Implementing CMMI for High-Performance with TSP/PSP
Information Systems & Software

• The Information Systems’ usage has experienced an exponential growth over the past years.

• IT Systems are much more software dependent than before.
Cost and Schedule Performance Trends - 2

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.

(This chart represents over 50,000 IT projects in large, medium, and small cross-industry world-wide companies tested by The Standish Group since 1994.)
Software Process Quality

• 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.
Cost to Fix Defects

Source: Construx Software Builders, Inc.
Implementing CMMI with TSP and PSP

- **CMMI** - for organizational capability
- **TSP** - for quality products on cost and schedule
- **PSP** - for individual skill and discipline
CAPABILITY MATURITY MODEL INTEGRATED (CMMI / CMMI)
CMM as a Remedy

- In the late 1980s and early 1990s the SEI developed the Capability Maturity Model (CMM) which captured organizational best practices for software development.
What is CMMI?

• The Purpose of CMM Integration is to provide guidance for improving your organization’s processes and your ability to manage the development, acquisition and maintenance of products and services.
CMMI In a Nutshell

**Staged**
- **Maturity Level 1**
- **Maturity Level 2**
  - REQM, PP, PMC, MA, PPQA, CM, SAM
- **Maturity Level 3**
  - RM, TS, PI, VER, VAL, OPP, OPD, OT, IPM, RSKM, DAR, OEI, IT, ISM
- **Maturity Level 4**
  - OPP, QPM
- **Maturity Level 5**
  - OID, CAR

**Process Areas (SE/SW/IPPD/SS)**
- Requirements Management (REQM)
- Project Planning (PP)
- Project Monitoring and Control (PMC)
- Measurement and Analysis (MA)
- Process and Product Quality Assurance (PPQA)
- Configuration Management (CM)
- Supplier Agreement Management (SAM)
- Requirements Development (RD)
- Technical Solution (TS)
- Product Integration (PI)
- Verification (VER)
- Validation (VAL)
- Organizational Process Focus (OPF)
- Organizational Process Definition (OPD)
- Organizational Training (OT)
- Integrated Project Management (IPM)
- Risk Management (RSKM)
- Decision Analysis and Resolution (DAR)
- Organizational Environment for Integration (OEI)
- Integrated Teaming (IT)
- Integrated Supplier Management (ISM)
- Organizational Process Performance (OPP)
- Quantitative Project Management (QPM)
- Organizational Innovation & Deployment (OID)
- Causal Analysis and Resolution (CAR)

**Continuous**

**Process Management**
- OPF, OPD, OT, OPP, OID

**Project Management**
- PP, PMC, SAM, ISM, IPM, RSKM, QPM, IT

**Support**
- CM, PPQA, MA, CAR, DAR, OEI

**Engineering**
- REQM, RD, TS, PI, VER, VAL
CMMI Staged Representation - 5 Maturity Levels

Level 1: Initial
- Processes are unpredictable, poorly controlled, reactive.

Level 2: Managed
- Processes are planned, documented, performed, monitored, and controlled at the project level. Often reactive.

Level 3: Defined
- Processes are well characterized and understood. Processes, standards, procedures, tools, etc. are defined at the organizational (Organization X) level. Proactive.

Level 4: Quantitatively Managed
- Processes are controlled using statistical and other quantitative techniques. Proactive.

Level 5: Optimizing
- Process performance continually improved through incremental and innovative technological improvements.
Examples

For the Requirements Management Process Area:

An example **Goal** (required):

“Manage Requirements”

An example **Practice** to support the Goal (required):

“Maintain bi-directional traceability of requirements”

Examples (suggested, but not required) of typical **Work Products** might be

Requirements traceability matrix or

Requirements tracking system
CMMI – TR’den Örnekler
Maturity Levels and Performance

• Many people believe that when you achieve a higher CMMI maturity rating that higher performance follows.

• Achieving a higher CMMI maturity rating doesn’t guarantee higher performance.

• The performance you achieve will depend on your implementation of CMMI.
CMMI is a model *not* a process. It describes *characteristics* of effective processes, not the processes themselves.

The trick is to translate the model into a high-performance, high-maturity implementation.

• Also, to do this *effectively* and *efficiently*. 
CMMI Implementation Challenges -1

There can be considerable performance variation in implementations of CMMI practices.

Why?

• The example practices don’t all have equivalent performance, e.g. informal reviews aren’t as effective as formal inspections.

• The same example practice can have substantial variation in performance, e.g. average inspection yield* can range from 30% to 70%.

• Complete, detailed, and timely performance metrics for the process are lacking and “what isn’t measured isn’t managed”.

