

Personal Software Process Introduction and Team Software Briefing

Peter Abowd
March 14, 2002
GL-SPIN

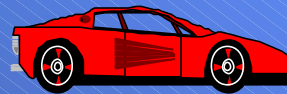
Trademarks and Service Marks

- The following are service marks of Carnegie Mellon University.
 - Capability Maturity Model IntegrationSM
 - CMMiSM
 - Team Software ProcessSM
 - TSPSM
 - Personal Software ProcessSM
 - PSPSM

- The following are registered trademarks of Carnegie Mellon University
 - Capability Maturity Model®
 - CMM®

Introduction : Background - 1

- n Peter Abowd
 - Automotive electronic/software engineers
 - 14 years at Ford Motor and Visteon Corporation
 - Several Product Releases
 - Audio System Software
 - Education
 - BSEE from University of Notre Dame
 - MSE from Carnegie Mellon University



Introduction : Background - 2

- n First PSP course at Carnegie Mellon
- n Performed Research with Watts Humphrey
 - Could PSP work for embedded automotive?
 - Estimate ROM, RAM and Cycle time better?
- n Developed an Embedded System
 - Built Bicycle Monitor software with PSP
- n Analyzed Results
 - Extended PSP process for Embedded Systems



Introduction : Background - 3

- n Trained Embedded Engineers in PSP
- n Employed PSP on Released Software
 - Encounter anticipated and unanticipated issues



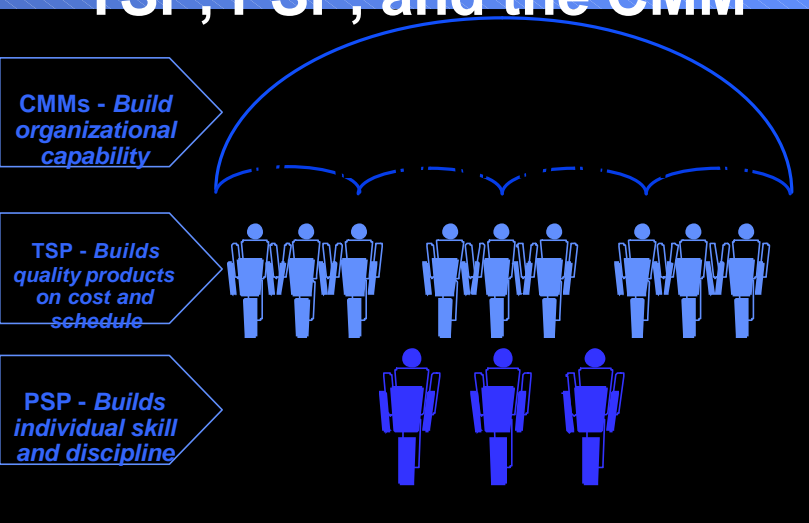
Introduction : Background – 4

- n Trained over 180 Engineers in PSP
 - >1400 programming assignments
 - Several Automotive and other organizations
 - Automotive OEM
 - Automotive Suppliers (Tier 1 & 2)
 - Airline : IT managers and developers
 - Glass Manufacturing: IT managers
 - Telecommunications: embedded software managers
- n Integrating into Level 1, 2 & 3 Organizations

Connecting CMM, TSP and PSP

- n CMM What Vs. PSP How
 - CMM is descriptive of what a process needs
 - PSP is prescriptive, how to perform a process
- n CMM Defines
 - What activities a capable process contains
 - Who shall perform the required activities
 - Order in which to introduce the activities

TSP, PSP, and the CMM



CMM KPAs in PSP and TSP

| Level | Focus | Key Process Areas (KPA) |
|--------------|--------------------------------|--|
| 5 Optimizing | Continuous process improvement | <ul style="list-style-type: none"> ∞ Defect prevention ∞ Technology change management ∞ Process change management |
| 4 Managed | Product and process quality | <ul style="list-style-type: none"> ∞ Quantitative process management ∞ Software quality management |
| 3 Defined | Engineering process | <ul style="list-style-type: none"> ∞ Organization process focus ∞ Organization process definition <ul style="list-style-type: none"> Training program ∞ Integrated software management ∞ Software product engineering ∞ Intergroup coordination ∞ Peer reviews |
| 2 Repeatable | Project management | <ul style="list-style-type: none"> ∞ Requirements management ∞ Software project planning ∞ Software project tracking ∞ Software quality assurance ∞ Software configuration management Software subcontract management |

