Changing Behavior:
The key to adoption of complex process technology

Dr. Gene Miluk
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA  15213

November, 2009
My goals for this presentation

1) Present new or different approaches to technology transition
2) Challenge your current thinking (changing change agents is hard)
3) Describe what I see is working in the field (and my thoughts on why)
4) Focus on the potential benefits to you and your organization inherent in these approaches to change
5) Describe my reactions and internalization of the approaches
Topics

- Current SEI Change Management Approach
- What's Needed
- A New Approach
- Bandura Social Learning
- Bayesian Belief Networks
Comprehensive System Change Model (IDEAL)

Typical Organization Structure

The Process Change Method
1. Organize and Prepare
2. Conduct Organizational Scan
3. Establish Technical Working Groups
4. Understand Project’s Current State
5. Redesign the Process
6. Develop Solution
7. Conduct Pilot(s) and Evaluate
8. Facilitate Organizational Learning

A Process Improvement Infrastructure
Core Teams are typically formed and given responsibilities and roles for managing, facilitating, and implementing a change effort from start to finish.

Enablers
- Management Steering Group (MSG)
- Engineering Process Group (EPG)
- Technical Working Groups (TWG)

Facilitators
- Management Steering Group (MSG)
- Engineering Process Group (EPG)
- Technical Working Groups (TWG)

Doers
- Management Steering Group (MSG)
- Engineering Process Group (EPG)
- Technical Working Groups (TWG)

Staffing the Process Infrastructure
My experience with using IDEAL:

• Takes too long (SEI time to move up)
• Costs too much
• Engineers don’t embrace it
• Hard to sell Management Value Proposition
The assimilation gap is the gap between the objective and the deployment.

Interested In?

A streamlined transition approach that provides:

- Compelling Management Value Proposition
  - Predictable Costs
  - Creeping Commitment
  - Quick results with measurable ROI
- Concentrated and Focused process investments
- Accelerated Learning Environment
  - New Processes, New Experiences, New Data, New Beliefs, New Behaviors
- Rapid Predictable Organizational Adoption
- Continually Measurable Results
Major Differences in Approach to Transition

• Concentrated Process
  ➢ Comprehensive Packaged Operational System of Integrated Processes
  ➢ Proven Performance
  ➢ Integrated Operational Measurement System (Individual level)

• Focused Implementation Strategy
  ➢ Unit oriented (Project/Team)
  ➢ JIT Concentrated 3 level Training
  ➢ Accelerated Learning Laboratory
  ➢ Effective Project/Team Launch Process
  ➢ Coaching and continued support
The process elements are adapted to the organization’s process.
Effective Project/Team Launch Process

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 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

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.
Operational Plans Implemented Processes

The TSP Launch Process

1. Randall: Product and Development Goals
2. Assign Roles and What Team Goals
3. Develop the Quality Plan
4. Build Baseline and Test Phase Steps
5. Hold Management Review
6. Hold Portfolio and Consolidated Plans
7. Contact Risk Assessment
8. Prepare Management Launch Report
Launch Postmortem

The TSP launch process produces necessary planning artifacts, e.g., goals, milestones, estimates, test plan, milestones, quality plan, risk mitigation plan, etc. The most important outcome is a committed team.

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

Quality Tracking

Resource Tracking

Earned Value Tracking

TSP Weekly Status Report

Changing Behavior

© 2009 Carnegie Mellon University
Focused Implementation: Building Organizational Capability Project-by-Project, Team-by-Team

EPG identifies gaps and potential improvements, and executes improvement strategies

**Training**  **Launch**

Corporate

Divisions, Departments, or Groups (4)

Projects (20)

Project data, improvement proposals, gaps

Baseline
Does it work for Organizations?

TSP Implements CMMI -2

An organization using TSP has directly addressed or implemented most specific practices (SP):
- 85% of SPs at ML2
- 70% of SPs at ML3
- 54% of SPs at ML4
- 26% of SPs at ML5
- 80% of ML2 and ML3 SPs
- 75% of SPs through ML5

Most generic practices are also addressed.

Based on SAIC/C of the latest version of TSP.

