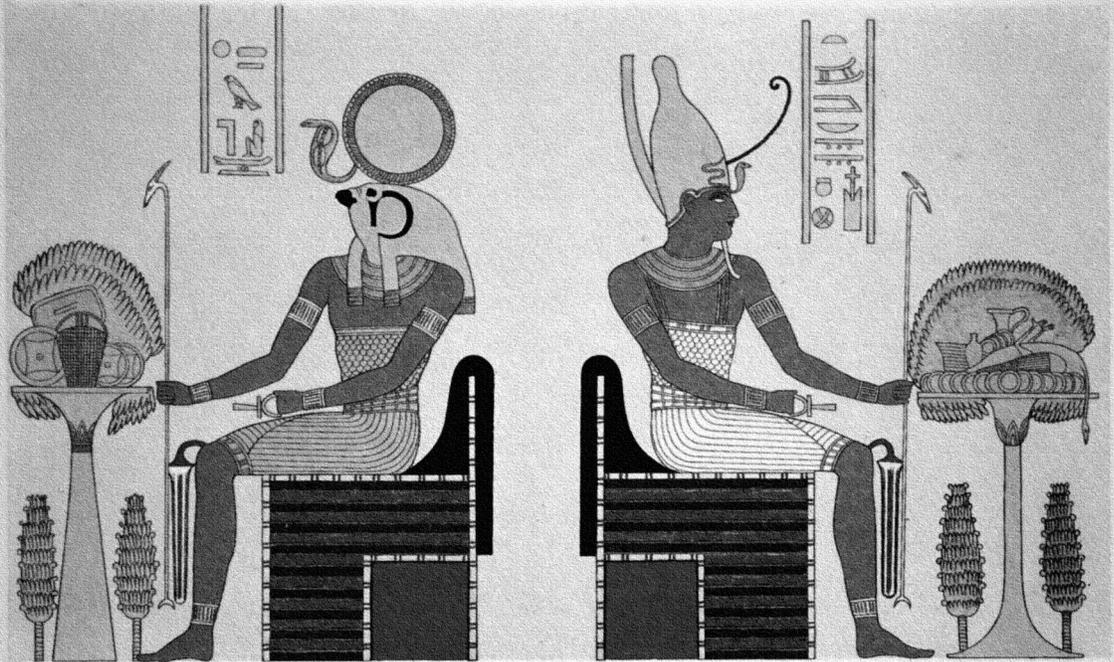


CROSSTALK

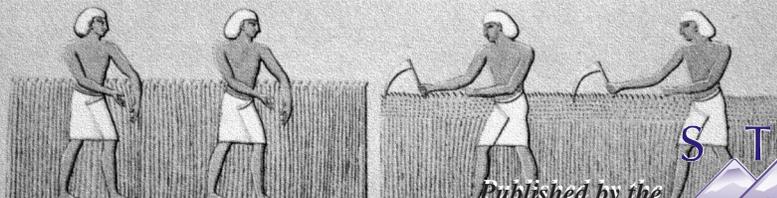
The Journal of Defense Software Engineering



Managing Change



In Your Culture



Published by the
Software Technology Support Center



Sponsored by the
Embedded Computer
Resources Support
Improvement Program
(ESIP)

November 1999
Volume 12 Number 11





CMMI Offers an Enterprise Focus for Technology Change Management

Lt. Col. Joe Jarzombek
ESIP Director



Technology change management represents the fusion of technology innovation and process management, and is a key practice of adaptive, higher maturity organizations. Managing changes in technologies and processes are among the foremost challenges for many organizations because process stability and continual improvement are difficult to balance. Technology change management is not an isolated activity; rather it is a process that touches many of the socio-technical activities in an organization. It includes business and work processes and technical systems. Models such as the Capability Maturity Model® (CMM®) provide a vital focus for technology change management. The integration of disciplines within a single model, the CMM Integration (CMMISM) offers a more encompassing focus to better support technology change management efforts for projects and enterprise-wide process improvement.

In his article, “Structured Approaches to Managing Change” (page 4), Mark Paulk offers three perspectives that may be of value in thinking about change management:

1. internally driven change [push] vs. externally driven change [pull]
2. change directed at products and services vs. those directed at design and production processes
3. incremental vs. revolutionary change

He notes that considering innovation management models can lead to significant changes in models, such as the CMM, that are widely influential in driving process improvement. Models also can influence strategic decision making by broadening and structuring executives’ thinking. An objective view of the challenges that must be overcome in adopting a new technology or process can be materially aided by structuring the analysis around models.

Linda Levine, in her article “Integrating Knowledge and Processes in the Learning Organization” (page 17), writes that paying attention to how people learn enables more effective change management. Learning and technology change management reinforce one another. When people are asked to change how they work, such as in adopting changes in technology or process improvement, then they are asked to learn. To be successful, learning organizations require the integration of process management, knowledge management, and technology.

