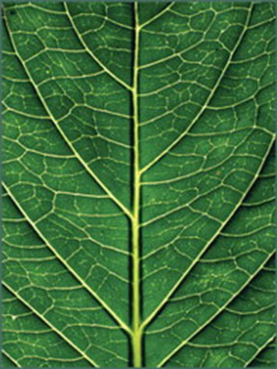
SEI SERIES IN SOFTWARE ENGINEERING

CMMI® Distilled



THIRD EDITION

A Practical Introduction to Integrated Process Improvement

Dennis M. Ahern

Aaron Clouse

Richard Turner

Carnegie Mellon
Software Engineering Institute

The SEI Series in Software Engineering

SEI Figure Credit List appears on page 249.

Library of Congress Cataloging-in-Publication Data

Ahern, Dennis M.
CMMI distilled : a practical introduction to integrated process improvement / Dennis M. Ahern, Aaron Clouse, Richard Turner.—3rd ed.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-321-46108-7 (pbk. : alk. paper)
1. Capability maturity model (Computer software) I. Clouse, Aaron. II. Turner, Richard, 1954– III. Title.

QA76.758.A397 2008
005.1068'5—dc22 2008008198

ISBN-13: 978-0-321-46108-7
ISBN-10: 0-321-46108-8
Text printed in the United States on recycled paper at Courier in Stoughton, Massachusetts.
First printing, April 2008

Preface

CMMI Distilled was originally conceived as a way to introduce the CMMI Product Suite and model-based continuous process improvement to a wide audience. Our goal was to offer a succinct, no-nonsense, minimal-jargon, wittily written, practical guide that was less than half the weight of the "official" CMMI book. We wanted to describe the origins of the model and give the readers some insight into how the 200-plus CMMI authors worked (and fought) to produce it. The first edition had a good deal of "why" information, aimed at folks who had used one of the three source models and who wanted to understand how CMMI differed from earlier models. Of course, it also had the "what" and "how" information about CMMI Version 1.0.

The second edition coincided (roughly) with the release of CMMI Version 1.1, so it included significant changes to the original "what" and "how" sections. At that point CMMI was no longer new and people were beginning to move away from the source models, so we removed some of the "why" material. To reflect the broader reach of CMMI and the need to support practitioners in acquiring sponsors for their improvement initiatives, the second edition added material aimed at managers rather than practitioners.

CMMI content and usage continue to evolve, so now we have created a third edition, building on the legacy of the first two. CMMI began as a tool for managing improvements in engineering development organizations, with a focus on systems and software. In CMMI Version 1.2, this attention to engineering has been strengthened by including explicit hardware-related information. More intriguing, however, are two new members of the CMMI Product Suite: one for the acquirer of systems instead of the developer, and the other for service providers instead of product builders. With these two additions, the potential scope of application for CMMI within an organizational enterprise has broadened significantly. At the same time, CMMI is finding use outside the traditional engineering field. For example, it has been adopted by enterprises as varied as medical facilities seeking to improve their patient care and government entities trying to build and improve their infrastructure.

We had three primary reasons for writing the third edition of *CMMI Distilled.*

1. We wanted to update the book to include changes to the CMMI architecture, content, and presentation, as well as the ongoing domain extensions to the model. The full, updated model content for version 1.2 is covered in the same way as in our previous editions. We use the graphics from the CMMI training material and describe the model components clearly and simply.
2. We wanted to further reduce the amount of historical information relating to the origins of CMMI. For those who may be interested, this material is well covered in the previous editions of this book.
3. We wanted to update and expand upon the practical advice we offer for those using the model. In this edition, we more fully address CMMI usage in tandem with Six Sigma, lean engineering, and other continuous improvement approaches. We also discuss some of the changes to the appraisal methods; specifically, we provide additional guidance on preparing for and managing appraisals, and on using the appraisal results as a powerful input to improvement activities.