Reliable Products

An analysis of 20 projects in 13 organizations showed TSP teams averaged 0.05 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-TB-014

Organizations Using TSP

Microsoft
Oracle
Softtek
Intuit
BDS
Adobe
Toshiba
Fujifilm
Adobe
IBM
Sun

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.

SEI Average: 6 Years
AV-8B: 2.5 Years

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 are cut from 4 months to 1 month or less.

Non-TSP

TSP

Organizations using TSP report productivity gains of 30% to 80% resulting in lower costs or more functionality in delivered software.
<table>
<thead>
<tr>
<th>Contact</th>
<th>Awareness</th>
<th>Understanding</th>
<th>Trial Use</th>
<th>Adoption</th>
</tr>
</thead>
</table>
| • Conversation  
• Website  
• Article | • Conferences  
• Books  
• Articles  
• Training | • JIT Training Focused on the projects and units implementing the processes (two weeks)  
• Three levels of training  
  • Executive  
  • Team Leader  
  • Practitioner  
• Advanced Learning Laboratory | • Packaged proven whole product Launch Process  
• Supported by a “COACH”  
• Instrumented  
• Implements the Processed learned in the Learning Laboratory on the actual project  
• Coach reinforces discipline throughout the project | • Project Based Rollout Strategy  
• Organizational Commitment  
• Organizational Support (EPG) |
Advanced Learning Laboratory

Training ++
Process Simulation
Individual Instrumentation
Immersion Therapy
Self Discovery

Behavioral modification
Challenge current beliefs
Change Behavior
Change Behavior generates new results
Belief Systems and Behavior

Belief drives behavior

How to change a belief?

Show results inconsistent with the belief
My Beliefs-My Data--My Journey

Think  Change  Improve
Consciousness Model and Bandura Social Learning

EFFORT

Unconsciously Incompetent  Consciously Incompetent  Consciously Competent  Unconsciously Competent
Bayesian Belief networks

Bayesian Inference Model: Allow the use of prior knowledge.

Let $P(h|\xi)$ be a degree of belief in $h$ given current state of information $\xi$.

New evidence $\tilde{e}$ is presented.

Update using Bayes’s Theorem:

$$P(h|\tilde{e},\xi) = \frac{P(h|\xi)P(\tilde{e}|h,\xi)}{P(\tilde{e}|\xi)}$$
Predicting Behavior based on Beliefs

The Technology Acceptance Model is an information systems theory that models how users come to accept and use a technology.

Simplified Acceptance Model based on Beliefs
Repeated for Contact, Awareness, Understanding, Trial use and Institutionalization

Benefit
Work
Continue

Concept of a BBN Model

Contact | Awareness | Understanding | Trial Use | Institution - alization
--- | --- | --- | --- | ---
Benefit | Benefit | Benefit | Benefit | Benefit
Work | Work | Work | Work | Work
Continue | Continue | Continue | Continue | Continue
Using BBN Model to Predict Future

Contact | Awareness | Understanding | Trial Use | Institution – alization
--- | --- | --- | --- | ---
Benefit | Benefit | Benefit | Benefit | Benefit
Work | Work | Work | Work | Work
Continue | Continue | Continue | Continue | Continue

For a particular client at a given point in the adoption curve, knowledge of any of the past or present scores can be used to predict the future scores!
Using BBN Model to Explain Past

For a particular client at a given point in the adoption curve, knowledge of a recent score can be used to explain what the historical, unknown scores most likely were.
Transition Survey

**Awareness:**

*Activity: Executive Seminar/ Team lead training*

PSP will benefit me/my organization:

<table>
<thead>
<tr>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>25%</td>
</tr>
<tr>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>50%</td>
<td>75%</td>
</tr>
<tr>
<td>75%</td>
<td>100%</td>
</tr>
</tbody>
</table>

PSP/TSP will work for me/my organization:

<table>
<thead>
<tr>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>25%</td>
</tr>
<tr>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>50%</td>
<td>75%</td>
</tr>
<tr>
<td>75%</td>
<td>100%</td>
</tr>
</tbody>
</table>

(For Partners) What percentage of clients continue on to understanding:

<table>
<thead>
<tr>
<th>0%</th>
<th>25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>50%</td>
<td>75%</td>
</tr>
<tr>
<td>75%</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Comments:**
Changing Benefit Profile

This distribution of the Benefit score is noticeably moving up across the adoption phases.
This distribution of the Work score is noticeably moving up across the adoption phases.
Changing Continue Profile

This distribution of the Continue score is noticeably moving up across the adoption phases.
Overall Trend of Average Responses

- Benefit
- Work
- Continue

X-axis represents different stages: Contact, Awareness, Understanding, Trial-Use, Institutionalization.

