An Introduction to CMMI and its Assessment Procedure

Seminar Paper
February 2006

Authors:
Martin Höggerl, Bernhard Sehorz
mailto:{mhoeggerl, bsehorz}@cosy.sbg.ac.at

Academic Supervisor:
O. Univ.-Prof. Dr. Wolfgang Pree
mailto:pree@softwareresearch.net
Abstract

A CMM is a process model of mature practices in a certain discipline. CMMI tries to integrate multiple CMMs. The old Software CMM is totally absorbed in CMMI. CMMI identifies 25 process areas in the software development process, each specifying a set of goals and practices, and it offers a continuous and a staged representation for each of its models. The continuous representation assigns capability levels to process areas, the staged representation assigns an overall maturity level to an organization’s development process.

CMMI is often said to favor large, bureaucratic organizations, and it is also criticized for its exclusive focus on the process.

CMMI is similar to but not easily comparable with the ISO/IEC 15504 (often referred to as SPICE).

The teams assessing an organization for CMMI compliance have to meet various requirements, such as special training and experience. We present two examples of a CMMI assessment for illustration purposes.
Contents

1 Introduction .............................. 1
   1.1 Process Models and Process Improvement ........................................... 1

2 CMMI ..................................... 1
   2.1 CMMI Basics .......................... 2
      2.1.1 The Structure of CMMI .......................................................... 2
      2.1.2 CMMI Representations ............................................................ 3
      2.1.3 Capability Levels ................................................................. 5
      2.1.4 Maturity Levels ................................................................. 5
   2.2 Criticism of CMM(I) ......................... 6
      2.2.1 Favoring the Very Large ....................................................... 6
      2.2.2 Leaving Out the Rest ............................................................ 7

3 CMMI and SW-CMM ....................... 7
   3.1 Differences between SW-CMM and CMMI ............................................. 8
      3.1.1 Multiple Disciplines ............................................................. 8
      3.1.2 Structural Changes .............................................................. 9
      3.1.3 Maturity Levels and Process Areas ........................................... 9

4 CMMI and ISO/IEC 15504 (SPICE) ...... 10
   4.1 Differences between CMMI and ISO/IEC 15504 .................................. 10

5 The Assessment Team .................... 11

6 CMMI Assessment Examples ............. 12
   6.1 A CMMI Assessment for HNIT Consulting Engineers by Students from the University of Iceland ................................................................. 12
      6.1.1 Requirements Management ....................................................... 13
      6.1.2 Measurement and Analysis ....................................................... 13
      6.1.3 Configuration Management ..................................................... 14
   6.2 CMMI-based Assessments for AVL Graz Instrumentation Test Systems by Kasse Initiatives ................................................................. 15

References .................................. 17
1 Introduction

In software development, three major components determine the quality of a product: the people that develop a software system, the technology that is employed by them, and the organization of the process of development [1]. In contrast to the former two components, the development process has been establishing itself as a major factor just recently—starting maybe about ten years ago—with the ever growing size and complexity of software projects. This may be due to the fact that development processes are primarily concerned with supporting management of software projects [2], and hence do not produce many desirable results in the form of products or the like. The “result” of a well-performed process may only become visible if one does not care about the process. In such cases, some symptoms of illness—high cost, late delivery, etc.—may (and most probably will) be perceived in a software project. An additional factor concealing the importance of process management is that is not strictly impossible to produce high quality software in time in an unmanaged process. It just demands much greater efforts and skills of the people involved [3].

While there is little doubt about the importance of people and technology, even now processes are not always accepted as being a major factor in software development [4]. Nonetheless, more and more organizations recognize the need for process management and start employing process models and the like.

1.1 Process Models and Process Improvement

As development processes gain more and more acceptance as having a heavy impact on software quality, various methods for modelling processes have been developed and are evolving constantly. Based upon such models, many organizations seek to assess their processes and raise the quality of their software by improving process performance. This is termed “process improvement” [1].

In this paper, we will take a look at one of those process models, namely the Capability Maturity Model (CMM) and its successor CMM Integration (CMMI). Beneath a general description, we explain the interactions between CMMI and ISO/IEC 15504, commonly (but erroneously) referred to as “SPICE”. We will also illustrate the CMM(I) assessment [2] by two examples.

2 CMMI

The CMM was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University during the late 1980s [2]. This work and the SEI as a whole are sponsored by the U.S. Department of Defense (DoD) [4]. Thus, CMM (and CMMI) are tailored to the needs and according to the characteristics of governmental organizations to a certain extent. That property is among the most often criticised features of CMMI (see Section 2.2).

According to the SEI, “A Capability Maturity Model (CMM) is a reference [process] model of mature practices in a specified discipline, used to improve and appraise a group’s capability to perform that discipline.” [4]. As this definition does not restrict itself on the software development process it may not come as a surprise that many different CMMs were developed for other disciplines, e.g. for systems engineering. Even for non-technical processes like managing and developing workforces CMMs were created (the P-CMM [5]).

