SEI-based Process Improvement Efforts at Digital

by

Neil L. M. Davies and Margaret M. Dumont

ABSTRACT

The Software Engineering Institute is chartered with advancing the state-of-the-practice of software engineering to improve the quality of the systems that depend on software. Digital has based its software process improvement program on the Capability Maturity Model and Software Process Assessment developed by the SEI. As software organizations gain process maturity, they produce higher-quality products. Case studies report the experiences and learnings of two software organizations at Digital that have introduced the SEI framework and methods into their process improvement efforts.

INTRODUCTION

During the late seventies and early eighties, the state-of-the-practice of software development and management at Digital improved significantly. Examples of these improvements include the following.

- Software and hardware architectures, notably the VAX VMS and the Digital Network Architectures, were developed.
- Higher-level languages (BLISS and C) were introduced into common use in systems development.
- Debuggers and language-sensitive editors were developed and used widely.
- Code management systems were introduced into widespread use.
- The phase review process for managing software projects was used extensively.

Although the complexity of software development projects has grown exponentially over the last few years, relatively few changes have occurred in the practice of developing and managing software projects. The lack of effective process management techniques impacted Digital's ability to predictably deliver quality software products that satisfy customers' expectations both in feature and time-to-market needs.

This paper describes the use of software process methods to improve the quality and predictability both in time and function of Digital's software products. Specifically, it

describes the approaches of two organizations actively involved in software process improvement efforts. In addition, it presents the conclusions drawn from case studies of their process improvement programs as well the challenges to be faced in the future.

SOFTWARE PROCESS IMPROVEMENT PROGRAM

The software process improvement program at Digital is based on the framework developed by the Software Engineering Institute (SEI). The SEI is a federally funded organization chartered with advancing the state-of-the-practice of software engineering to improve the quality of the systems that depend on software.

The SEI promotes the belief that software productivity and quality gains can be achieved through a focused and sustained effort toward building a process infrastructure of effective software engineering and management practices.[1] Case studies on process programs at Hughes Aircraft and Raytheon support this premise.[2,3] Although the importance of a quality process to the end quality of the product is gaining acceptance, this idea is not prevalent within software organizations. A strong fear still exists that development of a process structure is equivalent to the creation of a bureaucracy.

We chose the SEI's framework as the basis for our process improvement efforts because its focus is specific to software organizations. A key element of improving software process is the ability to develop effective structures and the discipline to manage the process. The SEI has developed a process framework and method that deal specifically with the complexity of software practices and organizations.

SEI Capability Maturity Model

The framework, known as the Capability Maturity Model (CMM), asserts that a project is an instantiation of the organizational processes in which it was developed. Therefore, to improve a project's predictability or quality, one must improve the structure and discipline of the process (or develop the process maturity) in which the project is developed. The capability of a process to deliver a quality product predictably is determined by how well the process is defined and how consistently it is applied.

As shown in Figure 1, the CMM framework defines five levels of maturity: Initial, Repeatable, Defined, Managed, and Optimizing. Each level is a building block for the next level. To see improvements, organizations must proceed from the lowest level to the highest level. Since each level is a precondition for the next, the organization cannot skip a level. Organizations can determine their process maturity and the processes they should

develop by undergoing an SEI process assessment.

SEI Process Assessment

The SEI has developed a method called the SEI process assessment to enable organizations to determine their process maturity. The assessment is used to determine process awareness in the organization and to devise an action plan for improvement of the process. The assessment involves all levels of the organization in a structured method aimed at building consensus on the primary problems the organization faces. A by-product of a well-run assessment is organizational agreement on the actions of how to address the problems. For more information on the process maturity framework and assessment, see Managing the Software Process by Humphrey.[4]

