CrossTalk

February 2007 **The Journal of Defense Software Engineering** Vol. 20 No. 2

CMMI®

CMMI

4 CMMI V1.2: What Has Changed and Why
This article provides a view of what has been included – and not included – in Capability Maturity Model Integration Version 1.2 (CMMI V1.2), describing the major elements of change for each CMMI product.
by Mike Phillips

8 Measure Twice and Cut Once
This article describes how the 309th Software Maintenance Group used Standard Capability Maturity Model Integration Appraisal Method for Process Improvement B to identify opportunities for additional improvements, educate key project personnel in the organization on best-practices, and obtain critical buy-in from project personnel.
by Rushby Craig

13 CMMI Level 2 Within Six Months? No Way!
The purpose of this article is to show that when an organization is already doing competent project management, the effort to benchmark that capability by using CMMI is almost straightforward, and it is possible to achieve a Level 2 CMMI appraised rating within six months.
by George Jackelen

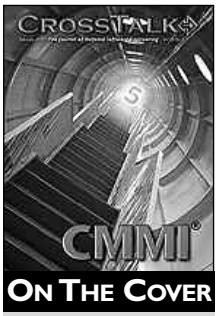
Best Practices

17 Future Directions in Process Improvement
This article describes how and why development processes, methods, and management must change and the likely future directions of process-improvement evolution.
by Watts S. Humphrey, Dr. Michael D. Konrad, James W. Over, and William C. Peterson

Software Engineering Technology

23 The *ImprovAbility* Model
This model helps users identify strengths and weaknesses that can be leveraged or avoided to help an organization get the most from its process improvement effort.
by Dr. Jan Pries-Heje, Jørn Johansen, Mads Christiansen, and Morten Korsaa

29 Applying International Software Engineering Standards in Very Small Enterprises
This article discusses the utilization of ISO/IEC JTC 1/SC7 in very small enterprises.
by Claude Y. Laporte, Alain April, and Alain Renault



ON THE COVER

Cover Design by Kent Bingham

Additional art services provided by Janna Jensen

Departments

3 From the Sponsor

12 Call For Articles Ad

16 More Online From CROSSTALK

22 Coming Events

30 SSTC 2007

31 BACKTALK

CROSSTALK

CO-SPONSORS:

DoD-CIO	*The Honorable John Grimes*
NAVAIR	*Jeff Schwalb*
76 SMXG	*Kevin Stamey*
309 SMXG	*Randy Hill*
402 SMXG	*Diane Suchan*
DHS	*Joe Jarzombek*

STAFF:

MANAGING DIRECTOR	*Brent Baxter*
PUBLISHER	*Elizabeth Starrett*
MANAGING EDITOR	*Kase Johnstun*
ASSOCIATE EDITOR	*Chelene Fortier-Lozancich*
ARTICLE COORDINATOR	*Nicole Kentta*
PHONE	(801) 775-5555
E-MAIL	crosstalk.staff@hill.af.mil
CROSSTALK ONLINE	www.stsc.hill.af.mil/crosstalk

CROSSTALK, The Journal of Defense Software Engineering is co-sponsored by the Department of Defense Chief Information Office (DoD-CIO); U.S. Navy (USN); U.S. Air Force (USAF); and the U.S. Department of Homeland Security (DHS). DoD-CIO co-sponsor: Assistant Secretary of Defense (Networks and Information Integration). USN co-sponsor: Naval Air Systems Command. USAF co-sponsors: Oklahoma City-Air Logistics Center (ALC) 76 Software Maintenance Group (SMXG); Ogden-ALC 309 SMXG; and Warner Robins-ALC 402 SMXG. DHS co-sponsor: National Cyber Security Division of the Office of Infrastructure Protection.

The USAF Software Technology Support Center (STSC) is the publisher of CROSSTALK, providing both editorial oversight and technical review of the journal. CROSSTALK's mission is to encourage the engineering development of software to improve the reliability, sustainability, and responsiveness of our warfighting capability.

S T S C

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address. You may e-mail us or use the form on p. 28.

517 SMXS/MXDEA
6022 Fir AVE
BLDG 1238
Hill AFB, UT 84056-5820

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the CROSSTALK editorial board prior to publication. Please follow the Author Guidelines, available at <www.stsc.hill.af.mil/crosstalk/xtlkguid.pdf>. CROSSTALK does not pay for submissions. Articles published in CROSSTALK remain the property of the authors and may be submitted to other publications.

Reprints: Permission to reprint or post articles must be requested from the author or the copyright holder and coordinated with CROSSTALK.

Trademarks and Endorsements: This Department of Defense (DoD) journal is an authorized publication for members of the DoD. Contents of CROSSTALK are not necessarily the official views of, or endorsed by, the U.S. government, the DoD, or the STSC. All product names referenced in this issue are trademarks of their companies.

CrossTalk Online Services: See <www.stsc.hill.af.mil/crosstalk>, call (801) 777-0857 or e-mail <stsc.webmaster@hill.af.mil>.

Back Issues Available: Please phone or e-mail us to see if back issues are available free of charge.

Axiomatic Improvement

An expert is an individual possessing special skill or knowledge representing mastery of a particular subject. Webster defines wisdom as accumulated philosophic or scientific learning, the ability to discern inner qualities and relationships, good sense, and a wise attitude or course of action. Even after more than 15 years of working in and managing a software engineering organization utilizing model based process improvement, I still cannot claim to be an expert on capability maturity models. However, I have accumulated some amount of wisdom regarding the subject. Drawing from those years of experience and acquired wisdom, I believe the following list of my thoughts and observations on the subject of software process improvement to be axiomatic:

1. There is always an opportunity (often a great need) for process improvement in software development endeavors.
2. Properly implemented model-based process improvements can produce real, frequently significant improvements.
3. The Capability Maturity Model and all of its derivatives are excellent tools, but none of them are perfect (there is always an opportunity for improvement even within the models themselves).
4. Implementing process improvement requires change, and affecting change is hard, tedious work.
5. Process improvement activities must be planned and managed like any other complex project in order to have any hope of success.
6. Not all organizations, projects, teams, or individuals accept change in the same way or at the same rate.
7. Not every attempt at process improvement works.
8. Every non-attempt at improvement is a guaranteed failure to improve – unfortunately, improvement doesn't occur on its own.

The articles in this issue of CROSSTALK do an excellent job of expanding on nearly all of my personal observations. Watts S. Humphrey, Dr. Michael D. Konrad, James W. Over and William C. Peterson's article *Future Directions in Process Improvement*, and Mike Phillips' article *CMMI V1.2: What Has Changed and Why* both address the need for process improvement models to constantly evolve and change. Rushby Craig's article *Measure Twice and Cut Once* and George Jackelen's article *CMMI Level 2 Within Six Months? No Way!* describe the necessity for rigorous planning and oversight required for successful process implementation efforts. Dr. Jan Pries-Heje, Jørn Johansen, Mads Christiansen, and Morten Korsaa's description of the ImprovAbility Model in *The ImprovAbility Model* is a fascinating description of a new tool used to evaluate an organization's ability to improve. The tool identifies the numerous variables that come into play when an attempt is made at implementing process improvements within software organizations. We conclude with *Applying International Software Engineering Standards to Very Small Enterprises*. While I wholeheartedly believe in mature processes for software development, we must still implement these processes in ways that make sense for individual organizations. I hope you will enjoy reading the articles as much as I did. They may not make you an expert, but I'm certain you will gain knowledge and wisdom.

Randy B. Hill
Ogden Air Logistics Center, Co-Sponsor

CMMI V1.2: What Has Changed and Why

Mike Phillips
Software Engineering Institute

This article provides a view of what has been included – and not included – in Capability Maturity Model® Integration Version 1.2 (CMMI® V1.2) for CMMI users who are familiar with the products. CMMI V1.2 products, including CMMI for Development, V1.2 (the model), Standard CMMI Appraisal Method for Process Improvement (SCAMPI℠), V1.2 (the appraisal method), and Introduction to CMMI, V1.2 (the training), was released on August 25, 2006. I describe the major elements of change for each of these CMMI products. Draft V1.2 products were approved, piloted, and revised to ensure that the proposed changes actually improved the quality of the model, method, and training materials – and did no harm to existing improvement efforts and investments already made by those who used CMMI V1.1. I also seek to add some idea of why many of these changes were made.

For the CMMI product suite, the development of V1.2 has improved in three dimensions for each of the products that comprise the product suite. In one dimension, the emphasis was to clarify and simplify. In the opposite dimension, the effort was to position each of the products for potential expansion of the life cycle or expansion into new and related areas of interest. Overarching these dimensions was a growing recognition that all of the elements of the product suite could be strengthened to increase user confidence that appraisal results accurately reflect genuine process improvement.

What Are the Major Changes?

The CMMI framework is a repository of elements from which CMMI products are built. For the framework, V1.2 improvements resulted in a new architecture that allows the creation of new groupings of CMMI products called *constellations*. The word *constellation* refers to a set of model components, training materials, and appraisal documents in the CMMI framework that covers an area of interest such as development, services, or acquisition.

The result for the V1.2 model is that what once was CMMI V1.1 was improved and is now part of the development constellation. Therefore, the V1.2 constellation, called CMMI for Development, has two member models: CMMI for Development and CMMI for Development + Integrated Product and Process Development (IPPD). Both models have 22 process areas (PAs). I address the PAs more thoroughly further in the article.

For the appraisal method, SCAMPI V1.2, improvements focused on the clarification of terms that had proven problematic, such as the use of *face-to-face* interviews in organizations that are virtual or have multiple and distant sites. The appraisal team has addressed requests for more flexibility in breaking up appraisal activities (particularly across multiple sites) without compromising the confidence in appraisal results. Also added are new approaches to broaden sampling across the organizational unit being appraised to build confidence in process institutionalization. Although SCAMPI B and C methods (less stringent appraisal methods than the more well-known SCAMPI A method, which do not result in maturity level or capability level ratings) were developed under the existing V1.1 approach, the thought regarding having several classes of ratings to make up an appraisal family (SCAMPI As, Bs, and Cs) has been clarified in the V1.2 release.

The training approach for V1.2 also got a start under V1.1. The CMMI Steering Group's agreement to have a single introduction to CMMI course rather than separate ones for the two representations of the model (staged and continuous) was accomplished early in the V1.2 development schedule. Today, the single course has been updated to reflect the model changes described in more detail to come.

At the Software Engineering Institute (SEI), we are applying similar improvements to related courses, such as the Intermediate Concepts of CMMI course that we use to groom CMMI subject matter experts, including Introduction to CMMI instructors and SCAMPI lead appraisers. To date, we have offered the Intermediate Concepts of CMMI course for those leading improvement efforts in their organization, even if they do not wish to become instructors or lead appraisers. We are now pursuing the creation of a CMMI Deployment and Interpretation course that will better serve this audience.

A new approach that was instituted with V1.2 is an online upgrade course. While we provide the essential elements of change in the CMMI model, the SCAMPI Method Definition Document, and the Introduction to CMMI training in material provided free on the Web site, we have added both the refresher material and more advanced training material in CMMI V1.2 Upgrade Training for all those who must be able to apply CMMI principles on appraisals. A more detailed CMMI V1.2 Upgrade Training course is available to those who are instructors or lead appraisers or are along the path toward being one. The course for instructors and appraisers is part of the annual partner/fee structure. The upgrade course, available for everyone else is available on the SEI Web site where users can register and complete the upgrade course online for $175.

Now Tell Me What the Actual Changes Are

Simplification: Three Fewer PAs for the Model, With IPPD and Supplier Sourcing Simplified

More than 80 percent of the appraisals performed using CMMI V1.1 used models that did not extend beyond systems engineering and software engineering (i.e., they did not use models containing supplier sourcing or IPPD), despite the use of team-based development (where IPPD practices would be useful) and of complex, multi-company developments (where supplier sourcing practices would be useful). The CMMI development team felt that by consolidating the material in each of the areas, it could improve the use

® Capability Maturity Model, CMM, and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
℠ SCAMPI is a service mark of Carnegie Mellon University.

of these practices while simplifying CMMI models.

An approach suggested by many change requests received from CMMI users was to combine Integrated Supplier Management (ISM), which comprised the supplier sourcing addition, with Supplier Agreement Management (SAM), which was part of the software and systems engineering portion of the models. While ISM was designed for an environment in which process understanding is maintained across organizations and SAM was designed for an environment that would not necessarily require such understanding, the overlap between these two PAs was troubling.

The resulting change for V1.2 is that the informative material was strengthened in SAM about effective sourcing, and two specific practices were added to address the kind of enhanced visibility of supplier progress that ISM covered. Since one specific practice, *Analyze COTS* (commercial off-the-shelf), was refocused as informative material within SAM and sub-practices in Technical Solution (TS), the net increase for SAM is one additional specific practice.

The two new SAM-specific practices are the following:

- Monitor selected supplier processes.
- Evaluate selected supplier work products.

These two practices are added with the understanding that the process monitoring and work product evaluation opportunities will be as described in the established agreements with the project's suppliers. Not all agreements will allow close scrutiny by the project and not all products provided by suppliers will need that level of scrutiny to avoid system development risk.

When the development team first sought to address IPPD in CMMI, we placed many of the concurrent engineering (i.e., a non-linear approach to product design and engineering) concepts throughout the model. We then used two approaches to address team-based behaviors. In the case of the Integrated Project Management (IPM) PA, we added two goals that were team-centric and would only be used if the IPPD was selected. We then added two additional PAs to capture team-based thinking: Organizational Environment for Integration (OEI) and Integrated Teaming (IT).

For V1.2, we determined that the approach could be simplified if we added a goal to Organizational Process Development (OPD) to address the organizational commitment to IPPD and then consolidated the material from IT into IPM. This simpler approach has greatly reduced the number of practices and PAs that are unique to team-based development. IPPD will now be addressed with only one approach for expansion – the inclusion of one additional IPPD goal in OPD (to address the organizational behaviors) and a *single* goal in IPM (to address the project behaviors). These two goals, which replace the five IPPD goals in V1.1, are the following (revision shown in Figure 1):

- Enable IPPD management (in OPD).
- Apply IPPD principles (in IPM).

Simplification: Eliminating Common Features and Advanced Practices

A legacy from the Capability Maturity Model® for Software (SW-CMM®) was the use of *common features* as a method of describing the different roles that generic practices fulfill in assuring institutionalization of the model's intent across the organization. While this concept may be useful in training, it complicates model depiction. We felt it was time to move to a simpler approach of simply numbering the generic practices. Therefore, V1.2 models no longer contain common features as a way to organize the generic practices.

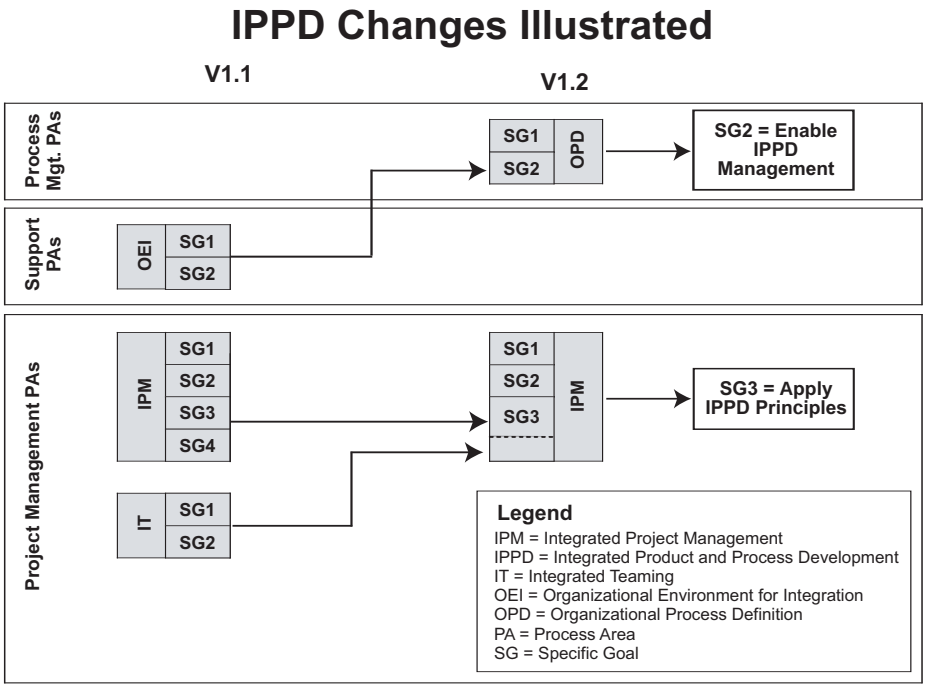
More difficult was resolving the legacy from the Systems Engineering Capability Model (SECM) Electronics Industries Alliance (EIA)-731, the advanced practices that we had placed in the engineering PAs. We felt that while the idea of advanced practices made sense, they were less valuable in the existing model structure because they added complexity without providing strong differentiation between base and advanced practices. Further, advanced practices seemed to complicate appraisals. Therefore, V1.2 models no longer contain advanced practices. All specific practices are now considered to be at capability level 1.

Expansion: Hardware Engineering Amplifications and Work Environment Coverage

A hardware engineering team was chartered with finding ways to ensure that CMMI adequately addressed the hardware aspects of product development that were sometimes perceived to be missing from earlier versions of CMMI. Much of this work is now reflected in additional hardware engineering examples throughout the model, sometimes within hardware engineering amplifications and sometimes in lists of examples representing multiple aspects of product development. This addition of examples resulted in a reduction in the total number of amplifications in the model.

We typically considered it better to cover product development examples together rather than seek to separate them into software examples, hardware examples, etc. Therefore, the additional hardware engineering material, when possible, was added as material that all would see as part of the development model, rather than an amplification that only some may read. The final result for V1.2 is that the hardware amplification (i.e., labeled *For Hardware Engineering*) were limited to only

Figure 1: *How IPPD Material Was Moved for V1.2*



six and the software amplification (i.e., labeled *For Software Engineering*) were reduced to only eight. An example of hardware amplification is found in Technical Solution, specific practice 2.1:

> **For Hardware Engineering:** Detailed design is focused on product development of electronic, mechanical, electro-optical, and other hardware products and their components. Electrical schematics and interconnection diagrams are developed, mechanical and optical assembly models are generated, and fabrication and assembly processes are developed.