∞ CMM Key Process Area addressed at the project level when using PSP and TSP

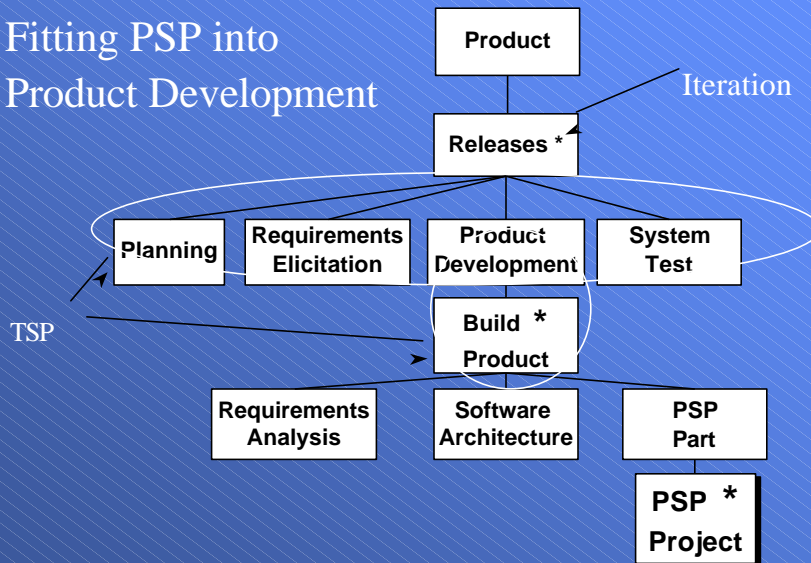
PSP Content Overview

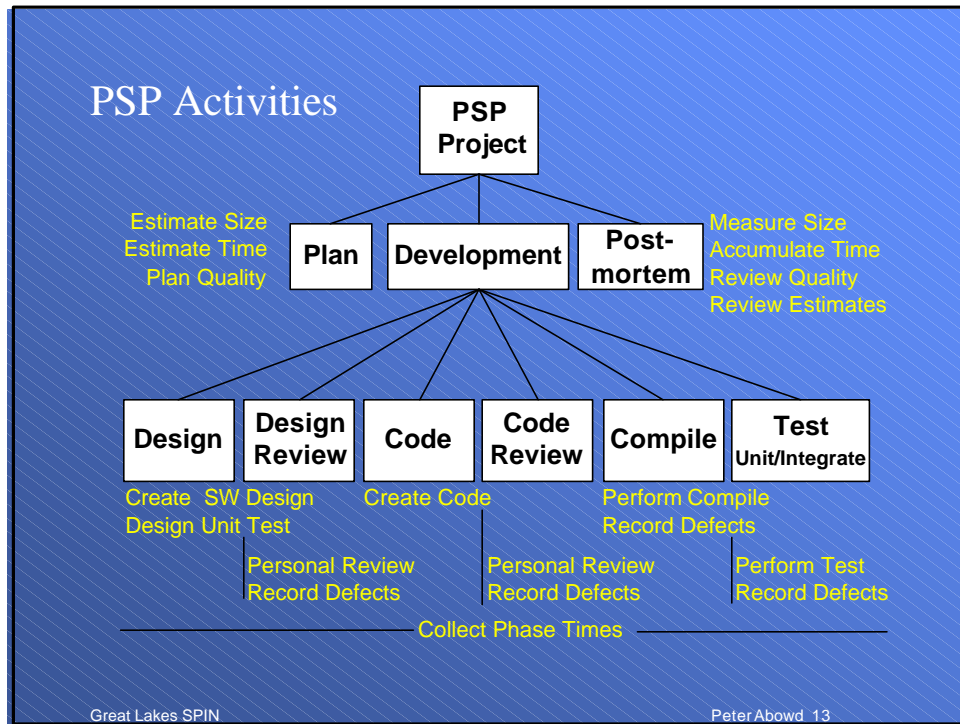
- ii PSP Provides
 - Explicit procedures to meet CMM requirements
 - Explicit techniques for measuring, planning, quality
 - Statistical analysis of historic data for planning
 - Analysis of defect data for process improvement
 - Defined unit testing and planned execution
 - Extensive software process training
 - Project planning & tracking procedures provided
 - Critical computing resource estimation procedure

Typical Software Cycle

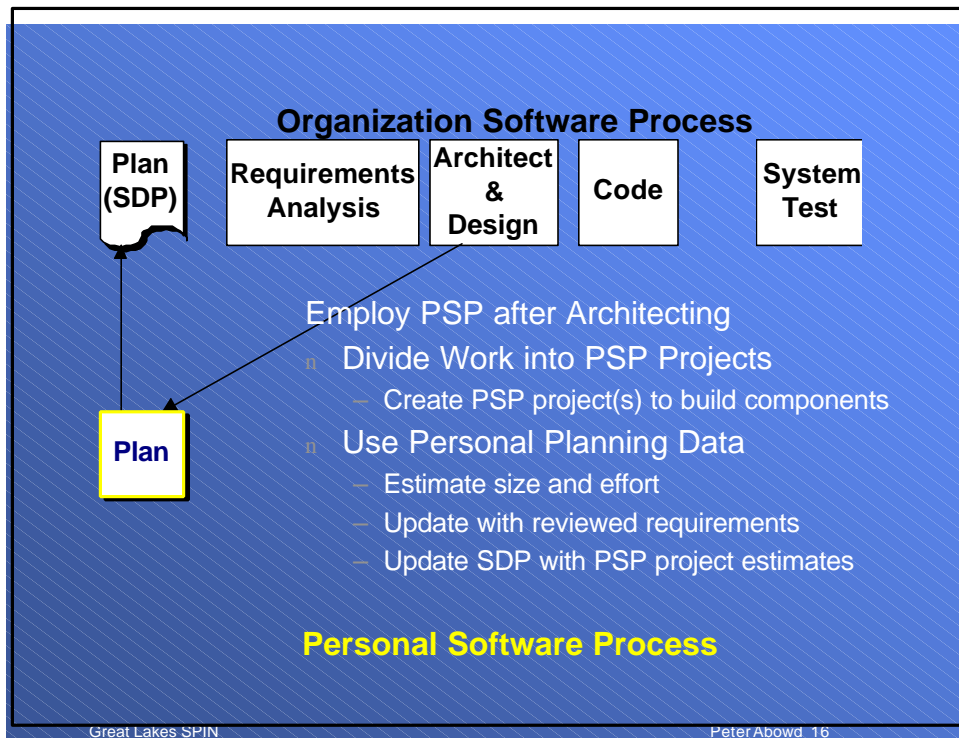
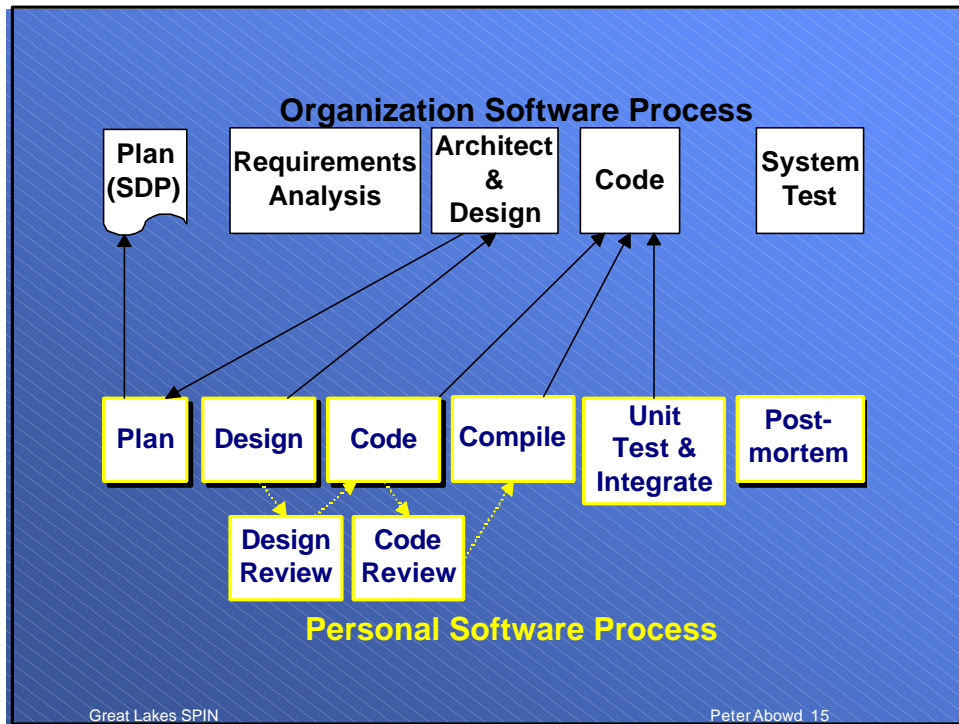
- n An Iteration of Software Releases
 - Plan, Requirements, Develop and System Test
- n Developers Iteratively Builds Pieces
 - Analyze, then architect and implement
 - PSP is utilized to implement pieces
 - Detailed design, personal reviews, code, compile and unit test

