Software Process Assessment
Using SEI’s Software Capability Maturity Model

Neal S. Coulter
College of Computing, Engineering, and Construction
University of North Florida
Can Software Development Practices Improve Dramatically? - 1

- Hughes Aircraft spent $445,000 from 1987-1990 and realized $2,000,000 annual savings.

- Schlumberger completed 94% of engineering projects on schedule in 1992, compared with 89% in 1991 and 51% in 1990. In 1989, 25% of total product defects were discovered by the customer; in 1991, 10% were customer reported.

• Raytheon realized $7.70 for every dollar invested and a two-fold increase in productivity and hired 100 additional staff members to meet the demand for new business.
• Raytheon realized a 190% increase in SLOC per person-month from 1988-1996; defect density dropped from 17.2 per KLOC of delivered code to 4.0 per KLOC. In two years rework costs dropped from 41% of project cost to 20%. Cost overruns for projects dropped form 40% in 1988 to 3% in 1991.
What Caused these Improvements?

Hughes, Raytheon, Motorola, Schlumberger and others reported such changes after assessing and modifying its software development process using the Software Engineering Institute’s (SEI) Software Capability Maturity Model (CMM).

Hence, CMM could be the reason for the improvements!
What’s In It for the Developers?

Less stress
  “The only ones questioning the value of Level 2 are those who have not achieved it”
  “A coherent culture exists at Level 3”

Better morale

Less turnover

Less overtime

Job security
The Software Engineering Institute

• Operated by the US Department of Defense (DoD)
• Opened in 1984 in Pittsburgh
• Affiliated with Carnegie Mellon University
• Mission is to provide leadership in advancing the state of practice of software engineering to improve the quality of systems that depend on software
Motivation for an SEI Software Process Model - 1

A review of 17 major DoD software contracts found that the average 28-month schedule was missed by 20 months; no project was on time. The $58 billion A12 aircraft program was canceled mainly due to software problems.

Many similar stories are common, of course.
Motivation for an SEI Software Process Model - 2

DoD concluded “few fields have so large a gap between current best practices and average current practice...today’s major problems with military software development are not technical problems but management problems... the understanding of software as a product and software development as a process is not keeping pace with growing complexity and software complexity...”
In November 1986, SEI and Mitre began developing a process maturity framework. In 1987, a brief description of the model was released by its primary architect, Watts Humphrey.

Methods were developed for
  Software process assessments (SPA)
  Software capability evaluation (SCE)
SPAs are intended for an internal process improvement program

SCEs are used to appraise contractors qualified to perform work

A questionnaire was developed to assist in evaluating organizations’ abilities to develop software.
In 1991, SEI evolved the maturity framework into the Capability Maturity Model for Software CMM, now Version 1.1.

The CMM
- Is based on actual practices
- Reflects the best state of the practice
- Reflects the needs of individuals
- Is documented
- Is publicly available (http://www.sei.cmu.edu)
Benefits and Risks of Model-Based Improvement - 1

Benefits
• Provides a common language
• Supports measurement
• Based on collaboration with hundreds of professionals
• Represents a consensus, but not total agreement
Benefits and Risks of Model-Based Improvement - 1

Risks

• All models are wrong; some models are useful” - George Box
• Not comprehensive; it barely touches on some nonprocess factors, such as people and technology
• Does not address application domains, advocate specific technologies, or suggest how to hire, train, and retain people.
CMM and TQM

CMM is built on Total Quality Management (TQM) principles. The goal of TQM is to meet the goals of the customer, now and in the future. CMM does not state the customer should be satisfied (or delighted) with the software product. It does state that the software supplier should build products that satisfy the customer’s needs as documented in the requirements allocated to the software component of the total system or product being supplied. Software development is usually on part of a business relationship.
The Basic CMM Model

- **INITIAL 1**: Disciplined process
- **REPEATABLE 2**: Standard, consistent process
- **DEFINED 3**: Predictable Process
- **MANAGED 4**: Continuously improving process
- **OPTIMIZING 5**: Maturity Levels

- M
- A
- T
- U
- R
- I
- T
- Y
- L
- E
- V
- E
- L
- S
Some Basic Definitions - 1

*Process* - A sequence of steps performed for a given purpose. What you do.

