18

Experience Using a Benchmarking Workshop to Explore One Route to Practical Technology Introduction

P. Fowler
Software Engineering Institute, Carnegie Mellon University
Pittsburgh, PA 15213 USA, Tel. (412) 268-7748, FAX (412) 268-5758, pjf@sei.cmu.edu

M. Patrick
Process Advantage Technology
Benicia, CA 94510 USA, Tel. (707) 745-6875, FAX (707) 746-5205, mac@patech.com

Abstract

This paper describes the concept of 'transition packages,' a proposed approach to practical technology introduction for software organizations, and why we chose to evaluate this concept in the context of the Software Capability Maturity Model. The paper also describes how this concept was considered in a practitioner-oriented workshop based on benchmarking principles, the processes and findings of that workshop, and the outputs of participant exercises. The final section provides analysis of the findings and suggests possible next steps in the creation of the first Requirements Management transition package for software engineering community use.

Keywords

Software technology introduction, software technology adoption, benchmarking, capability maturity model, software CMM, transition package, requirements management, whole product, key process area.

T. McMaster et al. (eds.), *Facilitating Technology Transfer through Partnership*

1 INTRODUCTION

The Software Engineering Institute (SEI) is a research and development center established in 1984 by contract between the United States Federal Government and Carnegie Mellon University, Pittsburgh, Pennsylvania. The SEI mission, broadly stated, is to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software. In addition to supporting and expediting the transfer of emerging software engineering technologies, the SEI also facilitates the broad introduction of best (or better) practices into the software engineering community. The goal of the latter work is the improvement of the general quality of software-intensive products through improving the approach to building these products, and through expediting the application of existing, mature technologies. This paper describes the background, design, and findings of a workshop held to evaluate one approach – the 'transition package' – to broad adoption of improved practice in the area of Requirements Management, a Software Capability Maturity Model (CMM[sm])* Level 2 Key Process Area (KPA) (Paulk 1995).

Our hypothesis is that by using transition packages, which would codify and systematize the adoption of technology and processes that support Key Process Areas, software organizations can achieve the objectives of the CMM Key Process Areas significantly more rapidly than at present. Current figures indicate it typically takes 26-35 months for organizations to move from Software CMM Maturity Level 1 to Level 2 (SEI Software Engineering Measurement Team 1996). Yet despite the value that might be derived from building on the experience of others, there is limited opportunity to do so. An informal analysis of the annual Software Engineering Process Group conference materials from 1992 through 1996 revealed few presentations† describing implementation strategy at the Key Process Area level of detail. Because of concern for disclosure of proprietary material, private sector organizations do not want to share their solutions. United States government organizations by law° are required to make all information publicly available, but informal technical interchange is the norm – it is rare to find organizations swapping artifacts such as templates for software requirements specification documents, examples of policy statements, or software tool vendor selection criteria. It is also rare to find technical interchange between government and industry organizations. Consequently, resources are expended by each organization in repeating the development work required to address achievement of improved levels of CMM maturity.

---

*Capability Maturity Model and CMM are service marks of Carnegie Mellon University.

†No proceedings are published for this conference series. Materials from the Software Engineering Process Group Conferences are typically reproductions of handouts and short, informal papers by practitioners. Materials are available only to attendees.

°Freedom of Information Reform Act of 1986, Pub. L. No. 99-570, 100 Stat. 3207 (1986).

This analysis is confirmed by the personal experience of the authors and by discussion with other SEI personnel consulting in the area of software process improvement. Most organizations appear to invent strategies for achieving CMM Maturity Level 2 from scratch with the exception of a few who obtain consulting support for the 'action planning' that typically follows CMM-based process assessments (McFeeley 96).

It was this informal analysis, plus a reading of Moore (1991) and related experience introducing software inspections at AT&T Bell Laboratories (Ackerman 1983) that led us to develop the transition package concept. It was the popularity and influence of the software process improvement movement within the software engineering and information technology communities that led us to consider focusing our initial efforts to develop the transition package concept on a Key Process Area. Adding to the attractiveness of this strategy, one of the authors with other SEI personnel who have been working to improve technology adoption methods have already had some success working with organizations building their own transition packages for internal use.* Some workshop invitees, as well as other organizations such as those described in Strauss (1994) and Grady (1987), have used similar ideas for introducing new software engineering methods, tools, and processes into their organizations.