Those who have read the previous editions of this book will not be surprised to find that we have included yet another literary offering (three songs) addressing CMMI and the world of continuous improvement. In fact, for those of you who may have missed the first two editions, the earlier "literary gems" can be found on the Pearson Web site at informit.com/aw under either "literary gems" or "superfluous material"—for some reason, the editors were still discussing this as of publication.

And so, in recognition of the explosive growth of CMMI as a process improvement tool of choice around the world, and having incorporated the most recent developments in the evolution of the model suite, we are pleased and excited to present this third edition of *CMMI Distilled*. We hope that it will continue to help our readers understand the CMMI Product Suite and use it wisely for their continuous improvement initiatives.

As always, we couldn't have put this third edition together without the support, wisdom, and patience of our wonderful wives. Pam, Debbi, and Jo—we still love you the best of all!

Dennis, Aaron, and Rich

April 2008

Chapter 3

The CMMI Concept

It must be remembered that there is nothing more difficult to plan, more uncertain of success, nor more dangerous to manage than the creation of a new order of things. For the initiator has the enmity of all who would profit by the preservation of the old institutions, and merely lukewarm defenders in those who would gain by the new order.
Machiavelli, *The Prince* (1513)

In this age, which believes that there is a short cut to everything, the greatest lesson to be learned is that the most difficult way is, in the long run, the easiest.
Henry Miller, *The Books in My Life* (1957)

The implementation of continuous improvement in an organization is possible using two or more single-discipline models. There are many advantages, however, in having just one model that covers multiple disciplines. For this reason, the U.S. Department of Defense—specifically, the Under Secretary of Defense for Acquisition, Technology, and Logistics—teamed up with the National Defense Industrial Association (NDIA) to jointly sponsor the development of Capability Maturity Model Integration (CMMI). In 2000, under the stewardship of the Software Engineering Institute (SEI) at Carnegie Mellon University, this effort produced the first integrated CMMI models, together with associated appraisal and training materials. The year 2002 saw the release of CMMI version 1.1. Version 1.2 was released in 2006.

This chapter begins with an overview of the kinds of information and guidance found in CMMI models. For those not familiar with any of the source models, it provides a good introduction to CMMI's scope and usefulness. We follow this overview with a discussion of CMMI objectives and history. Next comes information on the source models that were used in creating CMMI. Finally, we describe the CMMI project organization.

3.1 An Overview of CMMI

The CMMI Product Suite contains an enormous amount of information and guidance to help an organization improve its processes. But how does this information help? To answer this question, we start by noting that the CMMI models contain essentially two kinds of materials:

- Materials to help you evaluate the content of your processes—information that is essential to your technical, support, and managerial activities
- Materials to help you improve process performance—information that is used to increase the capability of your organization's activities

We start with a brief look at each of these types of materials.

3.1.1 Process Content

CMMI provides guidance for your managerial processes. For example, you should establish and maintain a plan for managing your work, and make sure everyone involved is committed to performing and supporting the plan. When you plan, you should define exactly how you will develop and maintain cost, schedule, and product estimates. When you do the work that you have planned, you should compare the performance and progress against the plan, and initiate corrective actions if actual and planned results are out of synch. You should establish and maintain agreements with your suppliers and make sure that both parties satisfy them. CMMI also incorporates information on managing project risk and on creating and managing teams.

CMMI guidance on technical matters includes ways to develop, derive, and manage requirements, and to develop technical solutions that meet those requirements. The integration of product components depends on good interface information, and CMMI reminds us that integration activities need to be planned and verified. In following the CMMI model, you should make sure that the products and services you develop are consistent with their original requirements and satisfy the customer's needs through verification and validation practices.

Support processes for technical and managerial activities are also addressed as part of CMMI. You should always manage the versions and configurations of intermediate work products as well as end products and services. You should have methods for ensuring that the processes you have defined are followed and the products you are developing meet the quality specifications you have established. You need to decide which information is important and establish ways to measure and track it. In some cases, you need to plan ways to resolve issues formally. You may need to figure out the root cause of serious problems with your products or key processes.