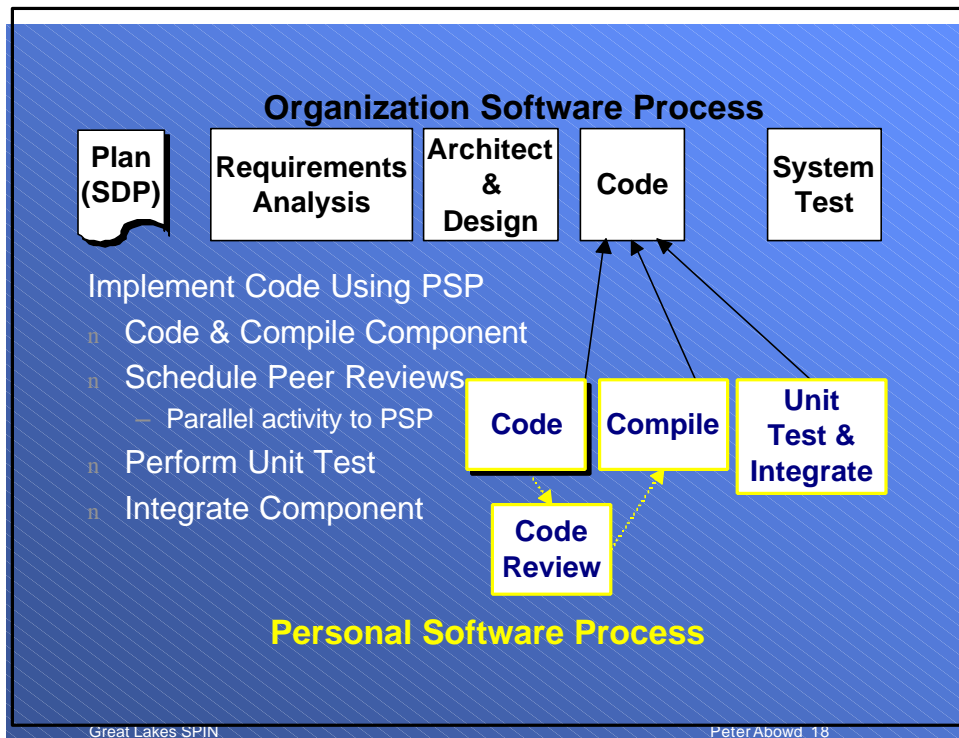
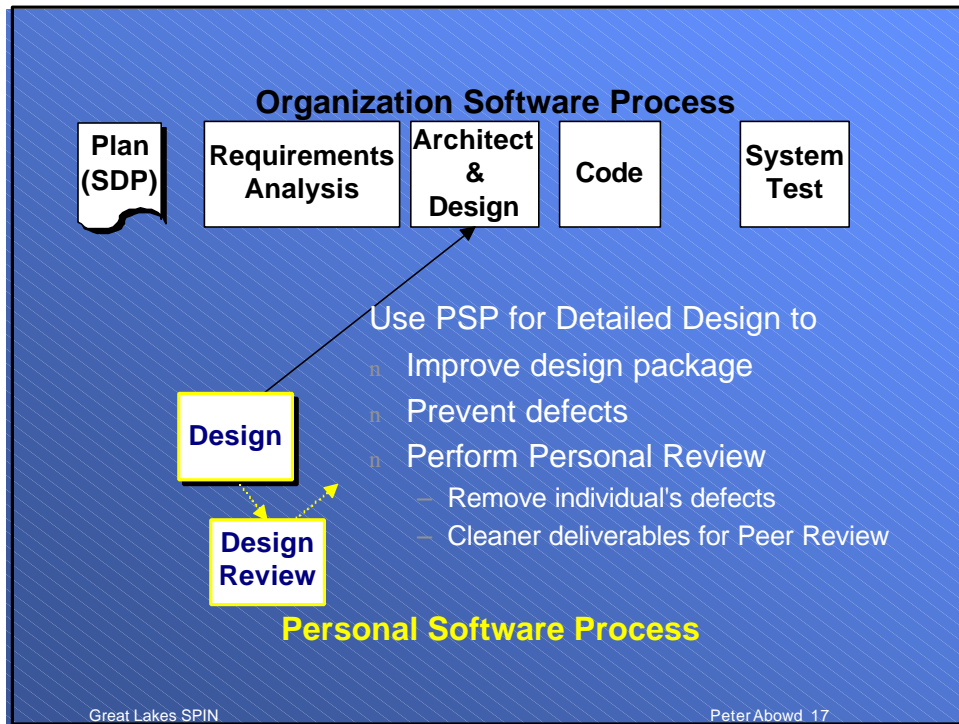
Fitting PSP into Product Development





- ## Using PSP
- n Perform Planning and Requirements Activities
 - Create overall plan with gross estimates
 - n Create Architectural (High Level) Designs
 - Identify software components to build or modify
 - n Use PSP to plan size and effort of pieces
 - Create a set of *PSP Project* plans
 - Use PSP plans to update original gross estimates
 - Execute *PSP Projects* Individually
- Great Lakes SPIN Peter Abowd 14





Organization Software Process



Perform a Postmortem for each PSP Project

- Compile and analyze metrics
 - Adjust Personal Planning Engine
- ...
- Start next PSP Project
 - Re-plan necessary?