At that point, the CMMI project was formed with the integration of three of these various CMMs among its major goals: the CMM for Software (SW-CMM), the Electronic Industries Alliance Interim Standard (EIA/IS) 731, and the Integrated Product Development (IPD) CMM [6]. Another goal was to enable the integration of future models [3]. (This is supposed to happen by adding new process areas.) As a result of these goals, CMMI came up with a model framework that could be parametrized depending on a

---

1 It should be noted that the terms “process model” and “process improvement” are also applicable to non-software processes, e.g. business processes [2]. Since we restrict ourselves to software processes these terms should be understood accordingly throughout this paper.

2 The SEI distinguishes between “assessment” and “appraisal”. The main difference between the two is that an appraisal only evaluates a process whereas an assessment is an appraisal done for the sake of process improvement. For the exact definition see any of the CMMI models [6]. We will not follow this distinction but use “assessment” in both of these cases.
selected set of disciplines that an organization deems most relevant in order to achieve their business goals. Currently, there are four disciplines available in CMMI: systems engineering (SE), software engineering (SW), integrated product and process development (IPPD), and supplier sourcing (SS) [4]. Section 3.1.1 provides more detail about the disciplines. The focus of our considerations lies on the SW discipline.

Beneath the actual process model, the SEI also released the CMMI Acquisition Module, a report that "defines effective and efficient practices for acquisition projects" [7], and the Appraisal Requirements for CMMI (ARC) that defines the essential requirements for appraisal methods intended for use with CMMI models [8]. In addition to ARC, there was also a Standard CMMI Appraisal Method for Process Improvement (SCAMPI) developed by SEI. Of course, the SEI also offers various training courses, ranging from introductions to CMMI to SCAMPI lead appraiser training [4].

2.1 CMMI Basics

When CMM was initially developed, its foundation was given by four simple axioms [9]:

1. [Process] improvement is possible and takes time
   This is supposed to occur by process inspection. According to [9], process inspection is the procedure determining how all stages, tasks, and stakeholders are linked in achieving some result.

2. Process maturity is defined by separate distinct stages
   This belief came as a result of various studies by the SEI during the late 1980's and early 1990's. Back then it showed that on whatever stage of maturity an organization's development process was to be found, examples of employed software processes were found.

3. [Process] improvement implies assigning a higher priority to some tasks
   This means that improvement in some specific process area requires a certain degree of maturity in another process area.

4. Process maturity will decrease if no-one pays attention to it
   This means that when a certain degree of process maturity is achieved it is not a good idea to rest on one's laurels.

Keeping those axioms in mind supports a thorough understanding of the topic, even though they are not referred to explicitly in the current CMMI models.

2.1.1 The Structure of CMMI

CMMI builds upon three key concepts: process areas, goals, and practices. Figure 1 illustrates the interaction of these structural elements.

CMMI identifies 25 so-called process areas in the development process [4]. Each process area defines a set of so-called specific goals and a set of specific practices that serve to fulfill the goals.

Concerning process areas, it has to be pointed out that CMMI's process areas will most likely not map one-to-one on the processes of a certain organization. Thus, it is vital to determine the best mapping of processes to CMMI's process areas. This is a matter of interpretation. In the models, the wording is: "Although process areas depict behavior that should be exhibited in any organization, all practices must be interpreted using an in-depth knowledge of the CMMI model being used, the organization, the business environment, and the circumstances involved." [9]

As mentioned above, specific goals and practices are defined by process areas. However, there is another kind of goals and also practices. The so-called generic goals and generic practices are equivalent to the specific goals and practices, with the exception that they are not specific to a certain process area. They are of concern to more than one process area. It is also worth noting that all practices that are meant to be performed for achieving a certain goal are sequentially ordered.

As an example, consider the process area Requirements Management (REQM). It defines a single specific goal "Manage requirements". The practices for this goal are:
1. Obtain an understanding of requirements
2. Obtain commitment to requirements
3. Manage requirements changes
4. Maintain bidirectional traceability of requirements
5. Identify inconsistencies between project work and requirements

It should be clear that no one can obtain a commitment to requirements that are not understood. So the ordering of practices makes perfect sense. We will resume this example in the next section.

2.1.2 CMMI Representations

Each CMMI model comes in two different “flavors”: a “continuous” and a staged representation. Both representations group the 25 process areas into distinct sets (four in the case of the continuous representation, five in the staged representation). [4]

The continuous representation then just assigns one of six capability levels to each process area. It does not grade the development process as a whole. Capability levels are explained in Section 2.1.3.