To better support technology change management, the CMMI integrates software and systems engineering processes that are critical to maturing organizational capabilities. CMMI serves as a more effective tool in helping to guide and assess integrated process improvement efforts. The next CMMI draft

will include integrated process and product development process areas. To achieve process improvement goals, higher maturity organizations will focus organizational improvement management on critical CMMI process areas composed of practices that are key to organizational transformation, such as organizational process focus, organizational process technology innovation (OPTI), and process innovation deployment (PID). These will help focus technology improvements and innovations that can measurably improve the organization’s processes. OPTI and PID involve identifying, selecting, and evaluating new technologies, and systematically transitioning incremental and innovative improvements into use. Understanding that managerial decisions regarding innovation are dominated by information relative to business objectives, people initiating and fostering innovation should assure that decisions regarding the selection of technologies and processes to be improved are based on the organizational business objectives. This better assures that organizational management will support the allocation of resources needed for continuous process improvement.

I encourage everyone to download CMMI draft version 0.2, released in August for public review. Go to www.sei.cmu.edu/cmm/cmms/cmms.integration.html. Review the staged and continuous representation of the model to determine which might better suite your organization’s needs. To better understand how the CMMI compares to the Software CMM, check the STSC-generated CMMI to SW-CMM 1.1 traceability matrix at STSC’s Web site at www.stsc.hill.af.mil; see news.

The Software CMM has helped organizations focus process change management efforts. Now, with CMMI supporting the integration of process management, knowledge management, and technology evaluation, organizations have a model that can be used to better focus integrated process and product change management to support improvement objectives of the enterprise.

Organizations can begin an iterative approach toward integrated process improvement by using the CMMI in pilot assessments that will provide insight about the value of the CMMI regarding how an integrated model can enhance organizational improvement objectives. Pilot assessments can also help identify “levers” of change that are key to “selling” and focusing future improvement efforts.

Regardless of which model an organization might use, change agents should consider the complementary use of frameworks and tools that now exist to support organizations in pulling together process, knowledge management, and technology to support organizational learning. Two such examples are

The Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark office to Carnegie Mellon University (CMU). CMMI is a service mark of CMU.

IDEALSM and IDEAL-based New Technology Rollout INTRo)¹. IDEAL (initiating, diagnosing, establishing, acting, and learning) is a model that provides a disciplined engineering approach for improvement, based upon the CMM, by focusing on managing the improvement program and establishing a foundation for a long-term improvement strategy. Designed to help organizations adopt and implement new technology, INTRo is a web-based process guide focused on making connections among business problems, value propositions, technology solutions, and their implementation. INTRo integrates the multiple dimensions of change and helps to fill skill gaps. This is significant to technology change management since many such efforts are complex and all-encompassing, requiring comprehensive knowledge and skills that are often not resident in a

single organization or team.

To be effective change agents within organizations, people must pull together process, knowledge management, and technology to support learning and successful change. CMMI, coupled with an appropriate framework or tool suite, can provide the key enabler for successful enterprise-wide technology change management efforts. ♦

1. IDEAL is a service mark of CMU; INTRo is a collaborative effort between the Software Engineering Institute (SEI) and Platinum/Computer Associates. Both IDEAL and INTRo are discussed in the January 2000 issue of *CrossTalk* in Linda Levine's article "Learning: The Engine for Technology Change Management." For more information see SEI's Web site.

CROSSTALK Wants to Hear from You

We welcome reader comments regarding *CROSSTALK* articles or matters pertaining to software engineering. Please send your comments and Letters to the Editor to crosstalk.staff@hill.af.mil or mail to

OO-ALC/TISE
Attn: *CROSSTALK* staff
7278 Fourth Street
Hill AFB, Utah 84056-5205

Limit letters to less than 250 words. Include your name, phone number, and e-mail address. We will withhold your name upon request.

How Have You Changed?

New job? Different e-mail address? Updated contact information?

In an effort to keep *CrossTalk's* mailing database up-to-date we would like to know what is new with you.

Fill out our easy-to-use form at:
<http://www.stsc.hill.af.mil/Crosstalk/crosstalk.html>
then click 'subscription form'

or respond in writing at:
fax: 801-777-8069
e-mail: stsc.custserv@hill.af.mil
mail: *CrossTalk*
OO-ALC/TISE
7278 Fourth Street
Hill AFB, UT 84056-5205



Your information will be promptly updated and you will continue to receive all the latest software engineering information from *CrossTalk* The Journal of Defense Software Engineering. (If you are a first-time reader please fill out the free subscription form located between pages 16 and 17 in this or any issue or visit our Web site at <http://www.stsc.hill.af.mil/Crosstalk/crosstalk.html>.)



On the cover: Throughout history, each culture has managed change in a different way. This month's theme depicts a culture that was known for its innovations, and illustrates the need to successfully integrate change.



Structured Approaches to Managing Change

Mark C. Paulk
Software Engineering Institute

Change management is crucial in today's fast-moving world. Three perspectives may be of value in thinking about change management: internally driven vs. externally driven change; change directed at products and services vs. those directed at design and production; and incremental vs. revolutionary change. A number of structured approaches have been developed for thinking about the implications of change. This paper summarizes three of those approaches: Geoffrey Moore's "crossing the chasm," Robert Fichman and Chris Kemerer's "assimilation gap," and Abdelkader Daghfous and George White's "innovation analysis model."

