Team Software Process

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

March, 2007

Software Engineering Institute | Carnegie Mellon

Topics

TSP Snapshot

Industry Performance

TSP Performance

TSP Concepts

TSP-I

Getting Started

Team Software Process Solution

The Team Software Process (TSP) is a solution to common engineering management issues.

- Project management
- Quality management
- Risk management
- Process improvement
- Communication
- Team performance improvement
- Quality of Work Life
- Performance measurement
- Standardization
- Crisis management

Team Software Process is Unique

Unlike other engineering methods, TSP

- Changes the behavior of individuals, teams, and the organization.
- Improves performance, with quantifiable benefits, on first use.
- Is a disciplined, yet agile approach to engineering.
- Has been applied to a broad range of application domains .
- Is scalable from very small to large projects and teams.
- Accelerates CMMI-based improvement, addressing most practices.

Team Software Process Product Suite

The Team Software Process Product Suite supports successful transition to practice.

- Management and engineering processes
- Measurement framework
- Deployment and support
  - Introduction strategy and plan
  - Training for management, engineering, coach, train-the-trainer
  - Tools
  - Certification
  - Community

Topics

TSP Snapshot

Industry Performance

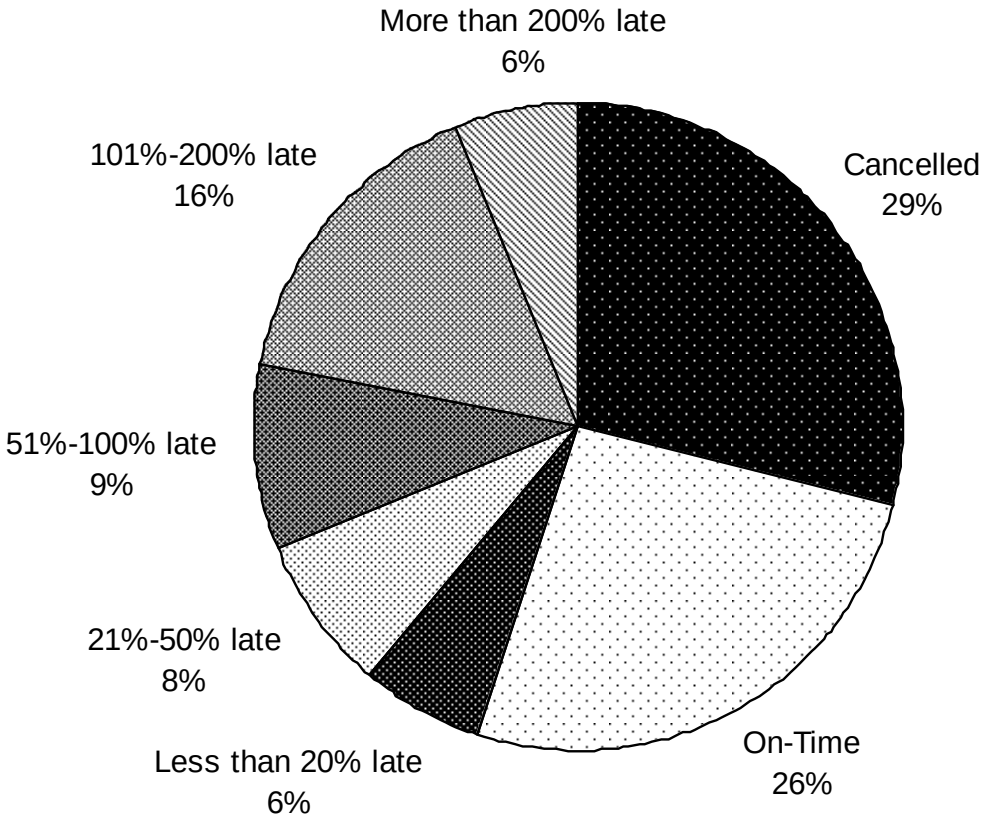
TSP Performance

TSP Concepts

TSP-I

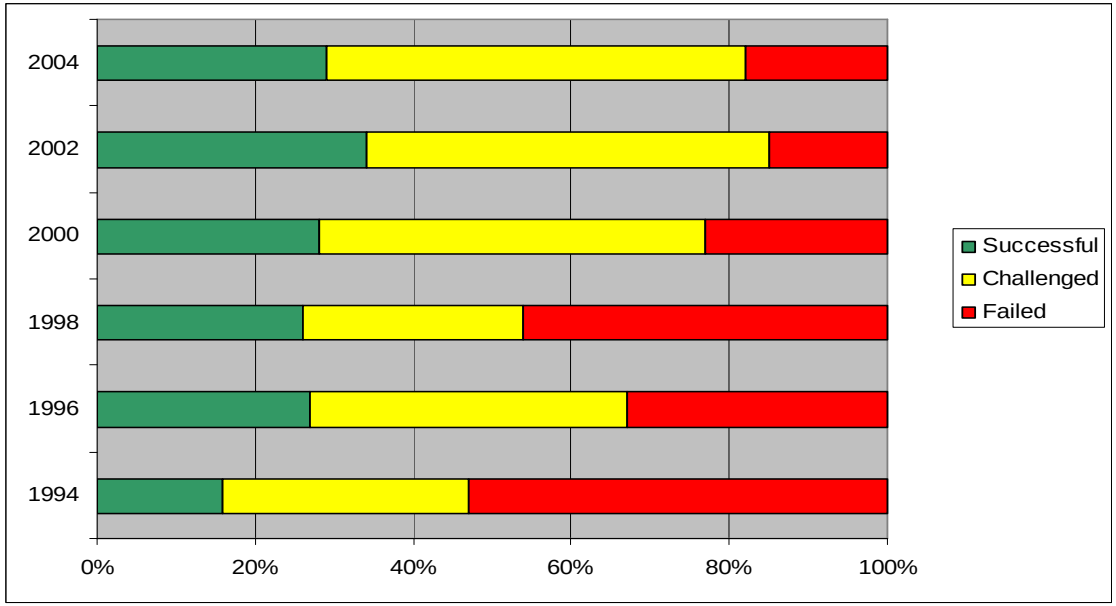
Getting Started

Software Industry Schedule Performance



* From the Standish Group, 2000

Cost and Schedule Performance Trends



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

Challenged projects estimated a 43% average cost overrun, time overruns of 83%, and delivered only 52% of required features and functions (in 2002).

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

Software Quality Issues

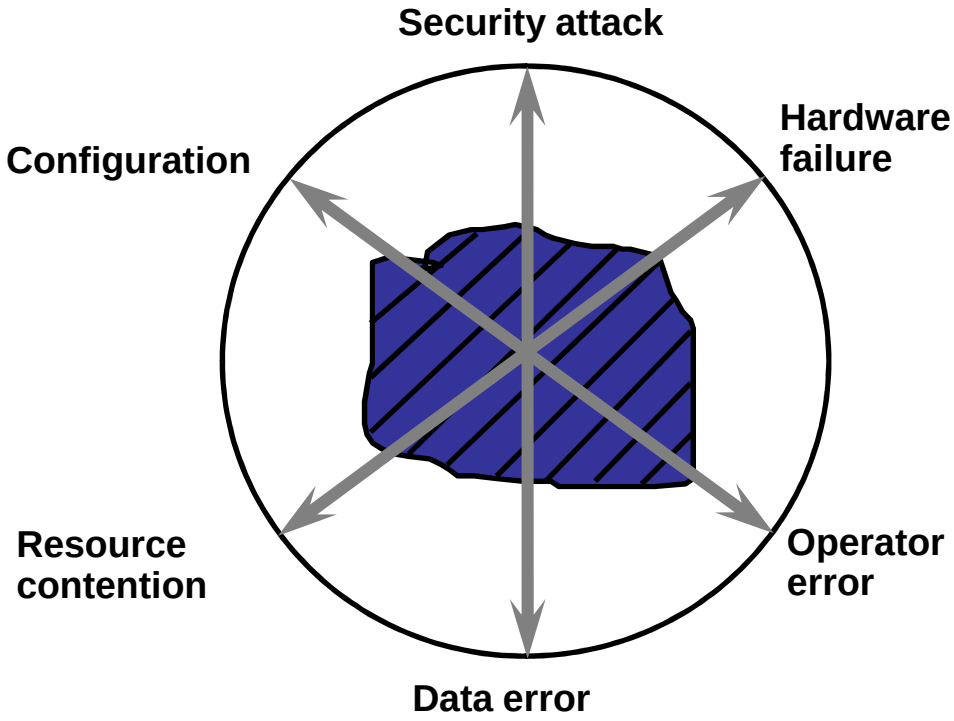
Software is the only modern technology that ignores quality until test.

Most software defects are found in or after test when defect removal is the most expensive and least effective.

This strategy results in buggy products and unnecessary rework, inflating development costs.

Software defects are also a principal cause of software security vulnerabilities.

The Problem with Testing



Safe region = tested (shaded)
Unsafe region = untested (unshaded)

Software Industry Quality Performance

Experienced engineers inject 1 defect per 7 to 12 lines of code during design and coding, and remove only about 80% of these defects before entering test.

Testing is about half of the cost of building software and at least half of the testing costs are avoidable rework.

Commercially available software that is widely used has from 1 to 5 or more defects per thousand instructions.

From 70% to 90% of reported vulnerabilities are caused by common design and coding errors.

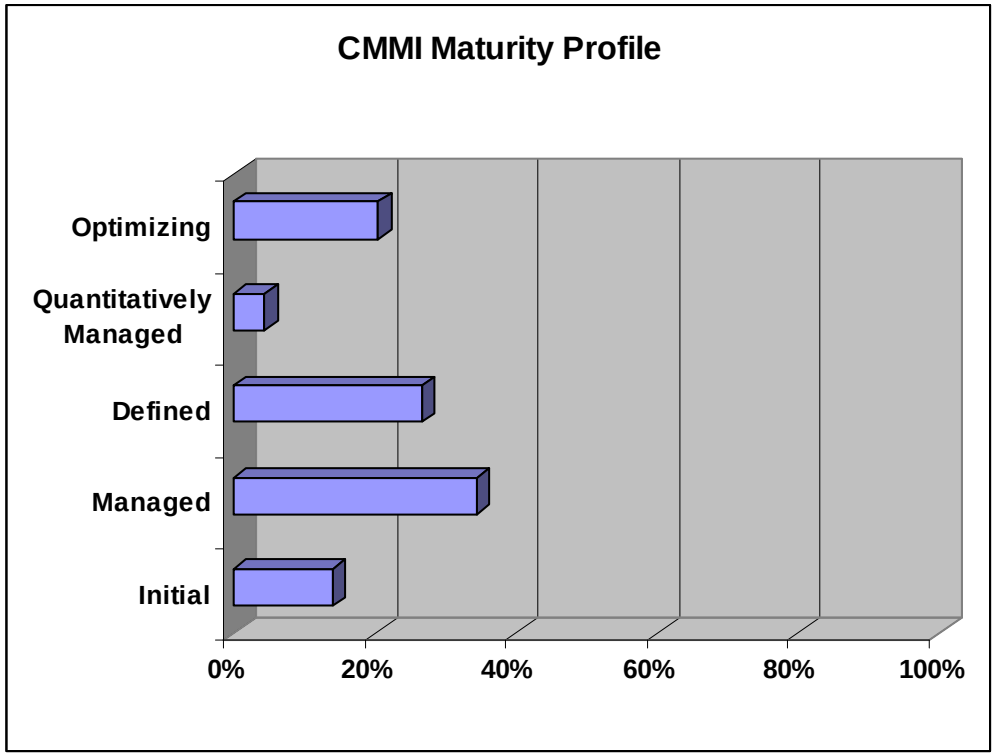
CMMI State of the Practice

The state of the practice is improving, but…

About 75% of reporting organizations are still not using quantitative quality management practices.