Post-mortem

Personal Software Process

**Next 7 slides included with permission
from the Software Engineering
Institute**

TSP Structure and Flow

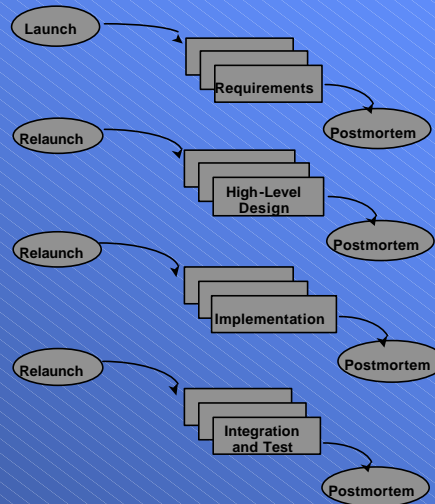
TSP has four principal development phases.

- Requirements
- High-Level Design
- Implementation
- Test

Each phase begins with a launch or relaunch.

TSP encourages iterative development strategies.

TSP permits whatever process structure makes the most business and technical sense.



Team-Directed Projects

- n TSP projects are team-directed.
- n Being a team-directed project means that the team has to manage itself.
 - plan and track their work
 - manage the quality of their work
 - work aggressively to meet team goals
- n The team must also show management and the customer that they are managing themselves.
 - frequently report status and progress
 - anticipate, plan for, and report on project risks

Team Member Roles

- The self-management responsibilities are shared among team members through eight defined team member roles.
 - Customer Interface Manager
 - Design Manager
 - Implementation Manager
 - Planning Manager
 - Process Manager
 - Quality Manager
 - Support Manager
 - Test Manager

- The team leader acts as team coach.

The TSP Launch

The TSP launch is a three to four day workshop used to kick-off each project phase.

The team leader and all team members participate.

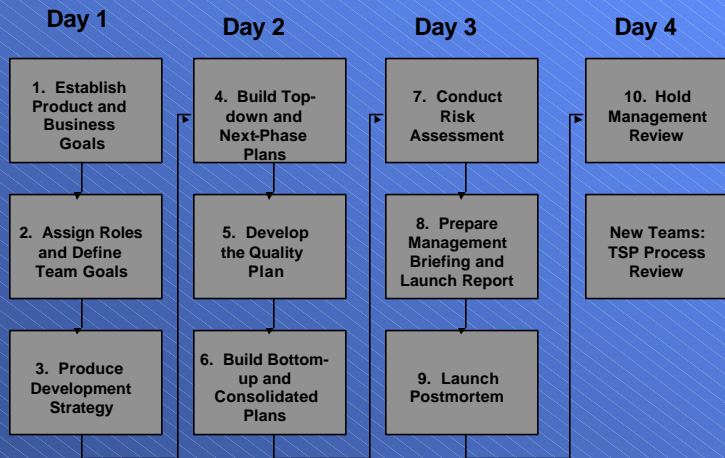
The launch workshop accelerates team-building.

The team establishes a common understanding of the work and the approach they will use.

They make a plan that they can commit to and obtain management support for their plan.

The most important launch outcome is a motivated team.

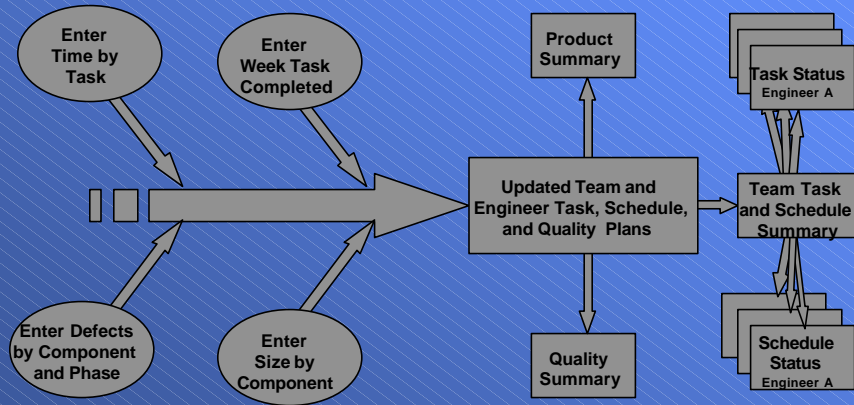
The Launch Process Meetings



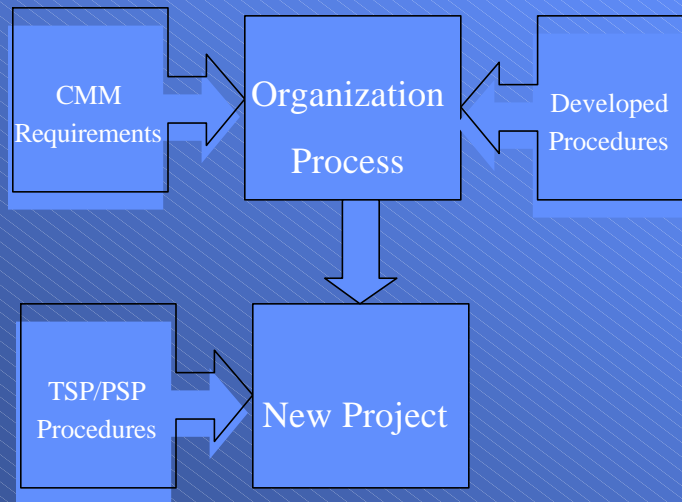
TSP Project Tracking -1

- n Project tracking in the TSP is based on the principles and measures used in the PSP.
- n The detailed team and individual plans facilitate precise project tracking.
- n Each team member is responsible for
 - gathering data on their work
 - tracking status against their personal plan
 - keeping the team informed
- n Each team member is also responsible for
 - the quality of the work they produce

TSP Project Tracking -2



Integrating TSP/PSP into Organization



Goals for TSP/PSP Based SPI

- Identify Elements of your process which can benefit from TSP/PSP integration
- Identify who performs certain Activities and if they can benefit from TSP/PSP procedures

Adding PSP to SPI Effort

- Use TSP/PSP Procedures to Introduce Level 5
 - CMM based SPI requires procedure development
 - TSP/PSP provides many effective procedures
 - Review Slide 10!

