

CMM^â Integration (CMMI) Framework

Roger Bate, Sandy Shrum

This article provides a high-level technical description of the CMMI framework, which is currently being developed to accomplish the goals of the CMMI Project. Keep in mind that the CMMI Project is not complete and, thus, the framework and other aspects of the CMMI product suite may change before they are released.

The CMM Integration (CMMI) Project has been underway since February 1998. Representatives from government, industry, and the SEI are combining several existing process improvement models under a single architectural framework. This framework will sort, combine, and arrange process improvement elements to form outputs that serve the individual needs of organizations. These elements include the core building blocks for process management and integration as well as elements specific to certain functional disciplines (such as software engineering or systems engineering).

The CMMI framework is one part of what is called the "CMMI product suite." In addition to the CMMI framework, the product suite consists of capability models, training products, assessment materials, a glossary, and tailoring guidelines. These informational elements are what populate the framework in a raw form and compose the outputs of the framework in their finished form.

The purpose of the CMMI product suite is to serve the users of capability models (CMs) and Capability Maturity ModelsSM (CMMs) better than the independently created CMs and CMMs now available. The framework's integrated approach will simplify

the process of understanding and using multiple models and provide integrated and tailorable process improvement tools for the development user community.

The process improvement models resulting from the framework will be called “CMs,” not CMMs. Other similar changes in terminology will be necessary to combine existing models and supporting materials into an integrated CMMI framework and product suite.

The CMMI framework development process

As [Figure 1](#) illustrates, framework developers from government, industry, and the SEI are engaged in a project to combine the wealth of information about software engineering (SW), systems engineering (SE), integrated product and process development (IPPD),¹ assessment techniques, and training techniques to develop a CMMI product suite. This suite will be capable of producing CMs tailored to the needs of user organizations.

¹ Integrated product and process development (IPPD) is a management technique that integrates all development activities ranging from product concept to product support. The IPPD approach uses multifunctional teams, called integrated product teams (IPTs), to improve the product and its development and sustainment processes. The goal of this improvement is to meet the organization's cost and performance objectives. IPPD evolved from concurrent engineering and is sometimes called integrated product development (IPD).

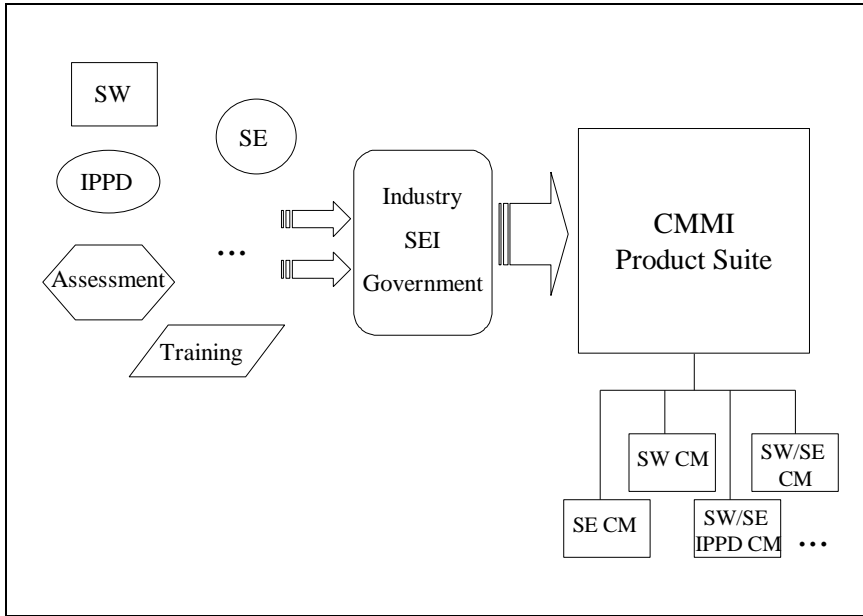


Figure 1: CMMI framework development process

Notice that the outputs at the bottom right of [Figure 1](#) show various combinations of software engineering, systems engineering, and IPPD. Over time, more elements will be added to the mix and thus will add to the available outputs of the framework. These outputs will include CMs covering additional disciplines (e.g., security systems engineering, hardware engineering) and various combinations of these disciplines. These CMs, because they are developed within the CMMI framework, will share common terminology, components, and assessment methods.