About 50% do not follow a defined process or use inspections.

SW-CMM results are similar.



The People Dimension

Software development is a creative, intellectual discipline.

The performance of a development organization is determined by the performance of its teams.

The performance of a development team is determined by the performance of the individual developers.

Few organizations know how to capitalize on team performance or their people's potential.

Topics

TSP Snapshot

Industry Performance

TSP Performance

TSP Concepts

TSP-I

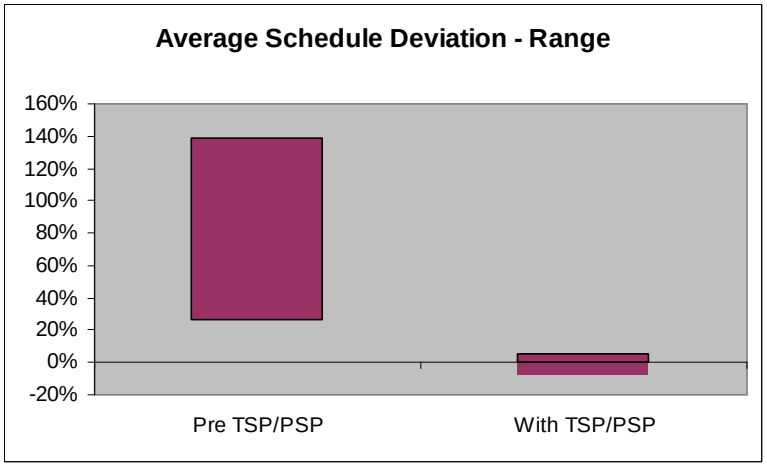
Getting Started

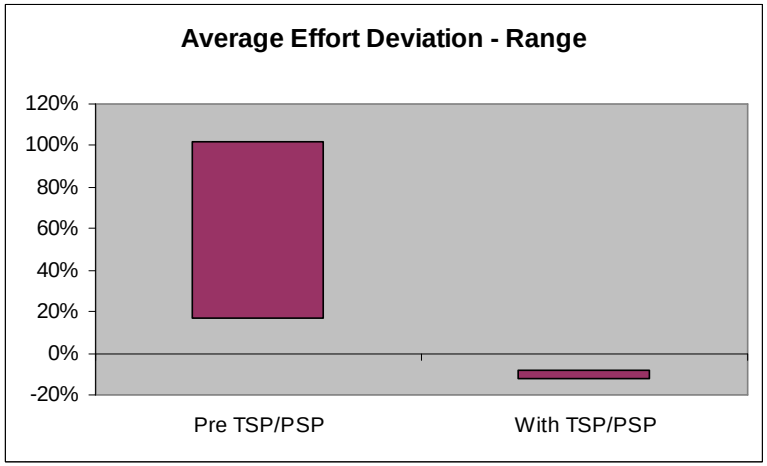
TSP Improves Predictability

Effort and schedule deviation are dramatically improved.

Schedule Performance	
Typical Industry	100%+ overrun
Study baseline	27% to 112%
TSP	< 10%

Effort/Cost Performance	
Typical Industry	100%+ overrun
Study baseline	17% to 85%
TSP	< 5%





TSP Improves Product and Process Quality -1

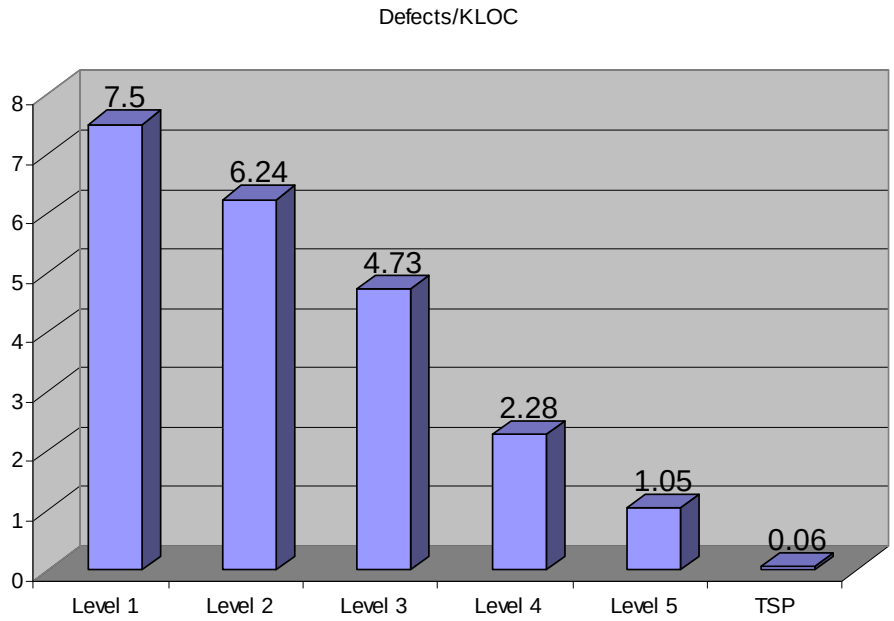
Performance Category	TSP Impact Study (2003)*	Typical Industry Performance
System test defects per thousand instructions	0.4 avg. 0.0 to 0.9	2 to 14
Released defects per thousand instructions	0.06 avg. 0.0 to 0.2	1 to 7
System test effort (% of total effort)	4% avg. 2% to 7%	40%

TSP Improves Product and Process Quality

An analysis of 20 projects in 13 organizations showed 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 Improves Quality of Work Life

"A more disciplined process allowed me to do a better job, and allowed me to balance my job with other aspects of my life."

"This project ended up a lot less stressful than other projects."

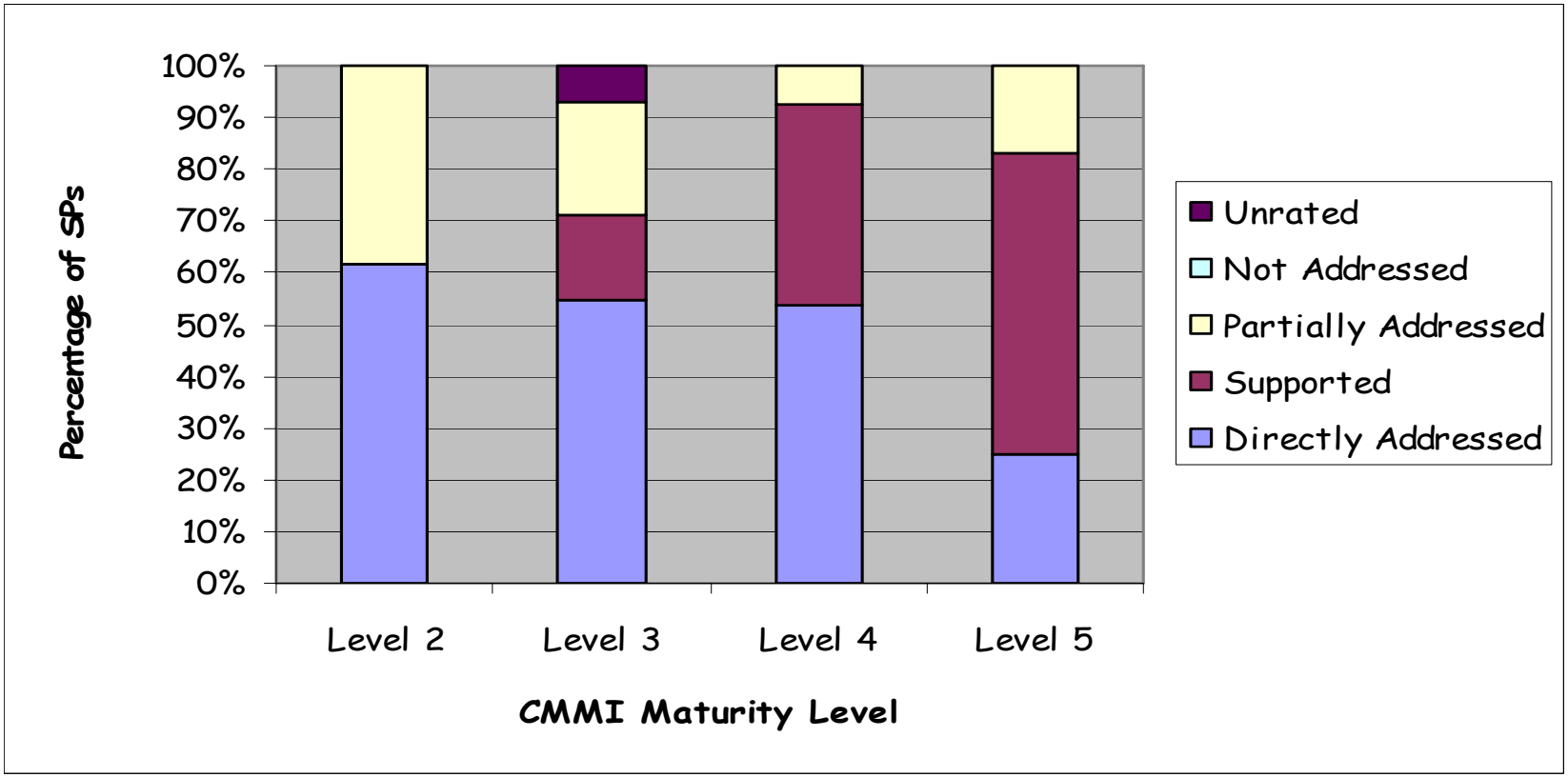
"Promotes a less stressful environment. Can track that the project is on schedule. Fewer defects are seen positively in the organization."

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

"I liked the level of detail that went into initial plan, and the constant awareness of the schedule. Allowed us to make adjustments as the project went on, instead of waiting for a major milestone."

"It was nice that management finally allowed the team to create the schedule."