Common Misconceptions

- Do CMM or TSP/PSP
- Must get to Level x then add TSP/PSP
- Train Team in PSP, Launch Project with TSP
 - Start working at Level 5! (even if your company isn't!)

Redefining Procedures

- n Mature Organizations
 - Already have many procedures in place
 - Don't want to disturb the process too much
 - Some provide what TSP/PSP provides
 - Compare their procedures to TSP/PSP
 - Typically TSP/PSP provides a more analytical procedure
 - Benefit of adopting TSP/PSP procedure
 - Training intact
 - This is an early step in the TSP Launch!
 - Already can measure improvement
 - So...Pilot the PSP and measure its impact

Redefining Procedures

- n Organizations with Recent Assessments
 - “Now we'll add PSP”
 - “But we just spent months defining similar procedures!”
- n Replace Quickly Developed Procedures
 - Those put in place “for the assessment”
 - Use the more mature TSP/PSP procedures

PSP Background Information

Preparing the Organization

- n Course Involves Significant Effort
 - 10 programs, 2 reports
 - 7 Process steps to learn PSP
 - PSP is introduced gradually
- n Engineers Require Time to Finish
 - Need to finish assignments after hours
 - Have work between and after sessions
 - In 5 day formats for Part I and II

What is a Personal Software Process

- n Customized Personal Working Framework
- n Defined Steps for Building Quality Software
 - Develop accurate time and size estimates
 - Develop efficient defect removal practices
 - Repeatedly develop high quality software...intentionally
- n Defined and Customized
 - Forms
 - Measurements
 - Standards
 - Checklists
 - Performance databases

What is a PSP (2)

- n Start of Course Student's Process
 - Informal Design
 - Code
 - Compile
 - Test (fixing bugs-rewrite code often)

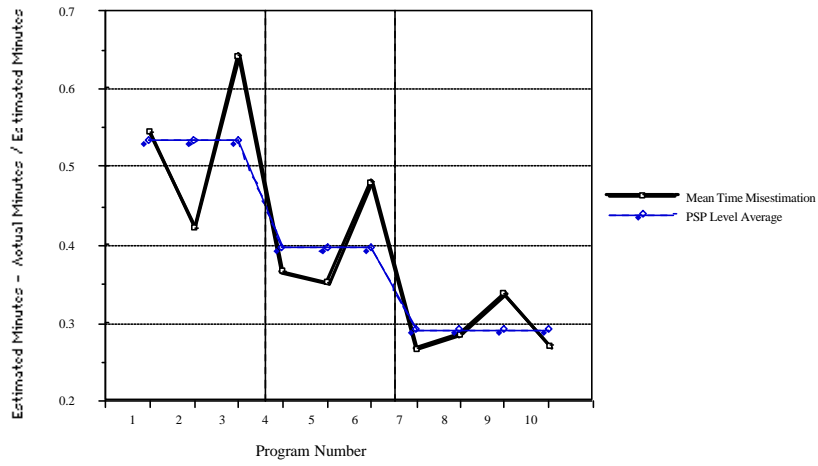
What is a PSP (3)

- n New Process for Student at Program 7
 - Planning Size, Time, and Quality of Program
 - Design
 - Design Review
 - Code
 - Code Review
 - Compile
 - Test
 - Postmortem
 - Analyze metrics on Size, Time and Quality

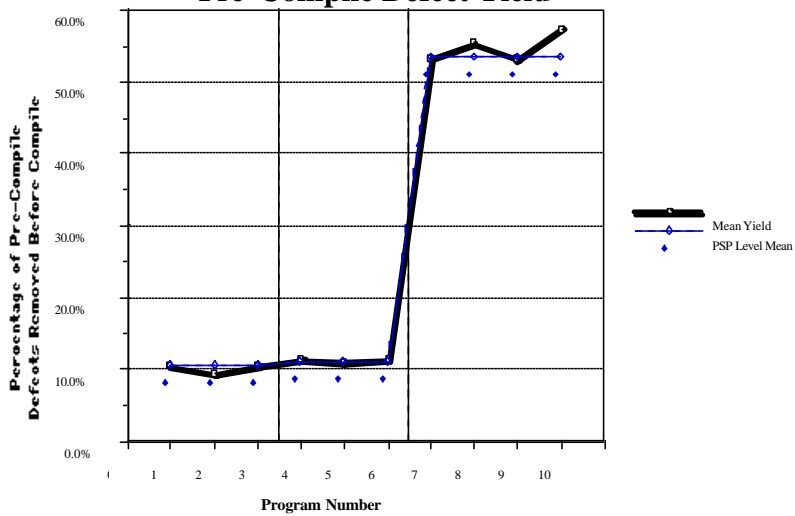
What is a PSP (4)

- n Developers Build Estimating Database
 - Linear regression models of estimates
 - Estimated size (X data) vs. Actual size (Y data)
 - Estimated size (X data) vs. Actual time (Ydata)
 - Standard table of Software *Objects* (Module) Sizes
 - Reflects developer's own style of writing code