*Software process* - A set of activities, methods, practices and transformations that people employ to develop and maintain software products.

*Software process capability* - Range of expected results by following a certain software process.

*Software process performance* - The actual results achieved by following a software process.
Some Basic Definitions - 2

*Software process maturity* - The extent to which a specific process is defined, measured, controlled and effective. Maturity implies a potential for growth in capability and indicates both the richness of an organization’s process and the consistency of its application across projects.
Maturity level - A well-defined evolutionary plateau toward achieving a mature software process. Each maturity level comprises a set of process goals that, when satisfied, stabilize an important part of the software process. Achieving each level of the maturity framework establishes a different component in the software process, resulting in an increase in the process capability of the organization.
CMM, Productivity, Quality, and Risk

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Optimizing</td>
<td>Productivity and Quality</td>
</tr>
<tr>
<td>4 Managed</td>
<td>Risk</td>
</tr>
<tr>
<td>3 Defined</td>
<td></td>
</tr>
<tr>
<td>2 Repeatable</td>
<td></td>
</tr>
<tr>
<td>1 Initial</td>
<td></td>
</tr>
</tbody>
</table>
CMM Level Summaries - 1

1. Initial - The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined and success depends on individual efforts and heroics.

2. Repeatable - Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
3. **Defined** - The software process for both management and engineering activities is documented standardized, and integrated into a standard software process for the organization. Called the organization’s *standard software process*. 
4. **Managed** - Detailed measures of the software process and product quality are collected and used for analysis and control.

5. **Optimizing** - Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.
So, Everyone but Us is Level 5, Right?

According to SEI’s reportable data, of the 379 organizations at 99 companies that are advanced enough to have process improvement programs in place and have conducted SEI maturity assessments, 73% don’t rate higher than Level 1.

The others have not applied.
# Advancing Through CMM Levels - 1

<table>
<thead>
<tr>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
<th>LEVEL 3</th>
<th>LEVEL 4</th>
<th>LEVEL 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PROCESS</strong></td>
<td>Few stable processes exist or are used</td>
<td>Documented and stable estimating, planning, and commitment processes at the project level</td>
<td>Integrated management and engineering processes are used across the organization</td>
<td>Processes are quantitatively understood and stabilized</td>
</tr>
<tr>
<td><strong>“Just do it”</strong></td>
<td>Problems are recognized and corrected as they occur</td>
<td>Problems are anticipated and prevented, or their impacts are minimized</td>
<td>Sources of individual problems are understood and eliminated</td>
<td>Common sources of problems are understood and eliminated</td>
</tr>
</tbody>
</table>
## Advancing Through CMM Levels - 2

<table>
<thead>
<tr>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
<th>LEVEL 3</th>
<th>LEVEL 4</th>
<th>LEVEL 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PEOPLE</strong></td>
<td>Success depends on individual heroics</td>
<td>Success depends on individuals; management system supports</td>
<td>Project groups work together, perhaps as an integrated product team</td>
<td>Strong sense of teamwork exists within each project</td>
</tr>
<tr>
<td>“Fire fighting” is a way of life</td>
<td>Commitments are understood and managed</td>
<td>Training is planned and provided according to roles</td>
<td></td>
<td>Everyone is involved in process improvement</td>
</tr>
<tr>
<td>Relationships between disciplines are uncoordinated, perhaps even adversary</td>
<td>People are trained</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Advancing Through CMM Levels - 3