TSP Overall Coverage of CMMI

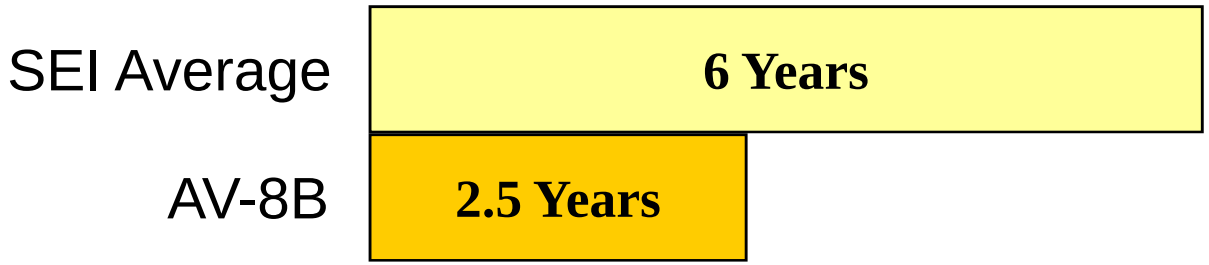


NAVAIR AV-8B TSP/CMMI Experience

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.



NAVAIR Benefits from TSP

Program	Size of Program	Defect Density (Defects/KSLOC))	Cost Savings from Reduced Defects
AV JMPS	443 KSLOC	0.59	$2,177,169
P-3C	383 KSLOC	0.6	$1,478,243

Level 5 and Continuous Improvement

	Project A (TSP)	Project B
Size (KLOC)	82	151
Duration (months)	31.8	43.0
Normalized (per KLOC)	$0.95K	$4.05K
Peer Review Exit Density	4.78	17.3
Delivered Defect Density	1.55	5.27
Integration / Acceptance Test Cost	$78.K	$612K
Time to Accept (months)	3.7	14.6

Source: Northrop Grumman (a CMMI Level 5 organization)

http://www.dtic.mil/ndia/2003CMMI/kent.ppt

Adoption

Organizations using, piloting, or preparing to pilot the TSP.

ABB
Accenture
Advanced Information Services
Advanced Maturity Services, Inc.
Alan S. Koch Consultants
Alliant
AMRDEC
Ascent
Bechtel-Bettis*
Boeing
Centre De Investigacion En Matamaticas
Census Bureau
Cluster File Systems
Computing Technology, Inc.
CQG, Inc.
CRSIP / STSC / DRAPER
Davis Systems
DEK
Delivery Excellence
DOE / Los Alamos
DOE / Naval Reactors
DPC Cirrus
Dynamics Research Corp.
Evince Media
Halex Associates
Heath Solutions, Inc.
Helsana
IBM Japan
IBM Mexico*
Idea Entity Corporation
Institute for Information Industry
Intuit*
Kaitatsu, Ltd.
KPMG
LogiCare
Los Alamos National Laboratory
M/A-Com Private Radio Systems, Inc.
Magellen*
Microsoft*
Misys
Motiva
MP SPI Solutions
NASA Langley
NAVAIR*
Naval Reactors*
NAVOCEANO*
NCS Pearson
Northern Horizons
Oakwood College
Oracle*
Prodigia S.A. de C.V.
PS&J Consulting - Software Six Sigma
QuarkSoft
Sage Software
SAIC
Samsung Electronics
Samsung SDS
Sandia National Laboratories
Satyam Computing Services
Scientific Atlanta
Softek*
SECC
STPP, Inc.
STSC
Tec de Monterrey
Trend Micro
TYBRIN Corporation - Air Logistics
Unisinos
University of Alabama / Huntsville
University of Queensland
Vicarious Visions

***Organizations SEI is currently working with**

Topics

TSP Snapshot

Industry Performance

TSP Performance

TSP Concepts

TSP-I

Getting Started

TSP Principles

Software is developed by individuals and teams. To improve overall performance, improve individual and team performance.

Empowered, team coaching, and rational management are the most effective way to manage engineering teams.

Process is a performance determinant…an ad-hoc chaotic process cannot produce predictable, high-quality results.

Measurement is necessary for high-performance.

To get a quality product out of test, you must put a quality product into test.

Quality must be measured and managed at every step.

A Software Product is a Team Effort

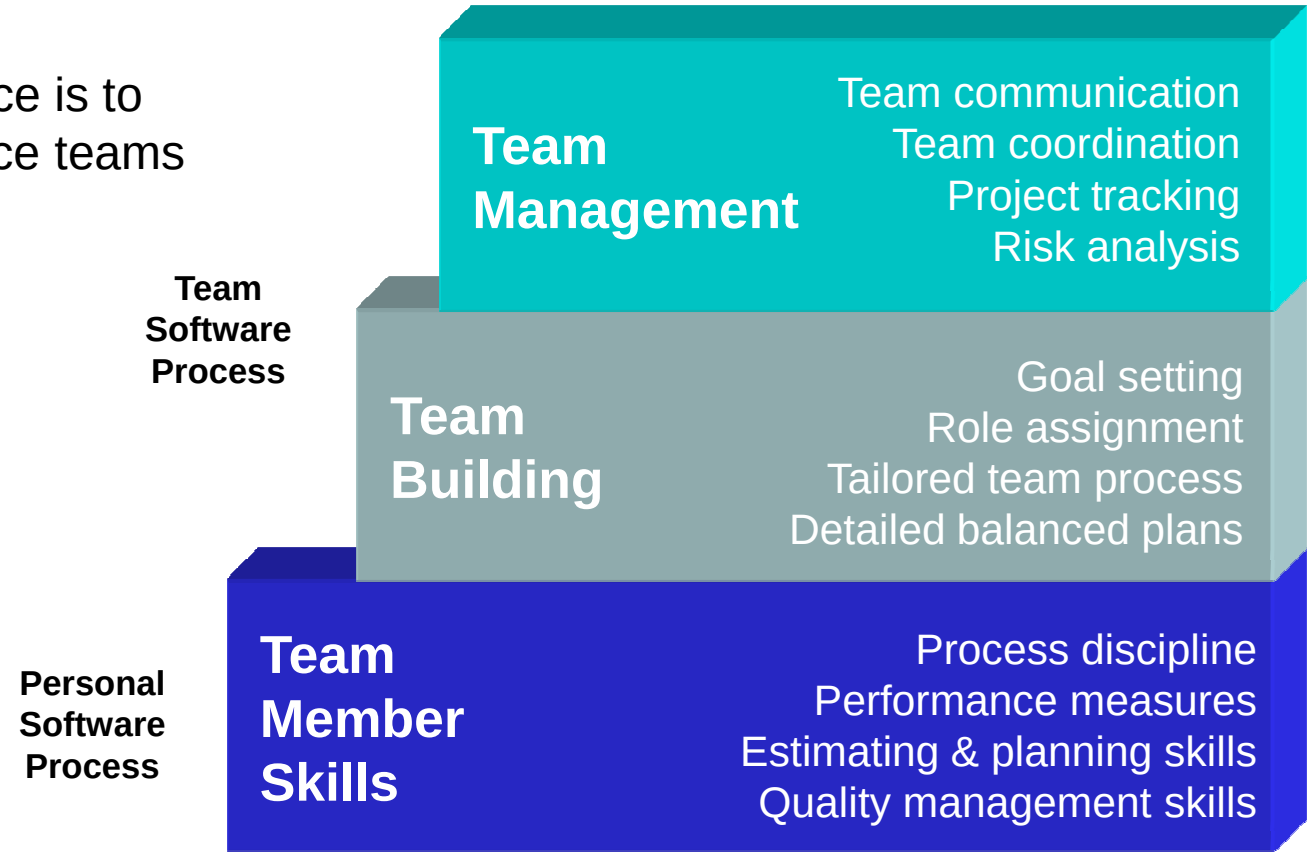
Software products are made from software modules or components.

These modules are designed, built, integrated, and tested by a team of software developers.

The team's skills, spirit, discipline, and commitment govern the results.

Building High-Performance Teams

The TSP strategy for improving performance is to build high-performance teams from the bottom-up.



Personal Software Process

The PSP is a process designed for individual use that applies to structured personal tasks.

With PSP, developers use defined and measured personal processes.

They gather size, time, and defect data as they work.

They use the data to

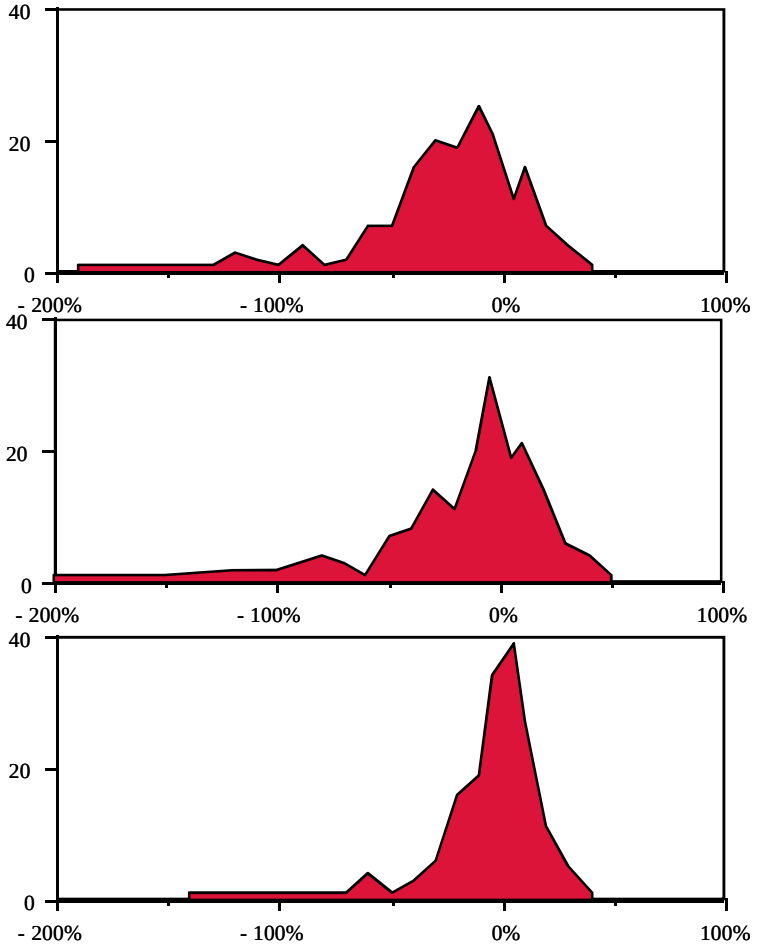
- plan and track their work
- manage the quality of the products they produce
- measurably improve performance