3.1.2 Process Improvement

Once processes have been established, improving them becomes a key goal. The improvement information in CMMI models includes the creation of a viable, improvable process infrastructure. To build this infrastructure, CMMI includes ways to get your organization to focus on defining and following its processes. Through training and standardization, you can make sure all members of the organization know their roles and understand how to execute them in the process. The measurement data you collect can be applied to improve your process performance, to innovate when processes need to evolve, and to ensure that your organization is able to meet changing needs.

Processes need to be planned just like projects, and it helps if the organization has given some weight and validity to process compliance through policy. You need to make sure that resources are available for trained, empowered people to perform the process. Those with a vested interest in a process need to be identified and involved. Work products that result from performing a process should be managed, the process documentation should be controlled, and progress against the process plan should be tracked as well. Someone should be responsible for objectively evaluating that the process is being followed, and management should be briefed periodically on process performance.

Processes become more capable when they are standardized across the organization and their performance is monitored against historical data. With capable processes, you can detect variations in performance and address potential problems early. Ultimately, you should be continuously improving the processes by identifying the root causes of process variability and finding innovative ways to make them better.

3.1.3 CMMI and Business Objectives

In Chapter 1, we identified some common organizational business objectives. Based on this brief overview of CMMI's process content and concern with process improvement, how could you expect CMMI to help your organization in meeting such objectives? Let's look at each objective individually.

Produce Quality Products or Services. The process improvement concept in CMMI models evolved from the Deming, Juran, and Crosby quality paradigm: Quality products are a result of quality processes. CMMI has a strong focus on quality-related activities, including requirements management, quality assurance, verification, and validation.

Create Value for the Stockholders. Mature organizations are likely to make better cost and revenue estimates than those with less maturity, and then exhibit performance in line with those estimates. CMMI supports quality products, predictable schedules, and effective measures to support management in making accurate and defensible forecasts. Process maturity can guard against project performance problems that could weaken the value of the organization in the eyes of investors.

Be an Employer of Choice. Watts Humphrey has said, "Quality work is not done by accident; it is done only by skilled and motivated people."[1] CMMI emphasizes training, both in disciplines and in process. Experience has shown that organizations with mature processes have far less turnover than immature organizations. Engineers, in particular, are more comfortable in an organization where there is a sense of cohesion and competence.

Enhance Customer Satisfaction. Meeting cost and schedule targets with high-quality products that are validated against customer needs is a good formula for customer satisfaction. CMMI addresses all of these

1. Humphrey, W. *Winning with Software.* Boston: Addison-Wesley, 2002.

ingredients through its emphasis on planning, monitoring, and measuring, and the improved predictability that comes with more capable processes.

Increase Market Share. Market share is a result of many factors, including quality products and services, name identification, pricing, and image. Clearly customer satisfaction is a central factor, and in the marketplace having satisfied customers can be contagious. Customers like to deal with suppliers that have a reputation for meeting their commitments. CMMI improves estimation and lowers process variability to enable better, more accurate bids that are demonstrably achievable. It also contributes to meeting essential quality goals.

Implement Cost Savings and Successful Practices. CMMI encourages measurement as a managerial tool. By using the historical data collected to support project estimation, an organization can identify and widely deploy practices that work, and eliminate those that don't.

Gain an Industry-wide Recognition for Excellence. The best way to develop a reputation for excellence is to consistently perform well on projects, delivering quality products and services that address user needs within cost and schedule parameters. Having processes that incorporate CMMI practices can enhance that reputation.

As you can see, CMMI comprises information that can make a significant impact on your organization and on the achievement of your business objectives. In the next sections, we'll discuss a different set of objectives—those that led to the development of CMMI itself. In addition, we explore the models that were used as the basis for the information CMMI contains, and something of the structure in place to manage it. More detail on CMMI contents is provided in subsequent chapters.