Work that explored future focus areas such as security and safety resulted in a proposal to include a new PA in V1.2 that covered the work environment (i.e., a work environment PA was proposed). However, further investigation revealed that we could cover the basics of work environment material just as we had for data management by creating two practices to address the concept.

These two practices were added to the same PAs as the new IPPD-related goals – OPD and IPM. A practice in OPD expects organizational attentiveness to effective work environment practices, and IPM expects deployment of these practices to the individual projects. These two specific practices are the following:

- Establish work environment standards (in OPD).
- Establish the project's work environment (in IPM).

Not Applicable PAs

With the release of V1.2, the potential for maturity level variability has been significantly reduced. In both V1.0 and V1.1, we described in Chapter 6 that PAs could be determined to be not applicable for organizational process improvement. One of the heritage models, the SW-CMM, had always allowed Software Subcontract Management (SSM) to be considered *not applicable*. The CMMI equivalent, SAM, was highlighted in the Chapter 6 discussion as the example of a PA potentially considered *not applicable* in CMMI.

The number of organizations seeking to exclude this type of PA from their appraisals dropped from 58 percent with the SW-CMM to 20 percent with CMMI, but we knew that some organizations, particularly small software developers, had no critical suppliers so that an allowance for exclusion remained important. However, the model text did not identify this as the only acceptable PA for consideration. We had a few other PAs declared *not applicable* for various reasons, but our view was that continuing to accommodate these exclusions diminished the confidence in the benchmark associated with maturity level appraisal results. (Appraisals using the continuous approach and not seeking staged equivalence, of course, allow any of the options desired for process improvement without providing potentially misleading results.)

Version 1.2 addressed this issue in both the model and the method. The V1.2 model no longer discusses *not applicable* status. The needed procedures for the appraisal team's determination are now part of the SCAMPI Method Definition Document. We will rely on the appraisal team to determine, prior to the appraisal onsite, if the SAM practices are needed in the organizational unit being appraised or not. The appraisal disclosure statement will include a statement about the lack of suppliers needing management, if the team makes that determination.

> *"Appraisals using the continuous approach ... allow any of the options desired for process improvement without providing potentially misleading results."*

Appraisal Validity Period

The CMMI Steering Group has determined that some sense of *lifetime* needed to be defined for CMMI appraisals. After extended discussions, the Steering Group determined that a three-year validity period, similar to that established for ISO 9000:2000, would be the most reasonable length of time. (We have frequently mentioned that there are often other significant reasons to question the maintenance of process capability, such as reorganizations or mergers and acquisitions.)

So how will this approach be phased in? The first part is easy. All future appraisals, both V1.1 and V1.2, will be considered valid for three years from the date of completion, as noted on the appraisal disclosure statement. When two years have passed without a new appraisal covering the organization, the SEI will contact the sponsor of the two-year-old appraisal to remind them of the three-year validity rule. At the three-year-point, publicly available appraisals on the SEI Web site <http://sei.cmu.edu/pars/> will be removed.

But what about already performed appraisals? Here, the planned availability of V1.2 causes a need for flexibility, as we want to encourage a smooth transition to the improved version. We therefore will consider existing appraisals older than three years valid for a full year after the release of V1.2, done in August 2006. This plan allows time to plan and execute appraisals using the V1.2 product suite. Further, we will continue to recognize V1.1 appraisals through most of 2007 in case the concerns about change are greater than what we currently expect.

Although we no longer publish SW-CMM appraisal results, we felt it appropriate to establish a *validity period* for these as well. The choice in this case, since all recognized appraisals had to be completed by the *sunset* of December 2005, was to choose a single date: December 2007. This plan leaves CMM users with some flexibility – more than a year and a half – to make the transition to CMMI, and to use either V1.1 or V1.2.

Discipline Distinctions

With the first two releases of CMMI, it was important to recognize which disciplines the models covered (e.g., software engineering, systems engineering), along with recognizing the heritage of the improvement models for each of the disciplines (i.e., material from the three source models: the SW-CMM, EIA 731, and the Integrated Product Development-CMM). However, over the years, these distinctions have become less important, and the unifying engineering development processes have demonstrated synergies that go beyond the original source models. We were also asked by users and the CMMI Steering Group to simplify the material.

The increasing number of possible model variations (e.g., CMMI-SE, CMMI-SE/SW/IPPD, CMMI-SW/IPPD), and therefore printed models, to address the various combinations of engineering disciplines made movement in that direction undesirable. Instead, we added amplifications for hardware engineering examples, but chose not to call out another model variation in the model name. Nor are multiple model documents available for users to choose from. Instead, there is one integrated model document containing the best development practices.

Changes to CMMI Beyond CMMI for Development

As we began to consider future coverage of organizational process improvement, we sought to maintain the greatest possible commonality among all the models created from the CMMI common framework of best practices. Figure 2 depicts the desire for commonality and needed specificity. This approach provides a way to avoid any CMMI model to grow too large for effective use.

Based on the initial efforts to maximize commonality among CMMI models, 16 of the 22 PAs of CMMI V1.2 comprise the process improvement CMMI Model Foundation for the three areas of interest currently being pursued: development, acquisition, and services. The 16 PAs (in alphabetical order) are the following:

1. Causal Analysis and Resolution (CAR).
2. Configuration Management (CM).
3. Decision Analysis and Resolution (DAR).
4. Integrated Project Management (IPM).
5. Measurement and Analysis (MA).
6. Organizational Innovation and Deployment (OID).
7. OPD.
8. Organizational Process Focus (OPF).
9. Organizational Process Performance (OPP).
10. Organizational Training (OT).
11. Process and Product Quality Assurance (PPQA).
12. Project Monitoring and Control (PMC).
13. Project Planning (PP).
14. Quantitative Project Management (QPM).
15. Requirements Management (REQM).
16. Risk Management (RSKM).

Each constellation includes the common parts of the 16 PAs above, with additions unique to the area of interest covered, or shared across some, but not all, of the constellations.

We recognized that even with the CMMI Model Foundation, we needed to allow some flexibility. No flexibility is allowed, however, for the *required* (i.e., specific goals and generic goals) or *expected* (i.e., specific practices and generic practices) components of the 16 PAs that make up the model foundation. Additions to these PAs will be allowed, just as the IPPD addition is allowed (and encouraged) in the development constellation.

In the informative material, we allow a little more flexibility so that typical work products can be added or substituted to fit a process area in each constellation. The only other substitutions or deletions allowed within these 16 PAs will be the informative material judged specific to development. This occurs in the current model in subpractices, where development-specific explanations are often found. These statements may be tailored to the needs of the new constellation. These include informative paragraphs below sub-practices and generic practice elaborations.

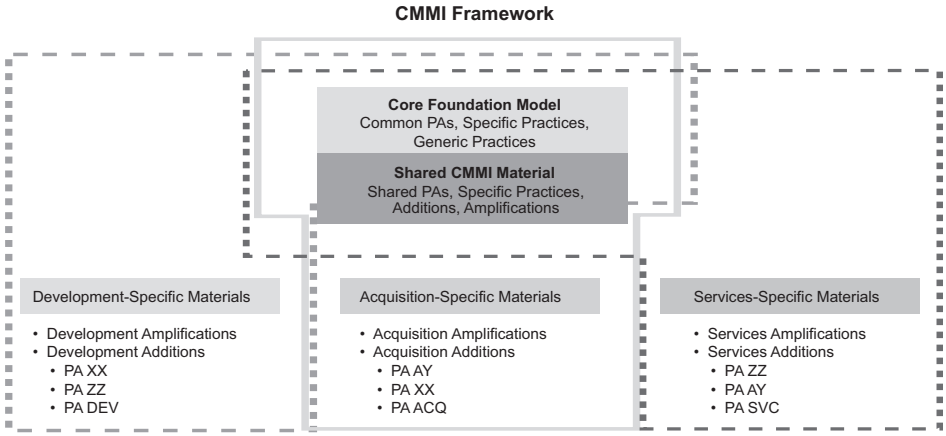


Figure 2: *How the CMMI Constellations Interact*

More tailoring is permitted to describe activities captured primarily in the engineering PAs of CMMI-DEV. While some of the constellations may share components with the engineering PAs in CMMI-DEV, the shared material may be arranged and grouped differently to meet the needs of the constellation's user base. If these adjustments change the PA in any significant way, the PA will be given a different name to avoid confusion in use, training, or appraisal. If two constellations find that a particular PA can be shared, then these PAs will be designed to capture that commonality as well. For example, the existing Verification or Validation PAs might be usable in one of the future models but not in others, so it would be shared across two constellations.

Summary

With V1.2, we sought to address a number of needed changes. Many of you, as CMMI users, gave us your thoughts on changes to improve CMMI. You may see, in the changes, something that you suggested. You may see areas changed in ways a bit differently than you suggested but similar in intent. And there may well be changes that you recommended, particularly expansions that we did not include this time.

Improvements will continue to be needed, and future updates to our constellations will continue to be made. We hope that this set of changes will simplify, add some needed coverage, and, most importantly, increase the confidence that the community appraisal results do represent faithfully the sincere efforts in process improvement that you and your peers have made in your organizations.◆

About the Author

Mike Phillips is the program manager for CMMI V1.2 at the SEI. Previously, he was responsible for transition enabling activities at the SEI. Phillips has authored technical reports, technical notes, CMMI columns, and various articles in addition to presenting CMMI material at conferences around the world. Prior to his retirement as a colonel from the Air Force, he managed the $36 billion development program for the B-2 in the B-2 System Program Office and commanded the 4950th Test Wing at Wright-Patterson AFB, OH. Phillips has a bachelor's degree in Astronautical Engineering from the Air Force Academy, a master's degree in Nuclear Engineering from Georgia Tech University, a master's degree in Systems Management from the University of Southern California, and a master's degree in International Affairs from Salve Regina College and the Naval War College.

SEI
4500 Fifth AVE
Pittsburgh, PA 15213
Phone: (412) 268-5884
Fax: (412) 268-5758
E-mail: dmp@sei.cmu.edu

Measure Twice and Cut Once

Rushby Craig
309 Software Maintenance Group, Hill Air Force Base

This article describes how the 309th Software Maintenance Group (SMXG) at Hill AFB, Utah, used Standard Capability Maturity Model® Integration Appraisal Method for Process Improvement (SCAMPI) B appraisals as means to identify value-added process improvements, educate key project personnel in the organization on best practices, and obtain critical buy-in from project personnel within the framework of an organizational transition from use of the Capability Maturity Model (CMM) to the Capability Maturity Model Integration (CMMI) as a model of best practices. Action plans were created based on weaknesses identified in the SCAMPI B appraisals and were used to baseline and track progress on the implementation of process improvements. A series of these SCAMPI B appraisals was followed by a SCAMPI A appraisal, in June 2006, when the 309th SMXG was awarded a CMMI Maturity Level 5 rating. Details on the strategy used and lessons-learned are shared.

Anyone with experience in carpentry has at some time made a mistake and cut a piece of wood too short for its intended purpose. What happens then? After some initial words of frustration, the piece is either scrapped, saved to be used for another part that is smaller, or the finished product becomes smaller in one dimension than initially planned.

I do a little woodworking as a hobby. Over the years, my dad has been my chief mentor. One of the lessons that he taught me – which I will never forget – is to measure and mark very carefully before cutting. I have found that it is a good idea to double (or even triple) check your measurements and markings to make sure that they are correct. When I have made the mistake of cutting a piece too long, I feel fortunate because at least I get another chance to cut it correctly the second time. Often though, I make the mistake of cutting a piece too short. When this happens, my dad will repeat an old adage that is simple to remember: *Measure twice, cut once*. These words always remind me to take extra care to follow some simple practices that help me make fewer mistakes.

You might say that the same principle applies to SCAMPI A appraisals in an organization. The time, effort, and cost associated with such an appraisal are substantial. Before going through with SCAMPI A, you want to have some confidence that you will meet the goals that you have for the appraisal. You do not want to go through the trouble of preparing for and holding a SCAMPI A unless you have some confidence that the goals of the appraisal will be met. The way to do that is to measure the readiness of the organization over a period of time. There are without a doubt different ways that organizational readiness can be measured. We as an organization thought quite a bit about how to prepare the organization for a SCAMPI A and came to some conclusions based on lessons learned from other organizations, and also from our own past experiences.

Background

Why a SCAMPI A Appraisal?

So why did we decide that we wanted to have a SCAMPI A appraisal? Well, that question begs some explanation and provokes a little history lesson.

In 2002, with the Software Engineering Institute (SEI) having already announced the future sunset of the CMM and the CMM Based Assessment for Internal Process Improvement (CBA-IPI) method, the 309th SMXG made a decision to transition from the CMM to the newer CMMI model. The organization had been rated CMM Maturity Level 5 in 1998, and it was time to have a formal reappraisal with the CMM. But rather than making a substantial investment in a CBA-IPI and then moving on to the CMMI, it was decided that the organizational strategy would be to transition to the new model with the goal of achieving Level 5 in a couple of years.

As an organization, the SMXG has a set of strategic plans and goals that have been established by our senior management with widespread input from below. One of the goals established was for the organization to be appraised at Maturity Level 5 in the CMMI. This goal was linked to several other goals in areas such as improvement in cost and schedule performance, quality improvement, organizational growth, business development, facilities expansion and improvement, etc. For them, maintaining Maturity Level 5 was a strategic investment because it ensured that they were following recognized best practices, that they were continuously focusing on improvement, and that their current and potential customers could have confidence that projects will be planned, executed, and monitored properly.

Transitioning to the CMMI

The transition plan from CMM to the CMMI for the organization was mapped out into three phases: Organization Documentation and Implementation, Product-Line Documentation and Implementation, and Appraise Maturity (see Figure 1).

Phase 1: Organization Documentation and Implementation

The goal of this first phase was to do the following: Review the current policy, plans, and processes used within the organization to determine where changes needed to be made to ensure compliance with the CMMI; update the applicable documents as necessary; and ensure implementation at the organization level.

Process Action Teams (PATs) were established and worked for approximately eight months, defining issues and finding solutions. The decision to use PATs was made in order to get involvement and buy-in to the needed changes from a broad cross-section of the organization. Four PATs were created, one for each category of Process Areas (PAs) in the CMMI: Process Management, Project Management, Engineering, and Support. These categories (used in the continuous representation of the CMMI) were convenient for partitioning the work that needed to be done by the PATs because of the common themes and threads within each set of PAs (even though they were using the staged representation). Each PAT had between six and 10 people

assigned from the ranks of projects, quality assurance (QA), the Software Engineering Process Group (SEPG), and management. All of these personnel were working on this task part-time, in addition to their regular duties. At the conclusion of this phase, the PATs submitted change proposals to the executive board of the organization for approval before implementation. The executive board and their technical advisors were responsible to ensure that recommendations and approaches were sound and consistent and to give final approval for implementation. Where conflicts existed between the recommendations of the PATs or changes were not accepted, the PATs were given assignments to work out these issues. Correction of these conflicts spanned nearly two months.

Phase 2: Product-Line Documentation and Implementation

Once organization-level policies and processes were in place, they needed to be implemented at the product line and project level. This phase was similar in many ways to Phase 1 except for the level at which implementation was focused. PATs were established in the product lines and included individuals who had participated in defining policy and processes at the organizational level in Phase 1. This ensured that the people who represented the product line in the organizational-level PATs were there to assist the product-line PATs in interpreting the changes.

The product-line PATs had to review their processes to find where gaps had been created by changes at the higher level, to find solutions to the gaps in the context of their business environment, to implement these in their processes, and to train their personnel on the changes. In many ways, this proved to be more difficult than the prior phase because it is usually at this level that the *rubber meets the road*, so processes had to be written and packaged in a way that made sense in the product lines. It was also more difficult to implement practices at this level because of the large numbers of people that had to train on, had to buy into, and had to start following these practices.

Phase 3: Appraise Maturity

In this phase, the goals were to accomplish the following:

1. Look closely at what had been implemented, make some judgments about how well the organization was satisfying CMMI practices, and provide some feedback to the organization that could be used to make corrections where practices were not implemented sufficiently.
2. When ready, hold a formal appraisal where rating of a maturity level in the CMMI could be made. The tools we decided to use to meet these goals were a series of SCAMPI B appraisals, followed by a SCAMPI A appraisal.

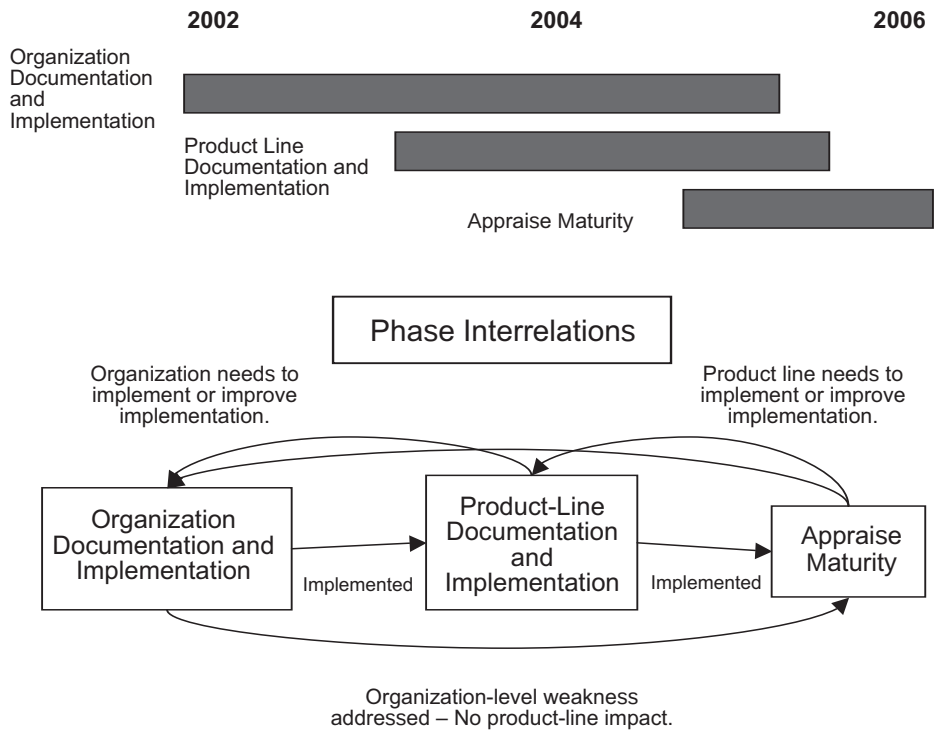


Figure 1: *CMM to CMMI Transition Strategy*

Within this phase, they implemented a plan for how the appraisals would be structured. You could say that the Appraise Maturity phase was made up of three rounds of appraisals. The first round was to use the SCAMPI B method and was intended to baseline where the organization and the projects being looked at were at that point in time. It provided a basis for the initial action plans. The second round of appraisals was also to use SCAMPI Bs, but this round was intended to measure, in an appraisal environment, how much improvement had taken place at approximately the mid-point of the Appraise Maturity phase. The third round in this phase was the SCAMPI A appraisal.

