Lecture 21: CMMI and the software process
(based in part on material by Dr. Peter Kolb)

Get ready for interesting English...

The plan for performing the organizational process focus process, which is often called "the process-improvement plan," differs from the process action plans described in specific practices in this process area. The plan called for in this generic practice addresses the comprehensive planning for all of the specific practices in this process area, from the establishment of organizational process needs all the way through to the incorporation of process-related experiences into the organizational process assets.

Boeing results

Over/Under Percentage

Without Historical Data

Variance between +20% to -145%

(Weekly level 1 & 2)

(Based on 120 projects in Boeing Information Systems)


Increased productivity and quality

Productivity Rate and Quality Performance

- For Software Programs

Percent productivity increased by 80% as error rates decreased.
CMMI goals

Emphasis on developing processes and changing culture for measurable benefit to organization’s business objectives

Framework from which to organize and prioritize engineering, people, and business activities

Supports coordination of multi-disciplined activities required to build successful product or application

Adds “Engineering Systems Thinking”

What is a CMM?

Capability Maturity Model:
A reference model of mature practices in a specified discipline, used to assess a group’s capability to perform that discipline

CMMs differ by:
- Discipline (software, systems, acquisition, etc.)
- Structure (staged versus continuous)
- How Maturity is Defined (process improvement path)
- How Capability is Defined (institutionalization)

NOT:
- Ready-made scheme or template for describing processes
- Methods for the processes

Bridging the divide

Integrates systems and software disciplines into one process improvement framework.

CMMI-SE/SW/IPPD/SS, V1.1
- Systems Engineering
- Software Engineering
- Integrated Product and Process Development
- Supplier Sourcing

Provides a framework for introducing new disciplines as needs arise.

The CMM Explosion

The first CMM (CMM v1.0) was developed for software and released in August 1991

Based on this success and the demand from other interests CMMs were developed for other disciplines and functions
- Systems Engineering
- People
- Integrated Product Development
- Software Acquisition
- Software Quality Assurance
- Measurement
- Others……

ISO 9001:2000 vs CMMI

ISO 9001:2000
- No explicit requirements for
  - Institutionalization
  - Creating and maintaining organizational process assets
  - Database of good and best practices
  - Misses details for the following process areas
    - Organizational Training (Lvl 3)
    - Risk Management (Lvl 3)
    - Decision Analysis and Resolution (Lvl 3)
    - Organization Process Performance (Lvl 4)
    - Quantitative Project Management (Lvl 4)
    - Organization Innovation and Deployment (Lvl 5)
    - Causal Analysis (Lvl 5)
Support of CMMI for ISO 9001:2000

Organizations at the CMMI Maturity Level 3 will be ready for ISO 9001:2000 registration with minor adjustments.

Organizations registered as ISO 9001:2000 compliant will require additional effort to reach the CMMI Level 2 or 3.

- The CMMI path leverages the investment an organization may have in ISO 9001.
- Provides additional benefits especially in institutionalizing the engineering discipline.
- Takes an organization to the quantitative management level of process improvements.

Model Representations

Management visibility by maturity level

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Process Characteristics</th>
<th>Management Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizing</td>
<td>Focus on continuous quantitative improvement</td>
<td>In</td>
</tr>
<tr>
<td>Quantitatively Managed</td>
<td>Process is measured and controlled</td>
<td>In, Out</td>
</tr>
<tr>
<td>Defined</td>
<td>Process is characterized for the organization and is proactive</td>
<td>In, Out</td>
</tr>
<tr>
<td>Managed</td>
<td>Process is characterized for projects and is often reactive</td>
<td>In, Out</td>
</tr>
<tr>
<td>Initial</td>
<td>Process is unpredictable, poorly controlled, and reactive</td>
<td>In, Out</td>
</tr>
</tbody>
</table>

Capability levels are cumulative

Because capability levels build upon one another, there can be no gaps.