SEI Guidelines for Process Improvement

Once the organization decides to introduce a process improvement program based on the SEI model and method, two questions require answers: (1) What does this mean? and (2) How do we get started? Process improvement work is unique and involves a level of abstraction beyond the usual work done in software organizations. This effort must be staffed with individuals who can blend organization knowledge with process improvement techniques. Unless the organization is serious about applying adequate resources to the effort, including a substantial amount of time and commitment from management, we suggest that the effort not be undertaken. The SEI has developed guidelines on staffing a Software Engineering Process Group (SEPG).[5]

In the next two sections, we offer our different experiences in implementing SEI-based process improvement programs as case studies from which other organizations can learn. In the first case study, an organization started with a small bounded improvement and used that to launch a process improvement effort that started with an SEI assessment. In the second case study, an organization built SEI concepts into existing quality processes to gain momentum for a process improvement program based on the SEI framework and SEI assessment.

CASE STUDY 1: USING AN SEI ASSESSMENT TO INITIATE THE PROCESS IMPROVEMENT PROGRAM

Undertaking an SEI-based process improvement effort is a huge task. The effort officially begins with an SEI assessment; however, we have found that months or years may be needed to prepare for an assessment. In our case, nine months passed from the time we began work to improve our processes until we considered an SEI assessment. Another four months was needed to complete the assessment. As our first step, we sought commitment

for change within the organization. To this end, we initiated a test involving a small bounded improvement plan.

Obtaining Commitment for Change

Often there is a perception in the organization that it is easy to change. In our experience, however, it is a difficult process even when an organization wants to change. To prepare for the larger process improvement effort, we devised a small bounded improvement effort to evaluate if the organization was ready to change. The test is beneficial in two ways. First, it gives the organization experience in dealing with change. Second, it creates energy for process improvement and helps to enlist sponsors within the organization.

The first improvement was to update the code management system. The organization had recently undergone changes in organizational structure and product strategy. These changes put new requirements on the system we used to build and integrate our sources. The improvement was to choose a new source management system and to establish its use in the development and release processes within one product release.

The success of our improvement plan was measured in two ways. First, the introduction of the code management system did not impact the schedule of the release in which it was introduced. Second, during the retrospective of the release, the new code management system was viewed positively by both the release management and engineering organizations. In addition, 30 percent of the people involved in the retrospective responded that updating the code management system was the highest positive change we made to the process. As a result of this success, we proceeded to the SEI assessment and SEI-based process improvement program.

Choice of SEI Model and Method

We chose to use the CMM and SEI assessment as part of an overall effort to improve the software development environment in our organization for two major reasons.

First, the CMM provided a framework for prioritizing process improvement efforts to develop the organization's capabilities. In the months prior to adopting the CMM, we tried unsuccessfully to agree on the priority of improvement in the organization. In time, we reached the point where we agreed that use of the CMM and SEI assessment would enable us to establish priorities for improvements. The major benefit we saw was that the assessment involved all levels of the organization from senior managers to individual contributors in the prioritization and implementation of changes. In addition, we considered the cross-functional involvement to be essential to sustaining the effort.

The second major reason we chose the CMM was its focus on the software industry. In the future, we hope to be able to benefit from the programs in risk management, software education, and software measures, now being developed at the SEI.

The assessment is designed to help determine the process areas that the organization must address in order to move up the capability levels of the CMM. In our case, the assessment was led by a trained SEI facilitator and a team of people within the engineering organization. We tapped the knowledge of approximately 60 people from within the organization through questionnaires, interviews, and free-form meetings. The data collected was analyzed and developed into a findings and recommendations document that was presented to senior management. This document is the basis for process improvement work in the organization. It is required reading for new managers at the staff level.

Extensions to the Framework of the CMM. The CMM has its roots in the government systems and defense-oriented areas of the software industry. It has only recently made inroads into the commercial software industry. Although it is the most complete method available for software process improvement, it makes certain assumptions about software development organizations that may not be true in the commercial sector. While implementing our software process improvement project, we found it necessary to extend the CMM.