The SMXG decided to select a number of projects that would represent the organization well in terms of workload performed and in terms of numbers of people assigned to them. Other projects in the organization that were not included in the SCAMPI B appraisals were not ignored. They were expected to implement the same changes as projects involved in the appraisals; the mechanism being used to monitor their progress was done using our standard internal QA audit function.

After much thought and discussion, they settled on six projects to participate in the SCAMPI Bs. The projects included in these appraisals were typically the larger and most mainstream in the organization. The same set of six projects was to be examined in both of the first two appraisal rounds.

Overlapping Phases

The three phases of the transition strategy overlapped each other as shown in Figure 1. The phases were not a true waterfall model where each phase would be completed before moving on to the next one. When the product lines started tailoring their processes to be compliant with new policy and organizational-level processes, it sometimes pointed out problems with these higher-level documents that needed to be addressed. This required that activities be done in Phase 1 again. Likewise, when appraisals were performed, it pointed out cases where organization, product line, or project approaches or documents were deficient and needed correction, and required reentry into the prior two phases.

Collaborative Approach to SCAMPI B

A couple of years before the expected date of the SCAMPI A appraisal, the SMXG selected a lead appraiser and began to talk with him. The lead apprais-

er who they chose was Dr. Miluk from the SEI. In the initial discussions, Miluk told them about some recent experiences leading what he called *collaborative* SCAMPI B appraisals with organizations within some large corporations in the defense and commercial communications industries, as well as with the SMXG's sister organization at Warner-Robins Air Logistics Center, the 402d SMXG. He told us how this new approach to SCAMPI B appraisals had worked well to not only identify areas of strength and weakness with respect to the model, but had also benefited the organization in other ways. Having had much experience in CMM and CMMI appraisals in the past themselves, they knew, like Miluk, that the appraisal results often are pretty clear to the SEPG and others well initiated in process improvement but are often confusing to lay people. Miluk explained that this new approach helps in several ways, including the following: provides projects with a better understanding of what the model is asking them to do; makes it more clear to projects what their weaknesses are and how to fix them; receives buy-in from key project personnel that weaknesses identified in the appraisal were valid. After much thought and discussion with senior management, they decided to go ahead and implement this approach.

Because they were going to be conducting a string of SCAMPI B appraisals over a period of 12-18 months, and since they had a number of authorized SCAMPI lead appraisers within the organization, they asked Miluk to lead the first SCAMPI B appraisal to train them on the collaborative approach. From there, they had resources available internally to lead the rest of the SCAMPI Bs.

Project and Organization Focused Mini-Teams

In SCAMPI appraisals, the work of examining artifacts and the responsibility of making preliminary judgments about the degree of CMMI model compliance is typically distributed among groups of two or three appraisal team members. These groups of appraisal team members are called mini-teams. The use of mini-teams allows for increased efficiency in the appraisal process.

One of the most important elements in the design of their collaborative SCAMPI B appraisals was to align the mini-teams primarily by project and to make sure that key project personnel were included in the mini-team looking at their project. Because a whole series of appraisals were happening in a relatively short period of time, there was no need to include organizationally focused PAs from the CMMI in all of the appraisals. This inclusion would have resulted in unnecessary appraisal redundancy; these PAs were examined in just two of the SCAMPI Bs. In these two particular SCAMPI Bs, they had a single mini-team focused on these PAs (e.g. Organizational Process Focus, Organizational Process Definition, Organizational Training etc.). The number of projects examined in each SCAMPI B varied, with the maximum being three projects plus the organization-level processes totaling four mini-teams.

Competencies

In determining the appraisal team members, the following three critical competencies were required for each mini-team:

1. **Project knowledge.** It was important to have someone on each mini-team who knew where to look for necessary artifacts when the provided ones were insufficient. Additionally, they wanted to have someone there who could explain in finer detail, if needed, the explanation for each practice provided in the program independent interfaces (PIIs). This competency was met using the project manager (PM) or a project lead engineer.
2. **Process improvement experience in the organization.** The importance of this competency is in maximizing buy-in to the appraisal findings. In the organization, the lead process improvement agents in the product lines are called Senior Technical Program Managers (TPMs). They are skilled, experienced PMs who have responsibility to mentor and assist the managers of projects within their product lines, along with leading process improvement efforts at that level of the organization. The SMXG decided to use these individuals to help meet this competency level because they knew that having their support was essential and that making them part of the teams on these SCAMPI B appraisals would help to ensure ownership of the appraisal findings and actions needed to address them.
3. **Appraisal and CMMI model expertise.** Each mini-team had an individual who was either a lead appraiser or had significant experience in appraisals, as well as interpreting and implementing the CMMI. This competency was addressed by using individuals who were members of the SEPG (who are lead appraisers), the internal QA group, or lead appraisers from the Software Technology Support Center (STSC).

> *"The use of mini-teams allows for increased efficiency in the appraisal process."*

Project-Focused Draft and Final Findings Briefings

Another important feature in the design of the collaborative SCAMPI Bs was to provide draft and final findings briefings at the project level to make sure identified weaknesses were clearly understood. The project leaders included in the appraisal team were given a conspicuous role in briefing the results of their projects as a means to add legitimacy to the findings and to help secure buy-in. In some cases, the appraisal team members representing the project were the presenters in the findings briefings. Where necessary, these individuals could explain to personnel from their own projects what the weaknesses meant, could discuss why correcting these weaknesses would add value to the project, and also could discuss with the PM ways to address and correct the issue.

After Each Appraisal

Now that a particular SCAMPI B appraisal was over, the real work of addressing issues and implementing changes was about to begin. One of the things that made our efforts to reach CMMI Level 5 more difficult than for other organizations was our size and relative diversity in our product line make-up.

This made our action planning and tracking of these plans critical. In our case, the implementation effort was focused more at the lower levels of the organization (product line and project levels) where more of our weaknesses were. The real challenge at the organization level was making sure that action plans were complete and were monitored to ensure that items would be completed in order to meet our timeframe goals for the SCAMPI A appraisal.

Action Planning

Plans for addressing appraisal weaknesses were created at the product-line level, and

also by the SEPG for organization-level weaknesses. These plans were then brought together at the organization level to provide an opportunity for review by all of the stakeholders. The plans included a work breakdown structure of tasks, personnel and resources assigned to each, effort estimates, schedules, risks (and mitigation plans), etc.

Tracking Progress

In tracking progress of the action plans, they used the following four types of charts: 1) action plan activity descriptions, 2) Gantt charts, 3) percent complete charts, and 4) charts that showed our current estimated risk of satisfying each practice in the CMMI based on the characterizations used in SCAMPI B appraisals (High [Red], Medium [Yellow], Low [Green]) as shown in Figure 2. After a short time, chart type 4 was given the moniker The *Red-Green* chart after the character from the Canadian television show (*The Red Green Show*) that is shown here in the U.S. on the Public Broadcasting Service. These were briefed every two weeks to senior management and provided them with detailed information about what issues were being worked and were scheduled to be worked, problems that were being encountered, and how well the tasks were being completed compared to the schedule.

In order to change the risk characterization on any practice on the *Red-Green* chart, a project had to have their remedy to the identified weakness reviewed by the SEPG (who have extensive CMMI model interpretation and appraisal experience) as a QA check. Only actions that could show artifacts which satisfied a particular practice were accepted toward changing a Red (High risk) or Yellow (Medium) to Green (Low). A partial solution to a weakness would permit a change in the *Red-Green* chart from Red to Yellow.

The SCAMPI A

The organization had made a goal to hold the SCAMPI A appraisal late in 2005. This goal proved to be a bit too aggressive. Some funding constraints were levied on them from the Air Force in mid-2005 that limited their ability for Process Action Teams (PATs) to function at the pace required to address all the issues that needed to be solved in that timeframe. So, the goal for holding the SCAMPI A was pushed back to June 2006. That proved to be a good thing in the end because it provided more time to do things right and to institutionalize changes. It also made efforts to prepare for the SCAMPI A a lot less pressured than if we had continued the march toward the original goal.

The SCAMPI A appraisal was performed according to requirements of the method. Some of the choices that we made in executing the appraisal were the following:

1. **Selection of focus projects.** Focus projects were chosen from each of the product lines in the organization. Projects chosen were representative of the majority of work performed in the product lines and represented 41 percent of the engineering personnel in the entire 309th SMXG.
2. **Appraisal team size and membership.** The appraisal team had 12 members with three members per mini-team. This is probably larger than the average for a SCAMPI A appraisal. The lead appraiser assigned a representative from each of the organization's product lines, a person with lead appraiser credentials – or a very experienced appraisal veteran – and an individual from an outside government organization with high-maturity experience to each mini-team.

   The members of the appraisal team from outside the organization were involved in one of the earlier SCAMPI B appraisals. This helped them to come up-to-speed on the structure of the organization and its product lines, and it familiarized them with some of the types of projects that exist in 309 SMXG and the processes used.
3. **An expanded readiness review.** A readiness review was held approximately four weeks prior to the planned start of the on-site period. The SCAMPI A method requires that the readiness review includes a review of the artifacts (or PIIs) and the plans to make sure that these are in order and that the appraisal has a reasonable-to-high probability of being completed according to plan. The readiness review included these activities but in addition, a very detailed examination of the artifacts was performed. The readiness review lasted five full days and included some method training and a full review of artifacts in the PIIs where preliminary characterizations were made on the instantiations examined. At the conclusion of the readiness review, the appraisal sponsor was given a list of additional information and artifacts needed along with the lead appraiser's assessment of the readiness to proceed to the on-site. The green light to proceed was given.
4. **Mini-team organization.** Mini-teams were assigned to examine all the non-organizational project-oriented PAs for a particular project, as had been done in the prior SCAMPI Bs. They found that for this organization this was the most reasonable approach. Another mini-team was assigned to examine the PA's that had an organizational focus.
5. **Interview Sessions.** The first week of the on-site period was spent entirely on briefings, interview sessions with the focus projects, tool demonstrations, and additional method training (where needed). During the early part of the second week, another set of interview sessions was held with functional representatives of a much wider group of additional projects. This was done to enrich the sampling within the organization to ensure consistency and institutionalization of CMMI model practices.

Figure 2: *Red-Green Chart Example*

General Goal 2	OPD	OT	IPM	RSKM	IT	OID
GP 2.1	L		L	L		L
GP 2.2	L		L	L		L
GP 2.3	L		L	L		L
GP 2.4	L		L	L		L
GP 2.5	L		L	M		H
GP 2.6	L		L	L		L
GP 2.7	L		L	L		L
GP 2.8	L		L	L		L
GP 2.9	L		L	L		L
GP 2.10	L		L	L		L

General Goal 3	OPD	OT	IPM	RSKM	IT	OID
GP 3.1	L		L	L		L
GP 3.2	L		L	L		L

Lessons Learned

The collaborative approach to SCAMPI B appraisals was very effective for the SMXG. It met the expectations that they had in terms of improving the ability of the appraisal team to find appropriate artifacts when the PIIs provided were less than perfect. This approach also improved buy-in to the weaknesses identified in the appraisals. The use of project leaders on mini-teams improved the ability of the project leadership to understand the concepts behind model practices and facilitated their commitment to implementing value-added improvements to address these weaknesses. As an organization, they have some experience using informal appraisals to prepare for formal appraisals or in piloting new models and definitely feel that this strategy works and this approach will be used again in the future.

The management of action plans using Gantt, Percent-Complete, and Red-Green charts in concert with Current Activity description slides was critical for SMXG in getting this large effort completed. Had it not been for the clear measures and frequent reporting, the goal of achieving CMMI Level 5 would not have been met, at least not in the timeframe that they had in mind.

The SMXG found that it was critical to have QA on the closeout of action items, and the changing of risk characterizations that were represented on the Red-Green charts. Without this step, they would certainly have had some things that were thought to be sufficiently addressed that would have come back again as a weakness in a subsequent appraisal.

PII preparation and maintenance through the string of appraisals that SMXG performed turned out to be a major issue. Projects who took part in the appraisals were responsible to prepare their own PIIs, and some were more efficient and skilled than others. The effort and cost of preparing PIIs was excessive through the Appraise Maturity phase. This is one area that they will be focusing some innovation on to improve efficiency and reduce cost. They have some ideas about potential tools and methods of leveraging from the internal QA audits and data mappings to help in preparing PIIs and plan to pursue these ideas over the next year or so.

Conclusion

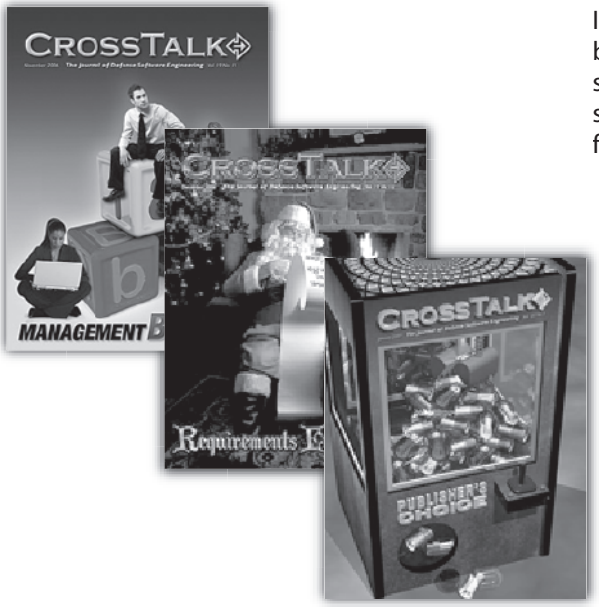
The adoption and institutionalization of the CMMI in the 309th SMXG was a journey that lasted approximately four years. It was a goal that would not have been reached without the continual support of all levels of management and the innovation and hard work of many individuals from all levels of the organization. The collaborative appraisal concept for SCAMPI B was a critical factor in transferring understanding of CMMI practices to projects and in obtaining buy-in to implementing them in a value-added manner.◆

About the Author

Rushby Craig is lead for the SEPG in the 309th Software Maintenance Wing, a CMMI Level 5 organization. He was a member of the CMMI model development team, and is an authorized SCAMPI lead appraiser, having either led or participated in more than 30 CMM/CMMI-based appraisals. Craig is also an authorized instructor for the SEI's Intro to CMMI course. For four years, he was an external software consultant with the STSC, assisting organizations across the U.S. in attaining their process improvement and maturity goals. Craig's other work experience includes project management, software engineering, and quality assurance. His education includes a Bachelor of Science in Electrical Engineering from the University of Utah and a Master of Science in Electrical Engineering from the Air Force Institute of Technology.

U.S. Air Force
309th SMXG
7278 4th ST
Hill AFB, UT 84056
Phone: (801) 775-2979
E-mail: rushby.craig@hill.af.mil

CALL FOR ARTICLES



If your experience or research has produced information that could be useful to others, CROSSTALK can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

Stories From the Field
August 2007
Submission Deadline: March 16, 2007

Service-Oriented Architecture
September 2007
Submission Deadline: April 13, 2007

Systems Engineering
October 2007
Submission Deadline: May 18, 2007

Please follow the Author Guidelines for CROSSTALK, available on the Internet at <www.stsc.hill.af.mil/crosstalk>. We accept article submissions on all software-related topics at any time, along with Letters to the Editor and BackTalk.

CMMI Level 2 Within Six Months? No Way!

George Jackelen
Global Analytic Information Technology Services, Inc.

Global Analytic Information Technology Services, Inc. (GAITS) decided to receive a Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI) Level 2 rating within five months. The purpose of this article is to show that when an organization is already doing competent project management, the effort to benchmark that capability by using CMMI is almost straightforward, and it is possible to achieve a Level 2 CMMI appraised rating within six months. This means there must be management support, the right CMMI project personnel, selection of the right effort(s) to be evaluated, and a CMMI appraiser who understands the company's effort and provides positive feedback.

Even though there was no contractual requirement, the GAITS owners decided in November 2005 to initiate a project to achieve a SEI CMMI appraised Level 2 rating within five months for a GAITS program.

The first thing the owners did was designate a mature effort for CMMI evaluation, i.e., a five-year Federal Aviation Administration (FAA) Independent Verification and Validation (IV&V) program. The program was chosen due to its requirement to use an internationally accepted process, i.e., the Institute for Electronic and Electrical Engineering (IEEE) 1012, *Software Verification and Validation*; the program was already active for almost two years; and the program's receipt of outstanding ratings from the GAITS quarterly customer satisfaction surveys. As a result of IEEE 1012, there was a built-in requirement to have a project plan, i.e., our IV&V plan, which management believed would be the foundation to achieve its CMMI goal. Without performing an internal appraisal, the program had a GAITS assumed level of maturity that would satisfy most, if not all, of the CMMI Level 2 requirements. Even though this proved to be true, they had a lot of work ahead.