Structure of the CMMI Staged Representation

Generic goals

- **Commitment to Perform**: Creates policies and secures sponsorship for process improvement efforts.
- **Ability to Perform**: Ensures that the project/organization has the resources it needs to pursue process improvement.
- **Directing Implementation**: Collects, measures, and analyzes data related to processes.
- **Verification**: Verifies that the projects and/or organization's activities conform to requirements, processes, and procedures.
CMMI terminology

Institutionalization

- Ensure the process areas are effective, repeatable and long lasting
- Provide needed infrastructure support
- Ensure processes are defined, documented, understood
- Enable organizational learning to improve the processes

Establish and Maintain

- This phrase connotes a meaning beyond the component terms; it includes documentation and usage.
- Work product
  - The term "work product" is used throughout the CMMI Product Suite to mean any artifact produced by a process. These artifacts can include files, documents, parts of the product, services, processes, specifications, and invoices.
- Planned Process
  - A process that is documented both by a description and a plan. The description and plan should be coordinated, and the plan should include standards, requirements, objectives, resources, assignments, etc.

CMMI terminology

Performed Process (Capability Level 1)

> A process that accomplishes the needed work to produce identified output work products using identified input work products. The specific goals of the process area are satisfied.

Managed Process (Capability Level 2)

> A "managed process" is a performed process that is planned and executed in accordance with policy, employs skilled people having adequate resources to produce controlled outputs; involves relevant stakeholders; is monitored, controlled, and reviewed; and is evaluated for adherence to its process description.

Defined Process (Capability Level 3)

> A "defined process" is a managed process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines; has a maintained process description; and contributes work products, measures, and other process improvement information to the organizational process assets.

Examples

The purpose of Integrated Supplier Management is to proactively identify sources of products that may be used to satisfy the project's requirements and to manage selected suppliers while maintaining a cooperative project-supplier relationship.
Examples

The purpose of Organizational Process Definition is to establish and maintain a usable set of organizational process assets.

(Organizational process asset: "Anything that the organization considers useful in attaining the goals of a process area.")

Examples

The purpose of Organizational Process Focus is to plan and implement organizational process improvement based on a thorough understanding of the current strengths and weaknesses of the organization's processes and process assets.

Process capability prediction

<table>
<thead>
<tr>
<th>Level</th>
<th>Process Characteristics</th>
<th>Predicted Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizing</td>
<td>Focus is on continuous quantitative improvement</td>
<td></td>
</tr>
<tr>
<td>Quantitatively Managed</td>
<td>Process is measured and controlled</td>
<td></td>
</tr>
<tr>
<td>Defined</td>
<td>Process is characterized for the organization and is proactive</td>
<td></td>
</tr>
<tr>
<td>Managed</td>
<td>Process is characterized for projects and is often reactive</td>
<td></td>
</tr>
<tr>
<td>Initial</td>
<td>Process is unpredictable, poorly controlled, and reactive</td>
<td></td>
</tr>
</tbody>
</table>

People implications

<table>
<thead>
<tr>
<th>Level</th>
<th>Process Characteristics</th>
<th>People Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizing</td>
<td>Focus is on continuous quantitative improvement</td>
<td>Focus on &quot;fire prevention&quot;; improvement anticipated and desired, and impacts assessed</td>
</tr>
<tr>
<td>Quantitatively Managed</td>
<td>Process is measured and controlled</td>
<td>Sense of teamwork and interdependencies</td>
</tr>
<tr>
<td>Defined</td>
<td>Process is characterized for the organization and is proactive</td>
<td>Increased reliance on defined process; investment in people and process as corporate assets</td>
</tr>
<tr>
<td>Managed</td>
<td>Process is characterized for projects and is often reactive</td>
<td>Overreliance on experience of good people - when they go, the process goes</td>
</tr>
<tr>
<td>Initial</td>
<td>Process is unpredictable, poorly controlled, and reactive</td>
<td>Focus on &quot;fire fighting&quot;; effectiveness low – frustration high</td>
</tr>
</tbody>
</table>