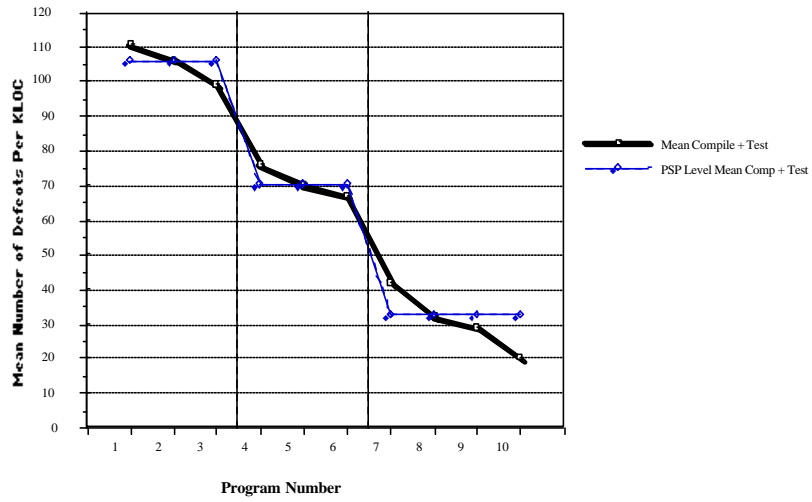
Effort Estimation Accuracy Trend



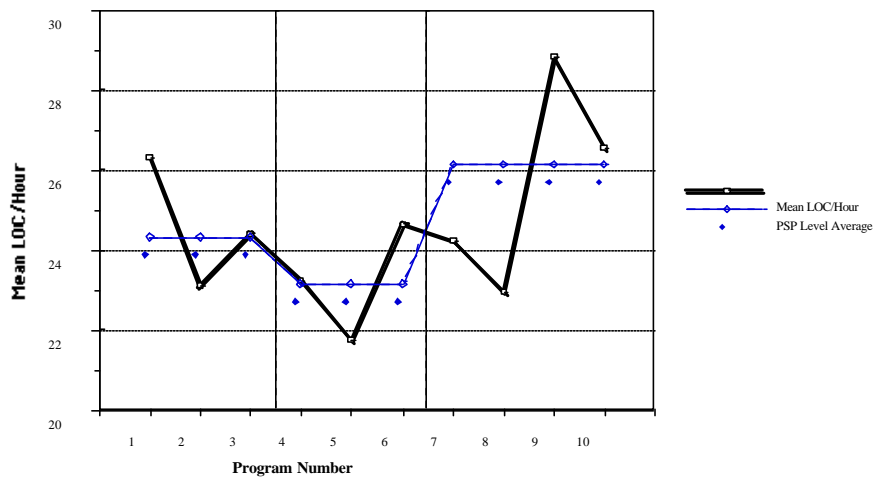
Pre-Compile Defect Yield



Defects Per KLOC Removed in Compile and Test



Lines of (New and Changed) Code Produced Per Hour of Total Development Time



Using PSP For Planning

Small Project - Small Team

- n PSP Trained Developers have Procedures
 - How to plan 100's of LOC and 1000's of LOC
 - 1 to 3 person teams use these procedures easily
 - PSP provides team the following
 - Planning and Tracking procedures
 - Critical resource estimating
 - Quality plans and measurement procedures
 - Include requirements activities in PSP planning phase
 - Need to define Validation procedures

Using PSP For Planning

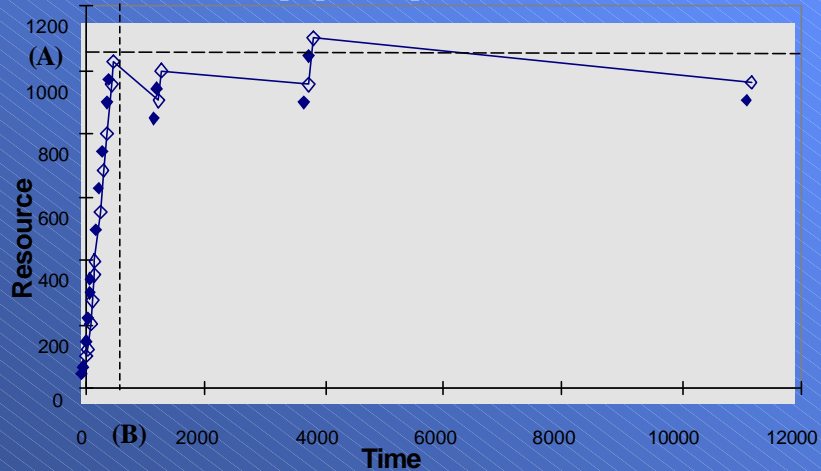
Large Project - Large Team

- n Requires Iterative Re-Planning
 - Initial overall plan with gross estimates
 - Detailed plan of next phase activities
 - Re-estimates based on new knowledge
 - Re-estimate based upon more detailed historical data
- n Require More Team Coordination Procedures
- n Need Requirements Phase and Procedures
- n PSP Tracking More Critical
 - Uses Earned Value technique
- n Need to Define Validation Procedures

Accounting for Project Type

- n Possible Types of Projects
 - Maintenance
 - Fixing defects and adding features
 - New development
 - Creating new applications
 - Experimental
 - Investigating new technology
- n Choose Appropriate Historical Planning Data

Use Appropriate Data



Appropriate Data

- n PSP Used Effectively to Time (B)
 - Previous project data
 - Predicted Feature 1 performance well
 - Feature 1 and previous data
 - Predicted Feature 2 performance well
 - etc.
- n Model Changes when Limit (A) Reached
 - Use post (B) data to predict other post (B) work
- n Effects both Maintenance and New Development
- n PSP Users can Maintain sets of Historic Data

Critical Resource Estimating

- n PSP Provides Statistical Analysis of Data
 - Can accurately predict LOC size
- n Embedded PSP Extensions
 - Predict ROM, RAM, EEPROM, Flash
 - Employs Basic Linear Regression Techniques

Coordinating Efforts

- n Resolving Individual and Team Responsibility
 - Individual must perform well and manage effort
 - Individual focus on defect prevention/removal
 - Achieve personal low defect density
 - Team focus on high quality of system
- n Effective Coordination - No Redundancy
 - Personal and peer reviews distinctly defined
 - Personal and team planning distinctly defined