Introduction

In today's world of rapid technological change, managing changes in both technologies and processes are among the foremost challenges for high-tech organizations. The emphasis in models such as the Capability Maturity Model for Software (CMM®) [1] and standards such as ISO 9001 on both process stability and continual improvement are difficult to balance.

Three perspectives may be of value in thinking about change management:

- internally driven change (push) vs. externally driven change (pull)
- change directed at products and services vs. those directed at design and production processes
- incremental vs. revolutionary change

Internally driven change comes about when an organization develops a new process or technology, in a research and development unit, for example, which is then "pushed" into use. Externally driven change occurs when an organization adopts a new technology or process that was developed elsewhere, i.e., "pulled" into use by demand. Note the focus on push or pull of technology in distinguishing between internal and external change. An alternative interpretation could be change-driven by the chief executive officer (internal driver) vs. that forced by the customer (external driver).

Product-oriented change is designed to be directly embedded in the products and services offered by the organization. An example is building software products with a graphical user interface (GUI) to broaden the market appeal of a product. Process-oriented change affects the way a product is designed or a service is provided. An example would be a usability engineering laboratory, where the effectiveness of various GUI designs are explored.

Incremental change and revolutionary change shade into one another, but the distinction can be dramatic between a *kaizen*-style¹ approach to gradual, cumulative improvement and business process engineering that starts over with a clean slate.

Combinations of these three perspectives are fairly common, e.g., internally driven innovation embedded in the organization's products that is intended to revolutionize the market,

such as Visicalc, or the adoption of an external technology to be incorporated into the production process as an incremental improvement, such as the Unified Modeling Language.

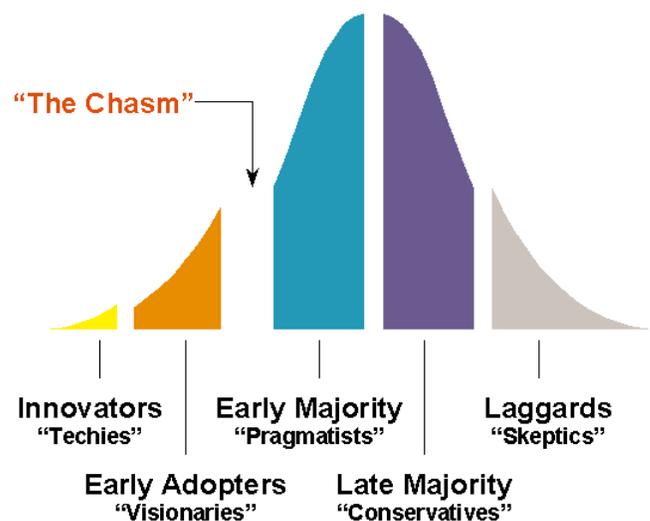
One of the challenges for a CMM writer is to determine which of these perspectives should be incorporated into the model as a determinant of organizational capability. For the Software CMM, we chose to focus on process-oriented change, regardless of source or magnitude. An executive, however, must address all of these aspects since organizational success — and survival — can be limited by failure from any of these perspectives.

It is somewhat surprising to note, therefore, that few high-tech organizations place significant emphasis on models and tools for structuring their thinking about innovation and change. An exception to this generalization is the emphasis in high-tech organizations on marketing their products to appropriate market niches. The work of Geoffrey Moore, for example, is both well-known and influential in high-tech companies.

Moore's Crossing the Chasm

A common model for characterizing the classes of people involved with technology adoption is that they can be listed as innovators (techies), early adopters (visionaries), early majority (pragmatists), late majority (conservatives), and laggards (skeptics) (see Figure 1). Moore extends this by identifying a "chasm"

Figure 1. *Crossing the chasm between visionaries and pragmatists.*



The Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark office to Carnegie Mellon University.

that separates early adopters and the early majority; a gap between two fundamentally separate phases in the development of a high-tech market [2]. The early phase builds from a few, highly visible, visionary customers, but transitioning to the mainstream phase, where the buying decisions fall predominantly to pragmatists, is a major challenge. The key to Moore's insight is characterizing the differences between these communities and how to proactively deal with them.

To address the chasm via organizational learning (i.e. process management) implies that the organization recognizes the existence of the chasm and the importance of addressing the needs of the early majority differently than from those of innovators and early adopters.

Fichman and Kemerer's Assimilation Gap

Fichman and Kemerer take a slightly different perspective by examining the assimilation gap between a new technology acquired by an organization, the traditional mechanism for measuring adoption, and its actual deployment and use [3,4]. Many researchers treat the acquisition of a technology as the adoption event, yet the failure to address the actual deployment makes a critical assumption about the last stages of the standard technology adoption curve illustrated in Figure 2.