Transition packages are based on the concept of a 'whole product' (Moore 1991). This approach goes beyond the conventional tactic of providing consulting support to adopting projects, to providing a suite of materials. A transition package might include a documented process for a Key Process Area, plus training and 'process assets' (such as plan templates). That is, a transition package is a designed and integrated set of components enhanced by a project-level documented process of introduction (including customization guidance) and an organization-level deployment strategy. Even a conservative estimate of a 10% gain in productivity due to the reuse of artifacts and documented processes in a transition package, when multiplied across thousands of firms, leads one to conclude that the concept is well worth exploring.

We are at the beginning of our investigation of the transition package concept. Our initial goal is to understand what questions to ask as we begin to test the viability of the approach. We need to understand what our potential customers think of the idea, and help them tell us where to start. We convened our workshop to learn whether and how well the approach has worked and whether and how it

---

*As represented in Fowler, P.J., Comparetta, C., García-Martin, I. and L. Levine. (1995) Panel discussion on co-development with Xerox Document Production Systems at the first IFIP Working Group 8.6 Working Conference on the Diffusion and Adoption of Information Technology, Oslo, Norway, October 1995. Also in Allen, I., Willett, A. and de Antonio, A. (1995) Panel at the September 1995 Software Engineering Symposium in Pittsburgh, PA. Also in Allen, I., Willett, A. and de Antonio, A. (1996) Panel at the 1996 Software Engineering Process Group Conference, Atlantic City, NJ.

can be packaged for sharing with organizations generally. In designing the workshop we drew upon tools from the quality community, such as benchmarking, and derivative tools, such as the Software CMM (Paulk 1995), the Technology Transfer Model (TXM) (Fowler 1997b), and IDEAL[sm*] (McFeeley 1996) models for software process improvement. We also used the 'whole product' concept (Moore 1991), to provide a framework within which to locate materials produced by participants in their internal solutions.

This paper describes the rationale for a transition package, how we perceive the Software CMM and similar models that facilitate technology introduction and adoption, how benchmarking principles supported our approach to developing the workshop to get input from the software engineering community, what the processes and findings of that workshop were, and the outputs of participant exercises. The final section provides analysis of the findings and suggests possible next steps in the creation of the first Requirements Management transition package for software engineering community use.

2 WHY CHANGE AGENTS NEED TRANSITION PACKAGES

We expect that KPA transition packages will be 'whole products' that provide detailed guidance for the introduction of software engineering methods, tools, and processes. This type of whole product consists of a core technology (such as a Software Quality Assurance process description or Requirements Management process description) and all the components that support moving an organization from non-use to routine, everyday use. (Components include process models, guides that describe how to perform a process for a Key Process Area, training ready to customize, scenarios for consulting to project and individual customers, document templates, help facilities, example policies, and more.) Creating these components is labor intensive and may be difficult for people on software change teams such as software engineering process groups (SEPGs) and process action teams (PATs) or technical working groups (TWGs) who are charged with introducing new processes, methods, and tools to comply with the goals of CMM Key Process Areas. Members of these teams often have problem-solving and other technical and management skills, but may not have had experience applying these skills to the process of managing technology-based change in organizations.

Further, many organizations' internal change agents are moving from supporting their 'early adopters' to supporting the majority of their adopter population, who constitute 68% of the adopters (Rogers 1983). These adopters require more carefully developed tools and support. The things that are important to the early adopters – for example, hands-on involvement, the chance to develop and tailor support processes and tools, and involvement in pilot tests and implementation planning – are of less interest to members of these larger populations. Moore calls

---

*IDEAL is a service mark of Carnegie Mellon University.

these latter groups 'pragmatists' and 'conservatives' (these are Rogers' 'early majority' and 'late majority,' respectively), and notes that they are concerned with the quality and reliability of processes and tools they are expected to learn to use. Change agents in this situation need tailorable materials as well as an adaptable process description. In fact, change agents may themselves be members of the majority population, and thus only part time in that role. In our experience, this type of change agent wants his/her job spelled out in detail, much as their own adopting customer projects do. So it is likely they would need a repeatable process for performing the tailoring and introducing the process and use of attendant materials that transition packages may be able to provide (Fowler 96).