Development goals

Before the development team created the framework, they identified the goals that the framework must meet. Overall, the CMMI product suite must be an improvement over the use of independently developed CMs and CMMs.

The goals of the CMMI Project are to eliminate inconsistencies, reduce duplication, and lessen the cost of implementing model-based process improvement. Further goals are to ensure that the products of the framework are easy to understand and use because they use common terminology, have a consistent style, follow uniform construction rules, and share common components among products. Finally, the team is trying to minimize the impact on those who are using existing models, assessment materials, and training materials.

Therefore, developers are striving to maximize commonality, maintain the look and feel of existing materials, provide assets for future models, design a logical organization for assets, and establish and follow configuration-management guidelines.

The vision of the CMMI product suite is that a framework user can generate capability models and their supporting training and assessment materials as needed from the framework's common elements and discipline-specific elements. In other words, if an organization wanted to improve its software engineering, systems engineering, and integrated product and process development, it could combine these three disciplines into one capability model supported by one set of training materials and assessment materials.

By eliminating unnecessary duplication of common activities, the job of training people to use the models will be simpler. The common terminology used in the models, training materials, and assessment methods will simplify the introduction of process improvement activities based on these models, thereby reducing the costs of adoption.

The structure of the framework

The CMMI framework is designed to provide an internally consistent set of common elements that apply to any discipline and that must be included in any CMMI product. These CMMI

products will support process improvement activities, including assessments and training. The CMMI framework currently consists of four parts: the input process, repository, control process, and output process. [Figure 2](#) illustrates the structure of the framework:

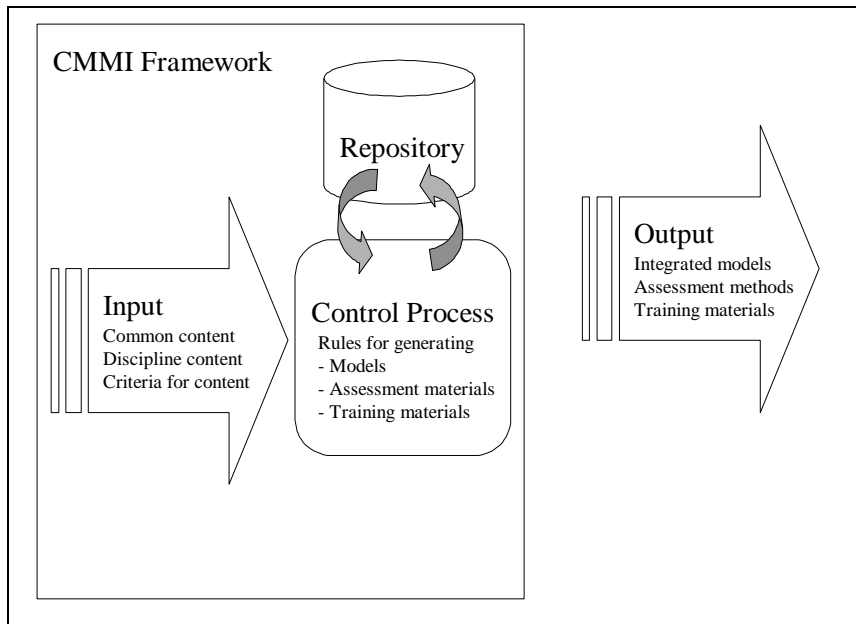


Figure 2: CMMI framework

The repository contains the components of capability models, training materials, and assessment materials, as well as construction rules and the conceptual architecture of the outputs. The control process combines the inputs and the rules for generating capability models, assessment materials, and training materials to produce output that can be applied to an organization's process improvement efforts.