The following process area groups (they are called categories in the models) are used in the continuous representation:

- **Process Management:** consists of:
  - Organizational Focus (OF)
  - Organizational Process Definition (OPD)
  - Organizational Training (OT)
  - Organizational Process Performance (OPP)
  - Organizational Innovation and Deployment (OID)

- **Project Management:** consists of:
  - Project Planning (PP)
• Project Monitoring and Control (PMO)
• Supplier Agreement Management (SAM)
• Integrated Project Management (IPM)
• Risk Management (RIM)
• Integrated Teaming (IT)
• Integrated Supplier Management (ISM)
• Quantitative Project Management (QPM)

• Engineering; consists of:
  • Requirements Management (REQM)
  • Requirements Development (RD)
  • Technical Solution (TS)
  • Product Integration (PI)
  • Verification (Ve)
  • Validation (Va)

• Support; consists of:
  • Configuration Management (CM)
  • Process and Product Quality Assurance (PPQA)
  • Measurement and Analysis (MA)
  • Decision Analysis and Resolution (DAR)
  • Organizational Environment for Integration (OEI)
  • Causal Analysis and Resolution (CAR)

In the staged representation, on the other hand, CMMI subdivides the process areas into five sequentially ordered maturity levels. When all process areas of one level and all the levels below are at a certain minimum capability level, an organization’s software process is said to have reached the respective maturity level.

Each of the two representations has its pros and cons, as well as its specific domain of application. Three points seem to stand out among them:

**Flexible vs. static order of improvement** When used for the sake of process improvement, the CMMI in the continuous representation allows an organization to choose any subset of process areas and improve them in arbitrary sequence. The staged representation does not allow that since each maturity level exactly defines which process areas have to be implemented. The improvement path imposed by the staged representation is assumed to be proven, though, because it is based upon numerous improvements and case studies.

**Process areas vs. entire organizations** The staged representation only allows comparisons of entire organizations through their respective maturity level. The continuous representation, on the other hand, allows comparisons on a process-area-by-process-area basis. A maturity level is concise but short, whereas a capability profile is provides more detail at the expense of expressiveness and length: Such a profile requires a lot more interpretation.

Having stated this difference, a typical field of application may have become obvious: If an organization requires its contract partners to meet certain CMMI-compliance criteria (as it is often done with SPICE), it will most probably expect the partner to use the staged representation and indicate CMMI-compliance through a maturity level.

**Migration issues** The staged representation allows easy migration from the old SW-CMM to CMMI, whereas it is easy to migrate from other continuous process models to the continuous representation.

---

3Actually, the process categories are also present in the staged representation but they do not have that much effect here.
4Speaking in terms of the models, this is not true. The staged representation does not know the concept of capability levels. However, even if the wording may be different, we feel that the effect is the same in both representations.
2.1.3 Capability Levels

A capability level defines a set of practices that must be implemented for a certain process area to reach the respective capability level. In most cases, that means that some goals are defined for a certain capability, and all practices corresponding to this goal have to be implemented. Sometimes, however, a specific goal has an associated practice that is termed an “advanced practice”. Such practices are only required for reaching higher capability levels. In the example using REQM, the goals \(\text{gg}_2\) and \(\text{gg}_3\) are advanced practices and required for capability level 2. The others are just required for reaching level 1.

For each non-trivial capability level, there is one generic goal that has to be achieved for the respective capability level. The generic goals (and the respective practices) of each capability level are the same for all of the 25 process areas. Specific goals are required only for level 1 (but remember that there may be specific practices required for higher levels).

The six capability levels used in the continuous representation are \([6]\):

0. Incomplete
   - Any process area that is not performed is considered incomplete.

1. Performed
   - Being on a performed level means that the specific goals of the process areas are achieved. The generic goal for this capability level is “Achieve Specific Goals”. It has a single associated generic practice “Perform Base Practices”. (A “base practice” is a specific practice required for level 1.)

2. Managed
   - On a managed level, the performance of the respective process area is managed. That means that there is a plan for performing it, resources are allocated, responsibilities assigned, expected work products are controlled, etc. The generic goal for level 2 is called “Institutionalize a Managed Process”.

3. Defined
   - A defined process is a managed process that is tailored from the organization’s standard processes according to the organization’s tailoring guidelines. The main distinction from a managed process is that a defined process requires an organization-wide standard process that can be adapted for a certain project or the like. A managed process does not require organization wide standards. It is possible only managed for a certain project. The third level generic goal is “Institutionalize a Defined Process”.

4. Quantitatively Managed
   - A quantitatively managed process is a defined process that is controlled using statistical and other quantitative techniques. Thus, predictability of the process performance should be achieved. The level 4 generic goal is called “Institutionalize a Quantitatively Managed Process”

5. Optimizing
   - An optimizing process is a quantitatively managed process that is changed and adapted to meet relevant current and projected business objectives. An optimizing process focuses on continually improving the process performance through both incremental and innovative technological improvements. Whereas a process on level 4 may perform predictable but maybe insufficient to establish objectives, an optimizing process will always reach objectives. Here, the generic goal is “Institutionalize an Optimized Process”.