3.2 CMMI Objectives

While CMMI has many business-related benefits, the CMMI project as defined by its sponsors was directed toward the development of more efficient and effective process improvement models. It had both initial and longer-term objectives. The initial objective (represented in version 1.1 of the CMMI Product Suite) was to integrate three specific process improvement models: software, systems engineering, and integrated product and process development. This integration was intended to reduce the cost of implementing multidisciplinary model-based process improvement by accomplishing the following tasks:

- Eliminating inconsistencies
- Reducing duplication
- Increasing clarity and understanding
- Providing common terminology
- Providing consistent style
- Establishing uniform construction rules
- Maintaining common components
- Assuring consistency with ISO/IEC 15504

In the update to CMMI version 1.2, one objective of the CMMI Team was to improve and simplify the model as it applies to engineering development activities.[2] A second objective was to expand the scope of the model beyond the world of development to include both acquisition and the delivery of services.[3] Figure 3-1 illustrates these objectives and the product line approach developed by the CMMI Team. It remains to be seen whether in the future other disciplines and constellations[4] will be added to the CMMI Product Suite.

To facilitate both current and future model integration, the CMMI Team created an automated, extensible framework that can house model components, training material components, and appraisal materials. Defined rules govern the potential addition of more disciplines into this framework.

From the start, the CMMI project had to find an acceptable balance between competing requirements relating to change. The task of integration, which by its very nature requires change from each of the original single-discipline models, meant that all model users could expect new ways of thinking about process improvement to be needed in a CMMI environment. At the same time, an equally strong requirement called for protecting the investments in process improvement made by

2. The "CMMI Team" encompasses all who were and are involved in the CMMI project, including the Steering Group, the Product Team, and the Stakeholder Group. See Section 3.4 for a description of the CMMI Project Organization.
3. CMMI-DEV designates the CMMI model constellation that addresses development activities, and with the addition of IPPD included the designation becomes CMMI-DEV +IPPD; CMMI-ACQ designates the model constellation as it applies to acquisition organizations; and CMMI-SVC designates the model constellation (still being developed as we go to publication with this book) as it applies to service providers.
4. A constellation groups together the parts of the model that apply to a special area of interest.

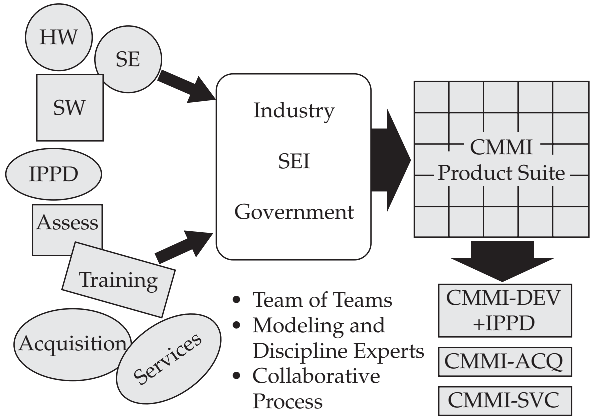


Figure 3-1: *The CMMI concept*

CMMI Milestones

1997	CMMI initiated by U.S. Department of Defense and NDIA
1998	First team meeting held
1999	Concept of operations released
	First pilot completed
2000	Additional pilots completed
	CMMI-SE/SW version 1.0 released for initial use
	CMMI-SE/SW/IPPD version 1.0 released for initial use
	CMMI-SE/SW/IPPD/SS released for piloting
2002	CMMI-SE/SW version 1.1 released
	CMMI-SE/SW/IPPD version 1.1 released
	CMMI-SE/SW/IPPD/SS version 1.1 released
	CMMI-SW version 1.1 released
2006	CMMI-DEV version 1.2 released
	CMMI-DEV +IPPD version 1.2 released
2007	CMMI-ACQ version 1.2 released

former users of those models, which meant controlling the introduction of new materials for each discipline. Judging from the significant rate of adoption throughout the world, we believe the CMMI project has achieved an appropriate balance between old and new.