The owners then designated a CMMI required sponsor from the senior managers to work with the CMMI project personnel as a channel of communications to other senior managers, and to ensure GAITS obtained the required CMMI project training, resources, and guidance. Next, they assigned a CMMI project leader who had experience as a process developer and who was an active member of the selected program. Finally, the owners assigned a CMMI project technical leader who was experienced with the FAA program and who could provide the CMMI project with technical and administrative support.

With the assistance of the mentor, Electronic Data Systems Corporation (EDS), as part of the Department of Defense (DoD) Mentor Protégé Program, GAITS selected an SEI-approved appraisal company to perform the CMMI appraisal. GAITS then selected a lead appraiser.

The GAITS assumption that the IV&V program could quickly be appraised at CMMI Level 2 had to be tested. If this assumption were not true, then more time would be needed.

By attending an SEI CMMI course and by reading books, the GAITS CMMI team realized the IV&V program had many of the needed artifacts/evidence. The perceived main problems were to fill in the gaps, to verify the artifacts met the requirements, to map the artifacts to the requirements, and to accomplish all of this within five months.

Most of the gaps consisted of documenting how we already did business in terms of the CMMI Process Areas (PAs). For instance, the IV&V plan did not address the needed details for the CMMI described Configuration Management (CM) process or the Management Analysis (MA) process. In other situations, gaps were caused by the need to find the physical artifacts, e.g., meeting minutes and documents addressing more than one CMMI PA. This was accomplished over three months; the team was confident they had the needed information for a CMMI Level 2 rating. However, the work was just beginning.

To improve the chances for success, the IV&V Program Manager (PM) agreed to allocate time during his weekly staff meetings for the CMMI project personnel to introduce CMMI, the reason the GAITS owners were willing to spend the time and money to receive a Level 2 rating, and to train the staff on the CMMI process and what to expect from a CMMI appraisal.

Practice Implementation Indication Description (PIID)

One of the critical steps was to develop a CMMI PIID; see Table 1 (page 14) for an example. The PIID identified the CMMI Level 2 PAs (column 1) and related specific and generic goals and practices (column 2), direct and indirect artifacts (e.g., documents), direct artifact title and the indirect artifact title columns, action items (direct artifact recommendations and the indirect artifact recommendations columns), history of key CMMI project activities (direct artifact comments and the indirect artifact comments columns and the direct artifact weakness/artifact collection issue column), and who was responsible for each CMMI project activity (the last column). In essence, a PIID is a traceability matrix between CMMI processes (the first two PIID columns) and the location of the related artifacts. The PIID was also used to track CMMI project progress.

The PIID direct artifact comments column also identifies the evidence within the identified artifact showing the specific CMMI requirement was satisfactorily met, e.g., what paragraph within a progress report addressed the communications of Project Monitoring and Control (PMC) progress to our senior managers or customer.

The PIID indirect artifact comments column is similar to the PIID direct artifact comments column but identifies the evidence within the identified artifact, showing artifacts are available to satisfy a CMMI indirect requirement.

Selected Program

The selected program involved the IV&V of an FAA critical, complex program involving aircraft flights throughout the United States. The IV&V program's staff size varies from year-to-year due to the annual FAA task order changes. Currently, there is a staff of 19 full-time personnel.

Even though the CMMI project per-

sonnel indicated the ability to be CMMI Level 2 appraised within five months would be impossible, an internal evaluation of the selected program showed the program was more advanced for a CMMI Level 2 rating than the CMMI project personnel initially thought. The ability to quickly develop, review, and correct the PA plans also helped, especially since the lead appraiser was one of the reviewers and provided very useful comments from a CMMI perspective that were very helpful and encouraging. The purpose of the lead appraiser's review was to identify areas not meeting the CMMI Level 2 requirements. After about three months of work, the CMMI project leader and the lead appraiser notified the sponsor that sixth months was needed to finish the CMMI project. The company's owners agreed to a one-month extension.

Roles and Responsibilities

The following provides information about how the CMMI team (CMMI project personnel, sponsor, and lead appraiser) worked together on this CMMI project.

The GAITS sponsor, a required CMMI appraisal position, provided the leadership needed to keep the CMMI project focused on the objective and provided needed communications to CMMI project personnel, other senior managers, and the lead appraiser. He also scheduled training for the CMMI project personnel and assumed the role of the acting PM when the PM left the company. Based on CMMI, the sponsor made changes to how the PM reported to the senior managers.

The GAITS FAA IV&V PM ensured compliance with the program's contract, vision, and objectives (without this coordination, the CMMI project would have failed due to conflicts between the IV&V program and the CMMI project). This included identifying appropriate program and company related artifacts, providing comments on how the CMMI FAA IV&V PA plans disagreed with the way the program operated, and providing recommended changes. He also obtained concurrence from our FAA customer and government stakeholders to utilize the FAA program for the CMMI appraisal. A key FAA IV&V PM activity was to provide CMMI training time during the program's weekly staff meetings. To improve communications between the program personnel and the CMMI project, he appointed PA managers to review and implement the PA plans.

The CMMI project leader managed the CMMI project and developed each of the PA plans and related documents, e.g., procedures and forms. Based on our environment, this was the most efficient way to develop the plans and to ensure compatibility between the plans and the program. Based on the CMMI project leader's experience with the PAs, process improvements, knowledge of CMMI and the program, and his past development and implementation of process plans, there was minimal rework and it was easier for the lead appraiser to deal with one person rather than a separate person for each PA plan. To improve the overall CMMI project, the CMMI project leader also created the initial Process and Product Quality Assurance (PPQA) plan, checklists, and forms. When it was time to perform PPQA audits, the CMMI project leader was excluded, per the lead appraiser, from auditing the PAs since a conflict of interest existed, i.e., the CMMI project leader might not provide *objective evidence* of what was found during the audit of plans the CMMI project leader developed.

The GAITS project technical leader provided backup to the CMMI project leader and kept the CMMI project leader informed of daily CMMI project activities. Whereas the CMMI project leader managed the CMMI project and developed the PA plans, the CMMI project technical leader's main role was to ensure the plans were implemented as described and to identify non-conformances. To accomplish this role, the CMMI project technical leader was assigned to perform the PPQA audits and to find and store the required artifacts. (NOTE: Since there would be a conflict of interest for the CMMI project technical leader to audit the PPQA PA, the FAA IV&V PM appointed another person to audit the PPQA PA.) The CMMI project technical leader also documented discrepancies discovered during the PPQA audits and followed through to ensure the identified corrective actions corrected the discrepancies. Since the PAs were being implemented based on documented plans, the CMMI project technical leader worked with the PA managers prior to and during the PPQA audits to modify the initial PA plans and audit checklists to correct errors or to improve the processes. The CMMI project technical leader also maintained the PIID by working with the lead appraiser and program personnel to document the location of artifacts and to resolve issues. This was a critical task and required many hours of work to ensure timeliness, consistency, and completeness, while working with others (e.g., PM, PA managers, and the lead appraiser) to ensure

Table 1: *Sample PIID*

CMMI Practice	Practice Statement	Direct Artifact Recommendations	Direct Artifact Weakness/Artifact Collection Issue	Direct Artifact Title	Direct Artifact Comments	Direct Artifact Path	Indirect Artifact Recommendations	Indirect Artifact Title	Indirect Artifact Comments	Indirect Artifact Path	Actions by Who and When Due
PMC Generic Practice 2.5	Train the people performing or supporting the project's monitoring and control process as needed.	None	Valid Artifacts	• FAA IV&V Training Tracker • PMP	• PMC 10/27 2005 and Management and Administration Training 11/4 • Section 5.10: Training	VSS – $/Projects/ FAA IV&V/4 Training/ Training Tracker.doc	Link in the meeting minutes and the training material.	Staff meeting minutes 11/11/05	New Business: First item is related to the Measurement and Analysis process area.	VSS ... FAA IV&V/9 FAA IV&V CM/IV&V Staff Meeting Minutes 11/11/05.doc	• PM • 12/5

everyone understood what was needed, when it was needed, and why it was needed. PIID updating was a daily task. As part of his PIID work, the CMMI project technical leader provided timely progress reports to the CMMI project team.

Prior to performing the official CMMI appraisal, the lead appraiser made several on-site visits to evaluate CMMI progress and to provide guidance. The theme of these visits was to determine if GAITS was ready for the appraisal and to identify our strengths and weaknesses. At first glance, this appeared to be a conflict of interest when it was not. The lead appraiser followed strict SEI rules, e.g., not developing artifacts. Instead, he performed informal appraisals to identify CMMI defects in what they were doing. Plus, when it was time to perform the official appraisal, a new SEI-trained appraiser came onboard to provide independent assessments. Since the formal appraisal results had to be a consensus of the appraisal team, the new appraiser could prevent an automatic approval based on just the lead appraiser's view or possible bias. At the same time, the lead appraiser and the group being appraised could be independently audited by the SEI if the SEI believed a possible conflict of interest existed or if the SEI appraisal rules were violated. In my experience, this is like the quality assurance and test groups stepping in to work as partners (not cops or watch dogs) with the development groups to ensure a quality product was being produced (e.g., to identify problems early), but still being able to objectively evaluate the products since these groups did not develop the products. In fact, having this early partnership results in better assessments and test cases, through a better understanding of what is being done, while minimizing the risk of failure late in a project. A partnership does not guarantee approval by an evaluating group. Instead, it improves communications and understanding. This is what the lead appraiser provided.

Some appraisers just identify defects without providing information to assist in resolving the defects. When identifying a deficiency or the need for a clarification, our lead appraiser provided recommended corrective action and positive advice. For us, the key was that the lead appraiser's information was enough for us to understand the problem and possible solution. The lead appraiser also asked questions so we would recognize a problem. At the same time, he reminded CMMI project personnel of items that were already discussed in other documents, thus eliminating duplications. An important function the lead appraiser performed was to identify items required and not required by CMMI Level 2. For example, some of the items in the process plans were for CMMI Level 5 and could not be supported by other process plans.

He also ensured the PA plans were developed for a service support program rather than a system or software development program. The difficulty here was that the CMMI model was oriented toward system or software development rather than service support programs, e.g., quality assurance, quality control, IV&V, and CM. As a result, some of the CMMI principles and artifact contents did not apply or had to be re-defined so we could implement the intent of the CMMI principles and artifacts from a service support perspective.

To provide continuity, the lead appraiser remained involved with the CMMI project from the beginning until the conclusion of the Standard CMMI Appraisal Method for Process Improvement (SCAMPI[SM]) for final appraisal evaluation.

Issues

The FAA IV&V program was finishing its second year when the CMMI project started. As a result, an item the lead appraiser initially had an issue with was that GAITS did not have a CMMI project-planning plan. To resolve this, the CMMI project developed a CMMI IV&V project management plan (PMP) that used the existing, official deliverable (the FAA IV&V plan) and added the necessary CMMI items. To make maintenance easier (since the contract is renegotiated each year to identify annual tasks, resource needs, and funding), the existing plan was made an attachment to the CMMI IV&V PMP. As a result, the FAA IV&V CMMI PMP referenced the FAA IV&V plan as much as possible and specifically addressed items not addressed by the FAA IV&V plan. Thus, the CMMI portion of the FAA IV&V CMMI PMP should remain static throughout the contract while only modifying the official FAA IV&V plan attachment to list negotiated tasking, resourcing, and funding for the upcoming year. All of this was still compatible with the IEEE 1012 IV&V plan template.

For the CMMI project personnel, the hardest concept to understand was the difference between the following (NOTE: these are my definitions):

- **A direct artifact:** An output artifact used to show a process was performed and completed as described.
- **An indirect artifact:** An artifact supporting a process, e.g., a process input. This is used to show a process was initiated. Thus, a direct artifact of one process could be an indirect artifact for another process.

Another issue was that the FAA IV&V program's products do not require pre-delivery coordination with other groups; especially since the IV&V products are normally reports documenting IV&V evaluations of products from the FAA and their development contractor. Therefore the IV&V program does not require a Configuration/Change Control Board (CCB). Instead, from the start, the program established a peer-review process to ensure program products (excluding proprietary products, e.g., products with pricing information) satisfied contractual requirements. As a result, the stated internal review process will document the peer-review results, followed by a final PM review just prior to delivery. This system has worked well for the program and was acceptable to the lead appraiser, especially since the only customer comments occur during the annual IV&V plan update when the contract is re-negotiated and new tasks are identified. The main point is that they have a very successful review/approval process that does not use a normal development approval group (i.e., CCB). The lead appraiser had to keep reminding himself that for a service support program, this was not a violation of CMMI principles.

For those wondering about the issue of making sure the changes are lasting, CMMI has a requirement that there be an appraisal within three years of the passing of an appraisal. Thus, a group can lose its CMMI status if the group does not continually maintain the correct artifacts.

Lessons Learned

Before starting an official CMMI appraisal project, an organization needs to perform an honest self-evaluation (or hire an outside, honest broker). One of the key outputs is a PIID. Using the PIID format, the CMMI deficiencies can be clearly listed and addressed. In GAITS' situation, they had most of the needed artifacts, but they were not organized to provide easy, documented, and logical access. For instance, some of the artifacts were on the hard drive of individual laptops. As a result, these artifacts were moved to a more central location. Some of the data and information was placed under restricted access since some of this data and information was proprietary (such as billable information and they had subcontractors with access to the database). Another issue with the individual laptop storage was the inconsistency of the file names within an

individual's database folder. As part of the CMMI CM PA, the CM manager developed a CMMI required standardized program repository and a standardized naming convention.

A major benefit of our CMMI appraisal effort was to clearly identify where information and data were to be stored. With the CM manager's development of a repository infrastructure, finding and retrieving program information and data greatly improved. This was also a great help for the new PM to quickly come up-to-speed about the program. At the same time, our people are better able to share information and data.

Conclusions

With the cooperation of organizational personnel and the lead appraiser, a CMMI Level 2 rating can be accomplished in less then 18 months without compromising how an organization operates. This does not mean every attempt to be Level 2 can occur within 18 months. As described earlier, there are many things that must fall into place.

Having a program with well-established processes can only speed up the appraisal process, especially if the program processes are similar to what the CMMI is looking for. This also helps speed up the process to develop PA plans. A major effort was for the CMMI project leader to document what those processes were and to compare the results with their requirements.

Having a person who is knowledgeable with the program/organization(s) being evaluated and very experienced with writing plans, procedures, and checklists can not only minimize issues discovered by a lead appraiser, but can also ensure these documents are quickly developed or existing documentation is corrected.

Having an almost full-time person (i.e., our CMMI project technical leader) being the PIID point-of-contact, creating and maintaining the PIID, and performing the initial PPQA audits also speeds up the process. This person should work directly with the lead appraiser and others and should also provide the sponsor and lead appraiser with status reports – weekly at first, but daily as the date of the SCAMPI approaches.

Ensure that the lead appraiser will work with your organization to understand your environment and to provide help rather than just provide a list of needed corrective actions. If the lead appraiser has pre-conceived notions about how an organization must operate, the CMMI project sponsor and leader must ensure these notions are corrected or a compromise can be reached. With the cooperation of the lead appraiser, the sponsor and the CMMI project personnel can help ensure success.

Acquiring a CMMI Level 2 rating is not cheap and cannot occur haphazardly. The main costs are organization personnel (in our situation, two almost full-time people and several part-time people) and paying for the lead appraiser and CMMI training. However, GAITS estimated the results, especially when the organization follows through to maintain at least the Level 2 rating, should pay for the CMMI investment within two years. Being organized and having artifacts to show defined processes are being followed helps organizations enhance competitiveness and reduce cost. For example, portions of the PA plans can be used within proposals.

The lead appraiser informed us that based on SEI rules, since the CMMI evaluated program represented over 67 percent of the IV&V division's work, the IV&V division was CMMI Level 2 rated. Thus, our rating was at a higher organizational level than we had planned.

As mentioned before, SEI requires that we will be re-evaluated at a later date to ensure we are maintaining at least a CMMI Level 2 rating. To help non-developmental system and software efforts, SEI has completed a CMMI supplement to address services rather than development efforts. This should greatly assist service organizations – like IV&V – that desire CMMI appraisal.◆

About the Author

George Jackelen works at GAITS, Inc. as a senior systems engineer on the referenced FAA IV&V program as CMMI project leader. He has been directly involved with IV&V for more than 10 years and quality assurance for more than 18 years. Jackelen spent 20 years in the United States Air Force on various information technology projects and more than 20 years with private industry doing work for federal and state agencies.

85 South Bragg ST 4th FL
Alexandria, VA 22312
Phone: (703) 866-2400
Fax: (703) 866-2423
E-mail: gjackelen@gaits.com

More Online from CrossTalk

CrossTalk is pleased to bring you this additional article with full text at <www.stsc.hill.af.mil/crosstalk/2007/02/index.html>.

Connecting Software Industry Standards and Best Practices: Lean Six Sigma and CMMI

Gary A. Gack and Karl D. Williams
Six Sigma Advantage, Inc.

Integration of Six Sigma and the Capability Maturity Model Integration (CMMI) is becoming fairly widespread, yet confusion remains about their relationship. Part One of this article includes several case studies that answer some of the more common questions. Part Two describes the relationship of Lean Six Sigma and Six Sigma's approach to improvement of existing products and processes (Define, Measure, Analyze, Improve, Control [DMAIC]), and Part Three examines the relationship between Design for Lean Six Sigma (used to develop new products and processes or major enhancements) and the CMMI Engineering Process Areas.

Software professionals, especially those working in the Department of Defense environment, face a somewhat bewildering array of relevant standards and best practices. As awareness and penetration of Lean Six Sigma in this environment have increased significantly over the last several years, we find many organizations struggling to understand and leverage the relationships between Lean Six Sigma and several other approaches to software process improvement, including CMMI.

Future Directions in Process Improvement

Watts S. Humphrey, Dr. Michael D. Konrad, James W. Over, and William C. Peterson
Software Engineering Institute

As systems become larger and more complex, and as increasing numbers of development programs are integrated and distributed, development processes, methods, and management must change. To keep pace with these changes, the directions of process improvement must also change. This article describes the likely future directions of process improvement evolution.