PSP Estimating Accuracy

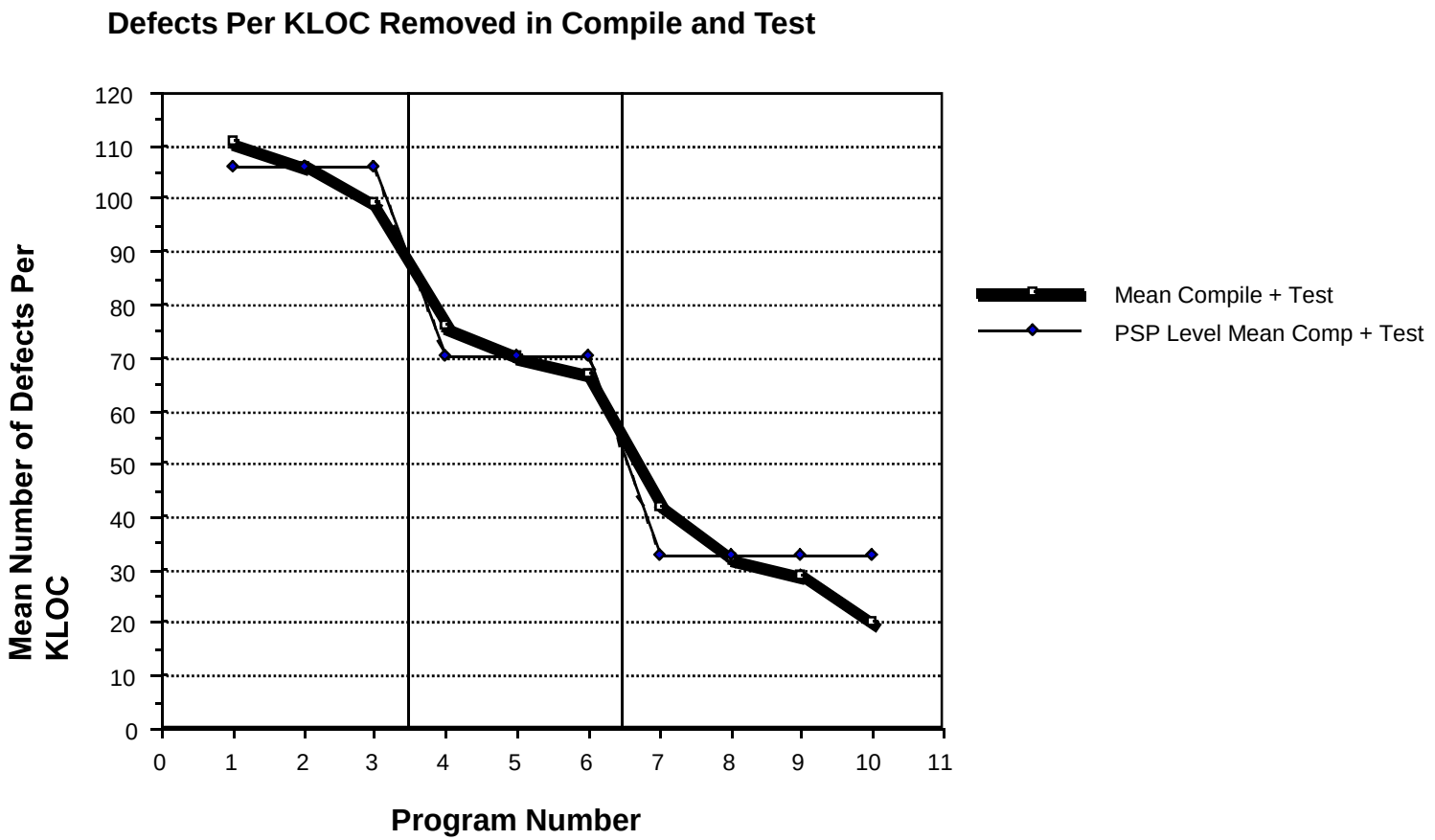
For the first 1/3 of the PSP training, the majority of students are under-estimating.

By the midway point students begin to achieve a balance of over- and underestimates.

By the end of the PSP training students achieve a much tighter balance around zero.



PSP Quality Results



PSP Conclusion

PSP training changes behavior.

Engineers convince themselves that process discipline, measurement, estimating & planning, and quality management will improve their performance.

They are then prepared and ready to apply these skills to their work.

Successful Teams

When development teams work well, they generally have successful projects.

To be successful, teams need

- clear goals
- established roles
- defined processes
- agreed-upon plans

Teams also must be motivated and personally committed.

Without these, teams will rarely be successful.

Self-directed Teams -1

Self-directed teams are in control of their own work.

Because they are in control, they are motivated to do a superior job.

The TSP builds self-directed teams.

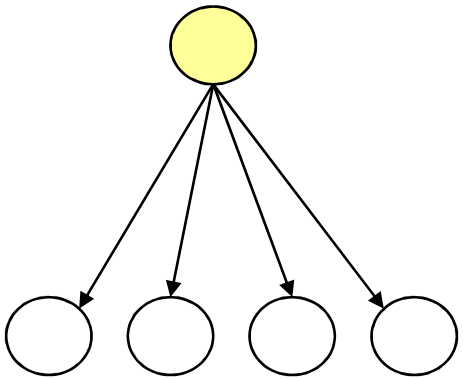
These teams

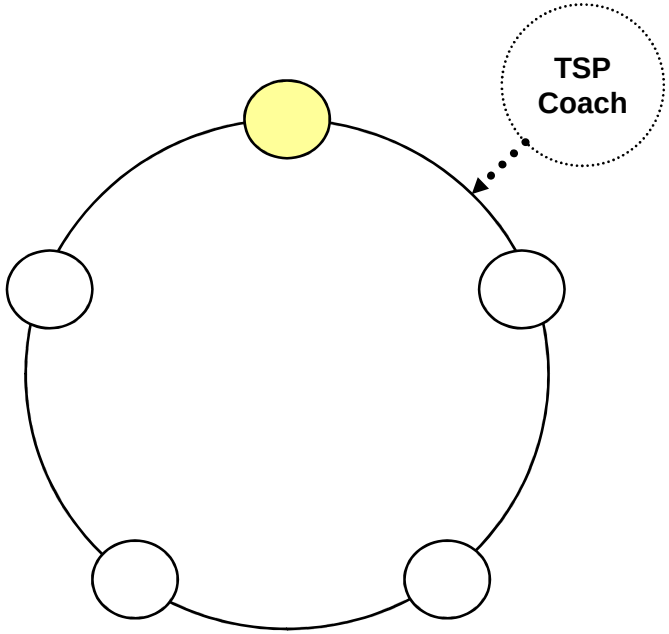
- have aggressive but realistic goals
- work to defined and measured plans and processes

The team members

- are skilled and trained for the job
- know their roles and responsibilities
- are personally committed to the work

Self-directed Teams -2





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.

Self-directed Team Roles

TSP implements self-management responsibilities through eight pre-defined management roles.

TSP Team Member Roles	
Planning Manager	Customer Interface Manager
Process Manager	Design Manager
Quality Manager	Implementation Manager
Support Manager	Test Manager

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

The team leader acts as team coach.

The TSP Coach also coaches and supports the team.

Team Building and Planning

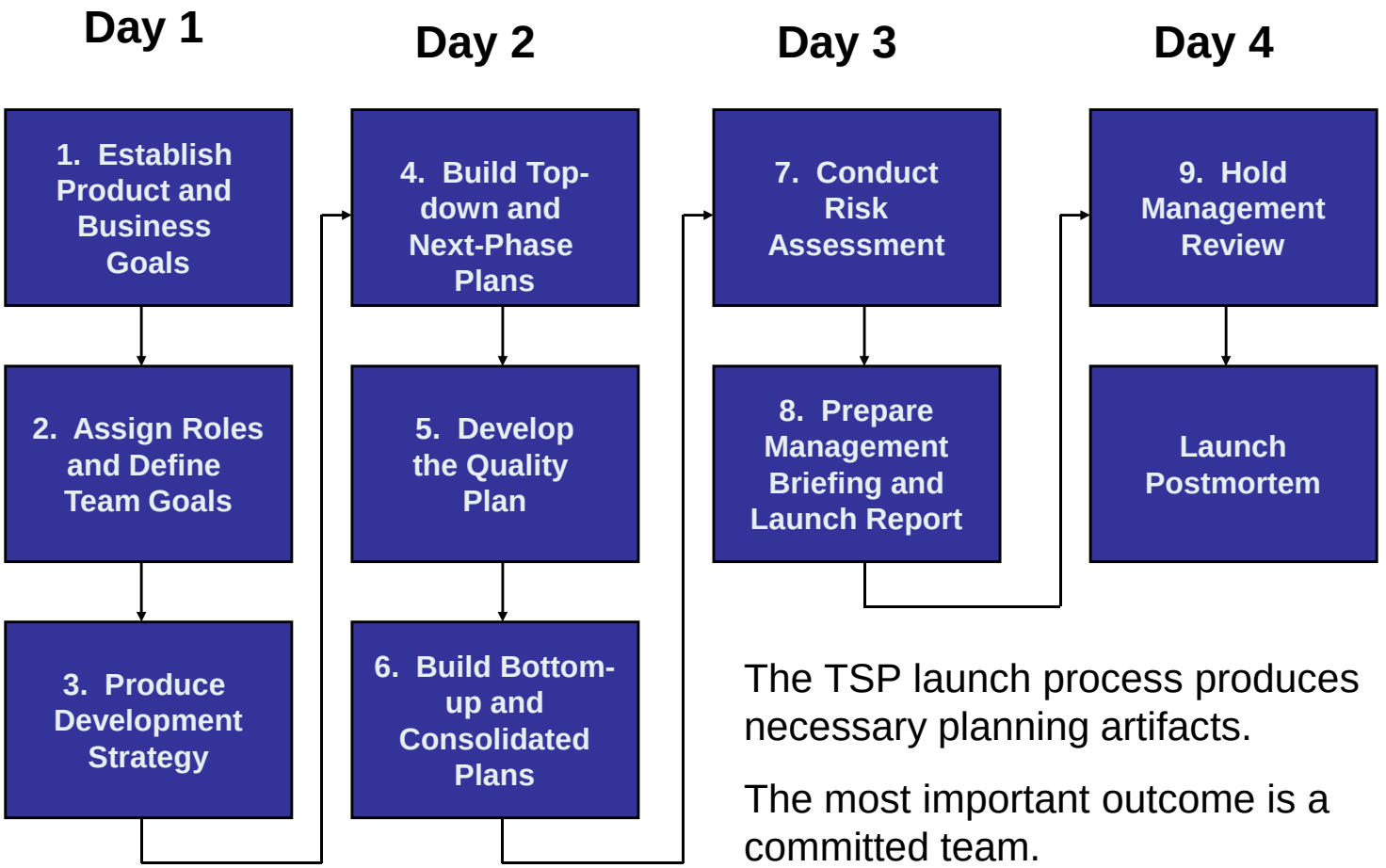
A “gelled” team is one with performance greater than the sum of its parts.

