A Configuration Manager's Perspective

by Ronald Starbuck
Output Technology Solutions

The Software Engineering Institute (SEI) dynamically broadened the Software Capability Maturity Model (SW-CMM) into an integrated suite (CMMI) of process models that now span the enterprise. This expanded comprehensive approach to software engineering through processes was the result of merging the updated unreleased SW-CMM with the other best features of software and system-related disciplines to form a single product maturity framework. This article looks at the CMMI from the perspective of a Configuration Manager to see what its influences were to Configuration Management (CM) since the release of the SW-CMM Version 1.1.

Background

The integration of the CMMI resulted in combining Version 2.0 of the SW-CMM, Draft C with the Electronic Industries Alliance (EIA) Interim Systems Engineering Capability Model Standard (SECM) and the CMM Integrated Product Development, Draft Version 0.98 (IPD-CMM) [1]. (See Figure 1.) The result is a suite of products in which CM continues to be a major building block in software/systems development. It continues to ensure the product life cycle commonality, compatibility, and consistency. Version 2, draft C of the SW-CMM was the major contributor to CM in the CMMI. It evolved from what has become the de facto standard for assessing and improving software engineering processes the SW-CMM, Version 1.1. This updated version was never released but included more than 180 user requests from lessons learned in SW-CMM implementation, defining a better understanding of higher software maturity, and achieving better consistency with the other software industry standards and terminology. The staged presentation of the CMMI is used for comparison in this article as its framework best reflects the architecture typified by the SW-CMM [2]. An organizational maturity level is established by a maturity framework structure in which the goals of a set of process area are attained.