Large and complex computer-based systems are critical to the economic and military welfare of the United States and to much of the industrialized world. These systems form the backbone of modern military, business, and governmental operations, and without their continued support, our societies would be severely inconvenienced and even threatened. Unfortunately, the development of such systems has been troubled, and the systems needed for the future will be vastly more complex and challenging. If history is any guide, attempting to develop these future systems with current widely followed practices will almost certainly yield unsatisfactory results.

In addressing the challenges of the future, it is important to consider three points. First, with few exceptions, the reasons that large-scale development programs have failed have not been technical [1, 2]. As the cancellation of two large and critical efforts demonstrates, these systems have almost always failed because of program-management problems [3, 4]. Second, the solutions to these program-management problems are known and have proven to be highly effective, but they are not widely practiced. Third, if these known and understood project management practices are not promptly and effectively adopted, future large-scale systems development programs will be completely unmanageable. Such systems will no longer be delivered late, over cost, and with poor quality; they will likely not be delivered at all. Proper use of sound processes would actually improve the cost, quality, and schedule performance of engineering organizations.

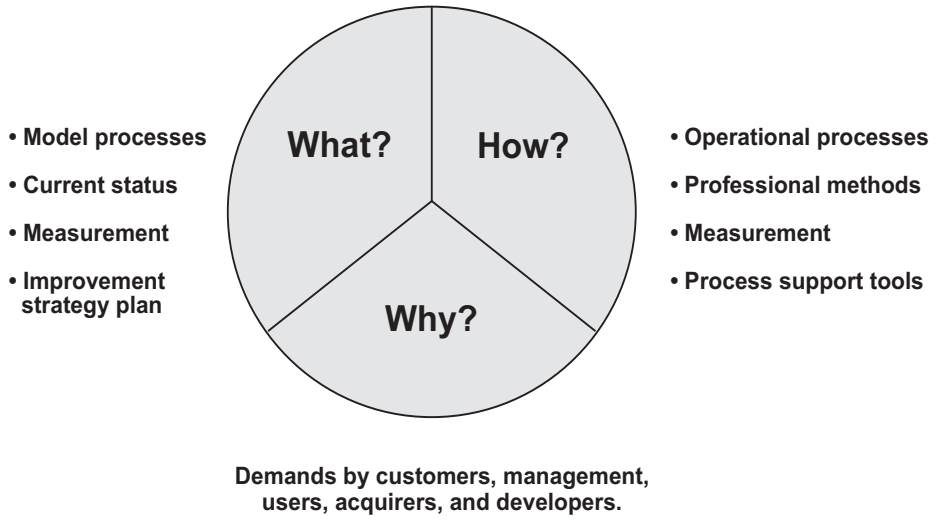
This article outlines the authors' current thinking on the likely future direction of process-improvement work and our strategy for addressing the challenges ahead. To explain the objectives of this article, however, it is necessary to revisit the original logic for the Capability Maturity Model® (CMM®) and Capability Maturity Model® Integration (CMMI)® work [5, 6, 7]. The original framework used by Software Engineering Institute's (SEI) Software Engineering Process Management (SEPM) Program to characterize its process-improvement mission can be explained in terms of *what*, *how*, and *why* perspectives as shown in Figure 1. The *why* in this figure describes the need to respond to customer, acquirer, user, or management demands for *better* software engineering practices and results. The *what* and *how* define SEPM's approach for addressing these needs.

Here, a specific *why* question might be the following: Why would I need to improve my process? The answer might be the following: Because, by following your current process, your organization consistently misses most commitments to its customers, acquirers, users, and management. A *what* question might then be the following: What can I do to get my organization to consistently meet its commitments? Part of the answer to this question might be the following: Require your organization to plan all of its development work. Then a *how* question could be the following: How should we plan the development work? Finally, part of the answer to this question could be the following: Train the developers in sound planning methods, have them and their development teams plan their own work, and then have management review and negotiate these plans with the developers and their teams.

The objective of this article is to identify the relationships of the key process *whats* and *hows* and to suggest strategies to guide development organizations of all types in introducing superior engineering processes and in using these processes to consistently obtain superior performance. In doing this, we address two increasingly important issues: First, organizations often focus on maturity levels rather than process capability. Second, this focus can result in high maturity ratings that do not lead to better organizational performance. To address these two issues, organizations need specific guidance on *how* to do development work. By design, CMMI does not now provide guidance on *how* to do development work.

The CMM was SEPM's initial approach to addressing the *what* dimension. It initially focused on software engineering and was later broadened to encompass other engineering fields such as systems engineering and software acquisition. This work has been codified in the CMMI model. Other aspects of the *what* dimension that are not discussed here are appraisal, measurement, and analysis.

Figure 1: *SEPM's Strategic Framework*



SEPM addressed the *how* dimension of software engineering with the Personal Software Process[SM] (PSP[SM]) and the Team Software Process[SM] (TSP)[SM]. When developers in CMMI-assessed organizations use the PSP and TSP, the assessors can use the data generated by the PSP and TSP to verify the proper use of mature development processes. The PSP and TSP are now being adapted for general use by acquisition, systems-engineering, and hardware and software teams.

It is easy to become confused about what is a *what* and what is a *how* in process improvement and how these aspects relate and fit together. Before addressing this issue, however, it is important to first describe what a superior process is and outline some of the current issues facing the process improvement community in general and the CMMI community in particular.

A Superior Process

The engineering processes of the future must meet the following five requirements:

1. Control development costs and schedules predictably.
2. Respond to changing needs.
3. Minimize development costs and schedules.
4. Be scalable from small to very large systems.
5. Produce quality products predictably.

These topics are covered in the following sections.

Control Development Costs and Schedules

Cost and schedule problems are not new, and many other fields have learned how to manage them. For knowledge work, the solution has always been the same:

- The people who will do the work estimate and plan that work.
- Sound methods and relevant data are used to plan, track, and manage the work.
- The progress of the work is precisely and regularly tracked.
- When progress falls behind plan, the problem causes are promptly identified and resolved.
- When the requirements change, all involved levels promptly re-estimate and revise the entire plan.
- Risks are anticipated and managed.

While one might question the applicability of this approach to systems, software, and hardware development, there is now ample evidence to demonstrate its efficacy. Nearly 20 years of experience with process improvement have shown that these principles, when properly applied at all levels, can have important benefits [8, 9, 10, 11].

Respond to Changing Needs

While responsively handling changing needs would seem like a simple issue, it is not. The problem is to be responsive to new information while continuing to meet prior cost and schedule commitments. In fact, it is just this trade-off that is responsible for many of the severe cost and schedule problems of large systems. To be responsive but still maintain project control, development groups must do the following:

- Examine every proposed change to understand its effects on the development plan.
- Pay particular attention to each change's impact on completed work, including the requirements, design, implementation, verification, and testing activities.
- Estimate all cost and schedule consequences of making the needed changes.
- Where the cost and schedule implications are significant or where they exceed the currently approved plan, get management approval before proceeding.

Minimize Development Costs and Schedules

The three most common ways to minimize development costs and schedules are the following:

1. Optimize project staffing.
2. Reduce the amount of work.
3. Minimize the amount of rework.

While the actions required to address points one and two are reasonably straightforward and do not need further discussion here, point three is not and is discussed later.

Be Scalable

A scalable process must follow principles and practices that are suitable for the sizes of the projects with which it will be used. To be scalable, a process must meet the following three criteria:

1. It must use robust and precise methods at all levels, especially at the working systems-, software-, and hardware-engineer levels.
2. For technical and management program decisions, the management system must be based on and give great weight to the knowledge and judgment of the development-level professionals and anyone else who has relevant information.
3. The process must consistently use data that are derived from accurate, precise, and auditable process and product measurements.

Produce Quality Products

The governing quality consideration for large-scale systems development is that a high-quality process will consistently produce high-quality products while a poor-quality process will generally produce low-quality products. The problem here is with the word *generally*. People tend to remember their occasional successes and forget their less memorable achievements. As a result, when a development group has produced a seemingly high-quality, small product with an unmeasured and poorly controlled process, the members tend to feel that they have proven that process and should continue to use it, even for larger-scale work. However, unless they have measured and statistically verified that this process consistently produces quality products and that it is scalable, they run a significant risk of getting poor-quality results. With really massive monolithic systems, just a small chance of getting a poor-quality component would compound almost certain quality problems for the overall system.

Current Process Improvement Issues

While engineering groups face many challenges, the process management and improvement communities must now address two current issues. First, with increasing marketplace pressure, organizations often focus on maturity levels rather than process capability. Maturity levels cannot comprehensively measure organizational capability. They can indicate management priorities and the degree to which an organization is attempting to address its process problems. They can also guide the search for risky process areas and help establish process improvement priorities.

We now see cases where high-maturity ratings do not always result in the rated processes being used on the subsequent projects. It is not that the appraisal process is faulty or that organizations are dishonest, merely that the maturity framework does not focus on *how* the work is actually done; it only addresses *what* is done. While this can be adequate when organizations are truly striving to achieve a superior process, a concentration on

[SM] Team Software Process, Personal Software Process, TSP, and PSP are service marks of Carnegie Mellon University.

maturity levels can cause organizations to ignore the *how* aspect of process improvement and adopt inefficient and poor-quality methods and practices.

The second issue concerns adjusting the CMMI framework and assessment methods to address this maturity-level problem. Without change, we can expect more cases where high-maturity ratings do not consistently lead to better organizational performance. Three lessons from experience suggest ways that help to ensure improved maturity levels consistently lead to improved organizational performance:

1. To properly control complex and precise work, everyone must manage to detailed and precise personal plans.
2. To predictably produce high-quality, large-scale systems, everyone must measure and manage quality.
3. The true measure of process improvement is then the degree to which the behavior of all of the working professionals and their teams reflects these practices.

To guide software developers in applying these lessons to their work, the SEI developed the PSP and the TSP and there is now substantial evidence that these methods can indicate the effective use of mature development processes [10, 12, 13]. The SEI is now adapting the PSP and TSP to general product-development work and investigating its use for acquisition work.

This, however, leads to a third issue: flexibility. CMMI does not define *how* to do development work. From the very beginning, the focus has been on *what* to do. The reason was that the original CMM was developed to guide the U.S. Department of Defense in source selection for software-intensive projects, and we did not want non-developers to specify how software development groups should do their work. While we had many ideas about how that process could be improved, we did not think that the users knew any better than industry how software should be developed. We also knew that, as current processes were improved, many new and creative methods were likely to be developed. By focusing exclusively on the *what* dimension, we hoped that CMM, and later CMMI, would not constrain process innovation. This continues to be our position.

Since we also believed that the *how* dimension of process improvement was important, we developed the PSP and TSP to provide guidance on how personal and team project planning, tracking, and quality management could be performed. Our concern now is to find ways to relate these practices and this guidance to the CMMI model without switching the focus from *what* to *how*. The need is to encompass both software and other development fields and not constrain development organizations as the technology advances. SEPM is working on these issues as we strive to improve the effectiveness of these methods.

Process Management Principles

Experiences with CMMI, PSP, and TSP have shown that, when an organization's management is convinced of the value of disciplined processes and properly implements an orderly and planned process improvement strategy, it can build the capability to successfully produce the truly large-scale systems of today and tomorrow. The five basic principles of these superior processes are as follows:

> *"CMMI does not define how to do development work. From the very beginning, the focus has been on what to do ... By focusing exclusively on the what dimension, we hoped that CMM, and later CMMI, would not constrain process innovation."*

1. All modern science and engineering is based on learning from prior demonstrably effective practices. Competent engineers and scientists know what experiments have been successful and base their personal and team processes and practices on this experience. They stay current with new process developments and do not waste time experimenting with processes that have already produced unsatisfactory results. Until developers consistently use defined and proven processes, they will waste their time relearning known truths. Similarly, in experimenting with new and improved processes and methods, competent process professionals build on the results of prior experience.
2. With an inefficient or ill-defined process, developers must follow poorly defined and inaccurate plans. With the data available from a modern process, plans can be both accurate and precise. With defined and sound processes and precise and accurate plans, developers need not waste their time trying to find out what to do next and can devote more of their efforts to creative technical work.
3. Development is a learning process, and unless this learning is codified and preserved, the resulting knowledge is generally lost. That is the reason developers should define, use, and continually improve their processes: to build on their own and other's experiences.
4. For individuals and groups to work together effectively, they must coordinate their activities. While very small groups may be able to accomplish this informally without defined processes and detailed plans, large groups cannot.
5. Quality work is not done by accident or mistake. Quality must be planned, measured, tracked, and managed. When it is, product defect levels are normally reduced by orders of magnitude [10]. With current commonly used practices, large software products typically have thousands of test defects and developers spend at least half of their time finding and fixing enough defects for the product to run the basic tests. Even then, finished products generally have many unidentified defects. While it takes considerable skill to find and fix defects in test, fixing defects is not creative work. High-quality work is only produced by people who strive to produce quality products with every step of their work.

To have a reasonable chance of being successful, the more challenging engineering programs of the future must address the critical systems-development problems of cost and schedule, requirements instability, process management, and quality management. In fact, the use of defined, planned, measured, and quality-controlled processes can help improve both the business and technical performance of large-scale programs. Later sections of this article describe the actions required to accomplish this. First, however, it is important to define the principles of quality management.

Quality-Management Principles

Newer systems-of-systems structures are now being considered for many programs because they offer many advantages. As demonstrated by the Internet, the best-known system-of-systems, such structures can be improved by many groups independently and they can generally withstand individual node failures without disabling the entire system. However, they also have disadvantages. For example, their high reliability and relative node independence requires substantial redundancy and, thus, higher costs for the individual system nodes. Such structures must also have relatively narrow bandwidth coupling among the nodes. This can severely restrict system performance.

For these reasons, most large complex systems are built as small collections of large monolithic nodes instead of as large collections of small independent nodes. This is because of the potential for increased performance, security, economy, or some other key property that could not be obtained by decoupling the system into relatively independent, small elements. The tight-coupling characteristics of large-scale systems generally result from optimizing the overall design and from minimizing redundancies and inefficiencies among the system's component parts. This results in closer coupling among the system's components and large numbers of critical interdependencies.

To produce superior systems and especially the types of large-scale systems needed in the future, all aspects of process quality must be addressed. This includes the business aspects of quality management such as cost, schedule, predictability, and risk management. It also includes the marketing aspects of quality like competitive superiority and customer satisfaction. Finally, particularly for large-scale systems, the process used must be effective for the scale of the work being undertaken. The quality methods required for a process to scale up to handle the kinds of large-scale systems-development work that we can expect in the future are based on the following four critical assertions:

1. To have any hope of producing quality work, the overriding process goal must be to prevent defects of all kinds before they are introduced into the system.
2. To produce quality products consistently, the process must also have the objective of removing all defects before test entry. The objective of testing then becomes to verify and validate the product – not to fix its defects.
3. To ensure an effective and high-quality process, that process must provide quality measures.
4. To manage a quality process effectively, everyone from the senior executives to the individual systems-, hardware-, and software-engineering team members must support and participate in that quality process.

These assertions are briefly discussed in the following sections.

Preventing Defects

The software- and systems-development communities have long focused on defect prevention, but they have not generally approached it as a quality-management activity. For example, efforts to improve requirements-development work are defect-prevention activities, as is the adoption of better design methods or the use of improved implementation or design tools. For these efforts to be most beneficial, however, they should include explicit defect and productivity measures and analyses to ensure that they are addressing the most critical defect sources in the most effective way.

Removing Defects Before Test

Removing defects before test is accepted as an essential element of all quality-management programs. By following traditional quality-management principles, organizations minimize rework, reduce project costs and schedules, and produce better products. The following principles of quality management are based on facts that have been demonstrated in every field where they have been tested, including software [13, 14, 15]:

- It costs more and takes longer to build and fix defective products than it would have taken to build them properly the first time.
- It costs more to fix a defective product after delivery to users than it would have cost to fix it before delivery.
- It costs more and takes longer to fix a product in the later testing stages than in the earlier design and development stages.
- It costs more and takes longer to fix requirements and specification errors in the design, implementation, test, and operational stages than in the earlier requirements and specification stages.
- It is least expensive and most efficient to prevent the defects altogether.

These quality principles are based on experiences with both small- and large-scale systems of all types. The only point of debate typically concerns requirements defects. Here, however, the issue is not really one of correcting known defects as much as with understanding the system's true requirements and how to achieve them. Where the requirements are not clear, it is often best to make a first cut at the requirements and then test the related functions in practice. Whenever the requirements for new or highly modified systems or platforms are known to be wrong, it is always cheapest to fix them at the earliest possible point in the process. However, if the requirements for a fielded system or platform are later found to be incorrect, the requirements and possibly even the entire system strategy may have to be re-evaluated.

Even though the above quality principles have been proven in every case where they have been properly applied, they are still not universally accepted. The reason is that engineering organizations often establish separate groups for product development, production, testing, and field repair. Therefore, while the total operation would save time and money by following sound quality practices, the added costs of quality work must be borne by some groups while the savings accrue to others. Since quality programs typically increase costs in the early program stages and reduce them in later phases, the early investments in quality programs are generally hard to justify, particularly when organizations do not have the quality data to support their introduction. Finally, unless managers and developers have personally experienced the benefits of an effective quality program, few are willing to make the necessary effort to do high-quality work.

Quality Measures Are Essential

Today's commonly used systems-, hardware-, and software-development processes do not incorporate quality measurement and analysis at the individual and team level. This is a crucial failing since with even rudimentary measures, the above-named facts would be obvious and, as a consequence, more efficient and sensible quality practices would have been adopted long ago. The simple fact is that without precise quality measurement and analysis at all working levels, no serious quality program can be effective. While developers can make rudimentary quality improvements without measures, achieving the defect levels required in modern complex systems must involve defect levels of a very few parts per million. Such levels are not achievable without complete, consistent, precise, and statistically based quality measurement and analysis.