Remember that satisfying a practice is no guarantee of performance.

*Yield is the percentage of defects found during the inspection
CMMI Implementation Challenges - 2

CMMI is a model of organizational capability and so implementations tend to be focused at the organization level.

There is increasingly less focus on the processes of

– projects and teams
– individual managers and developers

What does organizational performance depend on?

It is a function of the process performance of the projects and teams which is a function of the process performance of the managers and developers.
Process institutionalization is a key implementation challenge.

- coverage – the percentage of the organization that is using the process.
- process fidelity – the degree to which the process is practiced and measured as defined.
Example of impact on performance using inspections.

- **Assume:**
  - Only code inspections are conducted (process design flaw)
  - Only half the modules are inspected (process fidelity issue)
  - Only half of the projects are using the process (problem with coverage) — Inspection yield is only 30% (poor inspection practice)

- Only about 75% of the defects will be found before system test instead of the 95% that would be found if inspections were fully implemented.

- Cost to the organization...about 10 times more defects to find and fix in system test.
Implementing CMMI: Time to Move Up

Number of months move to next maturity level

Recommended time between appraisals

Period of Initial Appraisal

<table>
<thead>
<tr>
<th>Level</th>
<th>1 to 2</th>
<th>2 to 3</th>
<th>3 to 4</th>
<th>4 to 5</th>
<th>1 to 2</th>
<th>2 to 3</th>
<th>3 to 4</th>
<th>4 to 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orgs</td>
<td>12</td>
<td>99</td>
<td>22</td>
<td>16</td>
<td>3</td>
<td>55</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>1 to 2</td>
<td>15</td>
<td>154</td>
<td>28</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Implementing CMMI with TSP and PSP

CMMI - for organizational capability

TSP - for quality products on cost and schedule

PSP - for individual skill and discipline
PERSONAL SOFTWARE PROCESS (PSP)
Birth of PSP

- SEI Fellow Watts Humphrey decided to apply the underlying principles of the CMM to the software development practices of a single developer.

- The result of this effort was the Personal Software Process (PSP).
  - It is designed to be a CMM level 5 process for individual software developers.
What is a Process?

- A process is a defined set of steps for doing a job.
- A process guides your work.
- A process is usually defined for a job that is done multiple times.
- A process provides a foundation for planning.
  - A process is a template, a generic set of steps.
  - A plan is a set of steps for a specific job, plus other things such as effort, costs, and dates.
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 used the data
  – to plan and track their work,
  – Manage the quality of the products they produce
  – Measurably improve performance
PSP Steps

PSP0
- Current process
- Time recording
- Defect recording
- Defect type standard

PSP1
- Size estimating
- Test report

PSP1.1
- Task planning
- Schedule planning

PSP2
- Code reviews
- Design reviews

PSP2.1
- Design templates

TSP
- Team development
Process Planning Framework

1. Customer need
2. Define requirements
   - PROBE method
   - Produce conceptual design
   - Estimate size
   - Estimate resources
   - Product database
   - Productivity database
   - Produce schedule
3. Resources available
4. Develop product
   - Size, resource schedule data
5. Process analysis
6. Management
7. Tracking reports
8. Product delivery
PSP Changes Software Practice

- Software work is planned.
- Plans are based on processes and estimates.
- Estimates are based on historical process data.
- Software work is measured and tracked.
- Status is based on the data.
- Software quality is also planned, estimated, tracked, and managed.
Why Define and Use a Personal Process?

Benefits include

• Consistency
• Efficiency
• Basis for improvement
Using a defined personal process helps you to achieve consistent results.

- Your results are more likely to be similar each time that you use the process.
- Your work becomes more predictable.
Using a defined personal process helps you to be more efficient.

• It structures and guides your work.
  – Orders the steps
  – Avoids rework

• It keeps you focused on what needs to be done.
  – Fewer restarts
  – Manage interrupts

You can accomplish your work in less time.
By gathering data on your work, you can determine which steps

- Take the most time
- Cause you the most trouble
- Are the least effective.

With this information, you can identify opportunities for improving your results by making changes to your process.
PSP Improves Performance

Estimation accuracy
• Fewer underestimates
• More accurate estimates
• Estimates balanced around zero

Quality
• Yield improves by 2x to 3x
• Fewer defects in unit test, integration test, and system test
• COQ is flat or reduced
PSP Reduces Defects

Compile and Test Defects - from PSP Training

- Defects/KLOC vs PSP Assignment Number
- Curves for 1st, 2nd, 3rd, and 4th Quartiles
PSP Improves Productivity
TEAM SOFTWARE PROCESS (TSP)
Implementing CMMI with TSP and PSP

CMMI - for organizational capability

TSP - for quality products on cost and schedule

PSP - for individual skill and discipline
Birth of TSP

• It soon became obvious that, while excellent results were possible using the PSP, it was almost impossible to maintain the discipline required for PSP practices if the surrounding environment did not encourage and demand them.