Risk implications

<table>
<thead>
<tr>
<th>Level</th>
<th>Process Characteristics</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizing</td>
<td>Focus is on continuous quantitative improvement</td>
<td></td>
</tr>
<tr>
<td>Quantitatively Managed</td>
<td>Process is measured and controlled</td>
<td></td>
</tr>
<tr>
<td>Defined</td>
<td>Process is characterized for the organization and is proactive</td>
<td></td>
</tr>
<tr>
<td>Managed</td>
<td>Process is characterized for projects and is often reactive</td>
<td></td>
</tr>
<tr>
<td>Initial</td>
<td>Process is unpredictable, poorly controlled, and reactive</td>
<td></td>
</tr>
</tbody>
</table>

Specific and generic goals and practices

- **Total Quality Management (TQM)**: The overarching goal is to continuously improve the quality of products and services, and to eliminate defects and waste. TQM involves everyone in the organization and is driven by leadership and customer satisfaction.

- **Six Sigma** (SS): A data-driven methodology that focuses on reducing variability and defects in processes. SS uses a set of metrics and tools to identify and solve problems, and it is widely used in manufacturing, healthcare, and other industries.

- **Customer Focus** (CF): The goal is to satisfy and delight customers by understanding their needs and continuously improving the products and services to meet or exceed their expectations. CF involves gathering customer feedback and using it to drive improvements.

- **Process Improvements** (PI): The focus is on identifying and eliminating non-value-adding activities (Waste). PI involves process mapping, root cause analysis, and process improvement techniques like Lean and Six Sigma.

- **Product Improvements** (PI): The goal is to continuously improve the quality and functionality of products by identifying and solving problems at the design and development stage. PI involves using tools like FMEA, DFMEA, and Design for Six Sigma (DFSS).

- **Supplier Quality** (SQ): The objective is to ensure that suppliers meet or exceed the organization's quality expectations. SQ involves supplier selection, quality management, and continuous improvement.

- **Employee Focus** (EF): The aim is to create a supportive work environment that empowers employees and encourages them to continuously improve their skills and performance. EF involves training, development, and recognition programs.

- **Management Responsibilities** (MR): The role of top management is to set the vision, establish the organizational context, and ensure alignment. MR involves setting strategic goals, defining the organizational context, and communicating expectations.
**Generic goals and practices**

<table>
<thead>
<tr>
<th>Capability Level</th>
<th>Generic Goals</th>
<th>Generic Practices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Achieve Specific Goals</td>
<td>GP 1.1: Perform Base Practices</td>
</tr>
<tr>
<td>2</td>
<td>Institutionalize a Managed Process</td>
<td>GP 2.1: Establish an Organizational Policy, GP 2.2: Plan the Process, GP 2.3: Provide Resources, GP 2.4: Assign Responsibility, GP 2.5: Train People, GP 2.6: Manage Configurations, GP 2.7: Identify and Involve Relevant Stakeholders, GP 2.8: Monitor and Control the Process, GP 2.9: Objectively Evaluate Adherence, GP 2.10: Review Status with Higher Level Management</td>
</tr>
<tr>
<td>3</td>
<td>Institutionalize a Defined Process</td>
<td>GP 3.1: Establish a Defined Process, GP 3.2: Collect Improvement Defects</td>
</tr>
<tr>
<td>4</td>
<td>Institutionalize a Quantitatively Managed Process</td>
<td></td>
</tr>
</tbody>
</table>

**Generic practices**

The Generic Practices support institutionalization of critical practices for an organization to have a successful process improvement initiative:

- Processes will be executed and managed consistently
- Processes will survive staff changes
- Process improvement will be related to business goals
- The organization will not find itself continuously "reinventing the wheel"
- There will be the commitment to provide resources or infrastructure to support or improve the processes
- There will be historical basis for cost estimation