A key issue in creating transition packages may be where to obtain the components, and how to address reuse and adaptation. The workshop was intended to go directly to developers and customers of these transition package components to determine what issues and barriers they had already encountered. We also wanted to know how likely these people were to contribute artifacts they had built for either direct or tailored use.

3 THE SOFTWARE CMM AS A CONTEXT FOR FACILITATING SOFTWARE ENGINEERING TECHNOLOGY ADOPTION

In exploring methods and tools for improving software technology adoption, we seek to leverage the existing infrastructure and momentum of CMM-based software process improvement. The Software CMM is a five layer model of software management process maturity, organized to describe capabilities clustered into 'Key Process Areas.' Key Process Areas provide helpful 'chunking' of technology that can be the basis for planning software engineering technology adoption in organizations. We describe the CMM briefly, and show its roots in the quality movement, to illustrate how extensive the support for adoption of software engineering technology is in this context.

The initial stage in the CMM model is Level 1. It is characterized by ad hoc management processes, different for each project, depending on the particular people staffing that project. If the project succeeds no one knows precisely why: causes of failure are seldom investigated and corrected. In Levels 2 through 5, the CMM describes Key Process Areas that an organization must master to gain systematic control of the management of projects and to set the stage for systemic improvements in process and product quality. At Level 2, the 'Repeatable' level, projects individually master Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management. Meeting the requirements of these KPAs gives project leaders control over their products and provides senior and middle management with the information needed to manage and coordinate the different projects. Higher maturity level KPAs build further organizational capability, leading to the 'Defined' level, Level 3; the 'Managed' level, Level 4; and the 'Optimizing' level,

Level 5. In this 'stepped' model, each level builds on the level below. Details of this model are described in a range of sources (Paulk 1995, Humphrey 1989, Dymond 1995).

The origins of the maturity concept used to describe the stages of the CMM are attributed by Paulk (1995) to the prolific writer on the subject of quality, Philip Crosby (1979). Crosby's matrix describing 'The Quality Management Maturity Grid' contains five levels: uncertainty, awakening, enlightenment, wisdom, and certainty. Crosby's success improving quality at subsidiaries of International Telephone and Telegraph, largely in manufacturing industries, coupled this model of management maturity with a method for measuring and understanding the cost of poor quality, with the ethics and values necessary for a company to initiate and carry through the changes needed to deliver radically improved quality. The initial CMM applied these ideas to the professional practice of software development. Later versions of the CMM, developed through an extensive and open partnership between SEI and software development organizations* over a period of several years, converged on a consensus model that describes key practices that constitute mature software engineering.

The CMM and similar reference models such as those from the International Standards Organization (ISO) act as a map so that organizations can determine systematically what steps to take to improve. In fact, recently, the use of the CMM as both a diagnostic tool and as a guide for planning software process improvement (SPI) has been formalized by the SEI in the IDEAL model (McFeeley 1996). The five stages of this model (Initiate, Diagnose, Establish, Act, Leverage) are compatible with the four stages of the classic quality process improvement cycle: plan, do, check, act. This resonance is not accidental. Software process improvement builds upon successful approaches to process improvement in other fields.

Requirements management was selected by the SEI as the initial key process area (KPA) to explore in conjunction with developing the concept of a transition package because of its relative brevity in the Software-CMM. RM has only 3 activities, versus an average of 12 for the other Level 2 KPAs, and this means a

*The requirements for participating in the development of the CMM in partnership with the SEI were simple. One had to agree to contribute comments with explanations, review others' comments and make further comments, respect the views of all participants, etc. It was an open process – participants were self-nominated, and those who could command technical and managerial respect were listened to; others were not. The only drawback to this method is that some underfunded or small and isolated software development organizations may not have participated. The SEI continues to reach out to involve people from such organizations in the continuing work on the CMM; for example, see the SEI web site for the CMM Correspondence Group.

narrower change effort for the transition package to address.[1] In addition, the SEI wished to build on its experience collaborating on the development of the transition package concept for requirements management with two organizations. Also, requirements management is addressed early in the improvement plans for many organizations and it has the potential to improve their relationship to their customers.