As stated earlier, the CMM provides a set of levels that allow an organization to determine the maturity of its processes. Each level defines a set of key process areas (KPAs) required to reach that level's capability. For example, there are six KPAs at the Repeatable Level 2:

- Subcontractor management
- Software project planning
- Software project tracking and oversight
- Software configuration management
- Software quality assurance
- Requirements management

Each KPA is defined by a set of practices that cover the goals, the abilities and commitments to perform the process, the activities the organization must perform, and the mechanisms to measure and verify those activities.

The first extension we made to the CMM occurred during the

assessment process. The CMM does not address resource management and development, that is, employee development, changes in the way resources are applied to new processes, and communication within the organization. These are necessary to develop the practices required to implement a KPA. For example, to develop a project plan, one must be able to negotiate effectively to share resources among interdependent projects; or, to verify that an activity is performed, feedback loops must exist in the organization's communication processes.

Our findings indicated that the areas of commitment and communication needed improvement. The CMM describes attributes for these areas in each KPA; however, it provides no guidance on the goals, activities, and abilities of commitment and communication as process areas in their own right. We have some activity in each of these areas but have not successfully developed them into an integrated plan for the organization.

The next extension to the CMM required us to implement processes from the Defined Level 3, even though we had not achieved the Repeatable Level 2. First, we needed to establish an SEPG to carry out the activities to improve the process. Second, we needed to establish guidelines and methods for a training program. Without a training program, we could not ensure that the organization would have the abilities to perform KPAs at the Repeatable Level 2. Third, we needed to define the processes used in the organization. Definition of process and training are perceived by the organization as major causes of frustration. These areas tend to embody the organization's recognized need to change and its overall resistance to change. These two areas involve problems related to understanding how other functions in the group work, developing good peer-to-peer communications, and transferring responsibilities between people.

Finally, we introduced a KPA for the definition of the software development process. The CMM is based on first providing a good management framework and then developing the engineering framework. The assumption is that, as engineers, we tend to focus first on the engineering process for improvements. In implementing process improvement, we found that we needed a process model specifically for development of software components within our overall software product process.

Turning Recommendations into Actions. Our experience has shown that with organizations assessed at the Initial Level 1 of maturity, two aspects of turning recommendations into actions need to be considered. The first is the skill set of the people who develop the process improvements; the second is the framework for developing and delivering process improvements to the organization. We found that the individuals and teams who deliver process improvement must possess project management skills and organizational development skills.

Project management skills are essential because the environment does not otherwise foster the discipline or ability to create a set of plans from a set of recommendations. We structured the process improvement work into a project with a set of goals, objectives, and deliverables. The high-level goals and objectives were integrated into a set of long-range milestones. Currently, each person working on process improvement has a set of project plans that describe individual deliverables based on the project goals. The next step for the project is to attain the same level of detail in all the plans so that we can integrate the work as a single set of deliverables into the organization. Our recommendation to anyone starting a process improvement effort is to staff the effort with a strong emphasis on project management skills.

Organizational development skills are also essential. The process improvement team needs to assess the organization to determine the root cause of problems, to determine the rate of change for the process improvement efforts, and to institute feedback mechanisms to measure progress. In addition, the team needs to understand how to overcome resistance to change, to deal with change at all levels of the organization, and to sustain change at a manageable rate.

Our experience has convinced us that a framework is essential to develop and deliver process improvement to the organization. Our process improvement framework has three aspects:

- Skills development
- Process definition and improvement
- Operational environment and technology enhancements

For example, we had been working in the area of improving the organization's planning processes. After evaluating the existing planning processes, we determined that we would have to develop the organization's planning skills. First, we introduced a tool to enable people to implement schedules. Second, we developed requirements for the operational environment for the tool and process, specifically for access, archival, and retrieval of project-related information such as project plans and schedules. Third, we determined the requirements for training based on the needs of key individuals in the organization. Finally, we defined the organization's planning process and developed continuous improvement cycles for the process.