**For More Information About CMMI**

- Go to CMMI Website:
  - [http://sei.cmu.edu/cmmi](http://sei.cmu.edu/cmmi)
  - [http://seir.sei.cmu.edu/seir/](http://seir.sei.cmu.edu/seir/)
  - [http://jo.sei.cmu.edu/pub/english.cgi/O/323123](http://jo.sei.cmu.edu/pub/english.cgi/O/323123)
  - [http://dtic.mil/ndia](http://dtic.mil/ndia) (first annual CMMI Conference)
  - [http://www.fas.gov/aio](http://www.fas.gov/aio)

- Assistance for government organizations:
  - SW-CMM v1.1 to CMMI v1.1 Mappings
  - Software Technology Support Center
  - [http://www.stsc.hill.af.mil](http://www.stsc.hill.af.mil)

**CMMI**

Defines goals and practices shown to be useful to the software industry:

- Primarily directed to large organizations
- Focus on process: explicit, documented, reproducible, measurable, self-improving
- Essential to outsourcing industry
- Technology-neutral

**TSP, PSP**

Transposition to work of individual teams and developers

**Management support**

The initial TSP objective is to convince management to let the team be self-directed, meaning that it:

- Sets its own goals
- Establishes its own roles
- Decides on its development strategy
- Defines its processes
- Develops its plans
- Measures, manages, and controls its work
Management support

Management will support you as long as you:
- Strive to meet their needs
- Provide regular reports on your work
- Convince them that your plans are sound
- Do quality work
- Respond to changing needs
- Come to them for help when you have problems

Management support

Management will agree to your managing your own work as long as they believe that you are doing a superior job.

To convince them of this, you must:
- Maintain and publish precise, accurate plans
- Measure and track your work
- Regularly show that you are doing superior work

The PSP helps you do this

PSP essential practices

- Measure, track, and analyze your work
- Learn from your performance variations
- Incorporate lessons learned into your personal practices

What does a PSP provide?

A stable, mature PSP allows you to
- Estimate and plan your work
- Meet your commitments
- Resist unreasonable commitment pressures

You will also
- Understand your current performance
- Improve your expertise as a professional

PSP fundamentals

As a personal process, PSP includes:
- Defined steps
- Forms
- Standards
- A measurement and analysis framework for characterizing and managing your personal work
- A defined procedure to help improve your personal performance

The PSP process flow

Requirements
Planning
Design
Code
Compile
Test
Yes
Postmortem
Finished product
Log

Project and process data summary report
A progressive approach

PSP is introduced in six upward-compatible steps

At each step:
- Write one or more modules
- Gather and analyze data on your work
- Use results to improve your personal performance

Goals at each level

PSP0: Establish a measured performance baseline
PSP1: Make size, resource, and schedule plans
PSP2: Practice defect and yield management

PSP0 setup

PSP0 is a simple, defined, personal process:
- Make a plan
- Use your current design and development methods to produce a small program
- Gather time and defect data on your work
- Prepare a summary report

The six phases of PSP0

1. Plan
2. Design
3. Code
4. Compile
5. Test
6. Postmortem

Produce plan for developing program from requirements
Produce design specification for the program
Turn design into executable code
Translate into executable code
Verify that code satisfies requirements
Summarize & analyze project data

The steps

TSP
PSP
PSP1
PSP2
PSP2.1
PSP0
PSP0.1

Team development
Size estimating
Test report
Code reviews
Design reviews
Design templates
Task planning
Team development
Task planning

Objective:
- Demonstrate use of defined process for small programs
- Incorporate basic measurements in process
- Minimize changes to your personal practices

PSP0

Phase order

PSP looks like waterfall but is not.

Phase order is determined by dependencies:

- Can't test code before it's compiled
- Can't compile before it's written
- Can't use design if produced after code is written
- No reason to make a plan after you're done

Conclusion: start here with a plan.

Cyclic process flow

Programs that are large or programs or not well understood may require an iterative approach.

In this example, each module is separately coded, compiled, and tested.

The example uses PSP0 phases and 2 code-compile-test cycles.

Process measurement

To be useful, measurements should be:
- Gathered for a specific purpose
- Explicitly defined
- Properly managed
- Properly used

We measure to:
- Understand and manage change
- Predict or plan
- Compare one product, process, or organization with another
- Determine adherence to standards
- Provide a basis for control

Measurement objectives

Measurements only produce numbers.