How the framework functions

Imagine that the framework is a capability model generator. As a user of the CMMI framework, you would specify various options based on the needs of your organization, such as disciplines that need to be covered, staged vs. continuous, and inclusion of the IPPD environment. The framework would then generate the capability model that best meets your needs. Figure 3 represents the information that resides in the framework and how it is processed to produce tailored capability models.

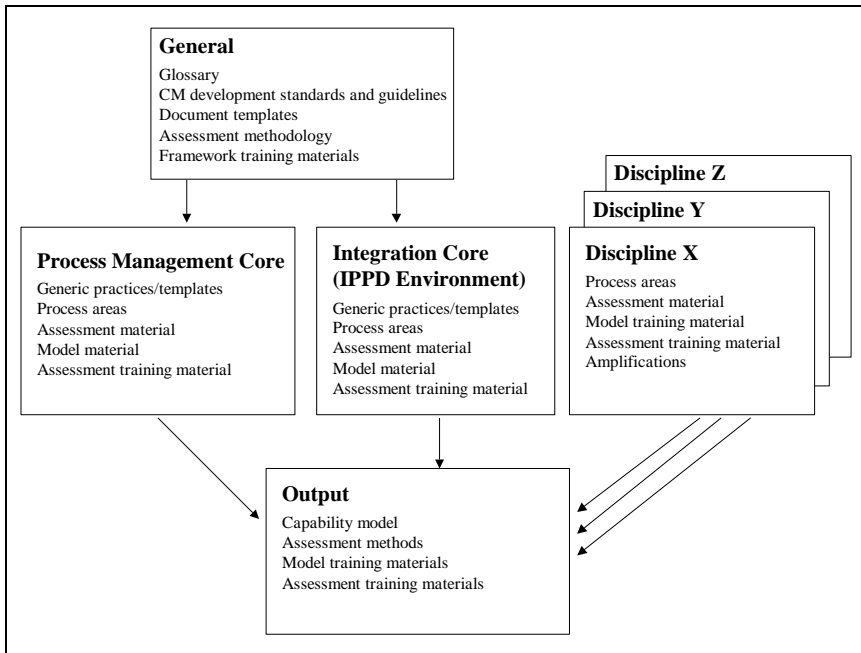


Figure 3: How the framework processes information

The entire middle row of boxes represents the essence of the process improvement information that is eventually used by the user organization. This information includes the process management core, the integration core, and any number of disciplines.

The *process management core* contains process management components that apply to all disciplines and all domains. These components are automatically included in your capability model.

The *integration core* contains information about IPPD, which can be applied in virtually any discipline or domain.

The *disciplines* represent specific information that you can select to include in your capability model. The initial CMMI product suite will include only two disciplines: software engineering and systems engineering. However, the framework is being designed so that it can accommodate new disciplines over time.

Essentially, the framework sorts, combines, and arranges information to make it useful for you and to tailor the information to your needs.

Staged versus continuous representations

In every process improvement model, regardless of representation, the basic building blocks are process areas. How these process areas are presented in the model can be considered its “representation.” The two dominant representations used in existing process improvement models are *staged* and *continuous*.

Naturally, framework developers had to decide whether to use staged or continuous representations in the framework's capability models. Each representation has its proponents and detractors in the development community.

In a *staged* representation, process areas are grouped into stages (or maturity levels). Each process area contains practices that, when performed, achieve the purpose of the process area. Within a stage, the institutionalization practices for all constituent process areas must be achieved to successfully achieve the entire stage. Once an organization has achieved the entire stage, it has

reached a capability maturity level. The Software Engineering Institute's CMM for Software (SW-CMM) is an example of a staged model.