Fichman and Kemerer point out that widespread acquisition of a technology is not necessarily followed by widespread deployment and use, as shown by the assimilation gap in Figure 3. Traditionally, innovation attributes such as relative advantage, complexity, and compatibility are viewed as the determinants of the rate and level of diffusion. Fichman and Kemerer propose that acquisition and deployment have different drivers, even though they are related processes. Acquisition is driven by the expectation of future benefits owing to increasing returns, but knowledge barriers impede deployment.

To address the assimilation gap via organizational learning (or process management) implies that the organization recognizes the difference between acquiring and deploying a technology and is proactive in tracking and addressing deployment issues. This means understanding the factors influencing returns to adoption (such as network externalities, learning-by-doing,

Figure 2. The standard technology adoption S-curve.

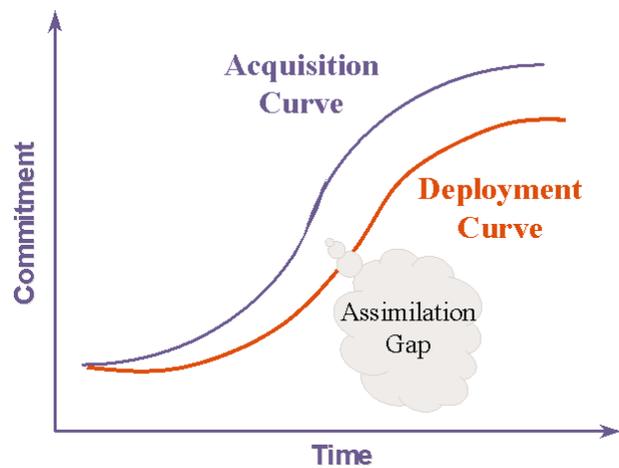


Figure 3. Fichman and Kemerer's Assimilation Gap.

and technological interrelatedness) and knowledge barriers (such as complexity and scaling).

The Daghfous and White Innovation Analysis Model

Daghfous and White have integrated a number of time-based approaches to characterizing the process of innovation that consider product and process evolution and marketing. They add an information axis and a focus on how information interacts with the demand and supply axes. The Daghfous and White model is primarily used by companies concerned with innovation management. The Daghfous and White innovation analysis model has three dimensions — product/process, application linkage, and information [5], illustrated as an "unfolded" three-dimensional graphic in Figure 4.

The product/process axis, also known as the supply axis, is the axis along which events proceed technically, from initial invention to successful innovation. This sequence of events can be characterized as:

- **INVENTIVE ACTIVITY** — new scientific principles or new combinations of existing principles provide substantial value-added power or flexibility over previous means.
- **EMBODIMENT ACTIVITY** — combination of new invention with existing complementary technology to deliver added value towards specified product performance.
- **OPERATIONAL ACTIVITY** — functions of finance, manufacturing, distribution, and maintenance adapting their processes to the new technology as necessary or advantageous; operationally delivering and sustaining the new product in customer use.
- **MARKET EVOLUTION** — the continuing processes of customers and suppliers jointly extending the usage of the innovation as far as possible, maximizing value-added and profit, under precise knowledge of the product/processes and applications linkages involved.

The applications linkage axis, also known as the demand axis, is the axis along which those events that define markets proceed, from initial definition of concept value to successful application

TRANSILIENCE — APPLICATION LINKAGE VS. PRODUCT/PROCESS

Concept Definition			
Utility Integration			
Product Penetration			
Market Evolution	Operational Activity	Embodiment Activity	Inventive Activity



SUPPLY — INFORMATION VS. PRODUCT/PROCESS

Precise Knowledge	Operational Activity	Embodiment Activity	Inventive Activity
Likely Optimization			
Plausible Plan			
Conceivable Scheme			

DEMAND — INFORMATION VS. APPLICATION LINKAGE

Precise Knowledge	Product Penetration	Utility Integration	Concept Definition
Likely Optimization			
Plausible Plan			
Conceivable Scheme			

Figure 4. *The Daghfous and White Innovation Analysis Model.*

of the innovative product. The customer’s question is, “Why would anyone want to do that?” This sequence can be characterized as:

- **CONCEPT DEFINITION** — the concept of a prospective innovation must define value in the context of customer applications.
- **UTILITY INTEGRATION** — to incorporate the value concept of an innovation, the user has to adapt the business activities into which it is integrated to maximize its utility.
- **PRODUCT PENETRATION** — the major growth of application usage requires validation of the utility as integrated into customer use, communication to firm and trade of values obtained, escalation of use in existing applications, and extension of use to new applications.
- **MARKET EVOLUTION** — see above. This phase is common to supply and demand.

The information axis deals with the transformation of uncertainty and ignorance into precise knowledge. Uncertainty means that the information does not exist to remove variance of expectations — the opposite of uncertainty is precision. Uncertainty can only be resolved by experience, by trial-and-error. Ignorance means the information is known or accessible elsewhere, but the innovator is oblivious and thus at a competitive disadvantage. The opposite of ignorance is knowledge. Ignorance is most efficiently addressed by study and analysis prior to operations.

