# To CMMI or Not to CMMI:
## Issues to Think About

Winifred Menezes
*Consultant*

*Version 1.02 of the Capability Maturity Model® Integration℠ (CMMI℠)-Systems Engineering/Software Engineering and CMMI-SE/SW/Integrated Product and Process Development were released more than a year ago. At the time this article was written, a cleaner and more stable version 1.1 was due to be released in January 2002. A number of organizations have decided to adopt the new model as a guide for their process improvement program (see <www.sei.cmu.edu/cmmi/publications/early-adopters.html> for a list of early adopters). Others are asking questions like: Should we adopt the CMMI? When is the right time to transition? Which model is suitable for our business? Which representation makes sense? This article describes a number of scenarios and discusses the pertinent issues for each. But first, it begins with some general information about the CMMI.*

The Capability Maturity Model® (CMM®) Integration℠ (CMMI℠) in its present form is a collection of best practices for the "development and maintenance" of both "products and services." The model was developed by integrating practices from four different CMMs – the "source models:" the CMM for software, for systems engineering, for integrated product development (IPD), and for acquisition.

Organizations can use the model as a guide for improving their ability to develop (or maintain) products (and services) on time, within budget, and with desired quality. During the past decade many organizations have used CMM and CMM-like concepts to bring order to their software development processes. The CMMI allows these organizations to continue focusing only on the discipline of software.
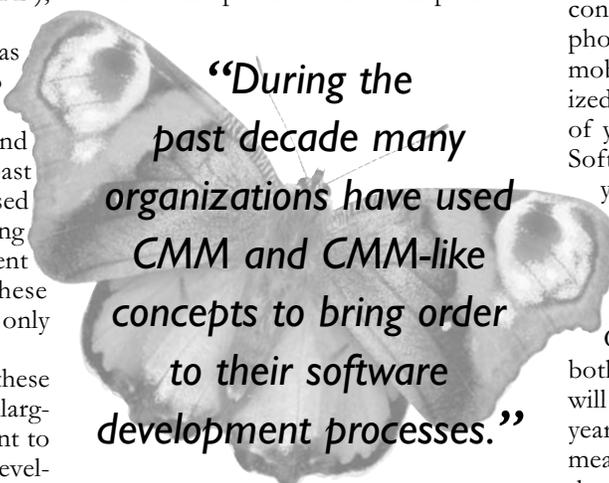
Additionally, it also provides these organizations the framework for enlarging the focus of process improvement to other areas that also effect product development – the discipline of systems engineering. During the past decade, new and effective concepts for organizing developmental work have surfaced and been adopted such as concurrent engineering or the use of integrated teams. Organizations using (or wishing to adopt these ideas) can also find support in the CMMI by using the model with integrated product and process development (IPPD) additions.

Finally, organizations that acquire components or services as a substantial part of development will find the acquisition additions useful. (CMMI-Systems Engineering (SE)/Software Engineering (SW)/IPPD/Acquisition (A) Version 1.02d draft is available for review and piloting.) CMMI-SE/SW, CMMI-SE/SW/IPPD as well as CMMI-SE/SW/IPPD/A are available at the Software Engineering Institute's (SEI) Web site <www.sei.cmu.edu/cmmi>.

## Representations

The CMMI has yet another complexity to it: the representations, staged and continuous. Philosophically there are two different approaches to process improvement. One focuses on the organization as a whole and provides a road map of successive stages aimed at improving the organization's ability to understand and control its processes. This approach is the basis for the staged representation.

The other approach focuses on individual processes, allowing the organization to choose which process or a set of processes need to have more capability. This is the approach of the continuous representation.

In theory the choice of processes is unconstrained, but in reality increasing the capability of a particular process necessitates that other processes have certain capabilities. So the continuous representation provides a few more routes on the process improvement map. We are talking of two representations – two different views of the same content. The rules for converting one representation into the other have been defined. So a choice of one representation does not preclude the use of another at a later time.

## Scenario 1: You Market a Product That Contains Software

Your organization develops and markets a product or a product component that contains software, for example a cell phone or the breaking system for automobiles. During the past decade, you realized that software was a key enabling part of your product; you have been using the Software CMM (SW-CMM) for some years now. In fact some units have been appraised at CMM Level 3, and say they have managed to reduce a good deal of rework.

Should you transition to the CMMI? SEI, who is the custodian of both the SW-CMM and the CMMI, says it will not support the SW-CMM after the year 2003. This pronouncement does not mean the SW-CMM will disappear, but the infrastructure that supports its use (e.g. training, authorization of assessors) will definitely be weakened.

However, there is another more compelling reason to transition to the CMMI. One of the source models that the CMMI was based on was the SW-CMM Version 2 Draft C. This version of the SW-CMM was an improvement on Version 1.1, which is what you and most other people are using. So the CMMI encompasses the experience and lessons learned from the previous 10 or so years of SW-CMM use.