3.3 The Three Source Models

To truly appreciate the significance of the CMMI accomplishments, you need to understand a bit of the history that led up to the development of the CMMI product suite. Of primary importance are the stories of the source models. Table 3-1 summarizes the three source models for CMMI.

Table 3-1: *Source Models for CMMI*

Model Discipline	*Source Model*
Software	SW-CMM, draft version 2(c)
Systems Engineering	EIA/IS 731
Integrated Product and Process Development	IPD-CMM, version 0.98

3.3.1 The CMM for Software

The character of software development sometimes seems closer to mathematics and art than it does to most other engineering disciplines. Software is inherently an intangible, intellectual development medium. No laws of physics govern its behavior; it is both marvelously and dangerously malleable. For this reason, it is critical that mature disciplines and processes be applied when working with software.

Software engineering and process management have been intimately associated since the pioneering work of Ron Radice and Richard Phillips in Watts Humphrey's group at IBM in the 1980s.[5] Basing their work on the tenets of the quality movement, Radice and Phillips led the

5. Radice, R. A., and R. W. Phillips. *Software Engineering, an Industrial Approach* (Vol. 1). Englewood Cliffs, NJ: Prentice Hall, 1988; Radice, R. A., J. T. Harding, P. E. Munnis, and R. W. Philips. A Programming Process Study. *IBM System Journals*, 2/3, 1999.

way in crafting a way to capture successful software development practices and then organize those practices so as to help struggling organizations get a handle on their processes and improve them. Given the nature of software development, it was not surprising that the large majority of the practices related to management discipline and processes.

Software Engineering

As defined in IEEE Standard 610.12, *software engineering* is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software—that is, the application of engineering to software.[6]

In 1986, Watts Humphrey, the SEI, and the Mitre Corporation responded to a request by the U.S. federal government to create a way of evaluating the software capability of its contractors. The group used IBM's concepts to create a software maturity framework, a questionnaire, and two appraisal methods. Over the next few years, this work was continued and refined.

In 1991, the SEI published the CMM for Software version 1.0, a model that describes the principles and practices underlying software process maturity. The CMM is organized to help software organizations improve along an evolutionary path, growing from an ad hoc, chaotic environment into mature, disciplined software processes. The CMM was used and evaluated for two years, and then revised and released as version 1.1 in 1993.[7] A similar revision was planned for 1997 as version 2.0;[8] this version was developed but never released as an independent model. However, the good work did not go to waste: The proposed revision was used as the source for the CMMI integration effort. In addition, two documents regarding software appraisals were used: the CMM

6. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Standard 610.12–1990. New York: Institute of Electrical and Electronics Engineers, 1990.

7. Paulk, Mark C., et al. *Capability Maturity Model for Software, Version 1.1.* CMU/SEI-93-TR-24; also ESC-TR-93-177. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1993; Paulk, M., et al. *The Capability Maturity Model: Guidelines for Improving the Software Process.* Boston: Addison Wesley, 1995.

8. For more on the history of the CMM for Software, see Paulk, Mark C. The Evolution of the SEI's Capability Maturity Model for Software. *Software Process—Improvement and Practice*, 1995, 3–15.

Appraisal Framework, version 1.0,[9] and the CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description.[10]

Software engineering's scope extends beyond the primary material contained in the CMM for Software to include software-related topics such as requirements elicitation, installation, operation, and maintenance. The CMMI model covers these areas in more detail through inclusion of appropriate material from the Systems Engineering Capability Model.

3.3.2 The Systems Engineering Capability Model

Systems engineering integrates all of the system-related disciplines so that systems meet business and technical needs in the most effective way, while striving to minimize local optimization and maximize return on investment. Another way of envisioning systems engineering is as the application of a set of rigorous engineering techniques to the solution of a complex technical problem.