4 THE ROLE OF BENCHMARKING IN THE DESIGN OF THE WORKSHOP

The people who attended our workshop are experienced enough in managing change in their organizations to know that if they don't have all the answers, they do know what many of the questions are. This group was interested in the workshop as a 'benchmarking' opportunity, even though we were not planning to be rigorous about following the classic benchmarking process described in Camp (1989). We approached benchmarking in the workshop in the sense of Spendolini's definition of benchmarking as 'learning' (Spendolini 1992). As a gathering of peers who each had something valuable to share with the others, the workshop provided a situation that encouraged interaction and integration of each of the participants' experience into the activities. Although the focus of the workshop was on applying shared experiences to the problem of defining the transition package idea and not on determining how each of the participant's processes worked for comparison to each other, we did calibrate the organizations by a number of factors. The information they were asked to bring to describe their organizations and processes* provided context that accelerated mutual learning, and will make it easier in future work to move toward direct benchmarking.

Benchmarking can be a useful technique for learning about possibilities and identifying the strengths and weaknesses of alternative processes before selecting or designing a new process or product. Benchmarking may follow a formal process and result in quantitative comparisons, or it may be less formal and be an occasion for sharing information in qualitative terms as it was in this workshop.

5 FINDING AND RECRUITING PARTICIPANTS FOR THE WORKSHOP

We sought to find participants who were early adopters and who were also interested in solving the problems later adopters encounter. To reach the

---

[1] Some research in technology adoption issues suggests that the "size" of a technology is a factor in the success of its adoption. Dorothy Leonard-Barton describes size in terms of the number of work units affected by the technology adoption (scope) versus the number of different categories of personnel affected (span). See [Leonard-Barton 1988].

*For details, see Fowler (1997).

appropriate people, we distributed a notice on the Software Process Improvement Network, an international electronic distribution list of software engineering process groups (these groups do internal technology adoption facilitation related to software process improvement based on Software CMM and ISO). Our notice asked for interest in participating in a workshop from those who had had recent experience (in the past 6 to 12 months) getting Requirements Management processes (plus tools, methods, policies, etc.) in place, in the context of the Software CMM.

Some eventual participants responded to this; others were invited because of ongoing relationships with the SEI. Each participant was screened informally by telephone against criteria that required them to be well along in the process of implementing Requirements Management practices in their organization.

Participants included twelve people from eight invited organizations ranging in size from eight to 60,000 employees, seven SEI staff members from three areas including software process and technology adoption, and two organizers/facilitators (one from the SEI, one external). Organizations were located in Europe, the United Kingdom, and throughout the United States, and represented both public and private sectors. Participants included relative newcomers to requirements management introduction and also very experienced corporate staff with experience introducing requirements management organization wide. One vendor organization building software process improvement tools also participated.

Participants were invited to present their work, beginning with providing context that included the type of organization, type of products or systems being built, product longevity, time between major product introductions or revisions, size of organizations (including numbers of managers and practitioners), number of management levels responsible for development, size of applications supported (in lines of code, total head count, function points or other measure), size and characteristics of user community, source of requirements (for example, marketing staff, contract, internal customer), and number of requirements supported.

Participants were asked to emphasize in their presentations how they introduced requirements management practices, methods and tools. Suggestions for topics to address in this area included: description of 'before' and 'after' requirements management introduction, steps for introducing requirements management, use of templates and examples, description of process model for enacting requirements management, criteria for selecting subject matter experts and vendors, tool selection criteria, and description of training materials and strategies.

Participants were also invited to bring and display examples of materials used in their introduction process. Materials requested and brought for display included meeting minutes (showing how membership in working groups or action teams changed over time as well the topics that were considered), reports on expenditures of effort and funds, example policies, software tool selection criteria, software tool

use guidance, on line demonstrations of tools and process asset libraries, process descriptions, and videos and other educational materials.