2.1.4 Maturity Levels

Maturity levels are collections of process areas. To reach a certain maturity level, all specific goals of the process areas of the level have to be achieved, as well as the generic goals for the respective level. In contrast to the continuous representation, no advanced practices exist here. To achieve a goal, all practices for this goal have to be implemented.
There are only five maturity levels (vs. six capability levels) but all the maturity levels higher than 1 are roughly equivalent to their capability level counterparts. The five maturity levels are:

1. Initial
   The CMMI describes processes at level 1 as “ad hoc and chaotic”. On this level, success depends on the efforts of the people. If they perform heroically, projects may succeed. However, projects will also often be abandoned and/or exceed budgets etc.

2. Managed (consists of: REQM, PP, PMC, SAM, MA, PPQA, and CM)
   As its name implies, level 2 is concerned with management: Requirements, processes, work products, and services are required to be managed at level 2. The status of the work products and the delivery of services are visible to management at defined points (for example, at major milestones and at the completion of major tasks).
   An important point is that level 2 ensures that practices are also retained during times of stress. Pressure of time will not result in dropping the practices.

3. Defined (consists of: RD, TS, PI, Ve, Va, OPF, OPD, OT, IPM, RiM, IT, ISM, DAR, and OEI)
   The defined maturity level is focused on organizational standards. Processes in a project are derived from the organizational standards. Even for the tailoring to specific projects, organization-wide standards exist. This ensures consistency among all processes within the organization.
   Also, processes are described in more detail and more rigorously at level 3 than at level 2.

4. Quantitatively Managed (consists of: OPP and QPM)
   This level introduces quantitative predictability of process performance (in contrast to qualitative predictability at level 3). Quantitative objectives for quality and process performance are set and controlled by statistical and other quantitative techniques.

5. Optimizing (consists of: OID and CAR)
   As the other levels, maturity level 5 is heavily resembles capability level 5. At level 5, the statistical predicted results are checked against the business objectives. If the results are insufficient, the process is changed to meet the objectives.

Obviously, any organization’s software process is always at least on the Initial level.

2.2 Criticism of CMM(I)

As any other methodology for creating software, the CMM and CMMI have critics. Among the features criticized, the following two items seem to stand out:

• CMM(I) seems to favor large and bureaucratic organizations
• CMM(I)’s exclusive focus on the process

We will now briefly address these issues.

2.2.1 Favoring the Very Large

This first point of attack is most likely caused by the fact that the SEI was sponsored by the U.S. DoD. Governmental organizations are known for being large, bureaucratic bodies, promoting form over substance. (Besides, many citizens often have a certain feeling of mistrust in “the government”.) Very often, such properties are also perceived with large enterprises like multinational corporations or monopolies. Another common feature common among such organizations is that they mostly deal with business customers—i.e. other enterprises or similar large organizations. In such a constellation, there is also often a lack of time to market.
Even Judy Bamberger, one of the key authors of the CMM, agrees “that the CMM reflects the large, aerospace, contract-software development environment it was originally intended to address” [10]. Thus, CMM(I) is undoubtedly easier applicable for measuring an organization’s capability of fulfilling software specifications rather than end user use-cases. It is also clear that CMM(I) needs less interpretation for a multinational corporation than a small software development studio.

Starting from such observations, it is often felt that CMM(I) promotes a bureaucratic attitude, claiming that anyone could develop good software, regardless of one’s intellect, by following a cookbook recipe. Such an attitude is said to raise the ratio of the number of managers to the number of developers in an organization. Bureaucracy is clearly also a major hindrance when having to meet due dates and having little time to market. Last but not least, bureaucracy suppresses creativity in the development process [4].

We believe that the key in addressing the bureaucracy issue is interpretation. As it was already stated, a small enterprise does need a lot more interpretation of the CMM(I) than do large organizations. Nonetheless, we think everybody can benefit from the CMM(I), provided it is adapted to one’s specific needs. This is discussed in detail by Judy Bamberger in her article The Essence of the Capability Maturity Model ([10]) in which she tries to help overcome some of the misconceptions about the CMM.

2.2.2 Leaving Out the Rest

The second point of attack is that CMM(I) solely concentrates on the process as a factor in software development, sparing out people and technology. It is sometimes criticized that CMM(I) promotes the process over all other issues (even over some core issues like coding software) and that implementing CMM(I) is thus no guarantee that a software project will succeed.

Besides from the fact that strict guarantees are hard to provide in any situation one has to acknowledge that “The CMM wasn’t intended to be all things to all people or cover all possible aspects of software and system development.” [10]. CMMI deliberately focuses on the process and omits people and technology. Thus, it covers not more or less than a third of software development—and does not claim to do more. It is of course also required to pay attention to people and technology in addition to manage one’s process. But implementing CMM(I) can significantly raise the probability of success in a software process.