<table>
<thead>
<tr>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
<th>LEVEL 3</th>
<th>LEVEL 4</th>
<th>LEVEL 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>Technology supports</td>
<td>New technologies are</td>
<td>New technologies are</td>
<td>New technologies are</td>
</tr>
<tr>
<td>Introduction of new technology is risky</td>
<td>established, stable processes</td>
<td>evaluated on a qualitative basis</td>
<td>evaluated on a quantitative basis</td>
<td>pro-actively pursued and deployed</td>
</tr>
</tbody>
</table>
# Advancing Through CMM Levels - 4

<table>
<thead>
<tr>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
<th>LEVEL 3</th>
<th>LEVEL 4</th>
<th>LEVEL 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M</strong></td>
<td><strong>E</strong></td>
<td><strong>A</strong></td>
<td><strong>S</strong></td>
<td><strong>U</strong></td>
</tr>
<tr>
<td><strong>E</strong></td>
<td><strong>M</strong></td>
<td><strong>A</strong></td>
<td><strong>S</strong></td>
<td><strong>U</strong></td>
</tr>
<tr>
<td><strong>A</strong></td>
<td><strong>S</strong></td>
<td><strong>U</strong></td>
<td><strong>M</strong></td>
<td><strong>E</strong></td>
</tr>
<tr>
<td><strong>S</strong></td>
<td><strong>U</strong></td>
<td><strong>M</strong></td>
<td><strong>E</strong></td>
<td><strong>A</strong></td>
</tr>
<tr>
<td><strong>U</strong></td>
<td><strong>M</strong></td>
<td><strong>E</strong></td>
<td><strong>A</strong></td>
<td><strong>S</strong></td>
</tr>
<tr>
<td><strong>M</strong></td>
<td><strong>E</strong></td>
<td><strong>A</strong></td>
<td><strong>S</strong></td>
<td><strong>U</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
<th>LEVEL 3</th>
<th>LEVEL 4</th>
<th>LEVEL 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement</td>
<td>Measurement</td>
<td>Measurement</td>
<td>Measurement</td>
<td>Measurement</td>
</tr>
<tr>
<td>Data collection and analysis is ad hoc</td>
<td>Planning and management data used by individual projects</td>
<td>Data are collected and used in all defined processes</td>
<td>Data definition and collection are standardized across the organization</td>
<td>Data are used to evaluate and select process improvements</td>
</tr>
<tr>
<td>Data are systematically shared across projects</td>
<td>Data are used to understand the process quantitatively and stabilize it</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Skipping Levels

Trying to skip maturity levels is counterproductive because each level is a necessary foundation for the next level. Processes without a proper foundation fail at the same time they are needed most--under stress--and provide no basis for future improvement.

Many developers have collected data that is characteristic of Level 4 organizations only to find they can not use it. However, some attributes of higher-level organizations, such as an SEPG, can be helpful in moving up levels.
Process Capability by Maturity Level

A process is under statistical control when repeating the work in roughly the same way will produce roughly the same results.
Internal Structure of Maturity Levels

Each maturity level has constituent parts.
Some Maturity Level Components - 1

A key process area (KPA) is a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability. To achieve a maturity level, the key processes for that level (and the lower levels) must be satisfied and the processes must be institutionalized.

Institutionalization is the building of a culture and infrastructure to support the methods, practices, policies, and procedures so that they are the ongoing way of doing business. The infrastructure is the underlying framework of an organization or system.
Some Maturity Level Components - 2

The *goals* of each KPA summarize its key practices and can be used in determining whether an organization has effectively implemented the KPA. The goals signify the scope, boundaries, and intent of each KPA.