6 OPERATION AND RESULTS OF THE WORKSHOP

On the first day each of the eight organizations participating presented their approach to introducing requirements management. Both presentations and organizations varied widely, but there were common issues and elements that became very clear in exercises later in the workshop (Fowler 97a).

At the start of the second day participants developed a definition for best practice:

> A best practice provides complete, feasible, and appropriate guidelines for executing an activity; it also provides a common procedure that improves its performance efficiently and effectively.

Best practice criteria for requirements introduction were then proposed, and included these (summarized):

- based on customer requirements
- appropriate for organization's target audience's maturity/context
- a well defined, formalized practice that can be easily applied across multiple organizations. or projects
- CMM compliant
- tailorable
- measurable
- effective/efficient
- includes templates and examples
- trainable
- user friendly
- proven effective in multiple applications or contexts.

Several participants next presented materials they brought to exhibit and described how these materials fit the 'best practice' criteria. This proved to be a very absorbing discussion about potential generalization of the tools for use by others. This exercise helped focus the information presented on the first day and gave people a means to discuss more clearly the possibility of sharing their artifacts, or reusing or adapting others'.

Participants then attempted to make explicit their assumptions concerning requirements management introduction. These assumptions, summarized, included:

- It is valuable and useful to have a requirements management transition package.
- It is possible to define the requirements for a requirements management transition package.

- All the main problems encountered by software developers can be identified.
- There is customer need for the transition package.
- Requirements management is difficult, confusing, time and labor consuming.
- A process for introduction will save time and money.
- The organization is ready and has the need for requirements management introduction.
- There are people who are responsible for managing requirements.
- The resources to support technology introduction exist.
- A process for introduction can be tailored and adjusted for business situations.
- Tailoring is required.
- There are training hours and training development hours committed to requirements management introduction.
- There is management support.
- People at different organizational levels see things differently.
- There is an SEPG or similar group that is a champion and oversees the implementation.
- Subject matter experts are present.
- The introducers understand requirements management or can easily obtain training and coaching.
- There is access to a customer (or surrogate) base.
- Templates and examples are useful.
- There is access to technology and tools.

To determine what else might be needed, and the sequence of use of items identified as potential components of a requirements management transition package, we combined elements of brainstorming with elements of affinity diagramming to compile a list of possible artifacts and to order the list, based on a generic development life cycle. Participants nominated 136 artifacts in sixteen groups across eight life cycle categories.

Participants were comfortable using the terms 'transition package' and 'starter kit.' On the third day the phrase 'blue box,' in analogy to 'shrink-wrapped' commercial software packages, was coined to describe what is given to project level change agents and became a popular nickname for a transition package.

At the start of the third day participants identified the following users and customers of a requirements management transition package.

- corporate SEPG
- SEPG leader
- SEPG members
- oversight bodies
- PAT/TWG

- training group
- software managers
- lead system engineer
- system engineers
- sponsor
- internal financial controller
- functional manager of software department
- chef de service (chief technical officer)
- project manager
- customer of software project
- customer – user
- customer 'buyer' – bill payer
- customer – product manager/systems engineers
- manager of requirements management database
- software engineer
- software quality assurance function
- configuration management manager
- software architecture function
- test team
- proposal manager
- source selection board
- subject matter experts
- external consultants
- vendors.

In a session on how to store, document, and make artifacts available as part of the transition package, participants considered issues, costs, technology, and differences between small and large organization strategies. This led to discussion of the customers (that is, what organizations would benefit) for transition packages. This list resulted:

- people in organizations who recommend (probably the SEPG)
- people in organizations with the money
- small company with limited infrastructure – the controller may be the person to recommend
- large companies that want licensable materials, where the corporate SPI sponsor would be a chief stakeholder
- software process improvement budget holders
- external consultants
- executive managers

- Department of Defense Software Development Centers (LCSAs/CDAD)
- training companies (or) departments within companies
- franchise holder
- universities
- standard-setting organizations
- SEI-like organizations
- organizations with only a few deployments needing little or no tailoring

The final activity in the workshop was the identification of viable next steps in developing a requirements management transition package. These are discussed in the next section.