We conclude the discussion of criticism with a complete quote of Judy Bamberger in The Essence of the Capability Maturity Model:

> The CMM wasn’t intended to be all things to all people or cover all possible aspects of software and system development. The view that guided me during my many years’ work as a CMM author, contributor, and reviewer was that the CMM was intended to provide one set of guidelines for managing software development projects and making improvements over time. This set of guidelines was based on best practices, software engineering discipline, real-world experience, and extrapolation from other industries. And, most importantly, this set of guidelines was just that—guidelines—not requirements or a checklist of “must do” items; the guidelines were intended to be interpreted, tailored, and applied within the culture and context of each unique organization.

3 CMMI and SW-CMM

In 2000, the SW-CMM was upgraded to CMMI. The SEI no longer maintains the SW-CMM model. As already mentioned, the CMMI project was formed to establish a framework to integrate current and future models and to build an initial set of integrated models, since the old CMM models

- were overlapping and contradicting
- had different structures, formats, terms and ways of measuring maturity
- caused confusion, especially when more than one was used together
- were difficult to integrate into a combined improvement program
- were difficult to use in supplier selection and sub-contracting
3.1 Differences between SW-CMM and CMMI

Obviously, it does not make sense to compare SW-CMM and CMMI in general, since CMMI is much more than a maturity model for software development. But a comparison in terms of SW-Engineering is reasonable. Hence, we will give a brief outline of what has changed during the transition from SW-CMM to CMMI.

3.1.1 Multiple Disciplines

The most obvious change is that the CMMI covers multiple disciplines. Currently the CMMI addresses four disciplines, which are characterized as follows [6]:

**Software Engineering (SW)** Software engineering covers the development of software systems. It focuses on applying systematic, disciplined, and quantifiable approaches to the development, operation, and maintenance of software.

**Systems Engineering (SE)** Systems engineering deals with the development of total systems, which may or may not include software. Systems engineers focus on transforming customer needs, expectations, and constraints into product solutions and supporting these product solutions throughout the life of the product.

**Integrated Product and Process Development (IPPD)** Integrated product and process development is a systematic approach that achieves a timely collaboration of relevant stakeholders throughout the life-span of the product to better satisfy customer needs, expectations, and requirements. If a project or organization chooses an IPPD approach, it performs IPPD-specific practices concurrently with other specific practices to produce products.

**Supplier Sourcing (SS)** The supplier sourcing discipline is applicable to projects that use suppliers to perform functions that are critical to the success of the project. Supplier sourcing deals with identifying and evaluating potential sources for products, selecting the sources for the products to be acquired, monitoring and analyzing supplier processes, evaluating supplier work products, and revising the supplier agreement or relationships as appropriate.

An organization may adopt the CMMI for software engineering, systems engineering, or both. The IPPD and Supplier Sourcing disciplines are used in conjunction with SW and SE. For example, a software-only organization might select the CMMI for SW, an equipment manufacturer might select the CMMI for SE and SS, while a systems integration organization might choose the CMMI for SW, SE, and IPPD.
3.1.2 Structural Changes

With the transition from the SW-CMM to CMMI a (relatively small) number of significant structural changes happened.

The most notable change is, that CMMI provides two representations whereas the SW-CMM knows only the staged representation. Some of the other CMMs were also continuous but none supported both representations.

Another change may seem unimportant at first glance: The transition from “Key Process Areas” (as they were called in the SW-CMM) to simple Process Areas. While the change in terminology is of course no big deal it should be noted that a process area in CMMI need not necessarily bear any relation with software development in contrast to the Key Process Areas in the SW-CMM. (Then, each Key Process Area in the SW-CMM had also “SW-” as a prefix to its name, indicating it was a software Key Process Area.)

3.1.3 Maturity Levels and Process Areas

The CMMI's maturity levels are defined the same way as in the earlier models, although some changes to the names of the levels were made. Levels 1, 3, and 5 retained their names, i.e., Initial, Defined, and Optimizing, but levels 2 and 4 are now named Managed and Quantitatively Managed, respectively, perhaps to more clearly emphasize the evolution of the management processes from a qualitative focus to a quantitative focus.

The CMMI contains 25 process areas for the four disciplines currently covered (see Figure 2). By comparison, the SW-CMM contained 18 process areas. Although many of the process areas found in the CMMI are essentially the same as their counterparts in the SW-CMM, some reflect significant changes in scope and focus and others cover processes not previously addressed.

Level 2 survived the transition to the CMMI relatively unscathed. Software Subcontracting has been renamed Supplier Agreement Management and covers a broader range of acquisition and contracting situations. Measurement and Analysis is a new process area that primarily consolidates the practices previously found under the SW-CMM’s Measurement and Analysis Common Feature into a single process area.