Coordinating Efforts - 2

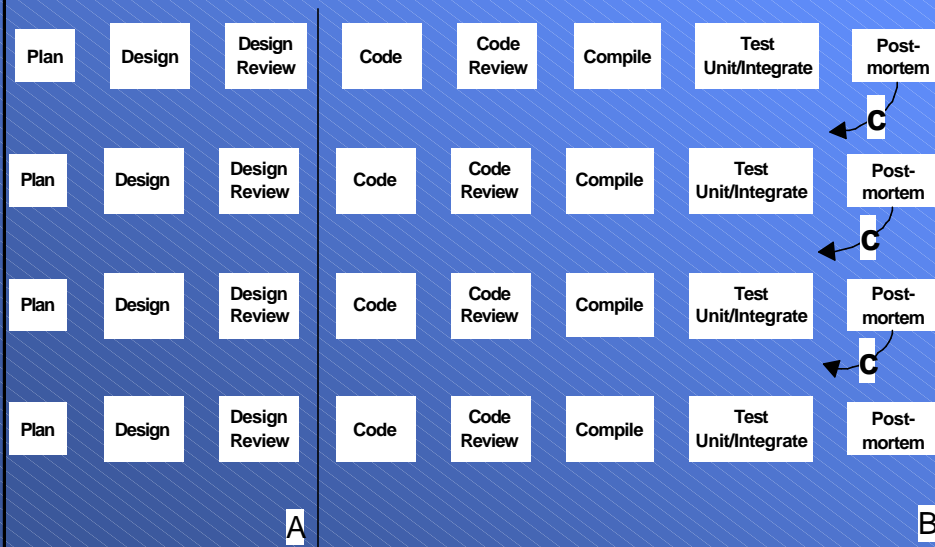
- n Team Members have Dual Responsibilities
 - Personal work
 - Peer-Related work
- n Poor planning of Peer-Related Work
 - Negatively impacts personal work
- n Resource Scheduling Critical
 - Conflict is not between PSP and Org. Process
 - Peer activities need planning consideration

Coordinating Efforts - 3

Personal and Peer Reviews Example:

- n Need to Coordinate for Maximum benefit
 - Personal reviews remove individual's defects
 - Peer reviews check for quality issues
- n Several Options available all assume
 - PSP project 7-14 days in length
 - Peer reviews are NOT instantaneous
- n Example: Build one component
 - Use four PSP projects
 - When should peer reviews occur?

Personal and Peer Reviews



Pause...Want More? Contacts

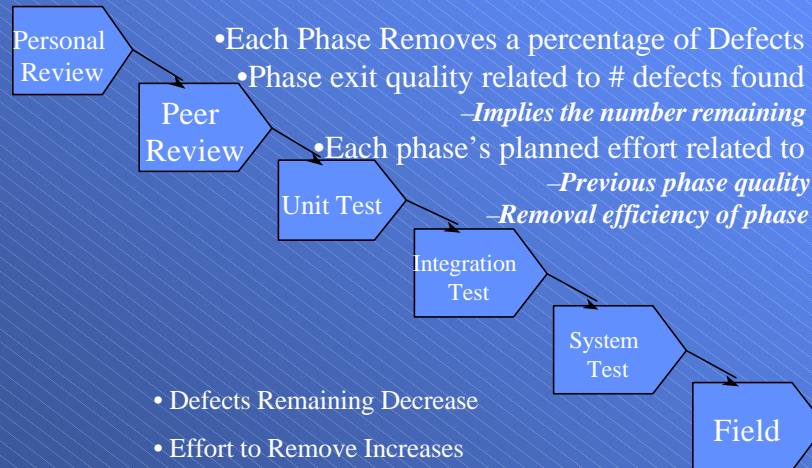
Peter Abowd
Visteon Corporation
(313) 755-4320
pabowd@visteon.com

Embedded Software Professionals
Embedded PSP Extensions and
Personal Planning Engine ©
248-921-4782
...more slides...

Measuring Improvement

- n Need to Understand Effect of Improvement
- n Need to Define this Strategy Early
 - When planning SPI activities
- n PSP's Effect may be Easily Measured
 - Employ The Defect Filter Model

Defect Filter Model



Utilizing PSP Data

- PSP Data can Provide the Following Insights
 - Typical Developer Defect Injection Rates
 - Personal Review Efficiencies
 - Defects removed per hour
 - LOC reviewed per Hour
 - Measure of Personal Quality and its Cost
 - Unit Test Defect Density
 - Unit Test Effort
 - Personal Review Effort

Utilizing PSP Data

- n Aggregate Individual Data
 - Characterize a team
 - Derive gross estimating data for early planning
 - Note: Requirements and Validation may not be included

Defect Measures

- n Yield
 - The percent of defects removed in a phase

$$Yield ? \frac{\# Defects Found}{\# Defects Found + \# Defects Not Found} * 100$$

- n Defect Injection Rate
 - Typical # of defects injected per 1000 LOC
- n Defect Removal Rates
 - # defects removed in one hour

Strategy

- n Measure ↗ Change ↗ Measure
- n Measure
 - How does the quality of my work impact the effort of those who use what I develop?
- n Change
 - Change individual practices
- n Measure
 - Measure downstream quality and effort
 - *Did it change?*