Y-axis represents percentage from 0% to 100%.
Some Initial Linear Models

Contact-Continue-Score = 4.3 + 0.85 * Contact-Work-Score
(Adj-Rsquare = 48%)

Understand-Benefit-Score = 41.1 + 0.49 * Awareness-Benefit-Score
(Adj-Rsquare = 36%)

Although we prefer adjusted Rsquare values in the 80%+ range, these single factor prediction models show promise.

Remember, Adj-Rsquare is the amount of behavior of the outcome explained by the modeling factor.
Questions?
Backup and Reference slides follow
Software Industry Project Performance

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.
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.
Competitive Advantage

As competition in the software industry increases, organizations seek:

- lower development cost
- shorter schedules
- more features per release
- predictable plans
- improved product quality
- fewer customer reported defects
- reduced staff turnover

Team Software Process supports these objectives.
Reliable Estimates

From a study published in 2000

- fifteen projects in four organizations
- CMM ML1, ML2, ML3, and ML5
- TSP improved effort and schedule predictability at all maturity levels

<table>
<thead>
<tr>
<th>Effort (Cost) Performance</th>
<th>Study baseline</th>
<th>TSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+17% to +85%</td>
<td>-25% to +25%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schedule Performance</th>
<th>Study baseline</th>
<th>TSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+27% to +112%</td>
<td>-8% to +20%</td>
</tr>
</tbody>
</table>

Source: CMU/SEI-TR-2000-015
Reliable Products

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
Reduced Rework

TSP System Test Performance Range and Average

- System Test Effort % of Total
- System Test Schedule % of Total
- Failure COQ

Range of a Typical Project

Source: CMU/SEI-TR-2003-014
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 are cut from 4 months to 1 month or less.

Organizations using TSP report productivity gains of 30% to 80% resulting in lower costs or more functionality in delivered software.
“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
Based on a SCAMPI C of the latest version of TSP

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.
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
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.

SEI Average 6 Years

AV-8B 2.5 Years
Organizations Using TSP

Advanced Information Services, Inc.
Centro De Investigacion En Matematicas
Chinasoft International, Inc.
COmputing TechnologieS, Inc.
Davis Systems
DEK International GmbH
Delaware Software, S.A. de C.V.
Delivery Excellence
Grupo Empresarial Eisei, S.A. de C.V.
Herbert Consulting
Hitachi Software Engineering Co., Ltd.
Idea Entity Corp.
InnerWorkings, Inc.
Instituto Tecnologico y de Estudios Superiores de Monterrey
It Era S.A. de C.V.
Kernel Technologies Group, S.A. de CV
Knowledge Partner QR Pvt. Ltd.
Kyushu Institute of Technology
L.G. Electronics
LogiCare
Motiva, LLC
National Aeronautics & Space Administration
Next Process Institute Ltd.
Praxis High Integrity Systems
Process & Project Health Services
Procesix
PS&J Consulting - Software Six Sigma
QuarkSoft
Sandia National Laboratories
Science Applications International Corporation (SAIC)
Siemens AG
SILAC Ingenieria de Software S.A. de C.V.
SKIZCorp Technology
Software Engineering Competence Center (SECC)
Software Park Thailand
STPP, Inc.
TOWA INTEGRADADORA S.A. de C.V.
TRX
Universidad Autonoma De Zacatecas
Universidad de Monterrey
Universidad Regiomotana A.C.
University of Aizu
U.S. Air Force (CRSIP/STSC)
U.S. Census Bureau
U.S. Navy Air Systems Command (NAVAIR)
U.S. Naval Oceanographic Office (NAVO)
Topics

Introduction

TSP concepts

- Self-directed teams and coaching
- Personal Software Process
- Process and measurement framework
- Comprehensive quality management

Team management with TSP

User experience

Getting Started
Key Features -1

Unlike many other software development methods TSP a uses self-directed team management style…the team owns the plan.

