

CMMI—V1.2 and Beyond

Mike Phillips

My last two columns have focused on appraisal issues (“myths” and “due diligence”), and it is a good time to update you on where we see ourselves going with the CMMI Product Suite. As many of you know, there is a CMMI Steering Group that controls CMMI Product Suite content. In Steering Group meetings, the future plans for CMMI are debated and approved. We are now moving into the initial phases of work on the next product suite revisions.

What changes will be in V1.2?

The upcoming revision to the CMMI Product Suite is called Version 1.2. The theme for the Version 1.2 update is “one book; one course.” The “one book” approach is based on the “blue book” in the Addison-Wesley series, *CMMI: Guidelines for Process Integration and Product Improvement*. The book’s authors (also CMMI Version 1.2 development team members) found that it was possible to combine both the continuous and the staged representations of the models into a single book—an improvement over the two separate representations released as technical reports on the SEI Web site for Version 1.1. This hard-bound book version of CMMI models will serve as the basis for our update to Version 1.2.

While some have desired resolution to a single representation, we think that both representations have shown value for different purposes. The staged representation provides a roadmap for organizational improvement; the continuous representation provides the ability to focus selective attention on specific problem areas and treat them effectively. So a book that contains both approaches together recognizes these complementary objectives for model use.

As for the course approach, by fall of this year, we will be piloting a single *Introduction to CMMI* course that includes coverage of both the staged and continuous representations. The original approach of having two separate courses, one for staged and one for continuous, made sense initially because it helped users to make the transition from other models. Users of the Software CMM were more comfortable attending the staged course, and users of standards such as EIA 731 and SPICE were more comfortable attending the continuous course.

However, now that CMMI has been available for nearly four years, those that attend the *Introduction to the CMMI* courses tend to be new to process improvement and have little or no experience with legacy models or standards. Their inexperience makes their choice of a course difficult. Students are uncertain about what a representation is, much less which representation to choose.

Additional changes to the CMMI Product Suite are currently being considered, such as further integrating supplier sourcing practices into Software and Systems Engineering model practices and other simplifications.

What are you planning to do to simplify the model?

Both users and developers have sought ways to simplify the model from its inception. We have plans to make changes that will simplify CMMI models for Version 1.2. First, we think that combining the two representations into one document for each model reduces complexity. We have been pleased to find that the book version has been well received by CMMI users and, believe it or not, has held the total number of model pages (as seen in the comparable technical reports) to a little over 700 pages.

We also plan to modify the architecture of CMMI models to eliminate common features and advanced practices. These two types of model components were passed down from the source models. Advanced practices are a type of model component that came from EIA 731. Common features are a type of model component that came from the SW-CMM. While each of these has had value in its source model, their value in CMMI is marginal, and their removal effectively simplifies the CMMI models. In addition to simplifying the numbering scheme, a few of the engineering practices will be consolidated, specifically those that are subsumed by a more advanced practice within the relevant engineering process area. Generic practices will continue as the main measure of capability levels but will no longer be characterized by their common feature identifier.

While no final decisions have been made, we are also investigating other consolidations based on change requests. One of these is the integration of the two supplier-related process areas—Supplier Agreement Management (SAM) and Integrated Supplier Management (ISM). This consolidation would not only eliminate one process area from the CMMI mix, but also would simplify the number of model choices by eliminating the supplier sourcing (SS) discipline as a separate choice when selecting a model. Such a change might also be perceived as increasing the difficulty of achieving maturity level 2, but is still worthy of consideration since supplier sourcing is not a true discipline like systems engineering and software engineering.

What kinds of clarifications are envisioned?

Thus far, we have organized the change requests by process area. While all process areas have change requests associated with them, the engineering and IPPD process areas have received the greatest attention. Therefore, these process areas are likely to change more than others for Version 1.2. In addition, we plan to clarify the importance of achieving customer satisfaction in the informative material. Those who are familiar with the emphasis on the “voice of the customer” in Six Sigma will recognize the importance of clarifying this coverage that was always intended as part of CMMI best practices.

Are there any expansions planned?

Many of the suggestions for expansion will require clear sponsorship to expand CMMI model coverage. Such sponsorship is described in the CMMI Concept of Operations, viewable on the CMMI Web site at <http://www.sei.cmu.edu/cmmi/background/conops.html>. One area that the CMMI Steering Group has approved for expansion of model coverage is actually a clarification—coverage of the hardware discipline. In some organizations, hardware development centers are not being included in CMMI-based process improvement programs because the organizations that manage these centers perceive CMMI models as being “just for systems engineering and software engineering.” We anticipate that by adding

hardware amplifications, typical work products, and other informative hardware examples, we can illustrate? demonstrate? assert? that the existing CMMI models are applicable not only to product and service development, but also to hardware development—and therefore all of the engineering disciplines involved in the development effort.

An expansion of model coverage that we will investigate is a broadening of the coverage of the development environment. Currently, a single process area, Organizational Environment for Integration (OEI), provides some coverage of the processes associated with the development environment. One of the source models, EIA 731, had many practices covering the engineering environment that are not included in CMMI best practices.