In a *continuous* representation, process areas also contain practices that, when performed, achieve the purpose of the process area. Generic practices are grouped into capability levels. These practices are added to the practices of each process area to attain a capability level for each process area. Capability levels are achieved process area by process area. Although the order in which process areas are addressed is not required to follow a particular sequence, the order may follow recommended staging. The Electronic Industries Alliance's Interim Standard 731, Systems Engineering Capability Model (SECM), is an example of a continuous model.

End users of the framework may prefer a CM that has a staged or a continuous representation. Likewise, there are two models that initially compose the input to the framework: one is a staged model (SW-CMM) and one is a continuous model (SECM). Given these existing models, the best way to serve the end user was to provide each model in both staged and continuous representations.

Products resulting from the CMMI framework

The products that result from the CMMI framework include capability models, training materials, and assessment materials.

Capability models

Each capability model consists of process areas, specific practices, generic practices, capability levels, stages, maturity levels, discipline-specific amplifications, and descriptive

material. Initially, the CMs that will be available from the CMMI framework are

- a Software Engineering CM
- a Systems Engineering CM
- an Integrated Software and Systems Engineering CM
- an Integrated Software and Systems Engineering CM that incorporates the principles of IPPD

These CMs will support both the staged and continuous representations.

Training materials

The training package you will receive from the framework will be tailored to the CM that you have chosen. Each training package will contain training for the CM (including both staged and continuous approaches), training for the assessment team, and training for lead assessors.

Assessment materials

The assessment package you will receive from the framework will also be tailored to the CM you have chosen. Each assessment package will contain materials for assessment planning (including requirements and methodology), data collection methods and tools for both staged and continuous approaches (e.g., questionnaires, interviews), analysis methods, and team qualifications.

Other materials

The framework also will provide materials that support development of CMs in other functional disciplines. These materials include a glossary, CM development standards and

guidelines, document templates, assessment methodology, and training materials on the use of the framework. Likewise, the core components (process management core and integration core) are available to be incorporated into the new CMs.

How will the CMMI framework be used?

The CMMI framework is designed to be used both by those who use CMs for process improvement in their organizations and by those who develop new CMs.

CM end users are interested in using the tailored capability models, training materials, and assessment materials for process improvement in their organizations. The framework's tailored CMs and supporting materials are used by end users to write policies, conduct process improvement programs, create process documentation, and persuade their organizations to adopt process improvement.

CM developers are interested in developing and maintaining CM products, validating their models, supporting state-of-the-practice reports, or entering new discipline-specific information into the framework. The CMMI framework will be an effective tool for constructing CMs that use common terminology, common elements, and standard construction rules.

The CMMI framework will provide the ultimate user of these products with several benefits: simple and consistent CMs, common terminology across multiple disciplines, and the ability to spread this useful improvement technology across all of their disciplines. What's more, the framework will ease the complex task of building new CMs and the associated training and assessment materials.

For more information

To learn more about what is happening with the CMMI project and development of the CMMI framework, visit the [CMMI Web site](#).

About the authors

Dr. Roger Bate was the chief architect of the Enterprise Process Improvement Collaboration (EPIC), which developed the Systems Engineering Capability Maturity Model (SE-CMM) and the Integrated Product Development Capability Maturity Model (IPD-CMM). Bate was a Texas Instruments (TI) fellow and the chief computer scientist for TI, where he headed a corporate-wide project to improve the software development process. Bate is also a fellow of the Association for Computing Machinery (ACM) and a fellow of the Society for Design and Process Science.

Sandy Shrum has been a senior writer/editor at the SEI since September 1995. Before working at the SEI, she was a senior information developer at Legent Corporation where she supported business-critical networking systems and maintained technical and user documentation. She has a BS in business administration and marketing from Gannon University and an MA in professional writing from Carnegie Mellon University.

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.

SM Capability Maturity Model, IDEAL, Interim Profile, Personal Software Process, PSP, SCE, Team Software Process, and TSP are service marks of Carnegie Mellon University.

[®] Capability Maturity Modeling, CERT, and CMM are registered in the U.S. Patent and Trademark Office.