TSP has an operationally defined process that is also owned by the team.

The process is supported by an integrated measurement framework to help the team track their work and improve their estimating abilities.

TSP emphasizes quality with comprehensive quality management practices.

- build the right product the right way to avoid rework
- put quality product into test instead of trying to test-in quality
Key Features -2

Complete engineering process – system requirements through acceptance test.

Scalable – small to large organizational settings and projects.

Tailorable – TSP is tailored or is adapted to support existing processes.

Provides immediate and measurable benefits on first use.

Role specific training, documented process, and tools.
Topics

Introduction

TSP Concepts

• Self-directed teams and coaching
  • Personal Software Process
  • Process and measurement framework
  • Comprehensive quality management

Team management with TSP

User experience

Getting Started
Management Styles

The principal management styles have been:

**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.
Sharing the Team Management Responsibilities

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.
The Team Leader’s Role

The team leader does not typically take one of the eight team member roles.

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 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.
The Impact of Self-Directed Teams

A self-directed team

• builds its own plans, negotiating trade-offs with management.
• owns its process and is committed to following it.
• measures and tracks its own work.
• knows precisely where it stands.

Because of this the team members are highly motivated to help each other meet their commitments and achieve their best performance.
Topics

Introduction

TSP Concepts

• Self-directed teams and coaching
• Personal Software Process
• Integrated process and measurement framework
• Comprehensive quality management

Team management with TSP

User experience

Getting Started
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 as 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

Team Software Process
- Teambuilding
- Risk management
- Project planning and tracking

PSP0
- Current process
- Basic measures

PSP1
- Size estimating
- Test report

PSP1.1
- Task planning
- Schedule planning

PSP2
- Code reviews
- Design reviews

PSP2.1
- Design templates

PSP0.1
- Coding standard
- Process improvement proposal
- Size measurement

Introduces process discipline and measurement

Introduces estimating and planning

Introduces quality management and design
PSP Estimating Accuracy

Majority are under-estimating

Balance of over-estimates and under-estimates

Much tighter balance around zero
Compile and 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

Program Number

Mean Minutes Spent Per LOC

Time Invested Per (New and Changed) Line of Code

PSP0

PSP1

PSP2

Design

Code

Compile

Test

298 developers

298 developers

298 developers
Topics

Introduction

TSP Concepts

- Self-directed teams and coaching
- Personal Software Process
- Process and measurement framework
- Comprehensive quality management

Team management with TSP

User experience

Getting Started
TSP Operational Processes and Measures

TSP is defined operationally.

- The processes provide guidance without being too detailed or inflexible.
- They are easily tailored to fit existing organizational processes.
- The measurement definitions are precise but also extensible.

Benefits

- Allows self-directed teams to own their processes.
- Instills process discipline rather than enforcing process institutionalization with auditing methods.
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 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.*
The TSP Development Process

Requirements → High-Level Design → Implementation → System Test

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

Four base measures
Apply to all processes and products
Estimates made during planning
Directly measured by team members while working

Schedule

Effort

Size

Quality

Source: CMU/SEI-92-TR-019
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.
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.
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
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.
What the Base 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, …
Topics

Introduction

TSP Concepts

• Self-directed teams and coaching
• Personal Software Process
• Process and measurement framework
• Comprehensive quality management

Team management with TSP

User experience

Getting Started
Testing Coverage

- Overload
- Hardware failure
- Configuration
- Resource contention
- Operator error
- Data error

Safe and secure region = tested (shaded green)
Unsafe and insecure region = untested (shaded red)
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.

How do you put a quality product into test?

**Quality Management!**
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.
Topics

Introduction

TSP Concepts

- Self-directed teams and coaching
- Personal Software Process
- Process and measurement framework
- Comprehensive quality management

Team management with TSP

User experience

Getting Started
Team Management with TSP

With the TSP measurement framework, teams know exactly where they stand in several dimensions.