Change requests have included comments that “coverage of the work environment is necessary to address safety- and security-related issues.” A few provocative change requests have suggested that we need to include some of the best practices associated with “business continuity,” particularly considering the experiences and organizational shocks experienced after September 11.

Most challenging about all of these ideas is that we must balance the desire for expanded coverage with the desire to make the CMMI models simpler, clearer, and smaller.

What if my organization doesn't develop, but provides services or operates systems?

This question has been a most challenging one. From the earliest development of CMMI, the desire to cover process improvement more widely was acknowledged. Today, organizations that focus on service delivery or the operation of systems can use the continuous representation to focus attention on all areas of the model that are relevant to them and improve their capabilities in those areas. But many organizations that do little or no development tell us that their customers (usually government) want to know their suppliers' maturity levels when selecting providers of services and support beyond the engineering development phases of the life cycle. These organizations would benefit if a way to achieve a maturity level existed for non-development, non-engineering organizations.

One of our ongoing objectives is to maintain confidence in the meaning of maturity level ratings. If one organization is a maturity level 3, it must be comparable to others who have also achieved maturity level 3. Simply declaring several of the engineering process areas “not applicable” while claiming maturity level 3 would erode the confidence in maturity levels. We are committed to maintaining the integrity of maturity levels while attempting to meet the needs of organizations that don't quite fit the engineering paradigm.

Our architecture team has investigated this and other issues and has defined architectural requirements necessary to accommodate future possibilities. This team decided that, with fairly minor adjustments, future models can be created to meet the needs of different domains. These needs will be met by “CMMI constellations,” or groups of related CMMI models that serve domains such as “development” or “services.” Each constellation can be created to reuse much of existing model content, but allows areas that are “not applicable” for a given domain (i.e., constellation domain) to be excluded.

Each constellation would include process areas and practices that are relevant to that domain and not to other domains. For example, in a service domain, the management of service level agreements is vital, but the focus on design, architecture, and product integration would not add value to the workforce. A “service management” process area category might replace the current “engineering” process area category, with a mixture of level two and level three elements. The current collection of models would be called the “development constellation,” and the new constellation in this example would be the “service constellation.”

Comment: Why do we spell the numbers out here but use numerals to designate maturity levels earlier in the paper?

By constructing future constellations from a common architecture, the commonalities across the constellations can be maximized. These commonalities are particularly important where companies and government agencies span multiple parts of the system life cycle. As one team member noted, “Some organizations must focus on multiple areas—development, service delivery, and operations—to be successful, and thus there may need to be a single constellation covering these areas across the enterprise.”

We want to maximize commonality for these multi-faceted organizations. Such an approach can aid process improvement across other boundaries. For example, we may have developers who maintain close relationships with the operators of the systems they provide. Using common approaches to enhance process relationships may be mutually beneficial to both groups.

Is there any plan to improve the appraisal method (SCAMPI)?

When we looked at the change requests submitted for the CMMI appraisal method, SCAMPI, there seemed to be two classes: (1) thoughtful recommendations for improvements that require extended consultation within the appraisal community, and (2) a smaller set of clarification changes critical to the use of the SCAMPI method that can be implemented relatively quickly. As a result, we are exploring the possibility of a SCAMPI update early in 2005 in addition to the Version 1.2 update, coinciding with the model update in mid-2006.

Is there a plan for expansion after the V1.2 release?

Yes, the possibilities of CMMI coverage for future constellations are already considered in the architecture, and coverage expansion can be initiated whenever suitable sponsorship is shown by the community. The *CMMI Concept of Operations* guides such model expansions; it requires the provision of expertise and funding to craft new CMMI practices, goals, process areas, and other artifacts needed to ensure that these new best practices can successfully be integrated into the CMMI Framework.

We envision that some proposed expansions may fall within the development scope of the current CMMI models, and thus might be numbered as “Version 1.x.” If the model expansion requires the addition of a new constellation, we would seek to treat it as a more major change, as we must ensure that the distinctions from the original development “parent” are clear.

About the Author

Mike Phillips is the Director of Special Projects at the SEI, a position created to lead the Capability Maturity Model[®] Integration (CMMI[®]) project for the SEI. He was previously responsible for transition-enabling activities at the SEI.

Prior to his retirement as a colonel from the Air Force, he managed the \$36B development program for the B-2 in the B-2 SPO and commanded the 4950th Test Wing at Wright-Patterson AFB, OH. In addition to his bachelor's degree in aeronautical engineering from the Air Force Academy, Phillips has masters degrees in nuclear engineering from Georgia Tech, in systems management from the University of Southern California, and in international affairs from Salve Regina College and the Naval War College.

The views expressed in this article are the author's only and do not represent directly or imply any official position or view of the Software Engineering Institute or Carnegie Mellon University. This article is intended to stimulate further discussion about this topic.

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.