Each of our process improvement efforts included the three factors from our project framework. These efforts were tracked by the organization to ensure that the schedule and resource needs of the work were met. In addition, process improvement work was prioritized according to the organization's business needs. The delivery methods for the process improvement work must be agreed upon and understood at all levels of the organization. This

provides the context and enables the work to be better understood in the day-to-day routines of the organization.

CASE STUDY 2: BUILDING SUPPORT FOR A FORMAL SEI-BASED PROCESS IMPROVEMENT PROGRAM INTO ONGOING PROJECTS

Initially, the amount of engineering time needed for a formal SEI-based process improvement program was intimidating to management and engineers. To demonstrate that the process could benefit the organization, we took several introductory actions. First, since the organization was already committed to project retrospectives, we introduced the basic SEI concepts into the existing retrospective process. Second, we worked with engineering management to ensure that formal quality planning was undertaken at the start of each project so that quality goals and processes were consciously selected. Third, we designed a metrics program to support our quest for maturity.

Project Retrospective

We developed a retrospective process based upon the principles in the SEI model for process improvement and applied it to our most recent product release. We wanted to ensure that we covered all the key elements in the SEI model (sponsorship, organizational preparedness, employee involvement, working first on KPAs at the Repeatable Level 2). As shown in Figure 2, the process was designed by the forerunner of the SEPG.

First, the SEPG met with the sponsor (the head of the engineering organization) to define the particular attributes of the SEI process we wished to integrate into our retrospectives. They included clear sponsorship, employee involvement in all aspects of the process, and creation of action teams to make improvements. The sponsor communicated to her organization the goals of the enhanced retrospective and her commitment to act on any findings.

Next, we designed and distributed a survey aimed at obtaining a broad view of what worked or did not work on the most recent large release. The retrospective team was assembled and conducted a facilitated meeting of the larger group to obtain an alternate view of what had happened during the project. The team used the findings from this meeting and the survey to develop a prioritized list of problems.

The following problems were identified as being applicable to both hardware and software.

- Design continued during debugging.
- Component quality ranged from faultless to untested.

- Check-in criteria were inconsistent.
- Check-in criteria were unclear and changed as the project progressed.

Team members discussed the problems in a series of structured interviews with the key people concerned with the release. The interviews focused on identifying the root causes of the problems. Sample root causes are listed below.

- Different assumptions were made about code freeze.
- Changes to check-in criteria were not communicated.
- Hardware was not available for tests early in the project; builds and tests were time consuming.
- Consistent success or failure was not rewarded or fixed.
- Known problems were allowed to continue.

The team then distilled these root causes into a set of findings that were fed back to the originators for confirmation and then to the sponsor for action. The findings from the retrospective team were the following.

- We planned only one release at a time.
- The overall testing model was unclear.
- Check-in procedures were unclear.

The final list of findings can be mapped to the Initial Level 1 of the CMM. The latter two issues relate to software quality assurance (SQA), and the first issue relates to the requirements definition.

The enhanced retrospective boosted our process improvement program. It showed that management needed to sponsor the project, that employee involvement facilitated the improvement plans, and that an SEPG was required to handle the results. In addition, the enhanced retrospective produced better results than a traditional retrospective. We recommend this process to other groups conducting process improvement programs.

Serendipitously, our retrospective was led by the manager of the next release. As we discussed the project's problems, he was heard to say, "We are doing the same thing in my release; I'd better talk to ...." We could not have asked for faster implementation! Furthermore, we changed our process to recommend that the manager of the next release participate in all retrospectives. We also believe that too much intuition was at work during the retrospective. At our next retrospective, we will closely compare the problem list with the key practices for our

CMM level before we produce a list of findings.

Quality Planning

Often the action plans from SEI, from other process improvement task forces, or from total quality control (TQC) teams are not carried forward to day-to-day project activities. A new technique is invented and prototyped by the action team and then turned over to the SEPG for widespread implementation. At this point, the process improvement usually ends. In other cases, a small group improvement activity may create an improved engineering process, but its success is unknown outside the immediate team.