To be useful, they must:
- Relate to business objectives
- Be properly interpreted
- Lead to appropriate action

If the business purposes for the measurements are not understood:
- The wrong data may be gathered
- Data may not be properly used
PSP measurements

Basic PSP data:
- Program size
- Time spent by phase
- Defects found and injected by phase

On every item, gather both actual and estimated data

Measures derived from these data:
- Support planning
- Characterize process quality

Estimating with PROBE

Stands for PROxy Based Estimating

Uses proxies to estimate program size and development time

A good proxy helps make accurate estimates

The PROBE estimating method

Start → Conceptual design

Conceptual design relates the requirements to the parts needed to produce the program

Parts categories:
- Reused: Can be used as-is
- Base: Exists, requires modifications
- Added: needs to be developed

Sizing parts

Reused part: Use actual size

Added part: define proxy
- Identify part type, e.g. parsing, GUI, network...
- Estimate number of items, e.g. routines
- Estimate relative size, i.e. very small, small, medium, large, or very large
- Find size of an item of this part type and relative size in the relative size table
- Estimated size = item size × number of items

Base part: start from actual size; estimate additions, deletions, modifications
**Objective:** introduce & practice methods for
- Making resource & schedule plans
- Tracking your performance against them
- Judging likely project completion dates

**PSP1.1**

Two new process elements:
- Task planning template
- Schedule planning template
Typically used for projects that take several days or weeks

**PSP2**

Objective: introduce
- Design & code reviews
- Methods for evaluating and improving quality of reviews

Two key capabilities added at this level:
- Design planning checklist
- Code review checklist

**Quality planning**

PSP2 introduces quality planning. This involves estimating:
- Total number of defects that will be injected
- Number of defects injected & removed in each process phase
- Amount of time for design and code reviews

& adjusting these parameters to ensure high-quality result

**Arguments for reviews over tests**

In testing, you must:
- Detect unusual behavior
- Figure out what the test program was doing
- Find where the problem is in the program
- Figure out which defect could cause such behavior

This can take a lot of time

With reviews you:
- Follow your own logic
- Know where you are when you find a defect
- Know what the program should do, but did not
- Know why this is a defect
- Are in a better position to devise a correct fix

**Code reviews**

General principles (not specifically from PSP):
- Uncoupled from evaluation process
- Meeting must have chair, secretary
- Chair is not supervisor
- Purpose is to identify faults
- Purpose is not to correct them
- Purpose is not to evaluate developer; keep focus technical
- Strict time limit (e.g. 2 hours)
- Announced sufficiently long in advance
- Participant number: 5 to 10
- Code available in advance, as well as any other documents
- Meeting must be conducted professionally and speedily; chair keeps it focused
Code review checklist

Reviews are most effective with personal checklist customized to your own defect experience:

- Use your own data to select the checklist items
- Gather and analyze data on the reviews
- Adjust the checklist with experience

Do the reviews on a printed listing, not on screen

The checklist defines steps and suggests their order:

- Review for one checklist item at a time
- Check off each item as you complete it

Design review principles

In addition to reviewing code, you should also review your designs

Requirements that you

- Produce designs that can be reviewed
- Follow an explicit review strategy
- Review design in stages
- Verify that logic correctly implements requirements

PSP2.1

Objective: introduce

- Additional measures for managing process quality
- Design templates that provide an orderly framework and format for recording designs

New process elements:

- Design review script
- Design review checklist
- Operational specification template
- Functional specification template
- State specification template
- Logic specification template

PSP: an assessment

Ignore technology assumptions (strict design-code-compile-test cycle) which is not in line with today’s best practices. Retain emphasis on professional engineer’s approach:

- Plan
- Record what you do both qualitatively and quantitatively:
  - Program size
  - Time spent on parts and activities
  - Defects
- Think about your personal process
- Improve your personal process

Tool support, integrated in development environment, is essential.