Being a member of a “gelled” team is motivating and fun, very satisfying but rare.

These teams start with the same issues that plague other teams, but along the way they develop a shared sense of success and how they will achieve it.

The TSP launch process builds “gelled” teams while planning the project’s work.

The TSP Launch Process



The TSP launch process produces necessary planning artifacts.

The most important outcome is a committed team.

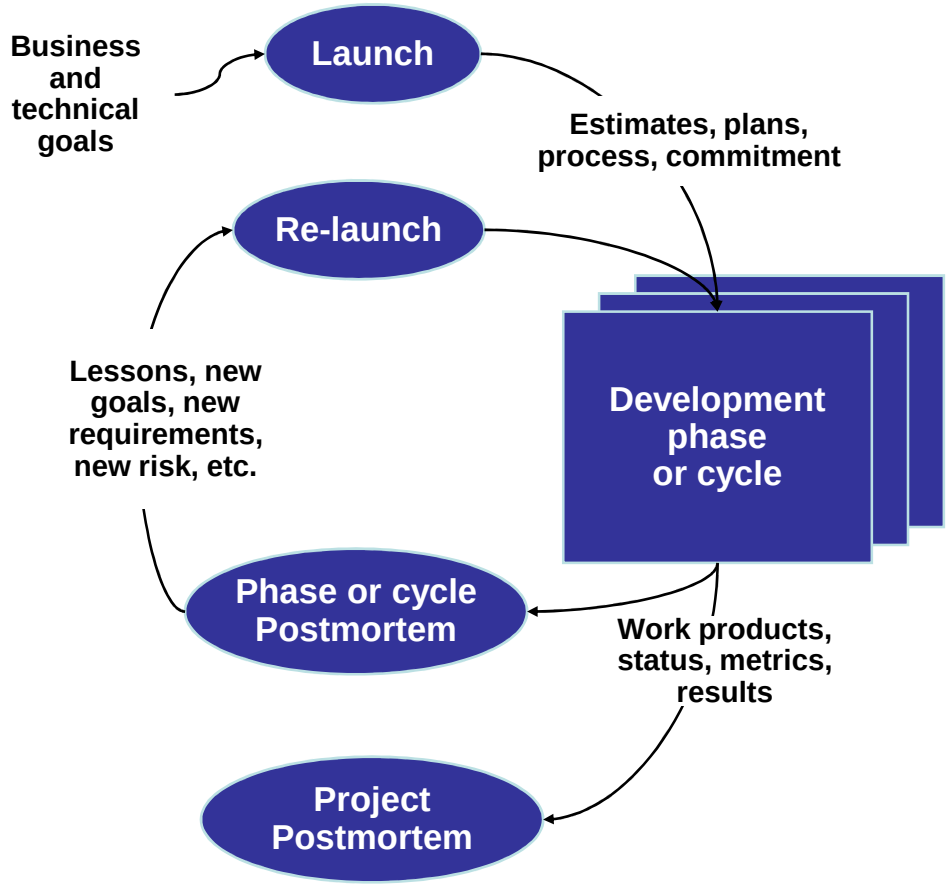
TSP Process Structure

The TSP process elements can be organized into whatever process structure makes the most business and technical sense.

The phases can be implemented iteratively in small cycles, in a spiral with increasing cycle content, or sequentially as in a waterfall,

TSP projects can start on any phase or any cycle.

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



The TSP Process

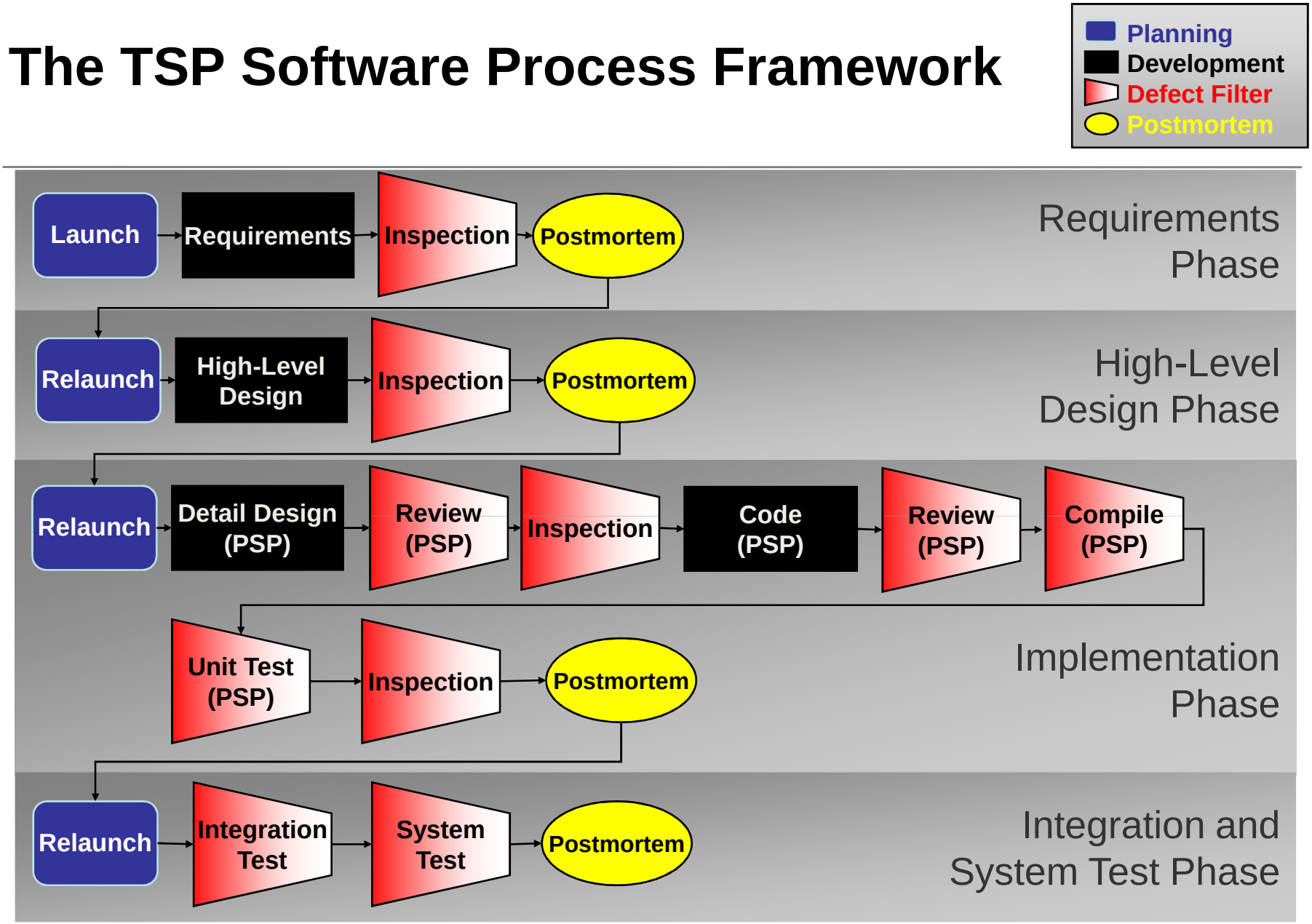
TSP is based on a defined process framework.

The process framework was designed to be integrated with, not replace, existing processes, methods, and tools.

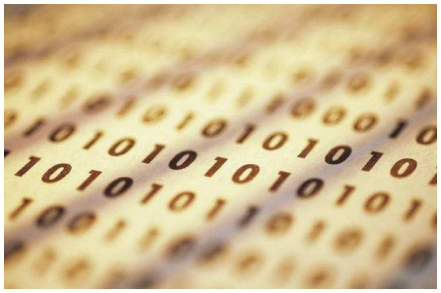
The TSP process

- guides the work instead of constraining it
- is straightforward and brief
- provides a basis for communicating the development approach

The TSP Software Process Framework



TSP Base Measures



Size

Effort

Quality

Schedule

The TSP Measurement Framework

TSP measurement framework includes

- four base measures: schedule, time, size, defects
- dozens of derived measures
- benchmark data for decision making and prediction

Unique features

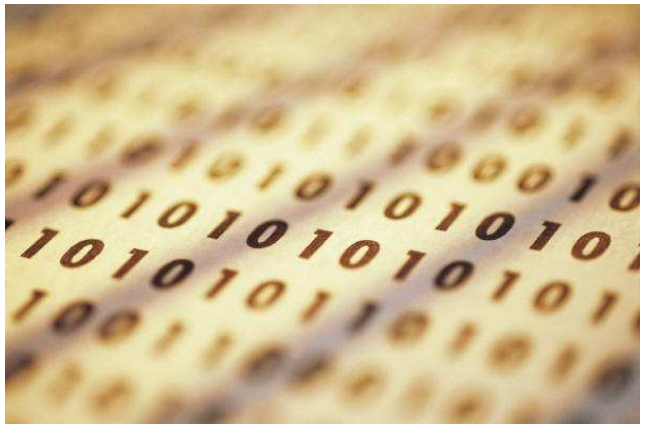
- developers gather and use the data
- quality metrics are emphasized

The metrics are detailed, but scalable, and readily support summaries as well as drill-downs, e.g.

- system, sub-system, component, object,…
- process, phase, task
- period

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:

ID	Assembly, Sub-Assembly, or Part Name	(A)ssembly or (P)art	Parent Assembly Name	Owner	Size Measure	Actual Size: Base	Deleted	Modified	Added	Reused	New and 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(SQL	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

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 Date: 2/3/2004
Team: PSP Ghost 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 **Date** 2/3/2004
Team PSP Ghost **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

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

Total Plan Hours: 318.9

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

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

Generate Task List | Update Task and Schedule

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

What the TSP Base Measures Provide

Sample of Derived Measures

Estimation accuracy (size/time)

Prediction intervals (size/time)

Time in phase distribution

Defect injection phase distribution

Defect removal phase distribution

Productivity

%Reuse

%New Reusable

Cost performance index

Planned value

Earned value

Predicted earned value

Derived Measures (continued)

Defect density

Defect density by phase

Defect removal rate by phase

Defect removal leverage

Review rates

Process yield

Phase yield

Failure cost of quality

Appraisal cost of quality

Appraisal/Failure COQ ratio

Percent defect free

Defect removal profiles

Quality profile

Quality profile index

TSP Management Process

Self-directed teams make their own plans, but they are also held accountable for knowing where they stand.

Tracking against the plan involves gathering the four base measures used in every TSP project.