- Schedule
- Resources
- Product quality

Teams use the data to

- manage their work
- anticipate and address problems early
- improve cost, schedule, and quality

The teams and their managers use the same data to manage the project as illustrated in the following sample of TSP charts and forms.
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.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline End Date</td>
<td>2/14</td>
</tr>
<tr>
<td>Current Plan End Date</td>
<td>4/25</td>
</tr>
<tr>
<td>Predicted End Date</td>
<td>5/16</td>
</tr>
</tbody>
</table>
# TSP Weekly Status Report

## TSP Week Summary - Form WEEK

<table>
<thead>
<tr>
<th>Name</th>
<th>Team</th>
<th>Date</th>
<th>Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carol</td>
<td>PSP Ghost</td>
<td>4/7/2003</td>
<td></td>
</tr>
<tr>
<td>Status for Week</td>
<td>Week Date</td>
<td></td>
<td>3/10/2003</td>
</tr>
</tbody>
</table>

### Weekly Data

<table>
<thead>
<tr>
<th></th>
<th>Plan</th>
<th>Actual</th>
<th>Plan/Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule hours for this week</td>
<td>151.0</td>
<td>86.0</td>
<td>1.76</td>
</tr>
<tr>
<td>Schedule hours this cycle to date</td>
<td>1526.0</td>
<td>1594.8</td>
<td>0.96</td>
</tr>
<tr>
<td>Earned value for this week</td>
<td>6.9</td>
<td>4.2</td>
<td>1.64</td>
</tr>
<tr>
<td>Earned value this cycle to date</td>
<td>79.5</td>
<td>84.3</td>
<td>0.94</td>
</tr>
<tr>
<td>To-date hours for tasks completed</td>
<td>1580.7</td>
<td>1568.1</td>
<td>1.01</td>
</tr>
<tr>
<td>To-date average hours per week</td>
<td>101.7</td>
<td>106.3</td>
<td>0.96</td>
</tr>
</tbody>
</table>

### Assembly Summary

<table>
<thead>
<tr>
<th>Assembly</th>
<th>Phase</th>
<th>Tasks Completed or Due</th>
<th>Resource</th>
<th>Task Plan Hrs.</th>
<th>Task Actual Hrs.</th>
<th>Earned or Plan Value</th>
<th>Planned Week</th>
<th>Plan vs. Actual Hrs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Form</td>
<td>CODEINSPE</td>
<td>Main Form Code Inspection</td>
<td>SA</td>
<td>1.5</td>
<td>2.4</td>
<td>0.1</td>
<td>10</td>
<td>0.63</td>
</tr>
<tr>
<td>OEMMOO Delivery.aspx UT</td>
<td>OEMMOO Delivery.aspx (FE-Server)</td>
<td>UNK</td>
<td>8.9</td>
<td>3.0</td>
<td>0.5</td>
<td>13</td>
<td>2.91</td>
<td></td>
</tr>
<tr>
<td>OEMMOO Delivery.aspx DLDINSPE</td>
<td>OEMMOO Delivery.aspx (FE-Client)</td>
<td>DINK</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OEMMOO Delivery.aspx CODE</td>
<td>OEMMOO Delivery.aspx (FE-Client)</td>
<td>CNK</td>
<td>7.5</td>
<td>5.7</td>
<td>0.4</td>
<td>14</td>
<td>1.32</td>
<td></td>
</tr>
<tr>
<td>OEMMOO Delivery.aspx CR</td>
<td>OEMMOO Delivery.aspx (FE-Client)</td>
<td>CNK</td>
<td>3.8</td>
<td>1.7</td>
<td>0.2</td>
<td>14</td>
<td>2.26</td>
<td></td>
</tr>
<tr>
<td>OEMMOO Delivery.aspx COMPILE</td>
<td>OEMMOO Delivery.aspx (FE-Client)</td>
<td>CNK</td>
<td>1.3</td>
<td>0.9</td>
<td>0.1</td>
<td>14</td>
<td>1.44</td>
<td></td>
</tr>
<tr>
<td>OEMMOO Delivery.aspx CODEINSPE</td>
<td>OEMMOO Delivery.aspx (FE-Client)</td>
<td>CNK</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OEMMOO Delivery.aspx UT</td>
<td>OEMMOO Delivery.aspx (FE-Client)</td>
<td>UNK</td>
<td>5.9</td>
<td>6.8</td>
<td>0.3</td>
<td>14</td>
<td>0.87</td>
<td></td>
</tr>
<tr>
<td>Query Object</td>
<td>TD</td>
<td>Query Object Test Development</td>
<td>MB</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>14</td>
<td>0.00</td>
</tr>
<tr>
<td>Query Object</td>
<td>CODEINSPE</td>
<td>Query Object Code Inspection</td>
<td>MB</td>
<td>0.0</td>
<td>1.2</td>
<td>0.0</td>
<td>14</td>
<td>0.00</td>
</tr>
<tr>
<td>Query Object</td>
<td>UT</td>
<td>Query Object Unit Test Dialog</td>
<td>MB</td>
<td>14</td>
<td>17</td>
<td>0.4</td>
<td>14</td>
<td>0.66</td>
</tr>
</tbody>
</table>
Quality Tracking