Systems Engineering

INCOSE defines *systems engineering* as "an interdisciplinary approach and means to enable the realization of successful systems."[11]

It is difficult to fully understand the scope of systems engineering without looking at the various specialty disciplines associated with it. In *Essentials of Project and Systems Engineering Management*, Howard Eisner lists 30 key elements of systems engineering. These elements include such diverse areas as mission engineering, architectural design, life-cycle costing, alternatives analysis, technical data management, operations and maintenance, integrated logistics support, and reengineering.[12]

The systems engineering material in the CMMI has a complex history. In a modern-day "Tale of Two Capability Models," two organizations

9. Masters, S., and C. Bothwell. *CMM Appraisal Framework (CMU/SEI-95-TR-001)*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1995.

10. Dunaway, D., and S. Masters. *CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description (CMU/SEI-96-TR-007)*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, April 1996.

11. INCOSE Winter Workshop, January 1996.

12. Eisner, Howard. *Essentials of Project and Systems Engineering Management*. New York: Wiley Interscience, 1997.

undertook to model systems engineering practices. In August 1995, the Enterprise Process Improvement Collaboration (EPIC—a group of industry, academic, and government organizations) released the Systems Engineering Capability Maturity Model (SE-CMM). EPIC enlisted the SEI and architect Roger Bate to lead the development. The team pulled its systems engineering expertise primarily from aerospace and defense industry corporations and from the Software Productivity Consortium. The result was a model based on the appraisal model architecture contained in draft versions of ISO/IEC 15504 that addressed engineering, project, process, and organization practices.[13]

Around the same time that the SE-CMM was under development, INCOSE created a checklist for evaluating the capabilities of systems engineering organizations based on various engineering standards. Over time, this checklist evolved into a full-blown capability model known as the Systems Engineering Capability Assessment Model (SECAM). SECAM extended the SPICE concepts of a continuous model but focused more specifically on systems engineering practices than the SE-CMM, using EIA 632, "Processes for Engineering a Model," as the fundamental reference.

Needless to say, an environment with two models developed by two reputable organizations that purported to address the same issues was ripe for a model war. Which model would emerge as the "standard" by which organizations could be evaluated? After a year of heated discussions, in 1996 EPIC and INCOSE agreed to work together under the auspices of the Government Electronic and Information Technology Association (GEIA) of the Electronic Industries Alliance (EIA), with the goal of merging the two models into a single EIA standard. The result was an interim standard EIA/IS 731, "Systems Engineering Capability Model" (SECM).[14] By issuing the interim standard, the systems engineering community could apply a single, common description of systems engineering processes to the CMMI project.

13. In January 1993, an international working group (WG 10) was formed as part of subcommittee 7 (SC7) of the ISO/IEC Joint Technical Committee 1 (JTC1) to create a standard for software process assessment. Piloting of working drafts was accomplished through a project called SPICE (Software Process Improvement and Capability dEtermination). The combined effort of WG 10 and the SPICE project resulted in the development of ISO/IEC 15504 as a draft standard. At the time of writing the third edition of this book, an international standard has been released, and it has been refocused to address "process assessment" generally—that is, it is not limited just to software process assessment.

14. *EIA Interim Standard, Systems Engineering Capability*. EIA/IS 731 (Part 1: Model, EIA/IS 731–1, version 1.0; Part 2: Appraisal Method, EIA/IS 731–2, version 1.0), 1999. This was converted to a full standard, EIA 731, in 2002.

Systems engineering in CMMI remains heavily influenced by EIA 731. While echoes of the controversy between SECM and SE-CMM found voice in CMMI discussions, the resulting systems engineering content reflects an even stronger evolution of the original concepts. It preserves some of the innovations of EIA 731 while providing a more consistent underlying architecture compatible with the emerging international standards.[15] The standard includes both the SECM model (Part 1) and an appraisal method (Part 2).

3.3.3 The Integrated Product Development CMM

The source model for integrated product and process development was a draft of the Integrated Product Development CMM, known as IPD CMM version 0.98. This model had been developed almost to the point of its initial formal release when the CMMI project began in 1998.