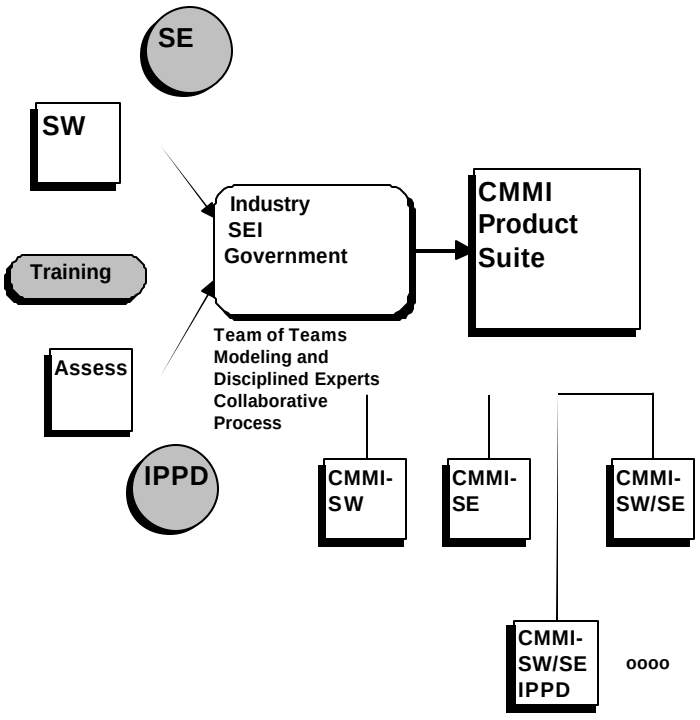

Figure 1. *Model's Creation*

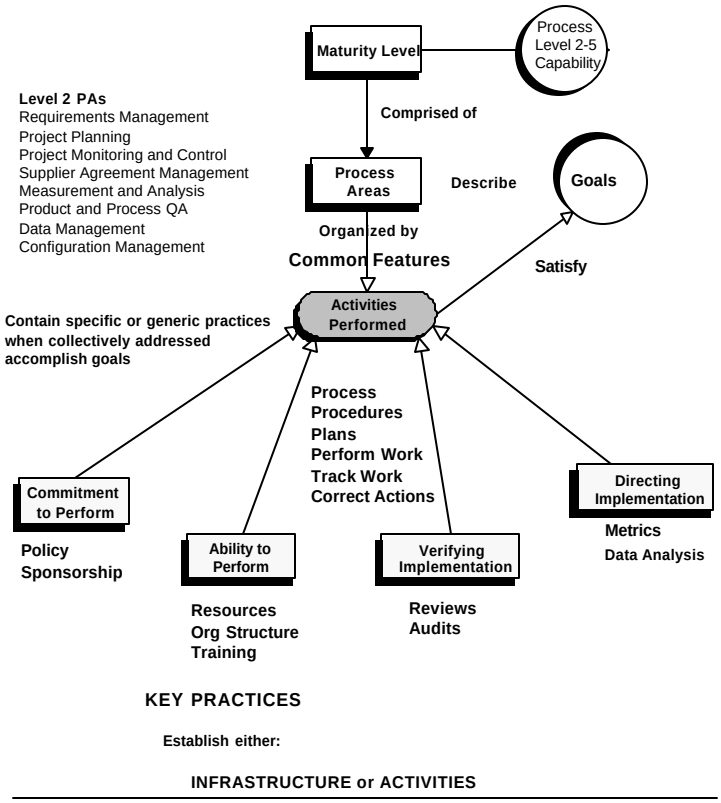

Figure 2. *Architecture of the Model*

Overview

The CMMI continues to embrace the same time-proven structure that evolved the SW-CMM, Version 1.1 into a world class model for software development. The staged presentation's structural design is still composed of five ascending levels of process maturity supported by defined Process Areas (PA) that collectively achieve a maturity level (see Figure 2). These clusters of related practices are performed collectively to achieve a set of objectives for each maturity level. They continue to be called common features, predefined attributes that signify whether the implementation and institutionalization of a process area are effective, repeatable, and lasting. A top-level check of the staged presentation of the model reveals the obvious changes since Version 1.1 of the SW-CMM:

- Most notable is the Maturity Level 2 name changed from "Repeatable" to "Managed."
- Removal of "Key" from "Key Process Area (KPA)."
- A different set of PAs comprises the CMMI. Two have been added to the six in the SW-CMM for Level 2. They are Requirements Management, Project Planning, Project Monitoring and Control, Supplier Agreement Management,

Measurement and Analysis, Product and Process QA, Data Management, and Configuration Management.
- The five Common Features now include a new key practice, Directing Implementation, that replaces the SW-CMM key practice of Measurement and Analysis. The five common features implementing the CMMI are now commitment to perform, ability to perform, activities performed, directing implementation, and verifying implementation.[1]

Detailed Look

It is obvious these aggregate changes in the integrated model will have an impact on the way in which CM is to be done. Today's CM encompasses a broad spectrum of interrelationships and associations with all the disciplines and business dependencies that make up the organization that produces the software and will require some retooling. A closer look at what these changes are to the Configuration Management Process Area follows:

The CMMI Maturity Level's "Repeatable" and "Key Process Area" names were changed, respectively, to "Managed" and Process Area." They should not have any effect on CM; however, they definitely entail a cultural transitional learning curve as the SW-CMM names have evolved to signify world-class definitions people have come to know as the Capability Maturity Model. In general some Process Area's are different in the CMMI as it modified some of the existing SW-CMM Key Process Area's and added two additional Process Areas, to bring the total to eight defining CMM Level 2 in the SW-CMM. It also changed the purpose statement of Configuration Management, "to establish and maintain the integrity of the products of the software project throughout the projects life cycle," [2] to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits [1]. They are:

- Combining and modifying the SW-CMM KPA, "Software Quality Assurance" (SQA) into "Product and Process Quality Assurance" (PA) seems to relax the former KPA of SQA. The purpose of Process and Product Quality Assurance is to objectively review activities and work products for their adherence to applicable require ments, process descriptions, standards, and procedures, and communicate the results to staff and management. It will have a slightly different role of visibility through objective reviews of the way products are developed.
- The "Data Management" PA is new. It provides administrative management of appropriate project data, both deliverable and nondeliverable project data and maintain its availability to the project staff and stakeholders. This and the CM PA are closely related. It addresses other data and focuses on data needs, the data development schedule requirements, date acquisition and control, and access to data. CM is focused on the rigorous control required for the technical work products.
- Changing the name from SW-CMM KPA "Software Project Tracking and Oversight" to "Project Monitoring and Control" better describes the PA's purpose—to provide adequate visibility into the progress of the project so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
- Changing the name from SW-CMM KPA "Software Project Subcontract Management" to Supplier Agreement Management" is just a name change. The purpose of Supplier Agreement Management is to manage the acquisition of products and services from sources external to the project to provide adequate visibility into a project's progress. This is so appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
- The "Measurement and Analysis" PA is new. Its purpose is to develop and sustain a measurement capability in support of management information. This PA was derived from the Measurement and Analysis Common feature to a PA as a definite lesson learned and as a requirement of the SECM. It centralizes organizations to implement measurement easier than if the equivalent practices spread across multiple PAs, as was done in the SW-CMM. It also is due in part for Level 4 requirements, [3]. Its addition to the model will add to the importance of configuration status accounting data and practices in some fashion.
- Project Planning's purpose is to establish and maintain plans that define project activities. While the KPA of SCM in the SW-CMM specifically identified a Software Configuration Management Plan, the CMMI relaxed this to cover the practices for performing CM functions. However, it better standardized what content is needed for establishing and maintaining plans to control the project.
- The CMMI uses "Directing Implementation," a new key practice based on the SW-CMM "Measurement and Analysis." Significantly, this key practice now implements management and analysis rather than saying they need to be done in the SW-CMM. This change enhances the key practice with action on what to do.