Each KPA is described in terms of *key practices*. Key practices describe the activities and infrastructure that contribute most to the effective implementation and institutionalization of the KPAs. The key practices describe “what” is to be done, but do not mandate “how” the process should be implemented.
Components of Key Practices

The practices that describe the KPAs are organized by common features that indicate if the implementation and institutionalization of a KPA are effective, repeatable, and lasting. The five common features are:

• Commitment to perform - Policies, sponsorship...
• Ability to perform - Resources, training...
• Activities performed - Plans, procedures, tracking...
• Measurement and analysis - Sample measurements...
• Verifying implementation - Reviews, audits...
KPAs by Maturity Level - 1

**Initial**
- Organization process focus
- Organization process definition
- Training program
- Integrated software management
- Software product engineering
- Intergroup communication
- Peer reviews

**Repeatable**
- Requirements management
- Software project planning
- Software tracking and oversight
- Software subcontract management
- Software quality assurance
- Software configuration management

**Defined**
- Quantitative process management
- Software quality management

**Managed**
- Defect prevention
- Technology change management
- Process change management

**Optimizing**
KPAs by Maturity Level - 2

Level 1 - No KPAs

Level 2 - Focuses on establishing basic project-management controls.

Level 3 - Addresses both project and organizational issues, as an organization establishes an infrastructure that institutionalizes effective software engineering and management processes across all projects.
KPAs by Maturity Level - 3

Level 4 - Focuses on establishing a quantitative understanding of both the software process and the software work products being built.

Level 5 - Focuses on issues that both organizations and projects must address to implement continuous and measurable process improvement.
KPA Template - 1

<Key Process Area X>

a key process for level n: <Level name>

The purpose of <Key Process Area X> is <statement>

<Key Process Area X> involves <summary> <Additional elaboration on Key Process X as appropriate>
KPA Template - 2

Goals

Goal 1  <Process summary statement as goal>
...
...

Commitment to Perform

Commitment 1  The project follows a written organizational policy for <X>
...
...
KPA Template - 3

Ability to Perform

Ability 1  A group that is responsible for <X> exists

...  

Activities Performed

Activity 1  <Activity performed in Key Process Area X> according to a documented procedure

...
KPA Template - 4

Measurement and Analysis

Measurement 1  Measurements are made to determine the status of the activities for <X>

...  

Verifying Implementation

Verification 1  The activities for <X> are reviewed with senior management on a periodic basis
A key process area for Level 2: Repeatable

The purpose of Software Configuration Management is to establish and maintain the integrity of products of the software project throughout the project’s software life cycle.

Goals
Goal 1 Software configuration management activities are planned
SCM Template - 2

Commitment to Perform
Commitment 1 The project follows a written organizational policy for implementing SCM

Ability to Perform
Ability 1 A board having the authority for managing the project’s baseline exists or is established
SCM Template - 3

Activities Performed
Activity 1  A SCM plan is prepared for each software project according to a documented procedure

... 

Measure and Analysis
Measurement 1  Measurements are made to determine the status of the SCM activities

...
SCM Template - 4

Verifying Implementation
Verification 1  The SCM activities are reviewed with senior management on a periodic basis

...
Complete assessment is a detailed task that requires organization wide participation. It consists of the phases:

- Select a team
- Complete the maturity questionnaire
- Analyze the responses
- Conduct a detailed site visit
- Produce a list of findings of strengths and weaknesses
- Prepare a key process area profile of the organization
CMM-Based Assessments - 2

1. Team Selection
2. Maturity Questionnaire
   - samples the CMM
3. Response Analysis
4. On-Site Visit
   - Interviews and document reviews
5. Findings
   - based on the CMM
6. KPA Profile
Maturity Questionnaire

• Based on CMM Version 1.1
• Organized by CMM KPAs
• Covers all 18 KPAs
• Addresses each KPA goal but not all key practices
• 6 to 8 questions per KPA
• Can be completed in one hour
• No standard scoring scheme
• Identifies issues to be explored further

Each question can be answered:
   Yes, No, Does not Apply, Don’t Know
Also place for comments
SCM Maturity Questionnaire

Has eight questions:

1. Are SCM activities planned for the project?

2. Has the project identified, controlled, and made available the software work products through the use of configuration management?

... 

5. Does the project follow a written organizational policy for implementing configuration management activities?

...