(SM\textsuperscript{CMM}) : Software Maintenance Capability Maturity Model

Alain April, Associate Professor
Department of Software Engineering
École de Technologie Supérieure (ÉTS)
1100 Notre-Dame West, Montréal,
Québec, Canada, H3C 1K3
+1 514 396 8682
+1 514 396 8684 fax
aapril@ele.etsmtl.ca
Overview

- The need for a software maintenance capability maturity model (SM^CMM)
- How the model was developed
- Overview of the model architecture
- What Can be learned and Future work
Need for SM-$^{C\text{MM}}$

- CMM and CMMi focus
  - Software Development and Maintenance Projects
  - Teams of developers
- Software Maintenance Specific Processes (SWEBOK)?
  - Transition
  - Service Level Agreements
  - Acceptance/Rejection of Change and Corrective Requests
  - Planning Maintenance activities
  - Supporting operational software
What current CMM could help?

<table>
<thead>
<tr>
<th>Year</th>
<th>Software Engineering CMM proposals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1991</td>
<td>Bootstrap</td>
</tr>
<tr>
<td>1992</td>
<td>Trillium</td>
</tr>
<tr>
<td>1993</td>
<td>CMM©</td>
</tr>
<tr>
<td>1994</td>
<td><strong>Camélia</strong>, automated testing (Kra94)</td>
</tr>
<tr>
<td>1996</td>
<td>TMM (Bur96), <strong>Zit96</strong>, Dov96</td>
</tr>
<tr>
<td>1997</td>
<td>Som97</td>
</tr>
<tr>
<td>1998</td>
<td>Esi98, Top98, Baj98</td>
</tr>
<tr>
<td>1999</td>
<td>Wit99, Vet99, Sch99</td>
</tr>
<tr>
<td>2000</td>
<td><strong>Cob00</strong>, Str00, Bev00, Lud00</td>
</tr>
<tr>
<td>2001</td>
<td><strong>Kaj01d &amp; 01e</strong>, Ray01, Sch01, Luf01, Tob01, Sri01</td>
</tr>
<tr>
<td>2002</td>
<td><strong>CMMI©</strong>, <strong>Nie02</strong>, Mul02, Vee02, Pom02, Raf02, Sch02, Ker02, Cra02</td>
</tr>
</tbody>
</table>
Step by step build SM-CMM

1. Understand the knowledge area
2. Look in standards to find processes, activities and best practices
3. Look to Framework and SWEBOK to create domains and KPAs
4. Look to Best practice guides and MMs for practice details
5. Build or Refine the model Architecture
6. Find a test site and conduct a trial of the model
7. Modify model as necessary
8. Review the content with Independent Experts
Sources to build SM\textsuperscript{CMM}:

- Zitouni/Abraham Maintenance Model
- Camélia model
- \textit{CM}^3 Corrective Maintenance Maturity Model
- ITIL Service Delivery and Service Support
- IT Service CMM
- Cobit
- Malcolm Baldrige
- ISO/IEC TR 15504 part 2 Spice
- ISO/IEC 12207, Information Technology Software Life Cycle Processes
- ISO/IEC 14764, Software Engineering, Software Maintenance
- ISO/IEC 1219, Standard for Software Maintenance
- CMMI® Software Eng. v1.1
- Process Area/Specific Practice
- Standards
- Best Practice guides and Maturity Models
Use of CMMi structure in SM\textsuperscript{CMM}:

- Contains the essential elements of effective processes for software related activities
- Contains a framework that provides the ability to generate multiple models and associated training and assessment materials. These models may represent:
  - software and systems engineering
  - integrated product and process development
  - new disciplines
  - combinations of disciplines
- Provides guidance to use when developing processes

Source P.Croll: 14th Annual DoD Software Technology Conference - IEEE-Sponsored Track - 1 May 2002
Standards, are consensus-based documents that codify best practice. Consensus-based standards have seven essential attributes that aid in process engineering. They:

- represent the collected experience of others who have been down the same road,
- tell in detail what it means to perform a certain activity,
- can be attached to or referenced by contracts,
- help to assure that two parties have the same meaning for an engineering activity,
- increase professional discipline,
- protect the business and the buyer,
- improve the product.

Source P.Croll: 14th Annual DoD Software Technology Conference - IEEE-Sponsored Track - 1 May 2002
Model Context (Scope)
SM$^{CMM}$ – Alignment to CMMi
### SM<sup>CMM</sup> – Resulting KPA’s

<table>
<thead>
<tr>
<th>SM&lt;sup&gt;CMM&lt;/sup&gt; Process Domains</th>
<th>Key Process Areas of Software Maintenance</th>
</tr>
</thead>
</table>
| Process Management               | 1- Maintenance Process Focus  
2- Maintenance Process/Service definition  
3- Maintenance Training  
4- Maintenance Process Performance  
5- Maintenance Innovation and deployment |
| Maintenance Request Management   | 1- Request & Event Management  
2- Maintenance Planning  
3- Monitoring & Control of maintenance requests  
4- SLA & Supplier Management  
5- Quantitative Maintenance Management |
| Evolution Engineering            | 1- Transition  
2- Operational Support  
3- Evolution & Correction of software  
4- Verification and Validation |
| Support to Evolution Engineering | 1- Configuration Management  
2- Process and Product Quality Assurance  
3- Measurement, Decision Analysis  
4- Problem Management and Causal Analysis  
5- Rejuvenation/Retirement Engineering |
SM<sup>CMM</sup> Architecture by levels

- Domains
  - Key Process Areas
  - Maturity Levels
- Roadmaps
- Best Practices
## SM<sup>CMM</sup> – Maturity Levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Level Name</th>
<th>Risk</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial</td>
<td>Very high</td>
<td>ad hoc maintenance process</td>
</tr>
<tr>
<td>2</td>
<td>Repeatable</td>
<td>High</td>
<td>basic request-based process</td>
</tr>
<tr>
<td>3</td>
<td>Defined</td>
<td>Medium</td>
<td>state-of-the-art process</td>
</tr>
<tr>
<td>4</td>
<td>Managed</td>
<td>Low</td>
<td>generally difficult to achieve now</td>
</tr>
<tr>
<td>5</td>
<td>Optimized</td>
<td>Very low</td>
<td>technologically challenging to attain</td>
</tr>
</tbody>
</table>
Roadmaps describe different facets of a KPA

1) Communications with the developer, the owner and the purchasing agent.
2) Management of the transition process.
3) Control of training and knowledge transfer during transition
4) Prepare documentation transfer (includes source code and outstanding problem reports)
5) Participate in user and acceptance tests
SM^CMM in summary

- Model in numbers
  - 4 Process Domains
  - 18 KPA’s
  - 74 Facets
  - 443 Practices with supporting text and references
- Public Domain soon!
Ongoing Work

- Release in a French Book planned 2005 (Loze-Dion)
- Evaluation tool built by Msc student will be available soon
- Knowledge Based system to support training is planned to start during 2005.
- Will be posted on our WEB site progressively during 2005 at: http://profs.logti.etsmtl.ca/aapril/English/Autres/index.html
Thank You

Alain April, Associate Professor
Department of Software Engineering
École de Technologie Supérieure (ÉTS)
1100 Notre-Dame West, Montréal,
Québec, Canada, H3C 1K3
+1 514 396 8682
+1 514 396 8684 fax
aapril@ele.etsmtl.ca