Polished Purpose Statement

The CM's purpose has evolved from SW-CMM Version 1.1, which was "to establish and maintain the integrity of the products of the software project throughout the project's software life cycle." Its purpose has been redefined as establishing and maintaining the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits in the CMMI. This definition to a configuration manager is classic configuration management at its best. It establishes and maintains the integrity of work products as they evolve through the full life cycle to ensure the bottom-line capability of CM to remanufacture products, the timeless and sometimes forgotten reason for doing CM. To me, CMMI configuration management is more consistent with its ancestry rooted in such government software standards as MIL-STD-483A, "Configuration Management Practices," released in 1970. It is a time-proven descendant of today's modern configuration management, but now process-oriented in the CMMI—today's trend for the application of CM.

Concerns

The latitude given by the CMMI to tailor how you do business can allow some flawed interpretations by inexperienced configuration managers. In relaxing language, these configuration managers could take the acceptable alternatives road allowed by two of its key practices.

The first concern is the ability to perform (AB1.) [1], "Plan the process, establish and maintain the requirements, objectives, and plan for performing the CM process." This means a configuration management plan (CMP). The alternative is a less acceptable set of processes. Without a CMP roadmap, these processes are not choreographed as to the order of steps from the development process until the final release of the product. A CMP purpose is to put all of these process sets into their proper life-cycle perspective [4].

The second concern is in the activity performed (AC5.) [1], "Establish and maintain the Change Request System." Your change management system should include a documented Configuration Control Board that fits your organization's needs—which the CMP should define. It provides a disciplined perspective of what is to be changed from a board of subject matter experts that has a big-picture view, or properly addresses what is proposed for change. The full board is used only when necessary, as not all changes have to go through all of the subject matter experts for a decision. The alternative to doing anything less is a subordinate way of making changes [4].

CM Conclusions

As re-instrumented in the CMMI-SE/SW, the staged representation continues the CM's time-proven functionality to fashion and maintain the necessary integrity to develop and reproduce software work products from baselines. The CMMI is more process-ordered than its predecessors and leads the trend for today's implementation of CM. It is one that features process dependencies or relationships with all of the disciplines involved in software development. Implementation success, as always, will depend on the sophistication of CM that the organization has already reached, and the experience level of its Configuration Manager. Less mature organizations could misinterpret some CMMI-acceptable alternative practices, resulting in something less than the best way to do things.

References

1. Public-release draft, *Staged Representation for CMMI*-SE/SW, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa. 1993.
2. Paulk, M. et.al, *Capability Maturity Model for Software,* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa. 1993.
3. Lucero, Scott, *Insight, The Army's Software Metrics Newsletter*, Spring 1999, The Capability Maturity Model Integration, An Interview with Bruce Allgood and Lt. Col. Jarzombek. Vol. 3, No. 4. [Ed note: reprinted this issue, page 4.]
4. Starbuck, Ronald, From the From Line, *Software Testing & Quality Engineering,* Vol. 1, Issue 6, 1999.

Note

1. See page 6 for more on this.

About the Author

Ronald Starbuck is a configuration manager at Output Technology Solutions, where he is involved in improving customer systems' department configuration management and software development process infrastructure. He has spent 21 years in government positions and agencies, such as lead programmer for the Navy's P-3 Orion Weapons Simulator., configuration manager at the Sacramento Army Depot for test program sets, and a software quality assurance representative for the Defense Logistics Agency

Output Technology Solutions
Attn: Ron Starbuck, Mail Stop 4150
1102 Investment Blvd.
El Dorado Hills, Calif. 95762
E-mail: Ronald_Starbuck@billing.com
Voice: 916-941-4250
Fax: 916-941-4064

Combined Automated Lessons Learned (CALL) Information Center

Lessons Learned are positive and negative experiences of value that are documented by a researcher, validated by a subject matter expert, and uploaded into an infobase maintained by NAWCAD 4.1.6.2 at PAXRIVMD. Lessons Learned are intended to inform people and influence intelligent decision making at all levels of business and all phases of production. The infobase is accessible on a Web site and access is authorized for military, government civilian, government contractor, and select academies of higher learning. The infobase contains about 5,000 lessons that cover 43 impact areas such as Acquisition Reform, Best Practices, Contract Administration, Design Engineering, Human Factors, Maintenance Engineering, Reliability, Safety, Test and Evaluation and Training on all Aerospace Programs/Platforms and Support Equipment. Lessons written for the Navy, Air Force and the FAA are present in the infobase. The World Wide Web site is located at www.nawcad.navy.mil/call

Points of Contact

Dawn Hassinger, CALL Manager
Voice: 301-342-2170
Fax: 301-342-4782

Ron French, CALL Administrator
Voice: 301-342-4284
Fax: 301-342-1247

Sal Pistachio, CALL Researcher
Voice: 301-342-4286
Fax: 301-342-1247

Mailing Address

Ron French
Naval Air Warfare Center Aircraft Division, Bldg. 2102
22689 Saufley Road, Unit 1
Patuxent River, Md. 20670-1620