Strategy

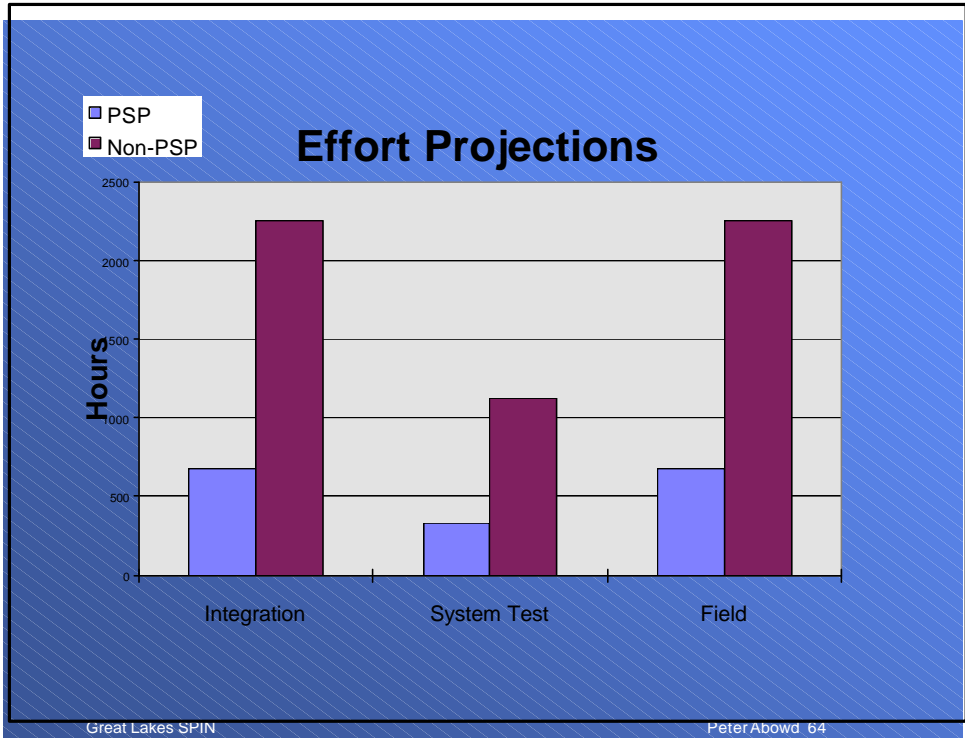
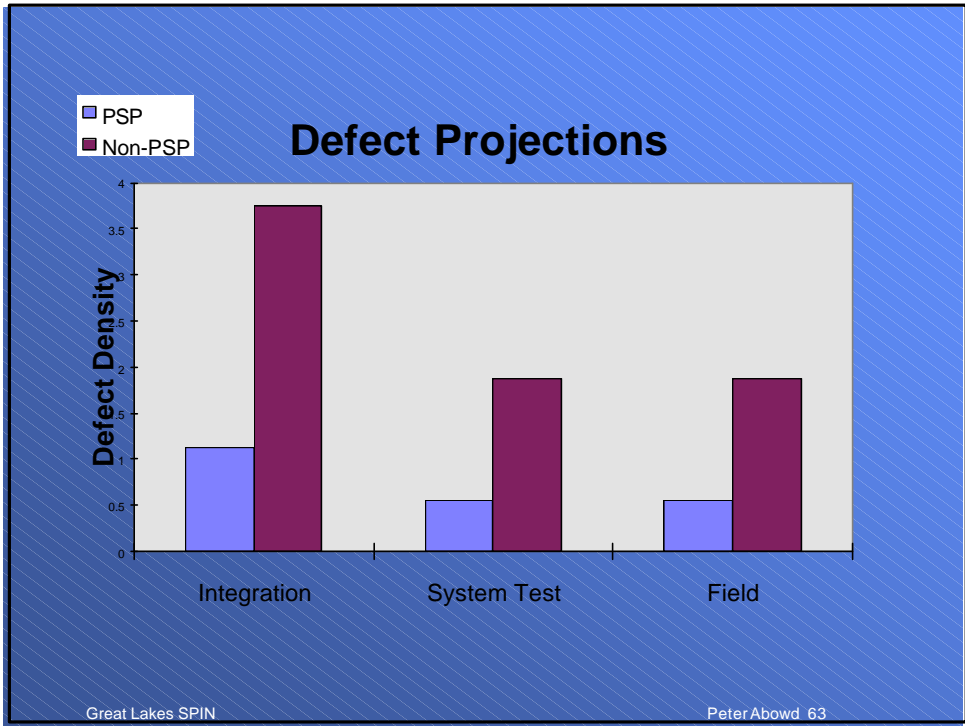
- n Given a Release of a Software Product
 - Define a developer independent test phase
 - Measure length of time release takes to pass
 - Count number of defects found by tests
 - Implement a Process Change on Next Release
 - Perform developer independent test again
 - Measure length of time release takes to pass
 - Count number of defects found by tests
 - Were desired results achieved?
 - Less time in test
 - Fewer defects found

Using A PSP Defect Model

| Removal Phase | Typical Yields | Project Yields | Typical Removal Def/Hr | Project Removal Def/Hr | Desired Review Rate LOC/Hr |
|-----------------------|----------------|----------------|------------------------|------------------------|----------------------------|
| PSP Review | 70% | | 7 | | 200 |
| Compile | 50% | | 15 | | |
| Unit Test | 50% | | 0.33 | | |
| Peer Review | 70% | | 3 | | 200 |
| Integration | 50% | | 0.1 | | |
| Validation | 50% | | 0.1 | | |
| Warranty | | | 0.05 | | |
| Defect Injection Rate | 100 | | | | |

PSP Defect Filter Model

| | | | | |
|-------------------|-----------------|------------------|----------------|-------|
| Estimated Size | 60000 | | | |
| Defect Projection | 6000 | | | |
| | Defects Removed | Defect Remaining | Defect Density | Time |
| PSP Review | 4200 | 1800 | 70 | 300 |
| Compile | 900 | 900 | 15 | 60 |
| Unit Test | 450 | 450 | 7.5 | 1350 |
| Peer Review | 315 | 135 | 5.25 | 900 |
| Integration | 67.5 | 68 | 1.125 | 675 |
| Validation Test | 33.75 | 34 | 0.5625 | 337.5 |
| Field Density | | 34 | 0.5625 | 675 |
| Compile | 3000 | 3000 | 50 | 200 |
| Unit Test | 1500 | 1500 | 25 | 4500 |
| Peer Review | 1050 | 450 | 17.5 | 900 |
| Integration | 225 | 225 | 3.75 | 2250 |
| Validation Test | 112.5 | 113 | 1.875 | 1125 |
| Field Density | | 113 | 1.875 | 2250 |



Measurement Summary

- n Individuals Affect Software Quality
 - Improvement significantly impacts product
 - Overall development effort
 - Time to market
 - Quality
 - Determining improvement requires planning
 - Apply the PSP defect filter model

Summary

- n Where PSP can Fit
 - Planning
 - Tracking and Oversight
 - Critical Resource Estimating
 - Software Product Engineering Procedures