Name:	CMMI : Guidelines for Process Integration and Product Improvement
Author:	Mary Beth Chrissis, Mike Konrad, and Sandy Shrum
Publisher:	Addison-Wesley
Date of :	2007
ISBN:	0-321-27967-0
BOK:	Software Engineering Processes; Program and Project Management
No. Pages:	595 plus appendices

There is a plethora of maturity models, standards, methodologies, and guidelines available to help organizations improve the way they do business. Most of them focus on a specific part of the business and do not take a systematic approach to the problems faced by the total organization. The CMMI provides an integrated model that transcends disciplines. It addresses practices that cover the product (or service) life cycle from conception through delivery and maintenance. CMMI emphasizes the work needed to build and maintain the total product (or service).

If you are new to process improvement or perhaps new to the concepts governing CMMI then you should read Chapter 1 because it provides an overview of process improvement and describes CMMI. Then you should skim Part Two, describing both generic and specific goals and practices. After that you may want to review the references in Appendix A to identify additional sources of beneficial information. Follow this with a reading of the acronyms and glossary to become familiar with the language of CMMI and then return to Part Two.

If you are new to CMMI but you do have experience with other process improvement models, you will notice many similarities. Read Part One to learn how CMMI is different from other process improvement models. Then as you read Part Two notice the best practices from other models with which you are familiar. Study the tips, hints, and cross references to see details and relationships that will assist in understanding CMMI.

If you are not new to CMMI you will quickly recognize the concepts and best practices. Focus on the tips, hints, and cross references in the process areas (Part Two) to discover ideas, relationships or details you may have overlooked previously.

Part One of this book describes the various approaches and models that govern the capability maturity model. It also describes the process area components and the multiple maturity levels. These are then followed by a description of the relationships among the process areas and information on using the CMMI models. The final section of Part One is a case study showing how Raytheon applied CMMI to services.

The authors have included practices that cover project management, process management, systems engineering, hardware engineering, software engineering, and other supporting processes used in development and maintenance. In addition, the construct may cover the use of integrated product teams for maintenance and development activities. The practices described are used by organizations in many diverse industries, including aerospace, banking, computer hardware, software, defense, automotive, and telecommunications.

There are 22 CMMI process areas and each one consists of required components, expected components, and informative components. Required components describe what the organization must

do in order to satisfy the process area. Expected components guide those who implement improvements or perform appraisals. Expected components include the specific and generic practices.

The authors treat each process area as a component; and each one, in alphabetic order, consists of:

- *Introductory notes* that describe the major concepts covered in the process area.
- *Related process areas* lists references to related process areas and reflects the high level relationships among the process areas.
- *Specific goals* describe the unique characteristics that must be present to satisfy the process area.
- *Generic goals* describe the characteristics that must be present to institutionalize the processes that implement the process area.
- *Specific goal and practices summary* provides a high-level summary of goals and practices that must be accomplished. This is informative.
- *Specific* practices describe the activities that are considered important in achieving the specific goal.
- *Typical work products* lists sample output from a specific practice. This is informative.
- *Sub-practices* provide a detailed description that provides guidance for interpreting and implementing a specific or generic practice. This is informative.
- *Generic practices* contain a description of an activity that is important in achieving the generic goal.
- *Generic practice elaboration* provides guidance on how the generic practice should be applied to the process area.
- *Supporting informative components* provide further information to describe the concept. This may be in the form of notes, examples, amplifications, or references.
    - *Notes* are text that provides detail, background, or rationale.
    - *Examples* consist of text that clarify a concept or described activity.
    - *Amplifications* are notes or examples that are relevant to a particular discipline.
    - *References* are a pointer to additional or more detailed information in related process areas.
- *Additions* can be informative material, a specific practice, a specific goal, or a process area that extends the scope of the model or emphasizes a particular aspect of its use.
- *Specific work products* describe artifacts that are developed while implementing the specific practices.

While Part One provides the models for implementing CMMI, Part Two describes in detail all of the generic goals and practices of CMMI. It focuses on those model components that address process institutionalization. Institutionalization implies that the process is ingrained in the way the work is performed and there is commitment and consistency in the way the process is performed. The progression of processes is as follows:

- A *performed process* accomplishes the work necessary to produce work products. This is equivalent to a Level 1 process.
- A *managed process* is a performed process that is planned and the performance of the process is managed against the plan. This is equivalent to a Level 2 process.
- A *defined process* is a managed process that is tailored from the organization's set of standard processes and provides a basis for planning, performing, and improving the project's tasks and activities. This is equivalent to a Level 3 process.

- A *quantitatively managed process* is a defined process that is controlled using statistical and other quantitative techniques. This is equivalent to a Level 4 process.
- An *optimizing process* is a quantitatively managed process that is changed and adapted to meet relevant current and project business objectives and is continuously improved by addressing common causes of process variation. This is equivalent to a Level 5 process.

Because of my own interest in verification and validation, these were the first process areas I turned to. Verification is the process of determining that something meets its requirements and validation is the process of determining that something meets the user needs and is usable by the user. At first look, they appeared to be correct and with an appropriate level of detail. However, further examination led to dismay when I noted that under SP 1.1 *Select Work Products for Verification* it identified various types of testing as work products. The Verification process should have identified the test artifacts (e.g., Integration Test Plan) for each type of testing as work products and should have stated that the various types of testing actually are encompassed in the Validation process. One of the "tips" in this section states that verification often means testing. Testing is a technique, not for verification, but rather for validation as are inspection, demonstration, and analysis.

Artifacts are typically verified against applicable standards. Artifacts such as a training guide or an operations manual may first be verified to determine that the contained information is adequate and complete and then be validated to determine that the information is useable. This reviewer considers it problematic that this book is at odds with verification and validation as described in IEEE Std 1012-2004, IEEE Standard for Software Verification and Validation. Systems and software are validated. In addition, the book, in the Part Two section describing the Validation process states that the end users are typically involved in validation. This is true when validation is for User Acceptance Test. However, it is not true when validation involves integration, system, performance, or stress testing.

While other process areas appear to conform to "best practices" I was disappointed that the authors frequently recommend that the process information be documented in the Project Plan. A process such as Configuration Management or Process and Product Quality Assurance might be better served by having its own plan. Perhaps the need for separate plans, as appropriate, and the identification of consensus standards to assist in the preparation of the plans will be addressed in the next version of this book.

Yet even with these points of disagreement, I still recommend this book because of its depth and detail of coverage.