The information available during successive phases of an innovation can be characterized as:

- **CONCEIVABLE SCHEME** — information and analysis on a potential innovation are sufficient to show that the innovation could succeed but are insufficient to show that it should succeed.
- **PLAUSIBLE PLAN** — prospects for innovation success are qualitatively positive.
- **LIKELY OPTIMIZATION** — all accessible information and complete analysis provide positive probability for success and for maximum values added.
- **PRECISE KNOWLEDGE** — comprehensive cumulative innovation provides precise knowledge of how innovation has succeeded to date and how its evolution should continue.

Managerial decisions regarding innovation are dominated by the lack of information. Lack of information is a major inhibitor to innovation, and addressing this lack is a direct consequence of innovation — although resolving uncertainty and ignorance require different approaches. Information gathering along the product/process axis usually results in removing ignorance. Information gathering along the application linkage axis usually results in removing uncertainty. It can be assumed that once information is learned, it will not be forgotten.

The ultimate objective, or bulls-eye, for an innovation is continuing market evolution under precise knowledge, with optimum products from optimum processes satisfying optimum demand. The sequence, if not the timing, of events is predictable using the Daghfous and White model.

To manage innovations that may be adopted externally via organizational learning or process management implies that the organization recognizes the importance of the information axis and how it impacts the other axes. This means analyzing where a technology or product is on the process/product and application linkage axes.

Conclusions

The Software CMM focuses on process-centered change, whether the change is incremental or revolutionary, internal or external. Product-oriented change is not considered within the scope of process management, although it is a fairly simple extension of the ideas in the CMM to incorporate product change management.

Considering the perspectives suggested by these and other innovation management models can lead to significant changes in models such as the CMM, which are widely influential in driving process improvement. They can also influence strategic decision-making by broadening and structuring the thinking of executives.

An objective view of the challenges that must be overcome in adopting a new technology or process can be materially aided by structuring the analysis around models such as the three summarized in this paper. Simply piloting a new technology and determining that it is likely to be beneficial is insufficient, as these models so aptly point out. ♦

About the Author



Mark C. Paulk is a senior member of the technical staff at the Software Engineering Institute (SEI)². He has been with the SEI since 1987 and initially worked on the Software Capability Evaluation project. He has worked with the Capability Maturity Model project since its inception and was the project leader during the development of version 1.1 of the CMM. He is

a contributor to the International Organization for Standardization's Software Process Improvement and Capability Determination (SPICE) project, which is developing a suite of international standards for software process assessment. Prior to his work with the SEI, Paulk was a senior systems analyst for System Development Corporation (later Unisys Defense Systems) at the Ballistic Missile Defense Advanced Research Center in Huntsville, Ala. He is a senior member of the Institute of Electrical and Electronic Engineers and a senior member of the American Society for Quality Control. Paulk has a master's degree in computer science from Vanderbilt University and a bachelor's degree in mathematics and computer science from the University of Alabama in Huntsville.

Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, Pa. 15213
mcp@sei.cmu.edu

References

1. Paulk, Mark C., Charles V. Weber, Bill Curtis, and Mary Beth Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*, ISBN 0-201-54664-7, Addison-Wesley Publishing Company, Reading, Mass., 1995.
2. Moore, Geoffrey A., *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers*, HarperCollins Publishers, New York, N.Y., 1991.
3. Fichman, Robert G., Chris F. Kemerer, "Adoption of Software Engineering Process Innovations: The Case of Object Orientation," *Sloan Management Review*, Winter 1993, pp. 7-22.
4. Fichman, Robert G., Chris F. Kemerer, "The Illusory Diffusion of Innovation: An Examination of Assimilation Gaps," Working Paper Series No. 746, Katz Graduate School of Business, University of Pittsburgh, November 1995.
5. Daghfous, Abdelkader, George R. White, "Information and Innovation: A Comprehensive Representation," Technical Report 91-4, Department of Industrial Engineering, University of Pittsburgh, 1991.

Notes

1. Kaizen is the Japanese word for an incremental change philosophy.
2. The SEI is a federally funded research and development center sponsored by the Department of Defense.

Call for Articles

If your experience or research has produced information that could be useful to others, *CROSS TALK* will get the word out. We welcome articles on all software-related topics, but are especially interested in several high-interest areas. Drawing from reader survey data, we will highlight your most requested article topics as themes for future *CROSS TALK* issues. In future issues, we will place a special, yet nonexclusive, focus on:

Cost Estimation

April 2000

Submission deadline: Dec. 4, 1999

Software Security

May 2000

Submission deadline: Jan. 4, 2000

Capability Maturity Model Integration (CMMI)

June 2000

Submission deadline: Feb. 1, 2000

We will accept article submissions on all software-related topics at any time; issues will not focus exclusively on the featured theme.

Please follow the *Guidelines for CROSS TALK Authors*, available on the Internet at <http://www.stsc.hill.af.mil>.

Ogden ALC/TISE
ATTN: Heather Winward
CROSS TALK Associate Editor/Features
7278 Fourth Street
Hill AFB, Utah 84056-5205

Or e-mail articles to stsc.custserv@hill.af.mil. For more information, call 801-775-5555 DSN 775-5555.