Level 3 has seen the most amount of reconstruction. Software Product Engineering, which, in the SW-CMM, covered nearly the entire range of engineering practices, has exploded into five process areas:

- Technical Solution covers design and construction.
- Requirements Development addresses analysis of all levels of requirements
- Product Integration addresses the assembly and integration of components into a final, deliverable product.
- Verification covers practices such as testing and peer reviews that demonstrate that a product reflects its specified requirements (i.e., “was the product built right?”).
- Validation covers practices such as customer acceptance testing that demonstrate that a product fulfills its intended use (i.e., “was the right product built?”).

Integrated Project Management covers what was previously addressed by Integrated Software Management and Intergroup Coordination in the SW-CMM.

Risk Management is a new process area, as is Decision Analysis and Resolution, which focuses on a supporting process for identifying and evaluating alternative solutions for a specific issue.

IPPD brings two additional process areas to level 3.

- Integrated Teaming addresses establishing and sustaining integrated product teams.
- Organized Environment for Integration focuses on the infrastructure and people management practices needed for effective integrated teaming.
The Supplier Sourcing discipline adds Integrated Supplier Management, which builds upon Supplier Agreement Management (level 2) by specifying practices that emphasize proactively identifying sources of products that may be used to satisfy a project’s requirements and maintaining cooperative project-supplier relationships.

Level 4 of the CMMI states more clearly what is expected in a quantitatively controlled process. Specifically, statistical and other quantitative techniques are expected to be used on selected processes (i.e., those that are critical from a business objectives perspective) to achieve statistically predictable quality and process performance. Software Quality Management and Quantitative Product Management in the SW-CMM have been replaced with two new process areas. Organizational Process Performance involves establishing and maintaining measurement baselines and models that characterize the expected performance of the organization’s standard processes. Quantitative Project Management focuses on using the baselines and models to establish plans and performance objectives and on using statistical and quantitative techniques to monitor and control project performance.

The focus and intent of level 5 has not changed dramatically with the release of CMMI. Process Change Management and Technology Change Management from the SW-CMM have been combined into one process area, Organizational Innovation and Deployment, which builds upon Organizational Process Focus (level 3) by emphasizing the use of high-maturity techniques in process improvement. Defect Prevention has been renamed Causal Analysis and Resolution.

With the increase in the number of process areas and practices, the CMMI is significantly larger than the SW-CMM; the staged representation of the CMMI-SE/SW holds a total of 80 goals and 411 practices, while the SW-CMM has 52 goals and 316 practices.

4 CMMI and ISO/IEC 15504 (SPICE)

In order to develop the ISO/IEC 15504 standard it was necessary to engage international experts in the process. As a result the SPICE project was established with a mandate to develop the first draft of the standard (handing over to the ISO working group) and to conduct user trails of the emerging standard before publication.

The SPICE project formally closed in 2003 and has been superseded by the SPICE Network [11] that is hosted by the SPICE User Group [12]. There has grown a specific habit of referring to the standard as ISO/IEC 15504 (SPICE) the standard being related to SPICE. ISO/IEC 15504 is often erroneously referred to as SPICE. The international collaborative effort to develop a standard has been underway (unofficially) since 1990 and officially since June of 1993.

The SPICE (Software Process Improvement Capability dEtermination) project is an additional effort staffed primarily by volunteers from around the world. The SPICE project has three goals:

- Assist the standardization effort in its preparatory stage to develop initial working drafts.
- Undertake user trials in order to gain early experience data that will form a basis for revision of the published technical reports prior to review as full international standards.
- Create market awareness and take-up of the evolving standard.

4.1 Differences between CMMI and ISO/IEC 15504

In this section, the main differences between CMMI and ISO/IEC 15504 are briefly sketched. These differences are:

Process Areas

Table I gives an overview over the available process categories in CMMI and ISO/IEC 15504.
<table>
<thead>
<tr>
<th>CMMI</th>
<th>ISO/IEC 15504</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering</td>
<td>Engineering</td>
</tr>
<tr>
<td>Support</td>
<td>Support</td>
</tr>
<tr>
<td>Project Management</td>
<td>Management</td>
</tr>
<tr>
<td>Process Management</td>
<td>Organisation</td>
</tr>
<tr>
<td></td>
<td>Customer-Supplier</td>
</tr>
</tbody>
</table>

Table 1: Process categories in CMMI and ISO/IEC 15504

Processes
There are several processes in ISO/IEC 15504 that have no direct equivalent in CMMI. These are:

- CUS.4 (Operation Process)
- SUP.7 (Audit Process)
- MAN.1 (Management Process)
- ORG.4 (Infrastructure Process)
- ORG.6 (Reuse Process)

Representation
In contrast to CMMI, ISO/IEC 15504 offers only one representation—a continuous representation.

Organizational capability
In CMMI, organizational capability is explicitly described in terms of maturity levels. This is not the case for ISO/IEC 15504, since ISO/IEC 15504 offers a continuous representation (and thus focuses on processes). In ISO/IEC 15504, organizational capability is implicit; it can be intuitively understood by looking at the organizational processes, the process attributes, and their dependencies.