7 AFTER THE WORKSHOP: NEXT STEPS

The result of the final – next steps – exercise in the workshop yielded insight into problems we will encounter and countermeasures we may employ in developing a Requirements Management transition package. We grouped the ideas and advice into six 'clusters':

First cluster

What are the real user needs? Who are the users? Were the workshop participants typical, or a special case? Participants offered suggestions for how to find out, such as a more detailed analysis, in the form of a model of the current process, of how people currently plan the introduction of Requirements Management, and how others adopt it.

Second cluster

Comments in this cluster emphasized the value of examples, and how to 'sanitize' examples so that organizations would donate them, and how to organize them once obtained. Participants suggested that all examples carry a warning to users: the risk is that some users won't be careful as to how they adapt examples, and some contributors may not be able to describe context in an adequately meaningful way.

In addition, comments echoed the results of the exercise where participants described the artifacts they brought with them in terms of the best practice criteria created earlier in the workshop. In one case, a manual for using a software tool was most interesting and useful because of its structure rather than its content. The attendant discussion considered how this type of information could be shared. Most participants regretted not spending more time on this exercise.

Third cluster

Strategies for the package, including proposals for transition packages for areas other than requirements management, were addressed in this group of comments, which mixed ideas about building the package with ideas about designing it. During the workshop, informally, the requirements management transition package got dubbed the 'blue box', with the notion envisioned that a shrink-wrapped package might arrive on a user's desk. Considerable discussion ensued

once this image was presented: would the package arrive alone, with a consultant, with a hot line, accompanied by training, etc.? Would it contain an integrated solution, or a set of components the user could configure as desired? What skills would the user need? How would the user need to adapt it to be successful? And what is the minimum set of materials and components needed? Most participants seemed to prefer the requirements management-specific focus for the package.

Fourth cluster

Some participants considered and suggested the ways and means of working together to create a requirements management transition package. Should it be built by the community, as with the SEI Systems Engineering Capability Maturity Model? Should it be built by a small working group? What role should the SEI play? What are the risks of a working group building the transition package, versus a single organization? The lack of consensus on a framework within which to construct the package was clear as participants struggled to agree on a way of framing the dozens of artifacts and activities identified as potentially part of a transition package, and a suggestion was made that smaller groups, each organized around a single model, should work together at a sequel workshop to address this.

Fifth cluster

This cluster is quite simple and pragmatic in its suggestion, strongly stated, that any requirements management transition package built be sharply scoped. There was a counterpoint of suggestions to be sure to set requirements management in the context of systems engineering and requirements engineering. A theme here seems to be that scope would constrain collaborators, as might be expected; some might prefer a narrower scope, others a broader one.

Sixth cluster

Befitting a group of savvy change agents, participants made a lively series of suggestions on how to position a requirements management transition package to engage collaborators and funders, including how to make selection criteria clear up front. The fact that only one member of the vendor community for software process improvement tools and services was represented at the workshop was noted, with the suggestion that discussions be held with vendor organizations as potential collaborators. The need for an almost immediate follow-on workshop was argued, with the suggestion that the next one might be held in conjunction with the SEPG 1997 conference.

8 CONCLUSION

As we anticipated in planning the workshop, convening a group of participants all operating in the context of the Software CMM and software process improvement expedited communication and work within the group. The benchmarking flavor of the workshop supported lively interchange. Small groups of participants were actively comparing notes by late in the first day, and this continued to the end. The 'whole product' concept, sketched briefly at the start of the workshop, helped

anchor discussions about what a requirements management transition package might be, who would use it, and how difficult it might be to build a good one. The exhibits were popular, although participants complained that they lacked both structured and unstructured time for examining these. The written feedback from participants solicited on a form at the end of the workshop indicated that participants were glad that they attended. Most reiterated what they had stated in the more public forum of the closing session on 'next steps.'