Addressing People Issues when Developing and Implementing Engineering Processes

Claude Y. Laporte
Yortar Technologies
Sylvie Trudel
Oerlikon Aerospace

This paper describes the approach used by a defense contractor to address the people issues raised when developing and implementing engineering processes. First, a brief description of the context is presented, then organizational mechanisms to better manage changes are described, and finally 16 lessons learned are presented.

Background

The organization is a systems integrator of an air defense missile system. More than 120 systems and software engineers are involved in the system's development and maintenance.

The organization had been ISO 9001 certified since 1993. In 1997 the organization had been certified as CMM® [1] Level 2 by independent assessors certified by the Software Engineering Institute. In addition to satisfying Level 2 goals, the organization met eight of the 17 Level 3 goals.

It was decided in 1995 that a formal systems engineering process had to be developed and implemented in order to seamlessly integrate disciplines associated with systems engineering. For an in-depth description of the process and its application in the organization, the reader is referred to other papers that have been published [2,3].

The organization thought that it also would benefit from a standardized project management process. In 1996 a working group selected *A Guide to the Project Management Body of Knowledge*®, developed by the Project Management Institute [4], as the framework for the organizational process.

The Management of Change

Since the management of change is a key element of a successful process improvement program, a series of mechanisms were put in place in order to facilitate the development, implementation, and adoption of processes, methods, and tools.

Organizational Process

Coordination

In early 1997, the thought was that implementing these processes would need organizational coordination and direction. A steering committee, the Process Action and Coordination Team (PACT), was established. The PACT is made up of vice presidents, the manager responsible for quality assurance, and the coordinator for process performance improvement.

The functions of the PACT are to:

- establish time-to-market, quality, costs, and product performance objectives to be supported by organizational processes
- set priorities in accordance with company vision and yearly objectives
- work as a liaison with executive committee
- establish consensus among different groups
- provide support for process performance improvement:
 - Review results of assessments and audits
 - Charter technical area working groups
 - Budget for resources for process groups
 - Monitor process performance

Process Ownership

A process owner is responsible for the processes' effectiveness and efficiency, methods, and tools. As an example, each year the process owner develops a process improvement plan (PIP). Process owners have also been delegated to review the tailoring of the process before a project is approved. Knowing that a project manager

and a process owner may have conflicting views about the tailoring of a process, a policy was written to handle such conflicts. In the event of a deadlock between a project manager and the process owner, both would present a risk analysis to a vice-president for the final approval of the tailored process.

Awareness Activities

To build the sponsorship level, the president of the organization attended an executive seminar on process improvement and two directors attended a three-day seminar discussing process, process assessment, and improvement. The coordinator for process improvement attended process improvement courses and conferences. There were briefing sessions and articles in each company's newsletter to explain the why, what, and how of process assessment and improvement and describing the progress made.

Meeting Guidelines

In order to facilitate the conduct of working group activities, the facilitators proposed a certain number of meeting guidelines to the members of working groups during their group's kick-off meeting [5] (Table 1). The facilitator read each proposed rule and asked participants if they agreed with the rule. Once the discussion ended, the facilitator reminded the participants that in the future, he would facilitate each meeting using the set of rules selected. After a few meetings, the facilitator invited participants to become

The Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.

- One conversation at a time.
- In the meeting or out, but not both (i.e. participants should make a commitment to participate in the meeting for the full duration).
- 100-mile rule (i.e. no interruptions; telephone messages are not allowed unless urgent)
- How decision will be made (e.g. by consensus, majority or minority rule, autocracy, or unanimity).
- Once a decision is made, participants support it inside and outside the meeting.
- Be as open as possible.
- Listen, with respect, to others and do not interrupt.
- Silence is consent.
- Few recreational stories.
- Differences are respected.
- Avoid blaming individuals.
- Come prepared to meetings.
- Publish minutes and action items at each meeting.

Table 1. *Proposed meeting guidelines.*

secondary facilitators, (i.e. when a participant observed a behavior which violated one of the meeting guidelines, he raised the issue with the offender). Eventually, a group can manage the “soft issues” without an outside facilitator. During meetings, a process owner focused on the content of a specific process while the facilitator focused on the process of developing a specific engineering process.

Decision Making

It was also decided that consensus decision-making was the preferred option. We defined consensus, according to the definition found in *The Team Handbook* [5]: consensus is not unanimity, consensus is based on the assumption that solutions are more likely to succeed if all of the key participants are “comfortable enough” with the outcome to move forward. From time to time “thumb voting” procedures [6] were used to make decision by consensus. This allows the following three alternatives: if the proposition is favored, the thumb is up; if someone can live with the decision, the thumb is to the side; and if someone cannot live with the decision, the thumb is down. In the later case, the members of the working group had to take time to understand the issues at stake and proposed an alternative that everyone could live with.

Team Evaluation

From time to time, members of the

working groups had to evaluate their group’s effectiveness. A survey [7] was distributed at the end of a meeting. Individually, members completed the survey and sent it to their group facilitator. The survey addresses the following issues: goals and objectives, utilization of resources, trust and conflict resolution, leadership, control and procedures, interpersonal communications, problem solving, experimentation, and creativity. Issues that members brought up were discussed to generate suggestions for improvement.