From the outset, the CMMI Team wanted to include the concept of integrated product and process development (IPPD) in the CMMI product suite. This concept was fundamental to many of the large member corporations of NDIA, and it was strongly supported by the Department of Defense (DoD).[16] Unfortunately, the definition of IPPD used in the CMMI requirements document was derived from the DoD's experience with integrated operation of government system acquisition programs—and acquisition was not an initial discipline for CMMI. This discrepancy led to some difficulty in addressing the IPPD tenets within the CMMI scope. Adding to the confusion was a lack of consensus in the industry (and among members of the CMMI Team) regarding the fundamental concepts and best practices of integrated product development. Because it represented a relatively new means of organizing and accomplishing engineering work, there were nearly as many definitions as there were organizations.

This problem was not unique to CMMI. Indeed, the team established by EPIC to develop the IPD CMM, which was supported by many of the same members of the SE-CMM team, struggled with IPPD concepts for more than two years before being subsumed into the CMMI effort. The final draft IPD-CMM was established as a source document for CMMI, but the draft never achieved the status of a finished product.

15. For example, see ISO/IEC 15504 and ISO/IEC 15288.
16. The U.S. Air Force played a significant role in the development of the IPD CMM.

Integrated Product and Process Development

CMMI defines *integrated product and process development* as a systematic approach to product development that achieves a timely collaboration of necessary disciplines throughout the product life cycle, to better satisfy customer needs, expectations, and requirements.

IPPD emphasizes the involvement of stakeholders from all technical and business functions throughout the product development life cycle—customers, suppliers, and developers of both the product and product-related processes, such as testing and evaluation, manufacturing, support, training, marketing, purchasing, financial, contracting, and disposal processes. Clearly, implementing IPPD affects more than an organization's engineering processes and practices. Because it is essentially a way of doing business, it may radically change organizational structure and modify leadership behavior.

3.4 CMMI Project Organization

During the development phase that led to the initial CMMI materials, the project was organized in terms of a Steering Group, a Product Development Team, and a Stakeholder Group. In all, it involved the efforts of more than 200 people over a period of more than six years. The three groups comprised representatives from industry, government, and the SEI. Representatives of the disciplines whose models were to be integrated into CMMI were included in all three groups.

The Steering Group produced a list of requirements for CMMI, which was then reviewed by the Stakeholder Group and subsequently used by the Product Development Team to guide its creation of the CMMI products. The Product Development Team was a cross-disciplinary group created for the initial development work; it was charged with ensuring that the viewpoints and interests of each discipline were adequately considered in the integration process. The Stakeholder Group reviewed the initial draft CMMI materials, with its work being followed by a public review of a second round of draft materials, prior to the release of version 1.0 in late 2000. Taking advantage of early feedback from version 1.0 users, and responding to more than 1500 change requests, version 1.1 of the Product Suite was released in 2002.

The CMMI Team

The following organizations supplied members to the CMMI Team: ADP Inc., Aerospace Corp., AT&T Labs, BAE Systems, Boeing, CIBIT, Comarco Sys., Computer Sciences Corp., EDS, EER Sys., Ericsson Canada, Ernst and Young, General Dynamics, General Motors, Harris Corp., Hewlett-Packard, Honeywell, IBM, INCOSE, Institute for Defense Analyses, Integrated System Diagnostics, Jacobs Engineering, JPMorganChase, KPMG Consulting, Lockheed Martin, MitoKen Solutions, MITRE Corp., Motorola, NASA, Norimatsu Process Engineering Lab., Northrop Grumman, Pacific Bell, Q-Labs, Raytheon, Rockwell Collins, SAIC, SEI, Siemens, Spirula, SRA, SSCI, TeraQuest, THALES, U.S. DAU, U.S. DCMA, U.S. DLA, U.S. DoD, U.S. DHS, U.S. FAA, U.S. GAO, U.S. MDA, U.S. NRO, U.S. NSA, U.S. Air Force, U.S. Army, U.S. Navy, and Wibas.