Everyone Must Participate in the Quality Program

The requirement that everyone must participate in the quality program is a direct consequence of the previously stated facts. To achieve defect levels of a few parts per million, quality must be a high priority for everyone. Quality work results only from a consistent striving for perfection. The individual development team members must all strive to produce defect-free work. Any undisciplined work by any acquisition professional, systems engineer, hardware developer, software developer, tester, or almost anyone working on or with the product can be a source of defects, and their work must be measured and quality controlled. If it is not, high-quality products simply will not be produced.

This is not only true for all of the development team members; it is also true for all of the support groups, managers, and executives. For example, an executive decision to skip some review, test, or check because *we don't have the time* will inevitably lead to poor-quality products. The only acceptable attitude at all levels of an entire program and in all involved engineering organizations must be *we don't have time to do it wrong!*

The CMMI and TSP Direction

The characteristics of superior engineering and management processes have now been clearly demonstrated in many fields, but it has become clear that to establish such processes, many engineering organizations need more precise *how-to* guidance than that provided by CMMI. For this reason, SEPM is exploring ways to provide such guidance. While the initial SEPM efforts in this direction were to establish a separate effort called the TSP, additional steps are needed, such as the following:

1. The TSP, which was initially focused on software teams, is now being broadened with a TSP Integrated (TSPI) project to include all types of engineering teams.
2. Because CMMI and TSP follow the same process- and quality-management principles, their joint use reduces TSP introduction costs and accelerates CMMI improvement. This has been demonstrated by the greatly accelerated process improvement schedules achieved by organizations that coordinate the use of both methods [11].
3. Work-to-date has demonstrated that there are substantial synergies between the CMMI and TSP methods. For example, an SEI project has mapped the TSP practices onto the CMMI framework and identified alignment gaps and overlaps [16]. Further, such joint work among the developers and users of the CMMI and TSP is planned.
4. Based on the work completed to date, the SEI will couple the CMMI and TSPI work more closely to provide a more coherent improvement road map for the process improvement community.

Conclusion

The current commonly used systems development methods have reached (or soon will reach) their feasibility limits, and continuing to develop the increasingly challenging and massive systems of the future with the most widely used methods of today is destined to failure. The danger is that with the rapid pace of technology, society could well be lulled into the false belief that the technical community is capable of building the systems we can technically describe. As these newer systems are used to support increasingly critical aspects of modern society, we then will likely face far more catastrophic system failures than we have experienced previously.

To successfully produce the large complex and critical systems of the future, development organizations must start to use more modern and consistently successful processes. These processes must follow sound management and quality principles. In particular, the people doing the work must plan their own work and that work must be precisely and continuously tracked. Everyone in the organization must be involved in and completely committed to the organization's quality management program, and management must recognize and reward superior work.◆

Acknowledgements

We thank the reviewers who made many helpful suggestions and comments on this paper. They are Bob Cannon, David Carrington, Caroline Graettinger, Jim McHale, Julia Mullaney, Dennis Smith, and Dave Zubrow. We also thank the CrossTalk staff and reviewers for their help in reviewing, refining, and producing this article.

References

1. Neumann, Peter G. Computer Related Risks. Reading, MA: Addison Wesley, 1995.
2. Petroski, Henry. To Engineer Is Human: The Role of Failure in Successful Design. New York: Random House (Vantage Books), 1992.
3. United States. General Accounting Office (GAO). Observations on FAA's Air Traffic Control Modernization Program <www.gao.gov/archive/1999/r199137t.pdf>.
4. Gross, Grant. "FBI Trying to Salvage $170M Software Package." Computerworld 14 Jan. 2005 <www.computerworld.com/databasetopics/data/story/0,10801,98980,00.html>.
5. Chrissis, Mary Beth, Mike Konrad, and Sandy Shrum. CMMI 2nd Edition - Guidelines for Process Integration and Process Improvement. Reading, MA: Addison-Wesley, 2007.
6. Humphrey, W.S. Managing the Software Process. Reading, MA: Addison-Wesley, 1989.
7. Paulk, Mark C., Charles V. Weber, Bill Curtis, and Mary Beth Chrissis. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA: Addison Wesley, 1995.
8. Gibson, Diane L., Dennis R. Goldenson, and Keith Kost. "Performance Results of CMMI-Based Process Improvement." CMU/SEI-2006-TR-004, ESC-TR-2006-004. Pittsburgh, PA: SEI, Carnegie Mellon University (CMU), 2006.
9. Humphrey, W.S. Winning With Software: an Executive Strategy. Reading, MA: Addison-Wesley, 2002.
10. Davis, N., and J. Mullaney. "Team Software Process in Practice." SEI Technical Report CMU/SEI-2003-TR-014, ESC-TR-2003-014. Pittsburgh, PA: SEI, CMU, 2003.
11. Pracchia, Lisa, "The AV-8B Team Learns Synergy of EVM and TSP Accelerates Software Process Improvement." CrossTalk Jan. 2004 <www.stsc.hill.af.mil/crosstalk/2004/01/index.html>.
12. Humphrey, W.S. PSP: A Self-Improvement Process for Software Engineers. Reading, MA: Addison-Wesley, 2005.
13. Humphrey, W.S. TSP: Coaching Development Teams. Reading, MA: Addison-Wesley, 2006.
14. Deming, Edwards W. "Out of the Crisis." Cambridge, MA: MIT Center for Advanced Engineering Study, 1982.
15. Juran, J.M., and Frank M. Gryna. "Juran's Quality Control Handbook." 4th ed. New York: McGraw-Hill Book Company, 1988.
16. McHale, James, and D. Wall. "Mapping TSP to CMMI." CMU/SEI-2005-TR-014, ESC-TR-2005-014. Pittsburgh, PA: SEI, CMU, Apr. 2005.

Coming Events

March 12-15
23rd Annual National Test and Evaluation Conference
Hilton Head Island, SC
www.ndia.org

March 14-15
SQuAD 2007 6th Annual Conference
Software Quality Association of Denver
Denver, CO
www.geocities.com/lbu_measure/conf.htm#p2

March 21-23
The ServerSide Java Symposium
Las Vegas, NV
http://javasymposium.techtarget.com/lasvegas/index.html

March 25-29
CNS '07
10th Communications and Networking Simulation Symposium
Norfolk, VA
www.scs.org/confernc/springsim/springsim07/cfp/cns.htm

March 25-29
SpringSim '07
Norfolk, VA
www.scs.org/confernc/springsim/springsim07/springsim07.htm

March 26-29
SEPG 2007
Austin, TX
www.sei.cmu.edu/sepg/2007

June 18-21, 2007
2007 Systems and Software Technology Conference
Tampa Bay, FL
www.sstc-online.org

Coming Events: **Please submit conferences, seminars, symposiums, etc. that are of interest to our readers at least 90 days before registration. E-mail announcements to nicole.kentta@hill.af.mil.**

About the Authors

Watts S. Humphrey joined the SEI after his retirement from IBM in 1986. He established the SEI's Process Program and led development of the SW-CMM, the PSP, and the TSP. During his 27 years with IBM, he managed all IBM's commercial software development and was Vice President of Technical Development. He holds graduate degrees in physics and business administration. He is an SEI Fellow, an Association of Computing Machinery member, an Institute of Electrical and Electronics Engineers Fellow, and a past member of the Malcolm Baldrige National Quality Award Board of Examiners. In a recent White House ceremony, the President awarded him the National Medal of Technology.

SEI
4500 Fifth AVE
Pittsburgh, PA 15213-2612
Phone: (412) 268-6379
Fax: (412) 268-5758
E-mail: watts@sei.cmu.edu

James W. Over is a senior member of the technical staff at the SEI. Since joining the SEI in 1987, he has worked in several technical areas, including TSP, PSP, Software Process Definition, and Software Process Modeling. His interests include software engineering, software process, and quality management. Over co-authored several SEI publications on software process definition and improvement. Before joining the SEI, he was director of information systems for Pulitzer Publications. Over has more than 34 years of technical and management experience in software engineering, and attended Northern Illinois University.

SEI
4500 Fifth AVE
Pittsburgh, PA 15213-2612
Phone: (412) 268-7624
Fax: (412) 268-5758
E-mail: jwo@sei.cmu.edu

Michael C. Konrad, Ph.D., is chairman of the CMMI Configuration Control Board and has been a team leader of CMMI model development since 1998. He was also a member of teams that developed Software CMM Vers. 1.0, Software Development Capability Evaluation, and International Organization for Standardization 15504 model requirements. Konrad has 24 years experience in software engineering, holding various positions in industry and academia. He received his doctorate in mathematics from Ohio University in 1978.

SEI
4500 Fifth AVE
Pittsburgh, PA 15213-2612
Phone: (412) 268-5813
Fax: (412) 268-5758
E-mail: mdk@sei.cmu.edu

William C. Peterson worked at IBM for more than 23 years, and has been with the SEI for 13 years. At IBM, he managed software development projects and led process improvement efforts. At the SEI, he is director of the SEPM program where he is responsible for the CMMI, the TSP, the Software Engineering Measurement and Analysis, and the International Process Research Consortium initiatives.

SEI
4500 Fifth AVE
Pittsburgh, PA 15213-2612
Phone: (412) 268-7736
Fax: (412) 268-5758
E-mail: wcp@sei.cmu.edu

The *ImprovAbility*™ Model

Dr. Jan Pries-Heje
IT University of Copenhagen

Jørn Johansen, Mads Christiansen, and Morten Korsaa
DELTA Axiom

*Too many improvement and innovation projects fail. We have studied characteristics of successful and failed projects. From this study, we derived 20 parameters that influence success and failure and used those parameters to build an Improvement Ability (*ImprovAbility*) Model, which is a model that can be used to measure an organization's or project's ability to succeed with improvement. After having built the* ImprovAbility *Model, we tested it in real life, learned from the experience, and improved the model. Further tests showed promising results. In this article, we report on the considerations and research behind* ImprovAbility. *Finally, we describe the method and how the model can be used in practice.*

Software Process Improvement (SPI) is about systematically evaluating current status in relation to software processes, doing something to improve, and measuring whether the things done improved the situation. Many information technology (IT) organizations have used considerable resources for SPI. However, investments in SPI often have not led to the changes and improvements as expected. For example, Goldenson and Herbsleb [1] found in a study of a fairly large number of organizations that had invested in SPI that 26 percent agreed that *nothing much has changed* and 49 percent declared themselves to be disillusioned due to lack of improvements. This study is not alone – several others have found that SPI initiatives can fail [2, 3, 4]. This leads to the research question that we address here: How can you improve an organization's ability to improve?

We believe it is possible and important to focus on the *ability to improve*, or if you like, improvability. In this article, we report on the findings from an in-depth study of successes and failures when improving and a model – called *ImprovAbility* – built from the results (see Figure 1). First, we describe our research methodology, a qualitative interview study with more than 50 interviews from four organizations followed by an action research undertaking to build a model of ability improvement. Second, we report the findings from the interview study and how our findings were grouped into 20 influential parameters. We then give an account of the model we developed based on the parameters and how that model can be used in two ways: One, to assess organizations' ability to implement innovations and improvements based on previous projects, and second, to assess ongoing projects to minimize the risks for the project henceforward.

™ *ImprovAbility* is a trademark of Delta Axiom.

Interview Study Research Method

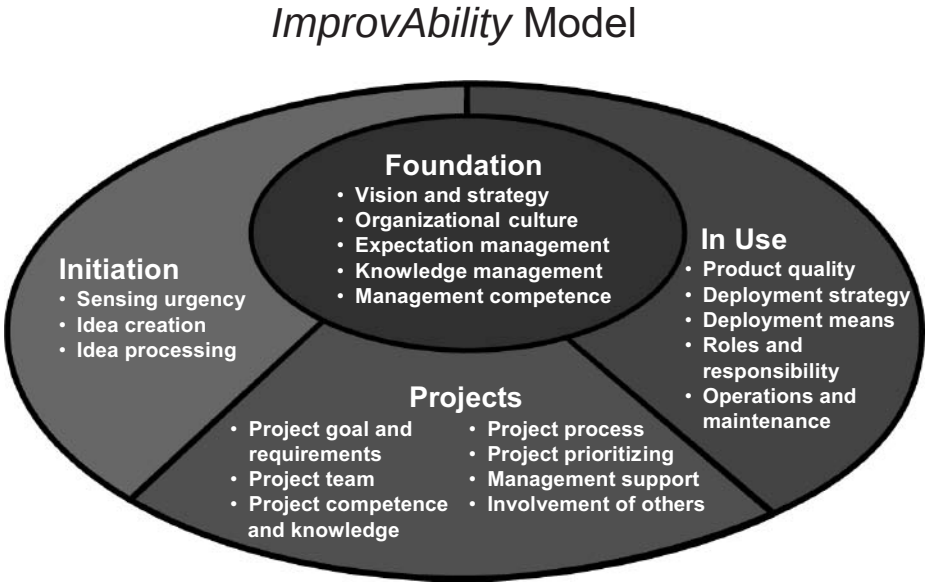
We selected successful and failed projects as an arena of particular interest from the viewpoint of improving the ability to improve. We can highlight two key reasons for this. First, we appreciate the learning that can be harvested by looking at projects in retrospect. Second, in opposition to many other studies, we decided to look at both SPI projects where other software developers are the users and at IT projects in IT organizations.

We used an existing research collaboration called Talent@IT[1] to select companies. There are four companies that participate in the research collaboration. Each of the companies was asked to appoint four projects, one successful and one failed SPI project plus one successful and one failed normal innovation project. Eventually, only 14 of the 16 projects asked for were available for our research; we included 12 scientific articles to widen the scope.

We then conducted interviews with personnel within the projects. We interviewed the project manager and one to two project members. We interviewed the sponsor or owner of the project, typically a manager in the organization. We interviewed the users; for an SPI-project, it signified other developers and for innovation projects, it typically signified end users. In 14 projects, we conducted more than 50 interviews in the period from summer 2003 to summer 2004.

Typically, every interview was conducted by two people and all interviews were transcribed and analyzed using Grounded Theory (GT) techniques. GT is a qualitative research methodology that derives its name from the practice of discovering theory that is grounded in data, i.e., this method does not begin with a theory, and then seek proof;

Figure 1: *Twenty Parameters in Four Groups for Success and Failure With Innovation and Improvement*



Vision and strategy	To what extent has the organization developed a business strategy and/or a vision that is decided and communicated?
Organizational culture	To what extent has the organization developed a culture that encourages improvement and innovation?
Expectation management	To what extent has the organization created systematic management of expectations in relation to both organizational changes and daily work?
Knowledge management	To what extent is knowledge systematically gathered, stored and used?
Management competence	To what extent has the organization developed the necessary competence at the management level?

Table 1: *Foundation Parameters*

Sensing urgency	To what extent is the organization able to sense the urgency for change? For example, because existing ways of working have become obsolete or because existing products are too old or maybe the organization has simply arrived in an untenable position.
Idea creation	To what extent is the organization able to identify, foster, and create many ideas for new SPI and IT processes or products? Preferably from many different sources such as user needs, new technology, or new strategies.
Idea processing	To what extent are new ideas captured and decided on?

Table 2: *Initiation Parameters*

Project goal and requirements	To what extent are project goals, expected benefits, and formulated requirements precise, unambiguous, and stable? Do the projects – developers as well as users – perceive their goals and the rationale behind as reasonable?
Project team	To what extent are the people allocated to projects highly motivated, and are they having the right attitude and profile for the projects? Is there a competent project manager on the team? Team sitting physically together and close to users? Does the team work as a team?
Project competence and knowledge	To what extent do the projects have the necessary technical knowledge? Domain knowledge? Development model and method(s)?
Project process	To what extent do the projects have good estimates, plans, follow-up, risk management, testing, and quality reviews?
Project prioritizing	To what extent are projects prioritized in relation to each other? And in relation to schedule, cost, scope and quality? Are priorities communicated and understood? Are priorities stable?
Management support	To what extent is management in the organization supporting the projects? This could include allocating the right resources at the right time, participating in a steering committee, or demanding results.
Involvement of others	To what extent are other stakeholders (than the team and management) involved? This could, for example, include early user involvement. External resources? Consultants? At the right time and in the right way?

Table 3: *Project Parameters*

instead, it begins with an area of study and allows the relevant theory to emerge from that area [5].

After having collected our interview data, we applied the three coding procedures of GT. According to [5], analysis in a GT approach is composed of three groups of coding procedures called open, axial, and selective coding. These procedures do not entirely occur as a sequence, but each overlaps the others and iterates throughout the research project.

The goal of open coding is to reveal the essential ideas found in the data. Open coding involves two essential tasks. The first task is labeling phenomena. This task involves decomposing an observation into discrete incidents or ideas. Each discrete incident or idea receives a name or label that represents the phenomenon. These names represent a concept inherent in the observation. The second essential open-coding task is discovering categories. Categorizing is the process of finding related phenomena or common concepts and themes in accumulated data and grouping them under joint headings, thus identifying categories and sub-categories of data.

In our analysis, we found 54 categories that all contributed to either the success or failure of a project. Three examples of categories are the following: user involvement, defect in product, and stakeholder involvement.

Developing a better and deeper understanding of how the identified categories are related is the purpose of axial coding. The first task in axial coding connects categories in terms of a sequence of relationships. For example, a causal condition or a consequence can connect two categories, or a category and a sub-category. The second task turns back to the data for validation of the relationships. This return gives rise to the discovery and specification of the differences and similarities among and within the categories. This discovery adds variation and depth of understanding.

The first part of the axial coding was done together by four people. Similarities and differences were noted and discussed. Categories and relationships were identified, discussed, corrected, and changed until a common understanding of the categories, sub-categories, and their relationships was reached. Concretely, we ended up with 19 categories. To distinguish the 19 categories from the 54 coming out of the open coding, we called them the *19 parameters*.

Selective coding involves the integration of the categories that have been developed to form the initial theoretical framework. Firstly, in selective coding, a storyline is either generated or made explicit. A story is simply a descriptive narrative about the central phenomenon of study and the storyline is the conceptualization of this story (abstracting). The storyline we ended up with was, in fact, a story that states that the ability of an organization to produce success and avoid failure – the ability to improve – depends on the organization's ability to cope with the following four groups of parameters:

- Parameters related to *initiation* of projects, i.e., ideas for new SPI or innovation projects.
- Parameters related to *projects*, from the very first hour until a result is taken into use.
- Parameters related to *results in use*, i.e.

from when the first user starts using the new process or product for the first time until full deployment. This can be a long period of time or a one-time delivery depending on the context.