### Your Product Life Cycle

When would be a good time to transition? The answer to this depends on your current experiences with the SW-CMM and process improvement, as well as your plans for the future. Are your software

> *"During the past decade many organizations have used CMM and CMM-like concepts to bring order to their software development processes."*

January 2002

www.stsc.hill.af.mil **9**

development groups constrained by budget, cost, and product decisions over which they have no influence? Is there a need to bring some structure into the total product-development life cycle? Is senior management planning to use integrated product teams (IPTs) for future development?

If your answer is yes to any of these, then you should begin to introduce CMMI now. You will need to plan for two different types of introduction. The first part of your transition plan is introducing process improvement concepts to the systems engineers, product management, customer representatives or similar groups. Expect the same amount of resistance that the software engineers once had, e.g., "My work is creative, it's different – I can't follow a process." The tools and techniques you initially used with the software groups should be useful.

The second part of your transition plan should concern those using the SW-CMM. If they are well on their way to the next maturity level, it might be better to make them aware of the CMMI. However, let them achieve the maturity level they were aiming for before working with the details of the CMMI. If their process improvement efforts are languishing, then maybe the CMMI will function as a catalyst.

In both cases you will need to think about the "new" process areas in the CMMI. Consider "measurement and analysis." Groups that have achieved CMM Level 3 or higher in the SW-CMM will already have some of the practices in place for this process area. Other new process areas, for example those in the engineering category, might require more effort. On the other hand, you may be pleasantly surprised to find that the software groups had these in place already. The objective of this two-pronged introduction should be that all relevant groups ultimately have the same level of process capability.

### Your Process Improvement Experience
Your systems engineering groups may already have the same amount of process awareness as your software groups. You may be one of the organizations that has tailored out the word "software" and applied the SW-CMM to non-software development too. In this case, the choice of when to transition should depend on the current effort of process improvement.

If groups are working towards a maturity level, make them aware of the new

model but wait for them to reach their maturity objective. If improvement work has reached a standstill, then CMMI may be the refocus point. However, first find out why improvement work is at a standstill before attempting to rally people around a slightly different flag pole.

Which model should be used? If you use or plan to use IPTs, then choose the CMMI-SE/SW/IPPD. If not, stay with the CMMI-SE/SW.

Finally, which representation should be used? Since you have experience with the SW-CMM, staying with the staged representation would be the easiest. If, however, your organization has fallen into a "level hunt" (levels for the sake of levels), then you may want to break the circle by using the continuous representation. Some organizations with a good understanding of processes and process improvement prefer the continuous representation because it provides more granularity and flexibility. Remember there are equivalency rules between the two representations, so you can get the benefit of both worlds.

Suppose, however, you are one of those organizations that has not used the SW-CMM. In this case the issue is not transition but adoption. If you want to start improving your processes, start with the CMMI by implementing the practices on all development groups within the limits of your improvement budget. Since you are new to the improvement game, you will find better guidance in the staged representation. So unless there is some compelling reason, choose the staged representation.

## Scenario 2: You Develop Only the Software Component
In this scenario, you are a software development unit within a larger enterprise. That is, other units develop requirements, some of which will be met by the software your group develops. Yet other units take what you deliver and integrate it with other components into a product or service.

You are interested in applying the CMMI to software only. This still means that the whole of CMMI-SE/SW (or CMMI-SE/SW/IPPD in the event you use or plan to use IPTs) is applicable. The process area descriptions contain amplifications for systems engineering, software engineering, and IPPD. The amplifications contain more information about how a practice could be applied within a particular discipline. You could skip the amplifications for systems engineering (and

IPPD), but the process area would still be applicable.

### The Best Time to Transition
When should you start using the CMMI? Just as in Scenario 1, that depends on whether you are transitioning from one of the source models or adopting a process improvement model for the first time. Similarly, the discussion from Scenario 1 regarding which model and which representation are applicable applies for this scenario, too.

The issues you will particularly need to think about are "use of" or "interpretations of" some of the engineering process areas, such as requirements development, product integration, and validation. Since other units in your enterprise are responsible for developing requirements, integrating the components, and validating that the product meets the customer needs, your unit will need to study these process areas and decide if there is a useful mapping between their practices and the scope of your unit's responsibilities.

## Scenario 3: Software Is Your Product
This scenario is a combination of Scenarios 1 and 2 with a twist. Your organization develops and markets a software product such as a word processor, financial system, networking software, or a game. You have most certainly heard about the SW-CMM. You may be on the verge of using the model and are now wondering about introducing a model that will be phased out after a few years. Or, you may have started using it either informally as a source of best practices or more formal as the basis for a sponsored and planned process improvement program.

Most of the discussion in Scenario 1 about transitioning or adopting the CMMI, as well as which representation to use, is applicable to you. Since you market your software product, you should also consider the "systems" part of the software product development. This includes issues like how the software will be used and how it will be marketed and delivered. The combination of management and engineering process areas would be a good guide in your process improvement work.

## Scenario 4: Your Software Process Improvement Is Based on ISO 15504
This scenario, really a variant of both Scenario 1 and 2, is that you have experience using ISO 15504[1] (also known as SPICE) as your guide to software process

improvement. The reference model in ISO 15504 covers the software life cycle, so any need to enlarge the scope of your process improvement to all product development would be one reason to move to CMMI. ISO 15504 has a continuous architecture, so you would probably find the continuous representation easier to use.