The role of the lead assessor
CMMI is heavily depending on the lead assessor, whereas there is not much difference between the lead-assessor and the other members of the team in ISO/IEC 15504. In CMMI, the whole responsibility for quality, trustworthiness, performance etc. is up to the lead assessor. In ISO/IEC 15504, every member of the team is responsible for his/her ratings.

5 The Assessment Team

Lead Assessor requirements:

- Introduction to CMMI Training
- Assessment team experience
- Advanced CMMI Training
- SCAMPI Lead Assessor Training or Upgrade Training (for current Lead Assessors)

The Assessment Team should also meet certain requirements to achieve best results:

- Assessment Team members should understand the business the organization is in and have an interest in seeing their organization improve its processes
- Assessment Team members must have a solid software engineering background, preferably with at least 10 years of software experience but no less than five
- The entire software lifecycle must be able to be covered by the collective experience of the assessment team
• One of the team members should have at least 6 years of management experience. The team as a whole should have at least 15 years of management experience.

• Some of the organizational assessment team members should truly understand the culture of the organization

• At least one member of the assessment team must have easy access to the organization’s Senior Management team

• Organizational team members should be selected to provide the best possible coverage of the business units domain and environment.

• The assessment team should include members that are comfortable with the concepts involved in Quality Assurance and Configuration Management

• Some of the assessment team members should have experience in measurement

• The organizational assessment team member's opinions and expertise are respected throughout the organization

• Each assessment team member should be willing and able to work on an assessment team

• All assessment team members should have good people skills and be able to conduct interviews in a non-threatening manner

• At least one assessment team member should have strong presentation skills and experience in making presentations to senior management

• An assessment team member's presence on the Assessment Team should not inhibit anyone who will be interviewed from speaking comfortably about process issues

• The individual serving as the business unit coordinator should have good organizational skills and be able to communicate comfortably with all levels of management

6 CMMI Assessment Examples

We will now illustrate the CMMI assessment process by two examples.

6.1 A CMMI Assessment for HNIT Consulting Engineers by Students from the University of Iceland

The CMMI assessment presented in this section [14] was done in February 2005 by Guðlaugur Kr. Jörundsson, Sveinbjörn Guðmundsson and Martin Höggður for the course Software Quality Management at the University of Iceland. The Organisation Unit (OU) selected was the Software Department of Hnit Consulting Engineers in Reykjavík, Iceland. Three process areas were assessed:

• Requirements Management (REQM)

• Measurement and Analysis (MA)

• Configuration Management (CM)

The goal of the Process Assessment was to give the students a little insight into how assessments are performed and to increase their understanding of CMMI. The organisation however got to know CMMI and was given an idea of which capability levels their processes were on.

The assessment had a two and a half week timeframe. First the assessment team (AT) had a preparation period to read about CMMI and the assessment process. In that time the AT prepared a nineteen-slide presentation to help the company understand the purpose and the schedule of the whole project.
The assessors presented the slides containing fundamental aspects of CMMI and an introduction to the assessment itself to the company on a presentation meeting. HNIT asked for a week for preparation and to collect evidence before the actual assessment meeting could take place. The company should always be granted some time to prepare itself for the assessment.

The time-frame for the assessment itself was just two hours. The majority of the time was spent on REQM, since the organisation had recently made some good efforts on that topic. The real meaning of MA, that focuses on time and cost, was discovered during the assessment. Unfortunately, the CM process area was not understood well enough to perform a meaningful assessment at all. The main problem with a CMMI Assessment is, that every assessment requires extensive knowledge and experience. Only experienced assessors are really able to decide whether the evidence collected is sufficient and to distinguish between documents that are worthless and documents that really prove something. The more often an assessment team performs assessments, the better its members get and they are able to produce more reliable results faster.

### 6.1.1 Requirements Management

The process area of REQM got the most attention because the company had already done some work in that specific area. The OU had mainly collected evidence to support REQM prior to the meeting. The OU got fine ratings for the specific goal of REQM.

**Evidence:**

- They presented documents on how they worked together with their customers to obtain an understanding of the requirements.

**Content:**

- The OU collects information about requirement changes and critical inconsistencies between project work and requirements.
- They also make efforts to maintain bidirectional traceability of requirements. So the REQM was rated to be at least on capability level 1.

**Results:**

- REQM is quite well managed and if it was not for the lack of managed training program for people the AT could have rated REQM to satisfy goal of managed process.
- The OU have defined the REQM process. The process is defined in HNIT's Quality Management handbook.
- Comments from the staff about the process are collected to be able to improve the process. The goal of defined process is satisfied.
- Since the goal of managed process was not quite satisfied the capability level of REQM is only on level 1. Improvement on training for people would increase the level to 3.
- But in consideration of the size of the company it would be fair to reason that capability level 3 is reached. So the final result for REQM is: capability level 3.