The cross-disciplinary team that produced the initial CMMI models included members with backgrounds in software engineering, systems engineering, and integrated product and process development. Most engineering organizations maintain these skills, but the manner in which they are aligned and interact varies across organizations. Thus the CMMI Team not only had to resolve differences between the three source models, but also had to bridge the cultural, linguistic, and professional differences separating engineering specialties and organizations. The bridges that had to be built in constructing the CMMI models serve as precursors of those that users of the models will need to construct to successfully support integrated process improvement and process appraisal.

During the CMMI development effort, the CMMI Team actively sought to keep balanced membership across the three disciplines. This move was supported by the strong interest espoused by the software and systems engineering communities. Thanks to the wide acceptance of the

CMM Software, strong advocacy was provided by the SEI and organizations that had used the model, understood its value, and wanted to see that value preserved in the integrated CMMI models. Likewise, in the systems engineering world, the International Council on Systems Engineering (INCOSE) advocated inclusion of systems engineering practices. Even the integrated product and process development community was represented on the CMMI Team, albeit with members voicing a somewhat wider range of views on how integrated product and process development should be handled than did the more established discipline representatives. In the end, this team of experienced and active people, each of whom brought his or her specific expertise and preferences to the table, came together to create the CMMI product suite.

Once the development phase of the initial CMMI product suite was complete, a new organizational structure was established (see Figure 3-2). That is, the CMMI Team evolved into the CMMI Product Team that exists today. This team has access to expert groups for software, systems engineering, integrated product and process development,

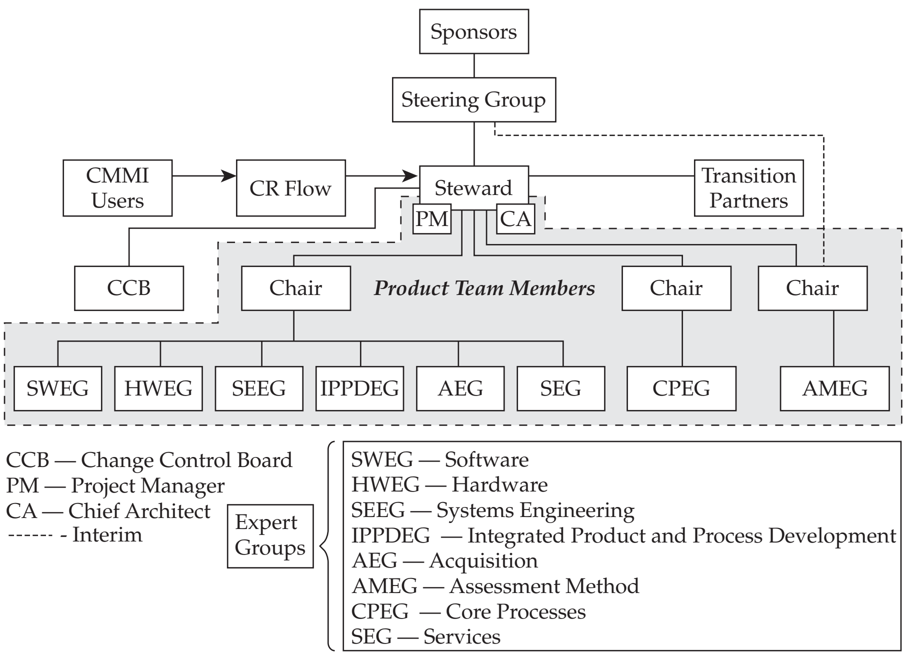


Figure 3-2: *Current CMMI project organization*

supplier sourcing, appraisals, and the core CMMI components. A configuration control board was established to guide CMMI evolution, and the SEI was named as the Steward of the CMMI Product Suite. In its role as Steward, SEI is responsible for maintenance and support of CMMI. As time goes on, new cross-functional teams of experts will be required to handle revisions of existing products and the subsequent work of adding disciplines to the CMMI framework such as the Services Constellation Team and the Acquisition Constellation Team.

Index

A

B

C

D

E

N

O

S

T

U

V

W