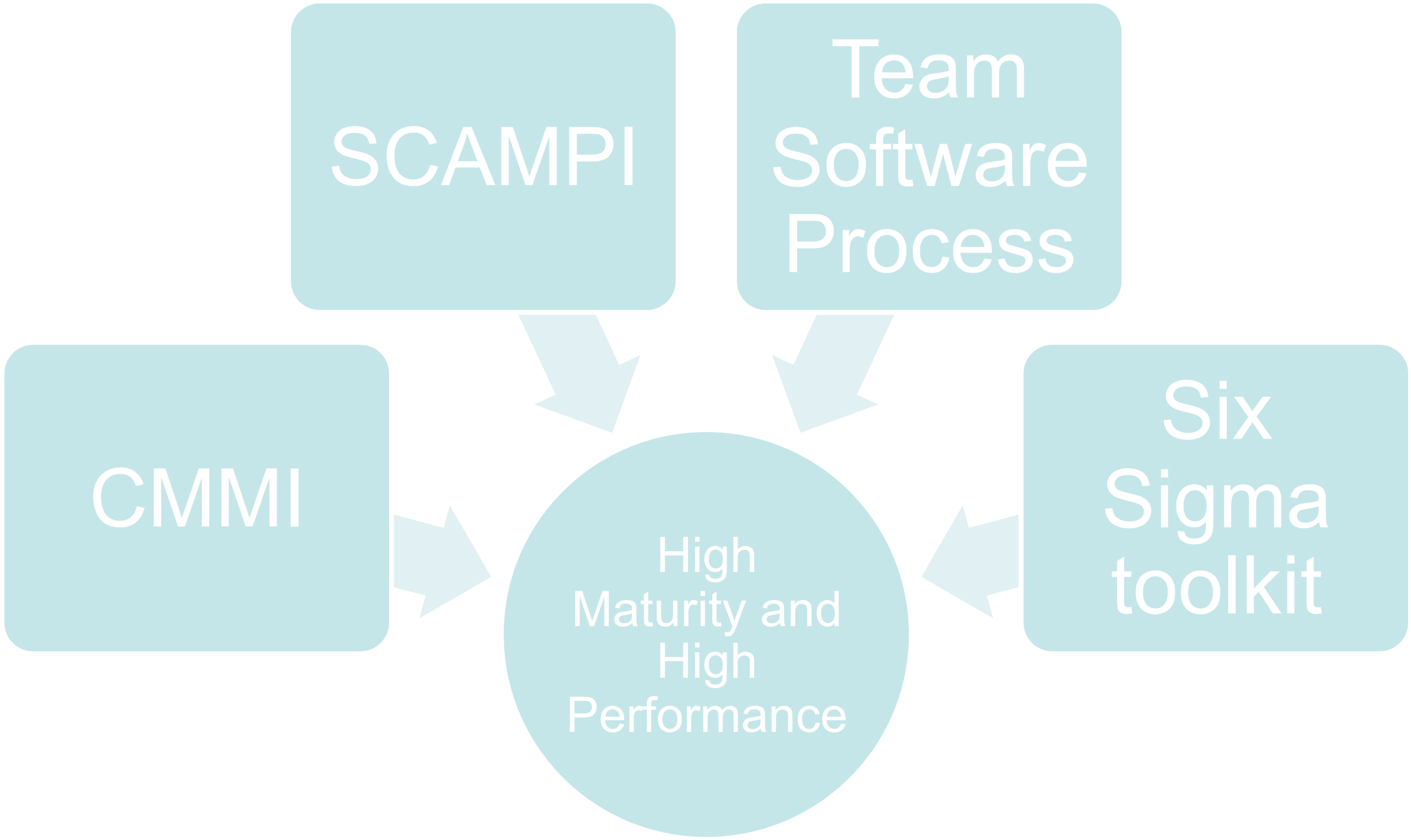
AV-8B is a NAVAIR System Support Activity.

They integrate new features into the Marine Harrier aircraft.

They used TSP to reduce the time to go from CMMI Level 1 to CMMI Level 4.



Fast Track to High Maturity and High Performance



TSP Implementation Strategy

TSP is implemented on a project-by-project or team-by-team basis

Start with two or three teams.

- train the team members and their managers
- launch these teams with TSP
- evaluate and fine tune the approach

From the time of starting the first training session, a team can be launched and up and running within 1 month.

This cycle is then repeated, increasing scope at a sustainable pace.

Selecting Pilot Projects

Pick 2 to 3 pilot projects.

- 3 to 15 team members
- 4 to 18 month schedule
- software-intensive new development or enhancement
- representative of the organization’s work
- important projects

Select teams with members and managers who are willing to participate.

Consider the group relationships.

- contractors
- organizational boundaries
- internal conflicts

Deployment Timeline

Task	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
TSP Executive Strategy Seminar	♦											
Leading Development Teams	♦											
PSP Fundamentals		♦										
Launch Initial Teams		♦										
Cycle Postmortem for Initial Teams				♦								
Re-launch Initial Teams				♦								
Train instructors and coaches		▪▪━	━▶									
Project Postmortem for Initial Teams						♦						
Train and launch remaining projects and teams at a sustainable pace.				■ ■	■━	━	━	━	━	━	━	━▶

The training schedule can be compressed to as short as one week for a faster start.

The gating factor for most organizations is the availability of projects.

SEI recommends training internal coaches as soon as possible.

Build Internal Capability

Organizations should develop internal capability to support TSP.

- SEI-certified TSP coaches are essential
- SEI-authorized trainers are optional as training can be outsourced

The initial pilot projects provide the “hands-on” experience with TSP.

Training and authorization requirements

- Coach – one week training course, exam, and a launch observation
- Instructor – one week training course and an exam

SEI does offer a support program where SEI leads the effort initially and internal staff observe, then internal staff lead and SEI mentors.

Training for Participants

Participant	CBT Option	Course	Notes
Executives and senior management	No	*TSP Executive Strategy Seminar*	1 day + optional ½ day strategic planning session.
Middle and first-line managers	No	*Leading Development Teams*	3 days
Software developers	Yes	*PSP Fundamentals* *PSP Advanced*	5 days 5 days (optional)
Team members other than software developers		*TSP Team Member Training*	2.5 days (will replace *Introduction to Personal Process in 2009)*
Instructors	No	*PSP Instructor Training*	5 days Pre-requisite training: *PSP Fundamentals and PSP Advanced or PSP I and PSP II*
Coaches	No	*TSP Coach Training*	5 days Pre-requisite training: *PSP Fundamentals and PSP Advanced or PSP I and PSP II*

Summary

TSP is an agile, high-performance, high maturity process for development teams.

It addresses the cost, schedule, quality, and predictability issues that most organizations face

TSP can be introduced quickly with near-term ROI.

TSP complements CMMI and has compelling results.

Questions?

Software Engineering Institute | Carnegie Mellon