- schedule
- size
- time
- defects

Status is then generated from these data. Examples:

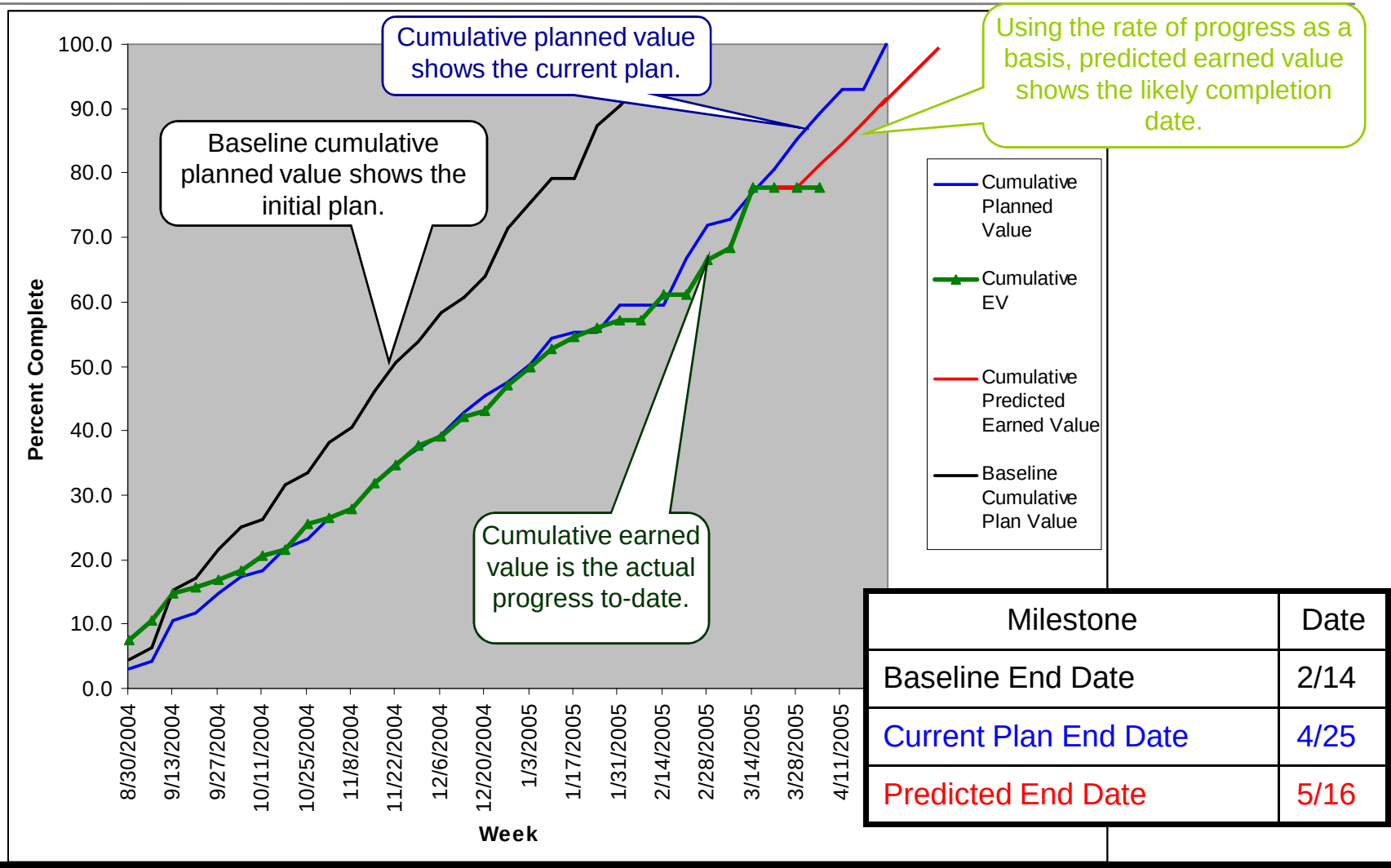
- earned value
- resources
- quality indicators
- weekly status

Earned Value

TSP uses earned value to track progress.

- A task's earned value (EV) is the task's estimated hours divided by the total estimated hours for all tasks in the project.
- The project earns the value for a task when it is completed, there is no partial credit.
- The EV credit is the same, regardless of the time the task actually takes.

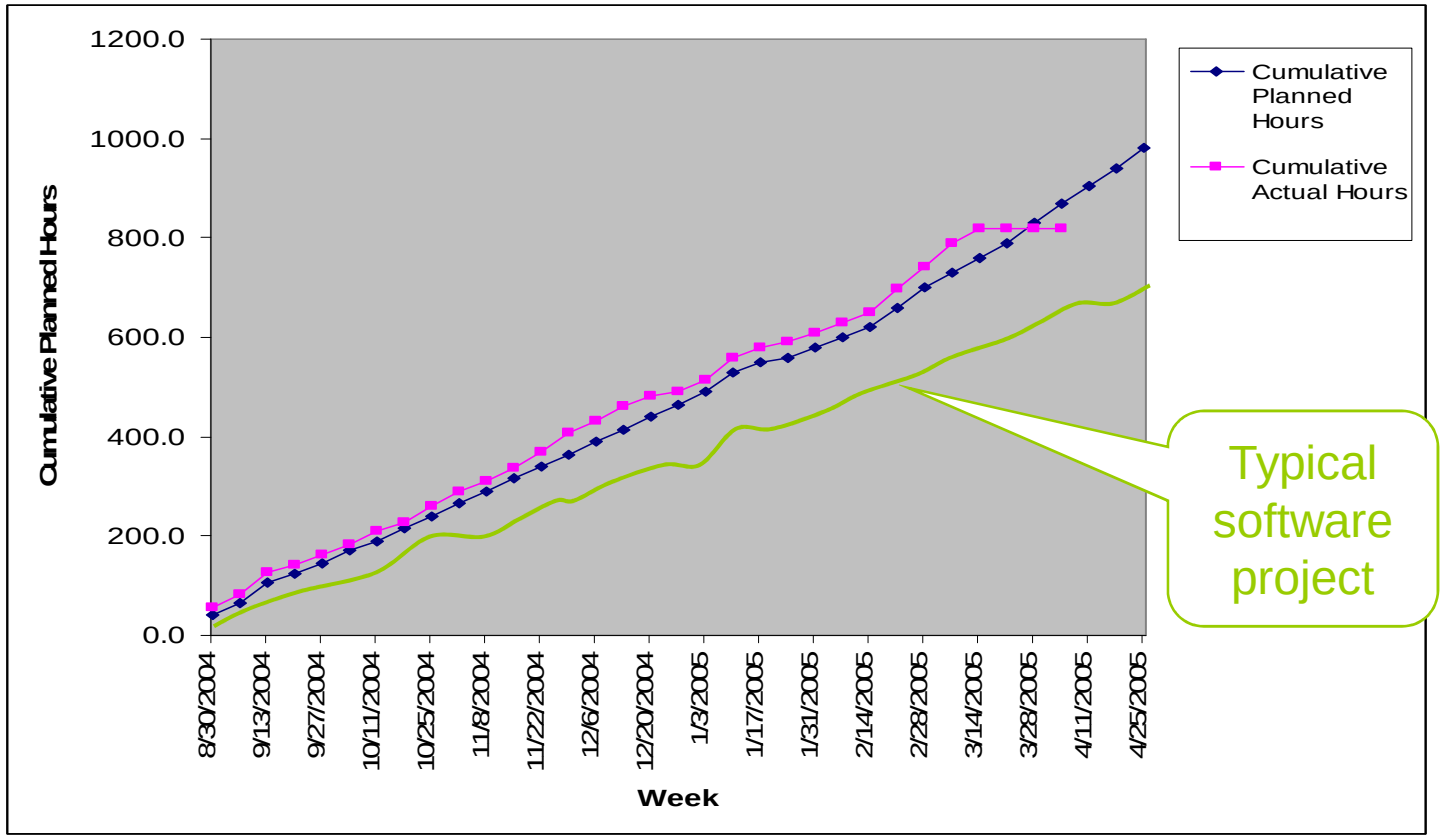
Earned Value Tracking



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

Resource Tracking

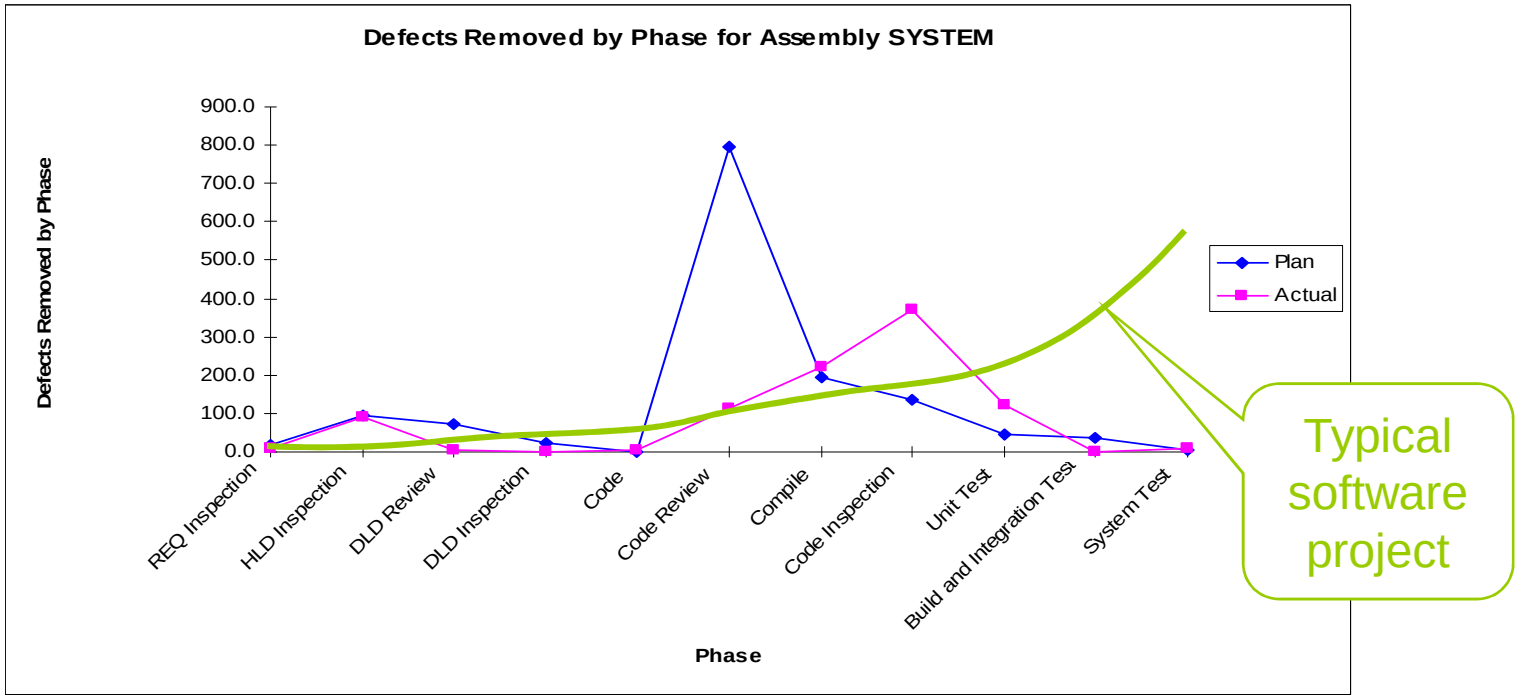
Cumulative plan and actual resource hours shows resource burn rate and potential source of slip