Team Charter

Each working group was managed like a project: it had a charter. The charter listed a budget, objectives, key players, roles and responsibilities, deliverables, risk issues, and expected schedule.

Lessons Learned

Certain lessons likely to be used by other organizations in the future are discussed below.

Lesson 1: Set Realistic Expectations for Senior Management

Appropriate expectations must be set prior to embarking on process development. The trap, especially for a low maturity level organization, consists of communicating to management the idea that a process improvement initiative will

be easy, fast, and inexpensive, has to be avoided at all costs.

A typical scenario looks like this: senior management hears about the benefits that attaining a maturity level could represent for the organization’s competitiveness. A project manager or an external consultant states, in order not to upset senior management, that such objectives are easily attainable. Senior management mandates middle managers to attain this objective in a short amount of time. If a formal process assessment is performed, a string of countless findings are surfaced to senior management. These are findings that developers had known about for a long time but which middle managers ignored due to the management mode of dealing continuously with the problems created (i.e. fighting fires). Then, senior management that had publicly announced its objectives suddenly realizes that it will take a lot more time and resources than initially estimated.

Three reactions are possible. Senior management may accept the findings and confirm that it will continue to support the objectives announced. Or senior management may announce discreetly that objectives will be lowered. Finally, it cannot accept the assessment findings and not put in place an action plan to correct the deficiencies highlighted by the assessment. This decision could have a destructive effect on the morale of the developers, since they know that the deficiencies they had been long deplored will continue to be ignored.

The lesson is to prepare a short action plan — some sort of a brief appraisal of the situation — preferably by someone who is not involved in the targeted sector. Estimate the time and resources necessary to perform a formal assessment, prepare and implement an action plan. It is better not to proceed to an assessment if it is not intended to deal with the findings. Once the problems are identified and publicized within the organization, if the management decides not to act, it sends a very bad message to practitioners.

Lesson 2: Secure Management Support

A second lesson for low maturity level organizations consists in realizing that

most of the assessment findings target the deficiencies of management processes. It is necessary to create an environment where the organization is ready to invest in implementing processes rather than blame its managers; in other words where the management is ready to fix the process, not the people. This is one of the reasons it is necessary to keep senior management informed so it can show understanding and full commitment when these findings are publicized within the organization.

Beside senior management buy-in, it is essential that middle management and first-line managers become strong supporters of the process improvement program. The strongest signal managers send is their day-to-day activities, because what a manager does talks louder than what a manager says. The developers must receive clear signals that the changes announced will be implemented and new practices will be enforced.

Lesson 3: Identify Management Needs, Expectations, and Understanding of the Problem

The involvement of process owners or managers is largely related to their understanding of the situation i.e. strengths and weaknesses of management and process. Once convinced that the current situation is undesirable, they will provide the leadership, direction, and momentum to implement solutions. They can also keep working groups focused on solving the right problems since it is very easy, after a few meetings, for a working group to start solving what it perceives to be the problems.

Lesson 4: Establish a Process Improvement Working Group before an Assessment

It is best if a small process group becomes active in process activities several months before the on-site assessment. The process group should take this time to familiarize itself with the models, such as the Capability Maturity Model (CMM) or the Electronics Industries Association (EIA)731 [8] and associated process improvement methods and tools. Ideally there should be one full-time person in the process group, while the other mem-

bers could be assigned on a part-time basis. Beyond their technical competencies, the members of the process group should be selected based on their enthusiasm for improvement and the respect they have within the organization.

Lesson 5: Start Improvement Activities soon after an Assessment

With regard to the development of the action plan, the organization should capitalize on the momentum gained during the assessment period. The organization does not have to wait for a completed action plan to begin process improvement activities. The implementation of certain improvements is an important motivation factor for all members of the organization.

Lesson 6: Collect Data to Document Improvements

Before and during the assessment, it is recommended that quantitative and qualitative data be collected. It will be used later to measure progress. One could obtain project data such as budgets and schedules, or measure the degree of customer satisfaction regarding product quality level. Since senior management will have made investments, it is important to be able to demonstrate that these investments have been profitable.

Lesson 7: Train all Users of the Processes, Methods, and Tools

Once processes are defined, it is essential to train all users. Otherwise, process documents will end up collecting dust on shelves. It is illusory to think that in addition to their workload, developers will study new processes by themselves. Training sessions also serve as a message that the organization is moving ahead and will require that its developers use these practices. During the training sessions, it is necessary to indicate that errors are bound to occur while using new practices. This will help reduce developers' level of stress when using these new practices. It would be wise if a resource person (i.e. a hotline) is available to help developers when they face obstacles while implementing new practices.

Lesson 8: Manage the Human Dimension of the Process Improvement Effort

The authors wish to make the reader aware of the importance of the human dimension in a process improvement program. The people responsible for these changes are often extremely talented engineering practitioners, however, not too well equipped in change management skills. The reason for this is simple: their academic training focused on the technical dimension and not on the human aspect. However, the major difficulty of an improvement program is precisely the human dimension.