Ideally, quality planning selects the processes to be used at the start of each project. Quality (process) plans close the gap between improved processes and project activities. We have asked each subsequent team to prepare a quality plan. The process for institutionalizing practices works well at our current CMM level. After we complete our first full SEI assessment and improvement cycle, we should see the necessity of these activities to achieve process maturity. The best quality plans are fully embedded in the release or project plan prepared by each team. We do not require a separate quality plan for each release, merely that the following questions are answered for each new release:

- What attributes of quality are important for this release?
- How will those quality goals be measured before and after the release?
- What are the goals for the product before and after the release?
- What processes will be put in place to ensure that the goals are met?
- What are the expectations for each component in a release and at what milestone?

For example, if the release is to have 10 percent fewer defects than the last release, then the questions above might be answered as follows. The defect reports from customers are important. The goals might be to have 10 percent fewer defect reports per 100 customers, to increase pre-release test coverage by 10 percent, and to continue testing until a rate of less than 1 defect per 1,000 hours of testing is achieved.

To ensure that the goals are met, formal code inspections for 100 percent of all new code would be introduced and regression testing coverage increased by 15 percent. All components would be required to meet this standard 2 weeks before integration.

Our early experiences with quality plans have confirmed our need for a more mature software engineering process. We have seen a tendency to "abandon quality to the quality person"; alternately, some plans have been rejected as "trying to tell engineering how to do its job." It is difficult to separate the testing plans from the quality plan. As a result, the early quality plans have focused on release criteria and have included large sections of background information justifying their very existence.

In the long term, we believe that the quality plan should cease to exist as a separate document and should be included in the overall project plan. In the future, quality plans will be created from known good practices in engineering. As we climb the maturity ladder, we will more and more use a repository of good practice as the basis for creating these plans. An SEPG will be chartered with maintaining the repository (or life cycle as we know it). The life cycle will be updated based upon SEI assessments, retrospectives, small group improvement activities, and so on.

The SEPG is aimed at long-term process improvement across multiple projects. The quality plan is the document to connect these general process improvements to day-to-day project work. Every project or release now has a person designated as responsible for quality. This person is responsible for liaison with the SEPG and bringing the best practices into the teams.

The Software Metrics Program

As shown in Figure 3, full benefit from metrics is experienced only when the processes are under real control, as at the CMM Managed Level 4 or above. In addition, measured SQA is one of the major criteria for attaining the Repeatable Level 2. Therefore we created a metrics program with a dual thrust: we instituted project- and release-related metrics of doneness, or SQA. We also created a metrics program throughout the organization to measure and track our long-term intent for process improvement. These process metrics are not pure because the underlying processes are not under rigorous statistical control; however, they provide a point of focus for the organization's improvement efforts. Our early efforts showed that the organization did not think in terms of processes whose yield can and should be measured over time. We need to start these metrics today so that we will have an effective collection system when we reach the Managed Level 4, and we will also have a population familiar with process management.

Organization-wide Metrics. We have tried to ensure that our metrics provide a business focus for our improvement activities throughout the organization. We have also tried to present the metrics in such a way as to promote continuous process improvement. We have metrics for product reliability,

performance, predictability of schedule, i.e., estimating quality factor (EQF), responsiveness to customers, and cost effectiveness. Each of the metrics is displayed in a format that embodies the Shewhart/Deming cycle (plan, do, check, act) as shown in Figure 4. In future quality planning sessions, we will review each plan for its impact on these metrics. The SEPG is responsible for preparing and analyzing these metrics.

SQA Metrics. Our SQA metrics are relatively simple and are based upon a convergence during a series of checkpoints at the end of our testing cycles. We are measuring test coverage, time under stress without failure, incident arrival rates, and unresolved incidents in the classic way. These measurements ensure that the product has been tested enough to ship. We are now starting to measure early quality indicators such as design stability, which predicts eventual SQA problems. The SEPG is defining improved metrics and is analyzing the effectiveness of our test programs. Day-to-day project decisions as to whether or not to ship are the responsibility of the project teams.