- Parameters related to the enterprise *foundation*, i.e., the environment and conditions for projects in the organization (e.g. organizational culture, management style and competence, and expectation and knowledge management).

The *ImprovAbility* Model

Our first model included 19 parameters, but testing the model revealed the need for one more parameter: operations and maintenance as indicated in the In Use group (see Figure 1, page 23).

The resulting model with 20 parameters in four groups looks like it is depicted in Figure 1. The core assumption behind this model is that the parameters identified from successful and failed projects can be used to identify an organization's ability to improve by encouraging activity that has shown to be related to success and avoiding activity that has shown to lead to failure.

Each of the 20 parameters in the model is described in Tables 1-4.

For each of 20 parameters in the four groups we have formulated a number of questions. The questions are based on our observations (the transcribed interviews plus the 12 scientific articles) and the grounded theory coding.

An Example of Questions for a Parameter

Let us, as an example, take the parameter *deployment strategy* from the In Use group. In Figure 2, we have shown the questions we derived for this specific parameter. The figure shows part of a spreadsheet that can be used to measure the *ability to improve* by an organization.

Process to Measure Ability With *ImprovAbility*

To bring *ImprovAbiliy* into use we designed a process to be used in an organization by assessors from outside the assessed organization. The process includes a number of meetings and activities as shown in Figure 3.

The method for gathering information during an assessment is inspired primarily by the Bootstrap method [6]. An assessment starts with a preparatory meeting, where, respectively, the assessors and key persons in the organization

Product quality	To what extent are new processes and products that are deployed of high quality? Few defects? User friendly? Low complexity? Compatible? Efficient? Have relative advantages for the user?
Deployment strategy	To what extent is a deployment strategy for new processes or products decided? Are the related plans followed also when deadline pressure arises at the end of the project?
Deployment means	To what extent is the optimal mix of information, communication, education and training, plus marketing of new processes and products applied? Optimal mix depends on the context and is planned as part of the deployment strategy.
Roles and responsibility	To what extent are roles and responsibilities in relation to deployment and use well defined and enacted?
Operations and maintenance	To what extent is it possible to operate the product or process? To what extent is it possible to maintain the product or process?

Table 4: *In Use Parameters*

Deployment Strategy		**N**	**P**	**L**	**F**	**NA**
1.	To what extent is a deployment strategy for new processes or products decided on and followed?	Score: 50				
1.a	To what extent is there a procedure for selecting a deployment strategy?			X		
1.b	To what extent are risks in relation to deployment uncovered?		X			
1.c	To what extent is there a plan for deployment (time, milestones, responsibility)?				X	
1.d	To what extent are deployment strategies and plans followed?	X				

Note: Excerpt from spreadsheet with questions used to measure the ability for the parameter deployment strategy. The scale used is *N* for not (counting as zero), *P* for partly (counting as 1/3), *L* for largely (counting as 2/3), and *F* for fully (counting as 3/3). The score is then calculated as a percentage of fully answers on. Here it is (2/3+1/3+3/3+0/3)/4*100 = 50. *NA* = Not Applicable and does not contribute to the score.

Figure 2: *Deployment Strategy*

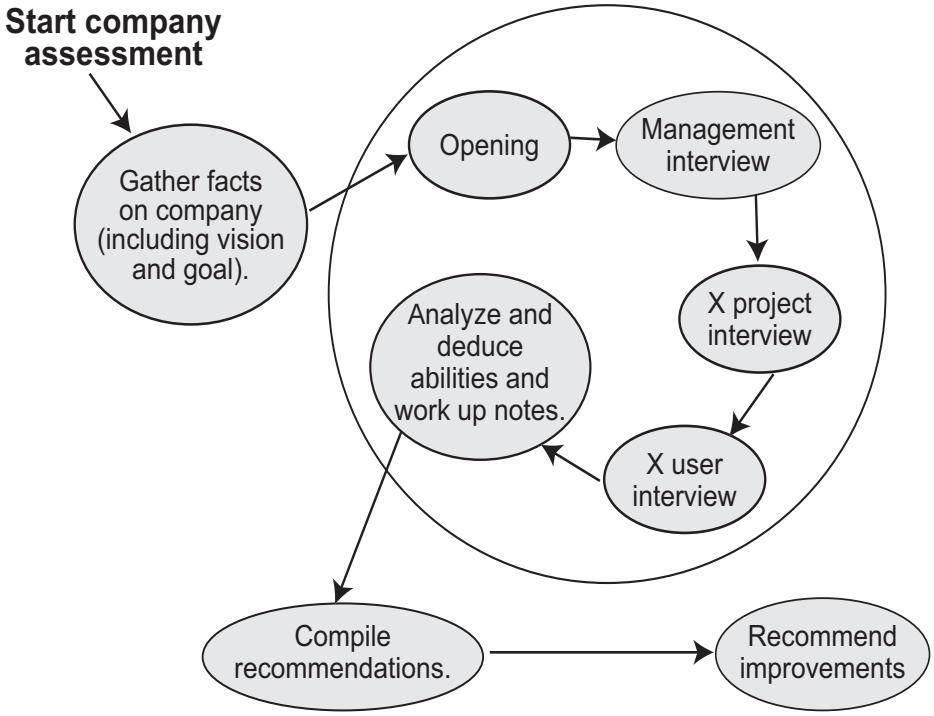


Figure 3: *How an ImprovAbility Assessment Is Conducted*

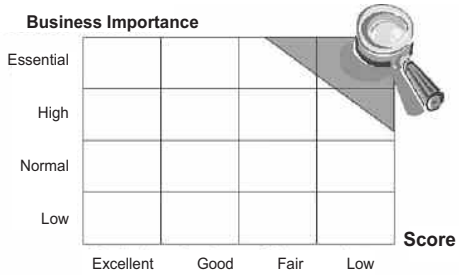


Figure 4: *Selecting Parameters for Improvement*

prepare for the assessment, gather facts on the organization, and clarify who is to say what at the opening meeting. This meeting is scheduled as one hour.

At the opening meeting, all persons involved should be present. At this meeting, the concept of the model and method, the purpose of the assessment, the plan and activities, the type of results, and the use and the results are explained in detail.

The data collection part of the assessment is a series of four hour interviews in the organization. Each interview includes two interviewing assessors and five to seven interviewees who are interviewed about each of the 20 *ImprovAbility* parameters. We start interviewing the management group and then follow with at least two project interviews in either process improvement or product development projects. The two project interviews must cover at least three projects. Finally we interview one or more groups of users of the same kind of products to make sure to cover the parameters from the *In-Use* group.

The interviews are carried out as open dialogues where the two assessors ensure that the discussions cover the subjects and all 20 parameters. After a group interview, the assessors answer the questionnaire in spreadsheet form(as shown in Figure 2). The spreadsheet generates a picture of strong and weak parameters on a scale from zero to 100. This is done for each interview.

When all interviews and scoring are complete, we have a measure of the strong (high scoring parameters) and weak (low scoring) areas in the organization. But in order to select parameters for improvement, it is also necessary to identify which parameters are important for the particular business. This is done during a prioritizing practice with management. In an open discussion, the managers are asked to prioritize the 20 parameters in four groups: very low importance, normal, high importance, and essential. Before they prioritize they are given two rules: at most three parameters must be essential, and at least three parameters should be low.

The 20 parameters are then positioned in a 4x4 matrix as shown in Figure 4. The x-axis represents the relative parameter score and the y-axis represents the priority given at the management meeting. In the upper right corner of the matrix, we now have the essential parameters with a low score and from that area we select three to five parameters for recommendations. It is here, for example, that we recommend that the organization focus their attention so they can improve their ability to improve.

To derive the concrete recommendation we use a catalogue of improvement methods and techniques. In fact as part of the *ImprovAbility* model we have a catalogue where for each parameter we can find inspiration on how to improve the concrete parameter. The catalogue is also a product of our coding of interview data for successful techniques and methods plus a literary study. A recommendation for the deployment strategy parameter could include – but are not limited to – the following:

Prepare deployment plans and make the following:

- Target group analysis (who, how many, when, how much) with an evaluation of the target groups pre- and post-condition.
- Risk analysis for deployment.
- Cost / benefit analysis.
- Definition of deployment roles and responsibilities.

During the assessment, factual data about the organization and its current strategic improvement initiatives are deducted. This is used to describe and illustrate the scope for the planned or already initiated changes. From studies of change management literature, we have identified 10 different change strategies. Some of the strategies have commonalities, others are quite different, and some are very much incompatible. It is therefore a difficult task for a company to choose the best change strategy, but as part of the research project we developed a spreadsheet based questionnaire to identify which strategy is best suited for a company facing a change. For example, Business Process Re-engineering (BPR) can be very useful in companies who are stuck and do not make money, where it would be a bad strategy to throw away all existing processes in companies who have their processes in place and make a lot of money. The best change strategy is identified during the management interview of the assessment and results in a prioritized list among the 10 change strategies in Table 5.

Finally, the assessors use all the collected data, parameter scores, the completed 4x4 matrix, the overall improvement practice, and the scope of strategic improvement initiatives to generate recommendations and produce a presentation for the closing meeting. The presentation is shown to management and afterwards shown to all involved in the

Table 5: *An Overview of the 10 Organizational Change Strategies*

Strategy	Definition
Commanding	Change is driven and dictated by (top) management. Management takes on the roles as owner, sponsor, and change agents.
Employee driven	Change is driven from the bottom of the organizational hierarchy when needs for change arise among employees.
Exploration	Change is driven by the need for flexibility and agility or a need to explore new markets, technology, or customer groups.
Attitude driven	Change is driven by a focus on organizational learning, individual learning and what creates new attitudes and behavior.
Metrics driven	Change is driven by metrics and measurements.
Optionality	Change is driven by the motivation and need of the individual or group. It is to a large degree optional whether the individual takes the innovation into use.
Production organized	Change is driven by the need for optimization and/or cost reduction.
Re-engineering	Change is driven by fundamentally rethinking and redesigning the organization to achieve dramatic improvements.
Socializing	Change in organizational capabilities is driven by working through social relationships. Diffusion of innovations happens through personal contacts rather than through plans and dictates.
Specialist driven	Change is driven by specialists, either with professional, technical, or domain knowledge.

assessment at the closing meeting.

Experiences Using the ImprovAbility Model

We have tested the model three times on the organizational level with promising results. In a medium size financial company, the manager of the IT Division (Chief Information Officer [CIO]) was most enthusiastic about the overall improvement strategy that we suggested. Based on our interviews we suggested that they used *attitude driven* and *socializing* as their main strategies for changing the organization and avoid re-engineering and commanding. The CIO called this the major *Aha!* experience for him as he had previously tried to create a burning platform, i.e., re-engineering and using a commanding strategy. In both cases, no changes really took place, so the CIO felt that the *attitude driven* and *socializing* change strategies made a lot of sense for him. At the closing meeting the CIO also committed to following the recommendations – not in detail but in principle. The other assessments were carried out in a large pension scheme enterprise and in the process department (SPI) in a privately owned software and systems company certified to CMMI Level 5. The results were appreciated as making good sense and reflecting their reality.

The Talent@IT partners identified a need for a special project level version of *ImprovAbility* where only a project team from an ongoing project is interviewed. In this case, the interviewees can only answer based on their expectations and experiences from previous projects. The outcome of the assessment is a focus on the risks for the project henceforth and the recommendations are used to reduce the risks of the project and increase their likelihood for success. We have tested the project version in nine projects from different business areas, covering projects of different size, complexity, and maturity level. We have seen a big variation in parameters for recommendation, but the data material is so far not big enough to spot any trends. However, we have seen that quite often *involvement of others* and the *deployment parameters* come up with weak scores, but further research has to confirm or invalidate that.

Conclusion

We are often asked how *ImprovAbility* compares with traditional maturity models like CMMI [7]. Our answer is that we have tried to group all the categories of our findings that were related to CMMI into the parameters of project team, project process, and project goal and requirements. This means for example, that if project process is selected for recommendation, the recommendation could include making a CMMI assessment to identify more precisely which processes should be improved first.

CMMI is a model that concerns the process behind product development and an assessment identifies which processes needs to be improved, i.e., what to change. *ImprovAbility* is not a maturity model but is a model that concerns the process behind *changing* the product development process. In other words, why do some have success with CMMI and others do not? So *ImprovAbility* is your concern if you want to identify how to organize and ensure success with CMMI based improvements, i.e., how to change. The organization assessment will help you improve the way changes are introduced into the organization, be it with new or improved processes or new product developments. Where the literature is full of change methodologies, *ImprovAbility* helps you define which one will work the best for you.

> *"ImprovAbility is not a maturity model but is a model that concerns the process behind changing the product development process. In other words, why do some have success with CMMI and others do not?"*

The *ImprovAbility* project assessment is very useful to assist running projects in becoming successful. For process improvements a CMMI or International Organization for Standardization 15540 assessment is very useful to identify which development processes needs improvement. Once this has been done and a project is launched, the *ImprovAbility* project assessment can identify how to plan and minimize risk for the improvement project.

Finally, even though we have now reached a stage where we find it fruitful to report our findings in this article, we recognize the need for more tests. We have, therefore, already planned a fourth action research testing to consolidate and improve the model. So the story will be continued …◆

References

1. Goldenson, Dennis R., and James D. Hersleb. "After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors That Influence Success." Technical Report CMU/SEI-95-TR-009. Pittsburgh, PA; SEI, Carnegie Mellon University, 2005.
2. El-Emam, Khaled, et al. "Modeling the Likelihood of Software Process Improvement: An Exploratory Study. Empirical Software Engineering." 6.3 (2001).
3. Blanco, et al. "SPI Patterns: Learning From Experience." IEEE Software (May/June 2001): 28-35.
4. Rainer, Austen, and Tracy Hall. "Key Success Factors for Implementing Software Process Improvement: A Maturity-Based Analysis." The Journal of Systems and Software 62 (2002): 71-84.
5. Strauss, A., and J. Corbin. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Beverly Hills, CA: Sage Publications, 1998.
6. Kuvaja, Pasi, et. al. Software Process Assessment and Improvement: The BOOTSTRAP Approach. Oxford, UK: Blackwell Publishers. 1994.
7. Chrissis, Mary Beth, Mike Konrad, and Sandy Shrum. CMMI – Guidelines for Process Integration and Product Improvement. Reading, MA: Addison-Wesley, 2003.

Note

1. The Talent@IT is a three-year research project (2003-2006) sponsored by the Danish Ministry of Science, Technology and Innovation. The project partners are the IT-University of Copenhagen (research responsible), the Approved Technological Service Institution DELTA Axiom (project management and *owner* of the *ImprovAbility* model) and the four Danish enterprises ATP, Danske Bank, Payment Business Services, and SimCorp. More information can be found on <www.talent-it.dk>.

CROSSTALK
The Journal of Defense Software Engineering

Get Your Free Subscription

Fill out and send us this form.

517 SMXS/MXDEA
6022 FIR AVE
BLDG 1238
HILL AFB, UT 84056-5820
FAX: (801) 777-8069 DSN: 777-8069
PHONE: (801) 775-5555 DSN: 775-5555

Or request online at www.stsc.hill.af.mil

NAME:_____

RANK/GRADE:_____

POSITION/TITLE:_____

ORGANIZATION:_____

ADDRESS:_____

_____

BASE/CITY:_____

STATE:_____ZIP:_____

PHONE:(____)_____

FAX:(____)_____

E-MAIL:_____

CHECK BOX(ES) TO REQUEST BACK ISSUES:

- OCT2005 ☐ SOFTWARE SECURITY
- NOV2005 ☐ DESIGN
- DEC2005 ☐ TOTAL CREATION OF SW
- JAN2006 ☐ COMMUNICATION
- FEB2006 ☐ NEW TWIST ON TECHNOLOGY
- MAR2006 ☐ PSP/TSP
- APR2006 ☐ CMMI
- MAY2006 ☐ TRANSFORMING
- JUNE2006 ☐ WHY PROJECTS FAIL
- JULY2006 ☐ NET-CENTRICITY
- AUG2006 ☐ ADA 2005
- SEPT2006 ☐ SOFTWARE ASSURANCE
- OCT2006 ☐ STAR WARS TO STAR TREK
- NOV2006 ☐ MANAGEMENT BASICS
- DEC2006 ☐ REQUIREMENTS ENG.
- JAN2007 ☐ PUBLISHER'S CHOICE

TO REQUEST BACK ISSUES ON TOPICS NOT LISTED ABOVE, PLEASE CONTACT <STSC.CUSTOMERSERVICE@HILL.AF.MIL>.

About the Authors

Jan Pries-Heje, Ph.D., works at the IT University of Copenhagen and is also a part-time professor at the IT-University in Gothenburg, Sweden and is responsible for research in the project reported in this article. Pries-Heje's main research interests are information systems development, software engineering, and software process improvement. He has carried out action research with industry on specific topics such as process improvement, high speed software development, IT project management, requirements specification, and successful organizational change with IT. Pries-Heje has a doctorate from Copenhagen Business School.

IT University of Copenhagen
Department of Design and Use of IT
Rued Langgaards Vej 7
DK-2300 Copenhagen S
Denmark
E-mail: jph@itu.dk

Jørn Johansen has more then 25 years experience in IT. He has worked for 15 years in a Danish company with embedded and application software as a developer and project manager. For the past 11 years, Johansen has worked at DELTA Axiom processes as a consultant, BOOTSTRAP, SPICE, and CMMI assessor. Jørn was project manager in the Talent@IT project developing the *ImprovAbility* model. He has a masters degree in electrical engineering.