Another reason to adopt the CMMI is that the revised ISO 15504 will no longer have a reference model.

The standard will have guidance for performing appraisals as well as compliance requirements for suitable reference models. Both the CMMI model as well as the appraisal method released by the SEI are ISO 15504-compliant. This means that if you were to be appraised by a rigorous appraisal process (like the one released by the SEI) you would fulfill any ISO-15504 requirements you may have.

## Scenario 5: Your Product Does Not Have Software.

The product or service you develop and market contains no software. Do you need to think about CMMI? The answer is yes; the CMMI applies to all product (or service) development.

Since you do not "do" software, you have obviously not bothered with the SW-CMM. But you may have used Electronic Industries Alliance (EIA) 731 (or one of its predecessor models, Systems Engineering Capability Appraisal Model or SE-CMM). Since EIA 731 is an interim standard and was one of the source models for the CMMI, there is reason to transition to the CMMI.

The continuous representation will be most suitable since that is what you are used to from EIA 731. You would not be interested in the software amplifications within the process areas, but the systems engineering and perhaps IPPD amplifications would be of use. So you would choose CMMI-SE/SW or CMMI-SE/SW/IPPD.

Wait until you reach a milestone in the current process improvement efforts before transitioning to CMMI.

## Conclusion

Every organization's particular situation is unique. In all probability none of the scenarios above will exactly fit your organization. However, this general discussion should give you some idea of what issues and problems you will face. The answers to the following set of questions are ultimately what should guide an organization's CMMI adoption strategy. Possible answers to these questions were the basis of the five scenarios above. Naturally your unique answers should guide your CMMI adoption strategy:

- What are the organization's business goals?
- What product/service does the organization develop/maintain?
- What is the product life cycle and development/maintenance organization?
- How much process improvement experience do the various units within the organization have?
- Would it make sense to enlarge the current process improvement effort to other parts of the development organization?
- When will the organization meet the next improvement milestone?

Good Luck!◆

## References

1. CMMI for Systems Engineering/Software Engineering, Version 1.02 (CMMI-SE/SW, Version 1.02) Continuous Representation CMU/ SEI-2000-TR-029, ESC-TR-2000-094, CMMI Product Development Team <www.sei.cmu.edu/cmmi>.
2. CMMI for Systems Engineering/Software Engineering, Version 1.02 (CMMI-SE/SW, Version 1.02) Continuous Representation CMU/SEI-2000-TR-028, ESC-TR-2000-093, CMMI Product Development Team <www.sei.cmu.edu/cmmi>.
3. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, Version 1.02 (CMMI-SE/SW/IPPD, Version 1.02) Continuous Representation CMU/SEI-2000-TR-031, ESC-TR-2000-096, CMMI Product Development Team <www.sei. cmu. edu/cmmi>.
4. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, Version 1.02 (CMMI-SE/SW/IPPD, Version 1.02) Staged Representation CMU/SEI-2000-TR-030, ESC-TR-2000-095, CMMI Product Development Team <www.sei. cmu.edu/ cmmi>.
5. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Acquisition, Version 1.02d DRAFT (CMMI-SE/SW/IPPD, Version 1.02d DRAFT) Continuous Representation CMU/SEI-20XX-TR-XXX, ESC-TR-20XX-XXX, CMMI Product Development Team <www.sei. cmu. edu/cmmi>.
6. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Acquisition, Version 1.02d DRAFT (CMMI-SE/SW/IPPD, Version 1.02d DRAFT) Staged Representation CMU/SEI-20XX-TR-XXX, ESC-TR-20XX-XXX, CMMI Product Development Team <www.sei.cmu.edu/cmmi>.
7. International Organization for Standardization and International Electrotechnical Commission. Information Technology: Software Process Appraisal. Geneva, Switzerland: 1998.

## Note

1. ISO 15504, a technical report published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), is expected to become a standard some time in the future. ISO 15504 consists of nine parts, covering the method to be used for assessing the software development processes. For an objective appraisal (or evaluation) processes are compared against a particular model or standard, called the reference model. The SW-CMM and CMMI are reference models as is part two of ISO 15504.

## About the Author

**Winifred Menezes** has more than 20 years experience in software engineering, software process improvement, and training. She has lived and worked in the United States since 1997. Previously, Winifred worked as a consultant at Cap Gemini and later at ABB Corporate Research, both in Sweden. At ABB she was instrumental in starting ABB Sweden's software process improvement program. Winifred is a member of the Capability Maturity Model® Integration℠ (CMMI℠) product team. She is also an authorized instructor of the "Introduction to Software-CMM" and the "Introduction to CMMI." In September 2000 she received a "certificate of appreciation" for her contribution to the CMMI product suite. She received a master's of science in mathematics from the University of Islamabad, Pakistan and a Fil Kand in computer sciences from the University of Upsala, Sweden.

**3406 Bitterwood Place, J202**
**Laurel, MD 20724**
**Phone: (301) 317-1425**
**E-mail: winifred.menezes@ieee.com**