CONCLUSIONS DRAWN FROM BOTH CASE STUDIES

We have drawn two conclusions based on our experiences using the SEI framework. Both conclusions apply whether the organization begins its process improvement efforts with an SEI assessment or uses the SEI framework in support of existing quality activities. First, involving people in the change process is important. At the Initial Level of the CMM, organizations are characterized by ad hoc processes. The processes are not described or enforced, and there is a high dependence on heroic efforts to meet schedules. At the Initial Level of maturity, people are the process. Lack of focus on the importance of people in improving the process causes confusion and chaos in the organization. Examples include:

- A process is not adopted or becomes a "jump through the hoop" exercise when people are unsure of how the change benefits their goals.
- Confusion and conflicts arise when the people involved in carrying out the process are not included in making changes to the process.

By involving people in the change process, we have found that new processes are adopted more quickly and are better suited to the work that people perform. In fact, the introduction of new processes becomes transparent to the organization.

Second, the use of the alternate method bolsters the primary process improvement method. For example, when we started with an SEI assessment in the first case study, we found that incorporating the SEI framework into our product retrospectives

raised the group's awareness of the SEI methodology. The SEI framework continued to reassert the importance of process improvement within the organization. In the second case study, we incorporated the SEI framework into ongoing activities. We concluded that, for future process improvement efforts, an SEI assessment would align the organization behind a single common vision and set of priorities.

CURRENT STATE AND FUTURE CHALLENGES

In this section we describe our current state and some of our next challenges in implementing the SEI-based process improvement programs.

Case Study 1 --- Formal SEI-based Process Improvement Program

As previously described, the process improvement program provided the assessment, an action team was formed, and we introduced improvements based on its recommendations. Our major learning from this program is that actual process change is risky to introduce in spite of strong organizational commitment and difficult to keep on track because factors that interact with the organization are changing. The change in business goals and restructuring within the organization had the highest impact on our process improvement efforts.

In implementing our process improvement efforts, we found that it was important to tie the improvements in our product process to the business goals of the organization. When the business goals changed, we were required to realign our priorities to meet those changes. For example, we set a business goal to meet the first revenue ship date for key hardware products. This required us to move from a sequential product release model to a concurrent release model, where we might have the development of several releases occurring in parallel, e.g., one or more functional releases and one or more hardware releases. This placed new requirements on our processes; as a result, we had to shift the priorities within the process improvement efforts.

Of the two changes, restructuring the organization had a greater impact for us. As a Level 1 organization, we had the practice of overreliance on a small number of people with special skills to perform critical functions. They understood and supported the process improvement work. The restructure resulted in these people leaving the organization or changing positions. Since many of the key sponsors for the process improvement work left the group, we had to rebuild support and sponsorship within the new management and organization structure. This had an impact on both the priority and the methods to deliver the process improvement work.

The basic problem in both changes was that we had no way to

transfer knowledge or skill sets during changes. We expect that the system in which we work will continually change and shift. Our major future challenge is to develop process improvements and support for these improvements that transcend changes to the system in which the organization exists. We intend to continue to bolster our SEI activities with the addition of metrics and quality planning to ongoing organization activities.

Case Study 2 --- Adding SEI to an Existing Process Improvement Program

Currently, the organization is focused on delivering two key products and on developing a new organizational structure. As a result, it has been difficult to maintain progress on major process improvements.

The retrospective process is now in use on all major releases of our products with positive results. The first action plans from the retrospectives took a long time to complete and are only being implemented today (August 1993). Metrics and quality plans are now in use by 100 percent of our releases.

We could have made faster progress throughout the improvement program if we had better fundamental knowledge about quality and process in our organization. The additional learning from retrospectives could have been more effective if we also had a broadly based education program in quality.