DELTA Axiom
Venlighedsvej 4
DK-2970 Hørsholm
Denmark
E-mail: joj@delta.dk

Mads Christiansen has 27 years experience with IT. He has worked for 19 years in a Danish company with embedded software and PC applications as developer and project leader. For the past eight years, Christiansen has been working as senior consultant at DELTA Axiom processes with a special focus on software process improvements, user centered design, and *ImprovAbility* assessment plus training of *ImprovAbility* project assessors. He has a masters degree in electrical engineering.

DELTA Axiom
Venlighedsvej 4
DK-2970 Hørsholm
Denmark
E-mail: mc@delta.dk

Morten Korsaa has focused his 16 years professional career on development processes and improving their efficiency. He has been globally responsible for process improvement activities in a 2500+ developer organization and has experienced a significant number of process improvement projects. Korsaa brought this experience, plus the experience coming from maturity assessments in more than 60 projects, into the development of the *ImprovAbility* model.

DELTA Axiom
Venlighedsvej 4
DK-2970 Hørsholm
Denmark
E-mail: mko@delta.dk

Applying International Software Engineering Standards in Very Small Enterprises

Claude Y. Laporte and Alain April
École de Technologie Supérieure

Alain Renault
Public Research Center Henri Tudor

At a time when software quality is increasingly becoming a subject of concern, and process approaches are maturing and gaining acceptance in companies, the use of International Organization for Standardization (ISO) systems and software engineering standards remains limited to a few of the most popular ones. However, these standards were not written for enterprises with fewer than 25 employees in mind. As they are difficult to apply in such settings, a new international standardization project has been mandated to address some of those difficulties by developing profiles and by providing guidance for compliance with ISO software engineering standards in very small enterprises (VSEs). A survey was conducted to ask very small enterprises about their utilization of ISO/IEC JTC 1/SC7[1] information technology (IT) standards and to collect data to identify problems and potential solutions to help them apply standards.

In Europe, 85 percent of IT sector companies have between 1 and 10 employees [1]. A survey of the Montréal area in Canada has revealed that close to 80 percent of companies that develop software have fewer than 25 employees [2]. More than 50 percent have fewer than 10 employees. There is a need to help these organizations, which are defined as VSEs, to understand and use the concepts, processes, and practices proposed by the ISO's international software engineering standards.

At the Brisbane meeting of ISO/IEC JTC 1/SC7 in 2004, Canada raised the issue of small enterprises requiring standards adapted to their size and maturity level. A meeting of interested parties was held with delegates from five national bodies, at which a consensus was reached on the general objectives:

- Make the current software engineering standards more accessible to VSEs.
- Provide documentation requiring minimal tailoring and adaptation effort.
- Provide harmonized documentation integrating available standards.
- Align profiles with the notions of maturity levels presented in ISO/IEC 15504.

It was also decided that a special interest group be created to validate these objectives, as well as to assign priorities and develop a project plan.

In March 2005, the Thailand Industrial Standards Institute (TISI) invited a number of software experts to advance the work items defined at the Brisbane meeting. A key topic of discussion was to clearly define the size of VSE that would be targeted by a future SC7 Working Group (WG). A consensus was reached to define our target VSE as IT services, organizations, and projects with between one and 25 employees. The major output of this meeting was a draft of a New Work Item (NWI) that would be discussed at the next SC7 Plenary meeting. A list of actions that could be undertaken by a future SC7 WG was also developed.

In May 2005, at the SC7 Plenary Meeting in Finland, a resolution was approved to ballot a proposal for the development of software life-cycle profiles and guidelines for use in very small enterprises. The following describes the mandate [3]:

- Provide VSEs with a way to be recognized as producing quality software systems, which would lessen the effort required to implement and maintain the entire suite of ISO systems and software engineering standards.
- Produce guides which will be easy to understand, short, simple, and readily usable by VSEs.
- Produce a set of profiles and provide guidance to VSEs in establishing selected processes.
- Address the market needs of VSEs by allowing for domain-specific profiles and levels.
- Provide examples of use.
- Provide a baseline for how multiple VSEs can work together or be assessed as a project team on projects that may be more complex than can be performed by any one VSE.
- Develop scalable profiles and guides so that compliance with ISO/IEC 12207 and/or ISO 9001:2000 and ISO/IEC 15504 process assessment becomes possible with a minimum impact on VSE processes.

The proposal was accepted, and 12 countries committed to participating in the new working group: Belgium, Canada, the Czech Republic, Ireland, Italy, Japan, Korea, Luxemburg, South Africa, Thailand, the United Kingdom, and the United States.

A new WG24 was established, made up of the following members, in addition to individuals sent by their national bodies:

- Tanin Uthayanaka (Thailand), who was appointed Convener.
- Claude Y. Laporte (IEEE Computer Society), who was appointed Project Editor.
- Jean Bérubé (Canada), who was appointed Secretary.

The TISI invited a Special Working Group, in September 2005, to prepare material to facilitate the start-up of the new working group. The main proposals of the meeting were the following:

- Requirements for International Standard Profiles (ISPs) based on technical report ISO/IEC TR10000-1 [4].
- A survey on VSE exposure and their need for software development life cycles.
- Approaches to profile development.
- Business models.
- Agenda for the first WG24 meeting.
- Draft strategic plan for WG24.

In October 2005, WG24 held its first working sessions in Italy to accomplish the following:

- Present the project to the official members of WG24.
- Finalize project requirements to constitute the project baseline.
- Gain consensus among WG members and obtain their commitment regarding the project.
- Process the comments received during the balloting of the NWI.
- Define the profile creation strategy.
- Identify lists of situational factors and business models.
- Build survey material to validate project requirements and to collect missing information from VSEs.

After the meeting, the survey questionnaire was translated into nine languages. In addition, a Web site, hosted by the École de Technologie Superieure, was developed to maximize the number of responses, which were collected between Feb. 20 and May 12.

In May 2006, WG24 members met at the SC7 Plenary meeting in Thailand. Two new countries, India and Mexico, sent delegates to

WG24. The main outputs of the meeting were the following:

- Four hundred thirty-seven responses were collected from 32 countries.
  - o Two hundred nineteen responses were received from enterprises with 25 employees or less.
- More than 67 percent indicated that it was important to be either recognized or certified (e.g. ISO, market).
- WG24 decided to prioritize the development of profiles and guides for organizations with 25 employees or less (total staff). These profiles and guides should also be usable for projects and departments of less than 25 employees.
- WG24 decided to propose separate profiles for the following:
  - o Enterprises with fewer than 10 employees.
  - o Enterprises with 10 to 25 employees.
- Evaluation of documents tabled by national delegations.
- Selection of the Mexican Standard [5] as an input document for the development of profiles and guides.

The next WG24 meeting will be held in Russia in May 2007. To complete the survey, go to: <http://iso-iec-sc7wg24.gelog.etsmtl.ca/Webpage/iso-iec-sc7wg24_english.html>. Username: isosurvey; Password: vse.

References

1. European Software Institute <www.esi.es/en/main/iitmark.html>.
2. Laporte, C.Y., et al. "Initiating Software Process Improvement in Small Enterprises: Experiment with Micro-Evaluation Framework, SWDC-REK." International Conference on Software Development, 27 May to 1 June, 2005, University of Iceland, Reykjavik, Iceland.
3. ISO. New Work Item Proposal – Software Life Cycles for Very Small Enterprises. ISO/IEC JTC 1/SC7 N3288. May 2005 <http://www.jtc1-sc7.org/>.
4. ISO. "Information Technology – Framework and Taxonomy of International Standardized Profiles – Part 1: General Principles and Documentation Framework." ISO/IEC TR 10000-1. 4th ed. 1998.
5. NMX-059-NYCE-2005. "Information Technology-Software-Models of Processes and Assessment for Software Development and Maintenance. Part 01: Definition of Concepts and Products; Part 02: Process Requirements (MoPro Soft); Part 03: Guidelines for Process Implementation; Part 04: Guidelines for Process Assessment (EvalProSoft)." Ministry of Economy, Mexico, 2005.

Note

1. ISO/IEC JTC 1/SC7 stands for the International Organization for Standardization/International Electrotechnical Commission/Joint Technical Committee 1/ Sub Committee 7.

About the Authors

Claude Y. Laporte and Alain April are software engineering professors at the École de Technologie Supérieure. Laporte is the editor of SC7 WG24 tasked with developing software life cycle profiles and guidelines for use in very small enterprises. April has contributed to ISO 9126 (Part 3), and is the associate editor of the Software Engineering Body of Knowledge software maintenance and quality chapters that have recently been published as an ISO/IEC TR 19759. Alain Renault is project leader at the Public Research Center Henri Tudor in Luxembourg. He is also a member of SC7 WG24.

École de Technologie Supérieure
Department of Software and IT Engineering
1100, rue Notre-Dame Ouest
Montréal, Québec
Canada, H3C 1K3
E-mail: claude.y.laporte@etsmtl.ca

Public Research Center Henri Tudor
29, avenue John F. Kennedy
L-1855 Luxembourg-Kirchberg
Luxembourg

The Joint Services
STC
Systems & Software Technology Conference

18-21 June 2007 • TAMPA BAY, FLORIDA

Systems and Software Technology - Enabling the Global Mission

Don't miss this must attend event for:

- Acquisition Professionals
- Program/Project Managers
- Programmers
- System Developers
- Systems Engineers
- Process Engineers
- Quality and Test Engineers

Save the dates and join us in Florida!

TAMPA Bay

www.sstc-online.org

Presentations will be presented in the following categories:

Rapid Response Capability
- Open Architecture
- Joint Rapid Acquisition Cell (JRAC)
- Disciplined Agile Development

Robust Engineering – Engineering for the Global Mission
- Systems of Systems Engineering
- Robust Software Engineering
- Software Product Lines
- Engineering for Manufacturing
- Adaptability

System Assurance – Addressing the Global Threat
- Information Assurance
- Software Assurance
- Anti-Tamper
- Open Source

Technology Futures
- New Computational Methods
- Time-Defined Delivery
- Technology Maturity

Communication Infrastructure
- Networks
- Interoperability
- Disaster Response

Enabling the Workforce
- Certification and Training
- National Security Personnel System (NSPS)

Hippocrates and the Oath

The recitation of the Hippocratic Oath, a rite of passage for physicians, outlines the ethical practice of medicine. It is widely believed that the oath was written by Hippocrates, the father of medicine, in the 4th Century B.C., or by one of his students [1].

Would software engineers have the integrity to take such an oath? If so, I suggest applying the concept of reuse to modify the classical version of the Hippocratic Oath [2] as follows:

I swear by Brooks, Booch, and Boehm, and I take to witness all the gods, all the goddesses, to keep according to my ability and my judgment, the following Oath.

To consider dear to me, as my parents, him who taught me this art; to live in cubicles with him and if necessary to share my computer with him; to look upon his children as my own brothers, to teach them this art if they so desire for caffeine and pizza; to impart to my sons and the sons of the guru who taught me and the bit-benders who have enrolled themselves and have agreed to the rules of the profession, but to these alone the passwords and root access.

I will prescribe applications for the good of my customers according to my ability and development environment and never do harm to anyone outside of clueless program managers.

To please no one will I unleash malicious code nor give advice which may cause a fatal crash.

Nor will I give a virus to abort a program.

But I will preserve the purity of my designs and my code.

I will not hack the operating system, even for customers in whom the requirement is manifest; I will leave this operation to be performed by system administrators, specialists in this art.

In every organization where I come I will enter only for the good of my salary, keeping myself far from all intentional ill-doing and all seduction and especially from the pleasures of the internet with jpeg or streaming video, be they free or premium.

All that may come to my knowledge in the exercise of my profession or in daily Googles, which ought not to be spread to competitors, I will keep secret and will never reveal, until sacked.

If I keep this oath faithfully, may I enjoy seclusion and practice my art, inspected by no men at any time; but if I swerve from it or violate it, may the reverse be my lot.

Modern physicians found Hippocrates' Oath a bit antiquated and in 1964 Louis Lasagna, Academic Dean of the School of Medicine at Tufts University penned a modern version of the Hippocratic Oath [3]. Maybe an adaptation of this modern version is more applicable?

I swear to fulfill, to the best of my ability and judgment, this covenant:

I will respect the hard-won scientific gains of those software engineers in whose steps I walk, and gladly share such knowledge as is mine with those who are to follow.

I will apply, for the benefit of the technophobes, all techniques required, avoiding those twin traps of over design and requirements nihilism.

I will remember that there is art to software as well as science, and that warmth, sympathy, and understanding may outweigh the program manager's schedule or the accountant's budget.

I will not be ashamed to say "what requirement," nor will I fail to call in technical support when the skills of another are needed for a system recovery.

I will respect the privacy of my customers, for their problems are not disclosed to me that the world may know. I'll leave that to Oprah. Most especially I must tread with care in matters of life and death. If it is given me to save a project, all thanks. But it may also be within my power to take a better paying job; this awesome responsibility must be faced with great humbleness and awareness of my own frailty. Above all, I must not play Bill Gates or Larry Ellison.

I will remember that I do not treat a feedback loop or a stealthy bug, but a sick human being who needs software, whose elusive requirements may affect the person's sanity and economic stability. My responsibility includes these related problems, if I am to get paid.

I will prevent inaccurate estimates whenever I can, for prevention is preferable to overruns.

I will remember that I remain a member of society, with special obligations to all my fellow human beings, those sound of mind and body as well as project managers and process zealots.

If I do not violate this oath, may I enjoy solitude and the best tools, feared while I live and forgotten with affection thereafter.

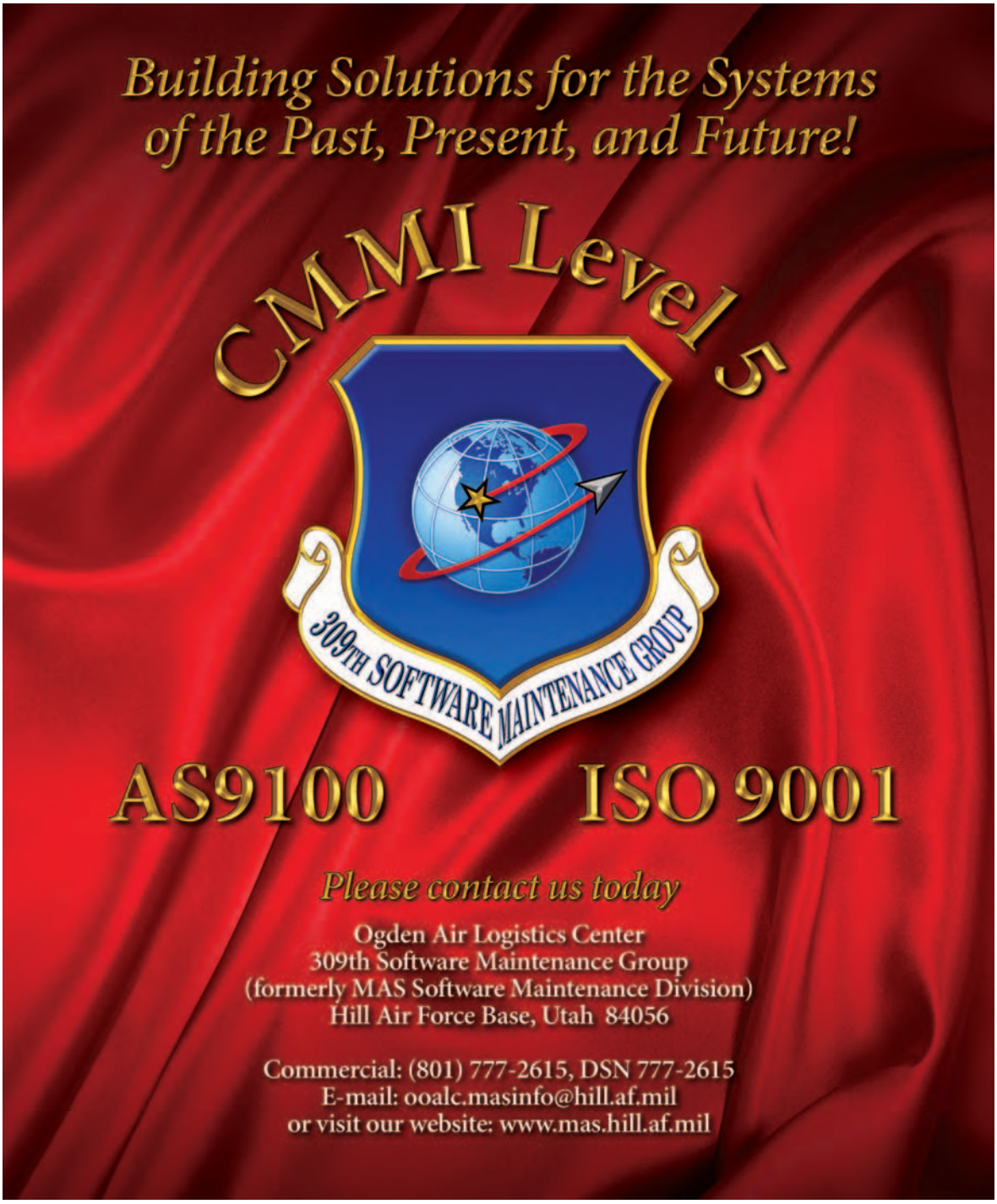
May I always act so as to preserve the finest traditions of my calling and may I long experience the joy of astounding those who seek my help.

Keep your own professional oath, or "… the home we never write to, and the oaths we never keep, and all we know most distant and most dear, across the snoring barrack-room return to break our sleep." [4]

References

1. Farnell, L.R. Greek Hero Cults and Ideas of Immortality. Kessinger Publishing, 2004
2. <http://en.wikipedia.org/wiki/Hippocratic_Oath>.
3. <www.pbs.org/wgbh/nova/doctors/oath_ modern.html>.
4. Kipling, Rudyard. Complete Verse. Anchor, 1998.

— Gary A. Petersen
Shim Enterprises Inc.
gary.petersen@shiminc.com



CrossTalk / 517 SMXS/MXDEA
6022 Fir AVE
BLDG 1238
Hill AFB, UT 84056-5820

PRSRT STD
U.S. POSTAGE PAID
Albuquerque, NM
Permit 737

CrossTalk is co-sponsored by the following organizations:

Homeland Security