• Humphrey then developed the Team Software Process (TSP) for the smallest operational unit in most organizations, the project team.
  – TSP was designed to be a CMM level 5 process for project teams.
Working in Teams

Successful teams are both satisfying and rare.

Although many teams come close to meeting their product and business goals, they often do so at the expense of the team members.

A “jelled team” works together smoothly and efficiently.

“A jelled team is greater than the sum of its parts and the enjoyment people derive from the work is greater than you would expect.”

Peopleware, DeMarco & Lister.

TSP is a proven-effective way of building jelled teams quickly.
The Team Software Process (TSP) is a software development process for engineering teams.

TSP is a process-based solution to common software engineering and management issues

• cost and schedule predictability
• productivity and product quality
• process improvement
TSP Teams are Self-Directed

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.
Integrating the “How-to” Technologies

• Used as implementation of CMMI practices, or to augment existing practices to improve performance and achieve higher maturity.
• Introduces self-direct team management style required for high performance teams.
Building High-Performance Teams

TSP builds high-performance teams from the bottom-up.
TSP Process Elements

Requirements
- Launch
- Relaunch
- Relaunch
- Relaunch

High-Level Design
- Requirements
- High-Level Design
- Detail Design (PSP)

Implementation
- Review (PSP)
- Code (PSP)
- Compile (PSP)
- Unit Test (PSP)
- Inspection
- Postmortem

Integration & System Test
- Integration Test
- System Test
- Postmortem
TSP Coverage of CMMI By Maturity Level

- **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.

Bar chart showing the percentage of SPs covered at different CMMI maturity levels.
TSP Improves Effort (Cost) and Schedule Predictability

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></th>
</tr>
</thead>
<tbody>
<tr>
<td>Study baseline</td>
<td>+17% to +85%</td>
</tr>
<tr>
<td>TSP</td>
<td>-25% to +25%</td>
</tr>
</tbody>
</table>

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

Source: CMU/SEI-TR-2000-015
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 System Test Performance Range and Average

Range of a Typical Project

Source: CMU/SEI-TR-2003-014
TSP Accelerates CMMI Improvement

The chart illustrates the percentage of SPs (Software Process) at different CMMI ( Capability Maturity Model Integration) Maturity Levels. The levels are as follows:

- **Level 2**: 70% Not Addressed, 30% Partially Addressed
- **Level 3**: 50% Not Addressed, 50% Partially Addressed
- **Level 4**: 30% Not Addressed, 70% Partially Addressed
- **Level 5**: 10% Not Addressed, 90% Partially Addressed

- **Unrated**: 0%
- **Not Addressed**: 0%
- **Partially Addressed**: 100%
- **Supported**: 0%
- **Directly Addressed**: 0%
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 CMM Level 1 to CMM Level 4. Similar results can be achieved with CMMI.
TSP is a practical, accessible method for achieving the benefits of high maturity process on a much accelerated schedule.

• Several organizations have used the TSP to move up the maturity ladder in much less time than reported averages, including one NAVAIR group that went from level 2 to level 4 in just 16 months largely on the strength of its TSP implementation.
• The results demonstrate that a high-performance CMMI implementation is possible with the use of PSP & TSP.
  – high–performance maturity
  – Predictable and improved cost and schedule
  – Near defect-free quality
  – Satisfied developers, managers and customers
The results demonstrate that a more successful CMMI implementations are possible with the use of PSP & TSP.

SUMMARY

- high –performance maturity
- Predictable and improved cost and schedule
- Near defect-free quality
- Satisfied developers, managers and customers
It’s not just about the visuals, but strengthening
New Picture Effects

Picture Color, Corrections, Artistic Filters, and Background Removal allow you to customize images for your story.
You can do a lot more with movies in PowerPoint 2011.

Movies can be embedded in your presentation. That’s right, no more lost movies!

Also, you can recolor the entire movie or apply a movie style easily. Your movies will look great!
What about the **beautiful transitions** you’ve been seeing?

They are new too.
Share Your Presentation

View your slides from anywhere!
Collaboration

Simultaneous edits on one single document. No more waiting your turn or managing multiple versions.

It’s about team work, and co-authoring will get you there.
Access Anywhere

» Check out the **PowerPoint Web App**

» Access slides wherever you are