Defect Removal Profile

The defect removal profile shows

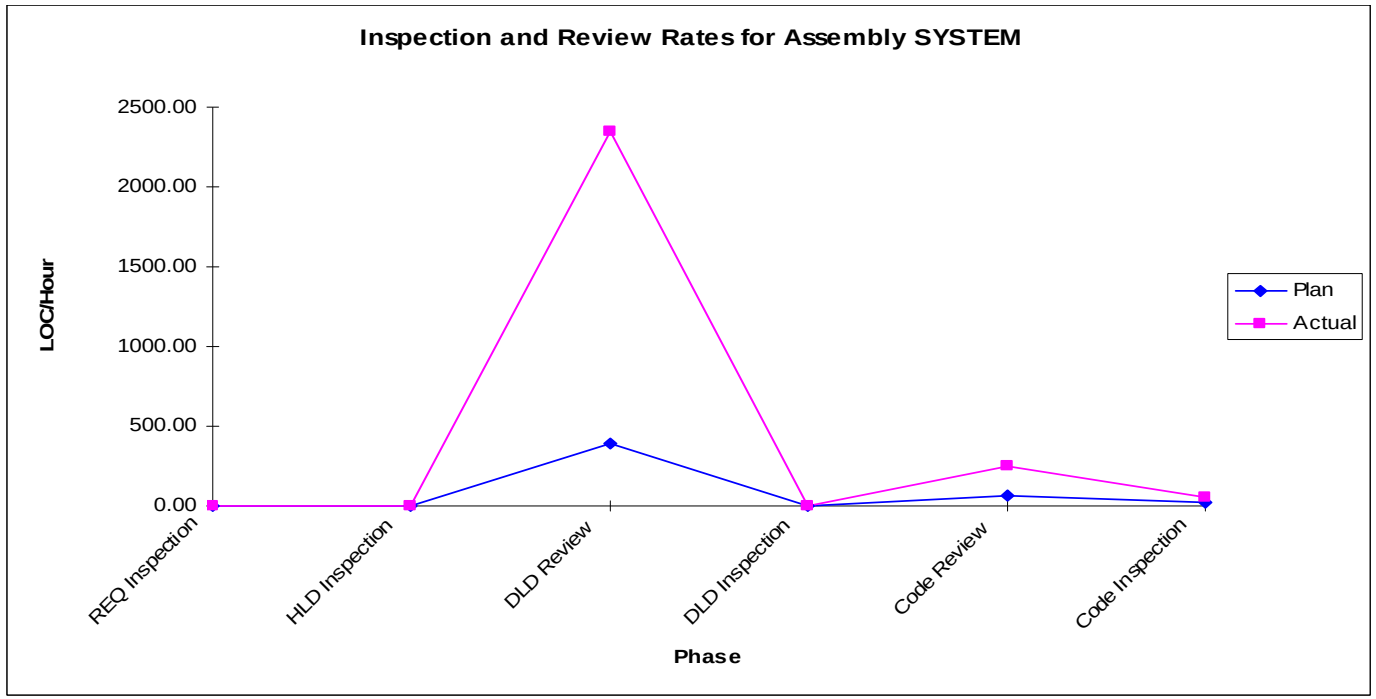
- plan and actual defects removed by phase
- early vs. late defect removal plan



Review and Inspection Rates

Review and inspection rates show

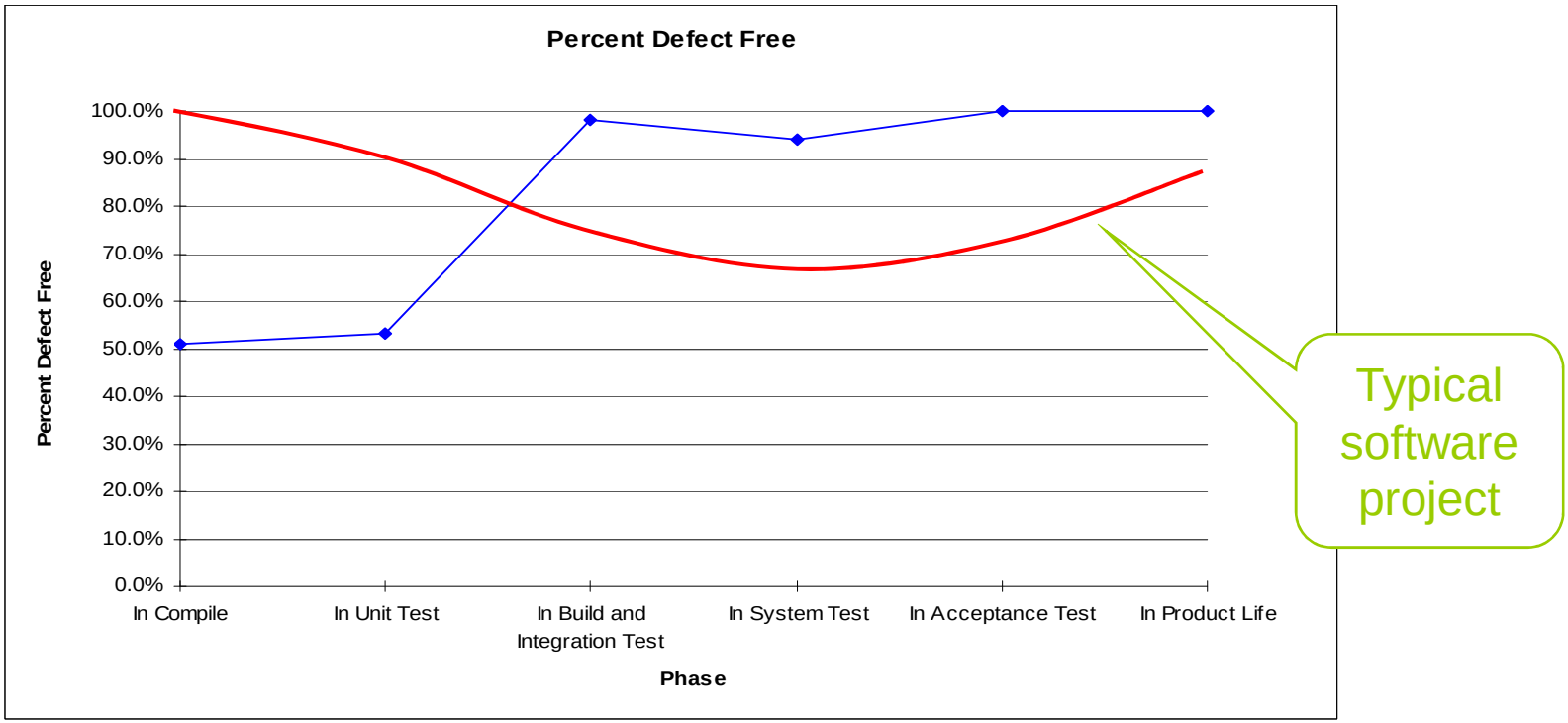
- reviews/inspections are being conducted
- likelihood that they are effective



Percent Defect Free

Percent defect free shows

- percentage of components that are defect free in test
- measure of pre-test process quality



Quality Benchmarking Example

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:

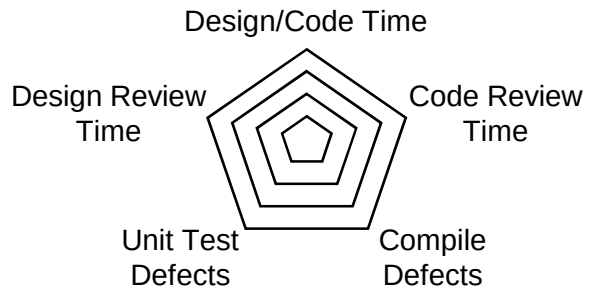
system → component → module → class.

Quality Profile Criteria

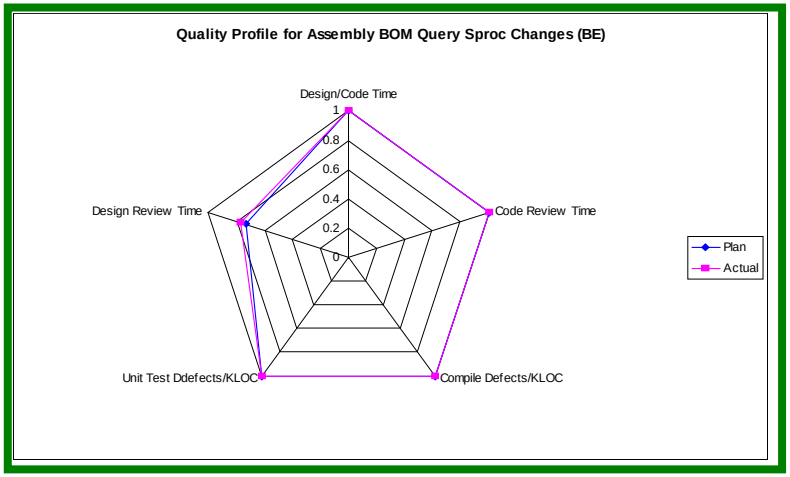
1. Design time = coding time
2. Design review time = ½ design time
3. Code review time = ½ coding time
4. Compile defects < 10 per KLOC
5. Unit test defects < 5 per KLOC

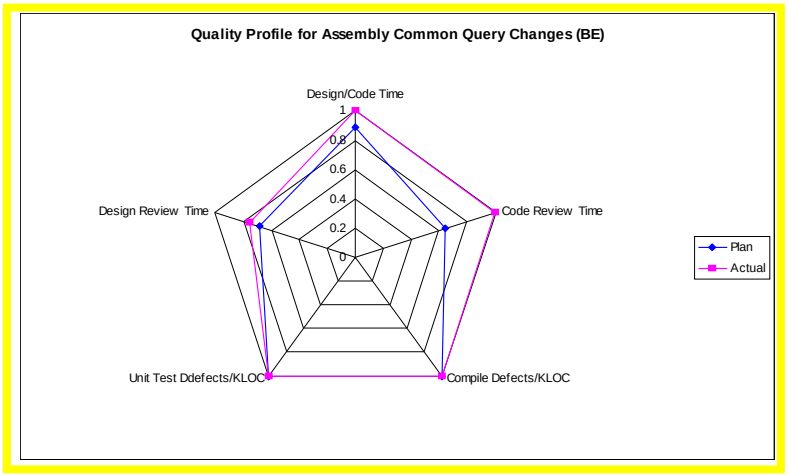
If satisfied, a criterion has a value of 1, and is drawn along the outer edge.