The retrospectives have produced real benefit and some goodwill toward process improvement. In addition, they have acted as an excellent way of educating their participants about the fundamentals of process management. We recently held the first meeting for the formal SEI program; both attendance and enthusiasm were high. The prototyping work with the retrospectives, however, has not overcome the concerns of the organization. For example, concern remains that an SEPG will take ownership of the process away from the engineering groups despite repeated assurance that it will not. The full benefits of quality planning and the metrics program and their connection to our breakthrough productivity objectives remain to be achieved.

We believe that the visible commitment for an SEI assessment is needed to galvanize the organization to achieve breakthrough levels of process improvement and higher benefits, and we are continuing with our formal SEI program. The initial organization-wide training is scheduled for the first week of September 1993, and the assessment is tentatively scheduled for April 1994.

ACKNOWLEDGMENTS

Neil thanks his team partners Bryan Jones, Brian Porter, Tom

Saleme, Nick Craig, and most especially his sponsor Laura Woodburn.

Both authors acknowledge Barbara Kelczewski, who helped edit this paper. She turned a random collection of thoughts and experiences into a format for communication.

REFERENCES

1. M. Paulk, B. Curtis, M. Chrissis, and C. Weber, Capability Maturity Model for Software V1.1 (Pittsburgh, PA: Carnegie-Mellon University, Software Engineering Institute, Technical Report, CMU/SEI-93-TR-24 ESC-TR-93-177, February 1993).

2. W. Humphrey, T. Snyder, and R. Willis, "Software Process Improvement at Hughes Aircraft," IEEE Software (July 1991).

3. R. Dion, "Process Improvement and the Corporate Balance Sheet," IEEE Software (July 1993).

4. W. Humphrey, Managing the Software Process (Reading, MA: Addison-Wesley Publishing Company, 1989/1990).

5. P. Fowler and S. Rifken, Software Engineering Process Group Guide (Pittsburgh, PA: Carnegie-Mellon University, Software Engineering Institute, Technical Report, CMU/SEI-90-TR-24 ESD-90-TR225, September 1990).

GENERAL REFERENCES

R. Ackoff, Creating the Corporate Future: Plan or Be Planned For (New York: Wiley, 1981).

T. DeMarco, Controlling Software Projects (New York: Yourdon Press, 1982).

A. Duncan and T. Harris, "Software Productivity Measurements," Digital Technical Journal, vol. 1, no. 6 (February 1988): 20-27.

D. Kauffman, Jr., Systems One: An Introduction to Systems Thinking (Future Systems, Inc., 1980).

M. Paulk, C. Weber, S. Garcia, M. Chrissis, and M. Bush, "Key Practices of the Capability Maturity Model V1.1" (Pittsburgh, PA: Carnegie-Mellon University, Software Engineering Institute, Technical Report, CMU/SEI-93-TR-25 ESC-TR-93-178, February 1993).

J. Thompson, Organizations in Action (New York: McGraw-Hill Book Company, 1967).

TRADEMARKS

Digital, VAX, and VMS are trademarks of Digital Equipment Corporation.

BIOGRAPHIES

Neil L. M. Davies Neil Davies is the OpenVMS quality manager and is responsible for creating metrics, goals, and programs to achieve engineering excellence in the OpenVMS organization. In this role, Neil works with Digital's engineering management to introduce the techniques established by the Software Enterprise Institute. Neil is also involved with the task force aimed at defining metrics for cycle time, applied time, and product quality and reducing the cost to qualify new systems. Prior to joining Digital in 1992, Neil was the software quality manager for all of Hewlett-Packard's computer systems.

Margaret M. Dumont Meg Dumont is the engineering manager for process development in the UNIX Software Group (USG). Meg helped organize a Software Engineering Institute Assessment of USG. She then created and staffed a process improvement group that is focused on the recommendations from the assessment. She is involved with the future direction and ongoing improvements within the quality function. Prior to this work, she created a standards engineering function within USG and was responsible for standards compliance in the DEC OSF/1 AXP operating systems V1.0 to V1.2. Meg joined Digital in July 1980.