Cumulative Defects Removed by Phase for Assembly SYSTEM

Defect Density by Phase for Assembly SYSTEM

Plan
Actual

% Defect Free

Percent Defect Free

Cumulative Defects Removed by Phase

Phase

Defects/KLOC

Plan
Actual

0.00
10.00
20.00
30.00
40.00
50.00
60.00
70.00
80.00
90.00
100.00

Planning
Requirements
System Test Plan
HLD Inspection
High-Level Design
DLD Review
Integration Test Plan
DLD Inspection
Detailed Design
DLD Review
DLD Inspection
Code Review
Compile
Code Inspection
Unit Test
Build and Integration Test
System Test
Assembly SYSTEM
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 = \( \frac{1}{2} \) design time
3. Code review time = \( \frac{1}{2} \) 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.
Using the Quality Profile
Topics

Introduction
TSP Concepts
Team management with TSP
User experience
Getting Started
The Business Case for TSP

The principal cost 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
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.

<table>
<thead>
<tr>
<th>Microsoft Schedule Results</th>
<th>Non-TSP Projects</th>
<th>TSP Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Released on Time</td>
<td>42%</td>
<td>66%</td>
</tr>
<tr>
<td>Average Days Late</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td>Mean Schedule Error</td>
<td>10%</td>
<td>1%</td>
</tr>
<tr>
<td>Sample Size</td>
<td>80</td>
<td>15</td>
</tr>
</tbody>
</table>
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?*
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
Xerox found that TSP quality management practices reduced the cost of poor quality by finding and removing defects earlier when costs are lower.
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
Topics

Introduction

TSP Concepts

Team management with TSP

User experience

Getting Started
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 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

This cycle is then repeated, increasing scope at a sustainable pace.
The training schedule can be compressed to as short as one month for a faster start.
The gating factor for most organizations is the availability of projects.
SEI recommends training internal coaches as soon as possible.
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
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.

- first SEI leads the effort and internal staff observe
- then internal staff lead and SEI mentors

Training and authorization requirements

- Coach – one week training course, exam, and a launch observation
- Instructor – one week training course and an exam
## Training for Participants

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

Stimulus for Change

Initiating

Set Context

Build Sponsorship

Charter Infrastructure

Characterize Current & Desired States

Develop Recommendations

Set Priorities

Develop Approach

Plan Actions

Establishing

Learning

Propose Future Actions

Analyze and Validate

Implement Solution

Refine Solution

Pilot/Test Solution

Create Solution

Acting

SM IDEAL is a service mark of Carnegie Mellon University.
<table>
<thead>
<tr>
<th>Contact</th>
<th>Awareness</th>
<th>Understanding</th>
<th>Trial Use</th>
<th>Adoption</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Conversation</td>
<td>• Conferences</td>
<td>• Books</td>
<td>• Org Sponsorship (MSG)</td>
<td>• Rollout Strategy</td>
</tr>
<tr>
<td>• Website</td>
<td>• Books</td>
<td>• Classes</td>
<td>• Change Agency (EPG)</td>
<td>• Training</td>
</tr>
<tr>
<td>• Article</td>
<td>• Articles</td>
<td>• Conferences</td>
<td>• Action Teams (PATS)</td>
<td>• Support</td>
</tr>
<tr>
<td></td>
<td>• Training</td>
<td>• Consultants</td>
<td>• New Organizational Processes/Innovation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Pilot Projects</td>
<td></td>
</tr>
</tbody>
</table>
The Technology Acceptance Model is an information systems theory that models how users come to accept and use a technology.