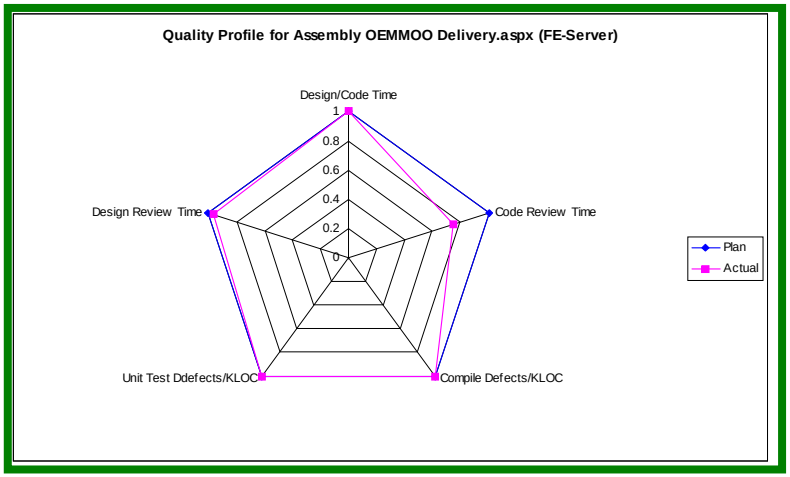
Quality Profile

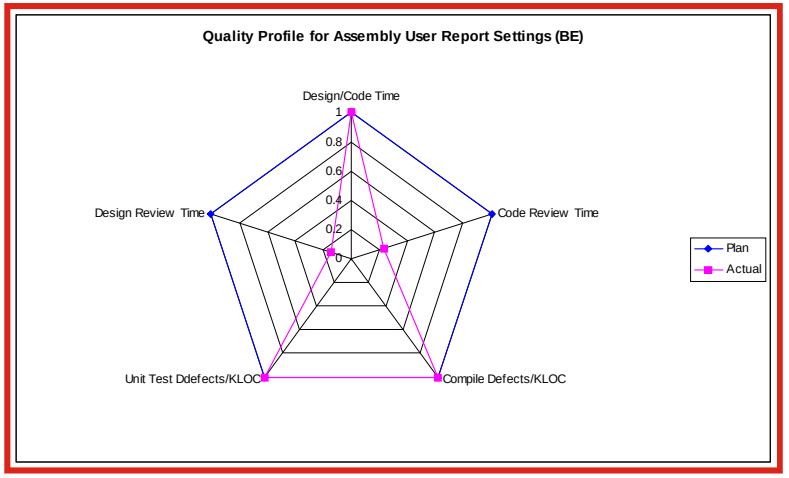


Using the Quality Profile









TSP Weekly Status Report

TSP Week Summary - Form WEEK

Name	Carol	**Date**	4/7/2003
Team	PSP Ghost		
Status for Week	15	**Cycle**	
Week Date	3/10/2003		

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.asp	UT	OEMMOO Delivery.aspx (FE-Server) U	NK	8.9	3.0	0.5	13	2.91
OEMMOO Delivery.asp	DLDINSP	OEMMOO Delivery.aspx (FE-Client) DI	NK	0.0	0.0	0.0	13	
OEMMOO Delivery.asp	CODE	OEMMOO Delivery.aspx (FE-Client) C	NK	7.5	5.7	0.4	14	1.32
OEMMOO Delivery.asp	CR	OEMMOO Delivery.aspx (FE-Client) C	NK	3.8	1.7	0.2	14	2.26
OEMMOO Delivery.asp	COMPILE	OEMMOO Delivery.aspx (FE-Client) C	NK	1.3	0.9	0.1	14	1.44
OEMMOO Delivery.asp	CODEINSP	OEMMOO Delivery.aspx (FE-Client) C	NK	0.0	0.0	0.0	14	
OEMMOO Delivery.asp	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

Topics

TSP Snapshot

Industry Performance

TSP Performance

TSP Concepts

TSP-I

Getting Started

Inter-disciplinary Teams

Standard TSP supports inter-disciplinary team settings

- The first industrial TSP pilot project included both hardware and software engineering
- The team tailored the TSP to meet the needs of both groups

Increased use of TSP has led to the need for a pre-tailored version of TSP for inter-disciplinary teams.

In 2006, SEI and NAVAIR initiated a collaborative effort to extend TSP to systems engineering and acquisition.

Approach

Develop a TSP process and measurement framework, training courses, and tools for systems engineering and acquisition.

Conduct pilot projects with NAVAIR on two systems.

- Harrier
- Hawkeye

Research Challenges -1

Process – developed prototype processes based on the as-is process, documented processes, and regulations, then integrated standard TSP components, e.g. TSP Launch Process, measures, etc.

Process Roles – developed new team member roles based on as-is process.

- Lab Test Manager
- Flight Test Manager

Training – a three-day TSP team member course was developed for team members that did no have software skills.

Research Challenges -2

Measurement

- Effort and schedule measures are unchanged.
- Size measures, based on TSP size measurement criteria, are now being evaluated by the pilot project.
- Defects remain the principal quality measure, but defect classification schemes and quality criteria are still being worked.

Tool Support

- Existing tools could not support the process and measurement tailoring requirements.
- A new support tool was developed; trial use is set for April, 2007

TSP-I Results

The first TSP-I Launch was held on 11 Sep 2006…Ran like a “normal” launch

- One team; 12 team members
- Two year overall plan
- Near-term plan is 6 months
- 475 tasks
- 22,000 task hours
- Gantt Chart didn’t provide visibility into all of the tasks that had to be completed
- Team members engaged in discussions of what the work would entail, dependencies, and what “task complete” meant.
- Team is preparing for a relaunch in April.
- Management team is already planning to apply TSP-I across all team involved with Harrier support.

TSP-I Issues

Issues:

- Level of granularity of the plan…more detail than before.
- Defining appropriate roles for SE Projects…Lab Test, Flight Test, ?
- Defining the SE process…latest version is substantially improved over process used at the first launch.
- Developing a quality plan…more study needed.

Topics

TSP Snapshot

Industry Performance

TSP Performance

TSP Concepts

TSP-I

Getting Started

Getting Started

Sprinkling a few PSP-trained engineers around an organization will not produce noticeable results.

Installing TSP in an organization requires

- a team-based improvement focus
- careful planning
- senior management involvement and sponsorship

Getting started

- train a few software teams and their managers
- launch the teams with TSP
- gather data and evaluate the results
- train change agents

TSP Product Suite Supports Introduction

Process Notebook

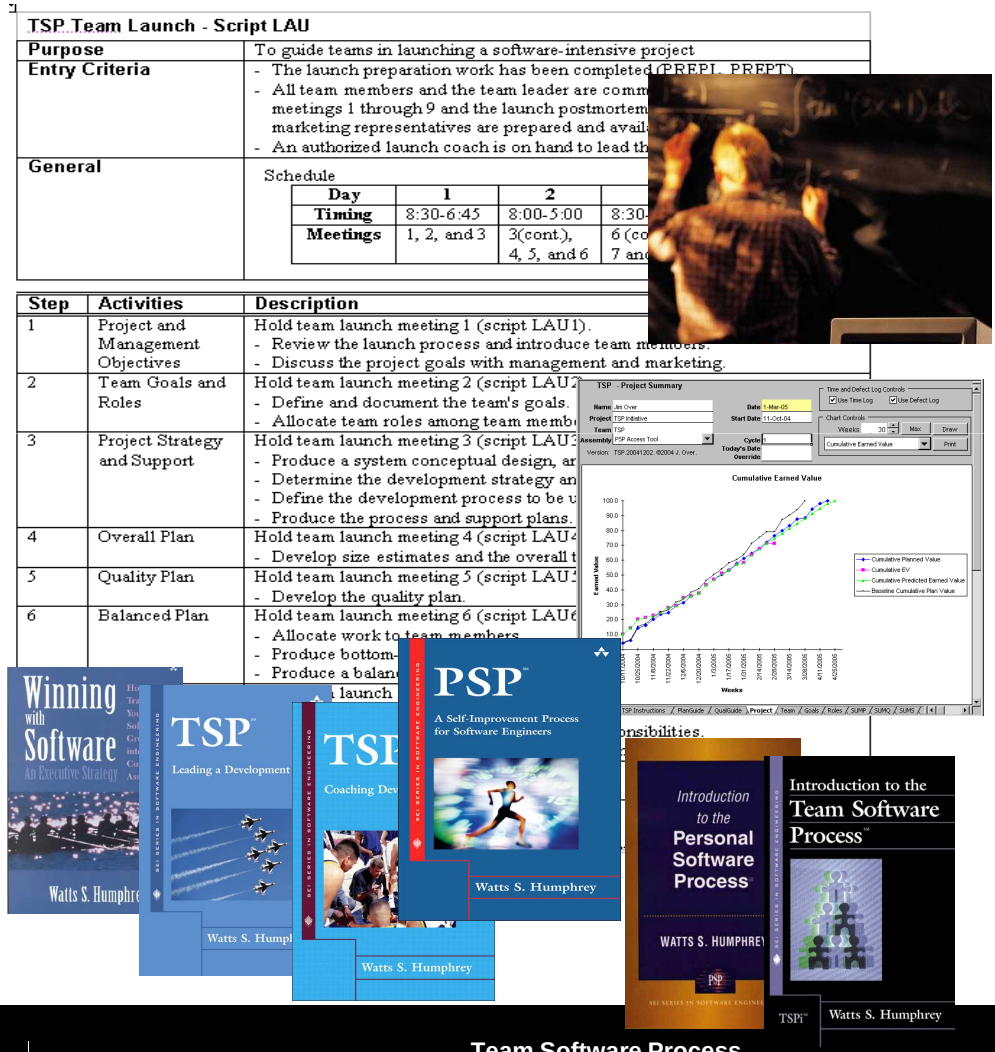
- Process scripts
- Planning and Tracking Forms
- Guidelines and standards

Training and Textbooks

- Executive/Management
- Engineering
- Coach/Trainer

Tools

- TSP Workbook
- PSP Workbook
- Coach/Trainer Workbook



TSP Introduction Strategy: Pilot Phase

Executive kickoff and planning seminar – 1 day

Management training – 2.5 days

Engineer training

- seven lessons (1-2 days each); flexible class structure
- teaches personal project management
- engineers learn to measure and manage quality
- required for effective team participation

Launch pilot projects using TSP

TSP Introduction Strategy: Rollout Phase

Internal transition agent training

- PSP Instructor – 5 days
- TSP Launch Coach (FY01) – 5 days

Plan and implement broad transition

- Make a plan, commit resources
- Work project-by-project, team-by-team
- Expand breadth of adoption on each cycle
  - Repeat training steps as needed
  - Build internal support as needed
- Manage and motivate change

Questions?

Software Engineering Institute | Carnegie Mellon