So, our wish to combine the best efforts of competent change agents to begin to produce a strong set of requirements for a requirements management transition package was fulfilled. The data harvested from the workshop is raw material for planning requirements management transition package development. Not surprisingly, the power of the group came through in the extent and sophistication of the detail in the issues they raised. Because there was general agreement that those who need the package most were probably not present, the concurrence of 18 experienced change agents that 'packaging' technology adoption is possible, and worth pursuing, has to be considered with caution. Nonetheless, not one participant said not to build one, and most anticipated further work on the requirements management transition package with relish.

9 REFERENCES

Paulk, M.C., Weber, C.V., Curtis, B. and Chrissis, M.B. (1995) *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley Publishing Company, Reading, Massachusetts.

SEI Software Engineering Measurement Team (1996) November 1996 bi-annual report, Process Maturity Profile of the Software Community, 1996 Update, in http://www.sei.cmu.edu.

Moore, G. (1991) *Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers*. Harper Business, New York.

McFeeley, R. (1996) IDEAL: A User's Guide for Software Process Improvement. CMU/SEI-96-HB-001. Software Engineering Institute, Pittsburgh, Pennsylvania.

Ackerman, A.F., Fowler, P.J. and Ebenau, R.G. (1983) Software inspections and the industrial production of software. In Hausen, H.-L. (ed.), *Proceedings, Symposium on Software Validation*, North-Holland, Amsterdam.

Strauss, S. and Ebenau, R. (1994) *Software Inspection Process*. McGraw-Hill, New York.

Grady, R. and Caswell, D. (1987) *Software Metrics: Establishing a Company-Wide Program*. Prentice-Hall, Englewood Cliffs, NJ.

Rogers, E. (1983) *Diffusion of Innovations*. The Free Press, New York.

Fowler, P. and Kasunic, M. (1996) Introducing KPAs: A Process Definition and Guide. In *Proceedings, Eighth Annual Software Technology Conference*. Software Technology Support Center, Hill Air Force Base, Utah.

Humphrey, W. (1989) *Managing the Software Process*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Dymond, K.M. (1995) *A Guide to the CMM*. Process, Inc., Annapolis, Maryland.

Crosby, P.B. (1979) *Quality Is Free: The Art of Making Quality Certain*. McGraw-Hill Book Company, New York.

Fowler, P.J. and Patrick, M. (1997a) *Proceedings of the Introducing Requirements Management Into Organizations Workshop: Requirements Management Transition Packages*. Special Report CMU/SEI-97-SR-001. Software Engineering Institute, Pittsburgh, Pennsylvania. To appear.

Fowler, P.J. SEI (1997b) *Introducing New Software Technology: A Workshop*. Course. Software Engineering Institute, Pittsburgh, Pennsylvania.

Camp, R.C. (1989) *Benchmarking: The Search for Industry Best Practices that Lead to Superior Performance* . Quality Press, Milwaukee, Wisconsin.

Spendolini, M.J. (1992) *The Benchmarking Book*. AMACOM, New York.

10 BIOGRAPHIES

Priscilla Fowler works on the translation of technology adoption and implementation research and practice into methods and tools for software change agents, such as in her current work on a prototype requirements management transition package. She is co-proposer, with Linda Levine, and founding Chair, of the International Federation for Information Processing Working Group 8.6 on Diffusion, Transfer and Implementation of Information Technology. She initiated and directed the AT&T Bell Labs Software Engineering Technology Transfer Program. Ms. Fowler has been a software engineer, project manager, and consultant at AT&T Bell Labs, IBM, and Knight-Ridder. She holds a BA degree from Drew University and is a member of the ACM.

Mac Patrick is a Visiting Scientist at the SEI in the Transition Enabling program area working on the transition package concept. Mr. Patrick also supports process benchmarking and improvement in software and general business organizations, via his own company, Process Advantage Technology. Prior to starting his own firm, Mr. Patrick helped originate the Pacific Bell software engineering process group. Mr. Patrick is a charter member and past president of the Bay Area Quality Assurance Association, and is a member of the ACM and IEEE. Mr. Patrick served in the US Air Force, and holds a BA degree from Saint Mary's College.