### 6.1.2 Measurement and Analysis

The process area of MA was the most difficult birth of all, not in terms of time or understanding, but in terms of misconceptions and tough discussions.

In the beginning, the wrong types of measurement were discussed. Lines of code, bugs per lines of code etc were considered as appropriate. The assessors were a bit disappointed, because the company did not use
any of the suggested measurements. During the conversation, new and even better parameters, namely
cost and time were discovered. It turned out, that HNIT made good use of those.

Evidence:
The problem was, that the assessment team did not get to see much of an evidence for this process area.
That was due to two things.

- Lack of time. Even if there had been a lot of evidence, the assessment team would not have had
  much time for investigation.
- The OU simply did not have much documentation to offer.

Content:

- They had relatively firm guidelines for cost and time overrun. For example, if a project is 4 weeks
  or more behind its schedule, the management has to be informed. So before that “deadline”, it is up
to the project leader to take corrective actions.
- A similar rule applies to the cost of a project. If the cost overrun is 20% or more, the matter has to
  be taken to the management as well.

Results:

- Even though the existing measurements seemed quite extensive at first sight, it was discovered that
  most of them were either not specified directly as a process, or simply were not enough to satisfy all
  the needs of the specific goals.
- In the end, there was not much missing to reach Level 1.
- In a comparatively small company it is possible to maintain guidelines without documenting and
  defining them, just by talking to the employees.

6.1.3 Configuration Management

The process area that caused everyone most headaches was CM. The teams were not familiar with the
used terms at all. Neither the assessors nor the company had any idea what CM was supposed to mean.
Some understanding of certain parts of CM was developed, but it was not possible to determine what
would justify the use of the word “configuration”.

Evidence:

- None

Content:

- None

Results:

- In the end it was decided, that even though the OU used several CM Tools like InstallShield and
  SourceSafe, the assessors would refrain from rating the whole process area.
6.2 CMMI-based Assessments for AVL Graz Instrumentation Test Systems by Kasse Initiatives

About the assessment team [13]

- Lead Assessor was a Senior or Principal Consultant with Kasse Initiatives
- Assessment Team Coordinator was the same for all five sites
- Additional assessment team members were first chosen from process improvement or quality management job focus and interest
- Assessment Team Members who were also interviewed were normally done so in a private interview by the Lead Assessor and Assessment Team Coordinator
- All AVL business sites had core assessment team members who attended the SEI Intro to CMMI

Each AVL Business Unit that was scheduled for an assessment first had a Process Improvement Awareness Briefing delivered (Software Engineering, SPI, CMMI, ...).

Every business unit had to fill out some questionnaires, for example about the company, about the readiness to change something, about the organisational structure and about documented processes. Kasse Initiatives suggests an online review of the documented processes. The “Process owners” or at least process lead developers should present or “walk through” the specific process area processes. Typical questions during a review of documented processes could be

- Would you please clarify or expand on a point?
- Would you please jump to the referenced procedure, guideline, template, or checklist?
- Would you please show project examples that have followed that procedure or used that template?
- Why was this information placed in this document and not another one that is more closely aligned to the CMMI way of organization?
- Would you please print a hard copy of that section of the procedure or please print out the entire procedure?

Such an online review, according to Kasse Initiatives offers several advantages:

- Reduces the time required to accomplish a detailed look at the documented processes
- The ones presenting the process are experts in where important ideas are located within the document
- Any issue can be required to be reviewed again with little loss of time
- Follow-up on referenced procedures, guidelines, templates, and checklists is immediate
- Contributes more quickly toward rating if this is one of the assessment goals
- Having “experts” describe their documented processes and other site assessment team members witnessing the answers eliminates or reduces the risk of long debates over the value of the documented processes in the later phases of the assessment

After the online review, it is the time to consolidate the observations. Each assessment team member reviews his/her own notes and makes observations. These observations will later be shared with the rest of the assessment team. For every category, each team member gets the chance to present his/her observations and opinion.

The approach chosen by Kasse Initiatives to present the assessment results for these CMMI-based assessments includes:
• Presenting the goals of the assessment
• Presenting the scope of the assessment
• Executive summary including strong process areas, weak process areas and general indication of the satisfaction of the Generic Practices
• Presentation of the Generic Practices for the set of process areas that follow
• Presentation of the process areas in terms of Plus Points, Weaknesses, Business Consequences, and Recommendations
• Presentation of background of related process areas that were not assessed but may have a dependency with those that were part of the assessment scope
References


The Wikipedia-articles on CMMI provides a good starting point for further research on the topic. Unfortunately, as of January 2006, the quality of the articles in the English Wikipedia is insufficient and inferior to the German articles.


[8] The Appraisal Requirements for CMMI, Version 1.1 (ARC, V1.1) by SEI. Available at the SEI website under http://www.sei.cmu.edu/cmmi/appraisals/appraisals.html


[14] Assessment for HNIT Consulting Engineers for the University of Iceland http://www.hi.is/