While preparing the technical part of the improvement action plan, the change management elements have to be planned. This implies, among other things: (1) a knowledge of the organization's history with regards to any similar efforts, successful or not; (2) the company's culture; (3) the motivation factors; (4) the degree of emergency perceived and communicated by (a) the management, (b) the organization's vision, and (c) the management's real support. The authors are convinced that the success or failure of an improvement program has more to do with managing the human aspect than managing the technical aspect.

Lesson 9: Process Improvement Requires Additional People Skills

In an organization that truly wants to make substantial gains in productivity and quality, a cultural shift will have to be managed. This requires a special set of people skills. The profile of the ideal process facilitator is someone with a major in social work and a minor in engineering. The implementation of processes implies that both management and employees will have to change their behavior.

With the implementation of processes, management will need to change from a "command and control" mode to a more "hands-on" or participatory mode. As an example, if the organization truly wants to improve its processes, a prime source of ideas should come from those who are working, on a daily basis, with the processes. This implies that manage-

ment will need to encourage and listen to new ideas. This also implies that the decision-making process may have to change from the autocratic style, e.g. "do what you are told" to a participatory style, e.g. "let us talk about this idea." Such a change requires support and coaching from someone outside the functional authority of the manager who has to change his/her behavior. Similarly, employees' behavior should change from being the technical heroes who can solve any problem, to team members that can collaborate and listen to others' ideas.

During the first few months of the introduction of a new process, a new practice, or a new tool, management and employees must acknowledge that mistakes will be made. Unless a clear signal has been sent by management and a "safety net" has been deployed to recognize this situation, employees may hide their mistakes. The result is that not only the organization will not learn from those mistakes but other employees will make the same mistakes. As an example, the main objective of a formal inspection process is to detect and correct errors as soon as possible in the project life cycle. Management has to accept that in order to increase the error's detection rate, results from individual inspections will not be made public, only composite results from many inspections (e.g. at least 10 inspections from different projects) will be made public. When management accepts this rule, employees should feel safe to identify mistakes in front of their peers instead of hiding them. The added benefit to correcting errors is that those who participate in an inspection will learn how to avoid these errors in their own work.

Facilitating behavior changes requires skills that are not taught in technical courses. It is highly recommended that the people responsible for facilitating change be given appropriate training. The authors recommend two books that may facilitate the management of change: the first one [9] gives advice to anybody acting as an internal consultant; the second one [10] gives the steps for developing and implementing a change management plan.

Lesson 10: Select Pilot Projects Carefully

It also is important to carefully select pilot projects and participants to the pilots, since these projects will foster adoption of new practices throughout the organization. As stated before, first-time users of a new process will make mistakes. If participants sense that mistakes will be used to learn and make improvements to the process instead of pointing fingers, the level of anxiety will be reduced and they will bring forward suggestions instead of hiding mistakes.

Managing the human dimension of the process engineering initiative is the component which not only fosters the adoption of change but also creates an environment where changes could be introduced at an increasingly greater rate. Members of the engineering organization now realize that managing the "soft stuff" is as important as managing the "hard stuff."

Lesson 11: Conduct Process Audits

Process audits should be conducted on a regular basis for two main reasons: first, to ensure that practitioners are using the process, and second, to discover errors, omissions, or misunderstandings in the process application. Process audits help to assess the practitioners' degree of utilization and understanding. As an example, a documentation management process was distributed and practitioners were asked to produce and update documents using this new process. It is widely known that engineers are usually not prone to documenting their work.

An audit was launched to measure process compliance. As expected, results

of the first audit were not exhilarating. The engineering manager kindly reminded engineers, in writing, to use the process. He also informed them that a second audit would be performed in the future. The results of the second audit are substantially better than the first audit (Table 2). The auditor gathered feedback and suggestions from engineers; this information would be used by the process owner to improve the process.

Lesson 12: Conduct Team Effectiveness Surveys

Usually people are not very likely to raise "soft" issues. Such tools [7] may promote open discussion with members of a group in order to improve its performance and provide the facilitators with information that helps them probe delicate issues. As an example, if the majority of a group reports that interpersonal communications are weak, the facilitator can probe the members and invite them to propose solutions. After a few meetings, the results of a new survey will show if the solutions really helped the team improve their communications.

Lesson 13: Start a Process Initiative from the Top-Level Process

The process improvement initiative was a bottom-up exercise (i.e. first software process was developed, then systems engineering). Historically, this was the selected strategy because, in 1992, only the software CMM was available; then came the systems engineering CMM and after that, the Project Management Body of Knowledge. If an organization had to start a process initiative today, it would simplify process integration to start from

Table 2. Results of audits performed on the documentation management process.

Activity	Results from First Audit	Results from Second Audit
Comments made by reviewers	38 %	78%
Approval matrix completed	24%	67%
Effort log completed	18%	33%
Review checklist completed	5%	44%
Configuration management checklist completed	5%	27%
Distribution list completed	38%	39%
Document formally approved	100%	100%

the top by developing the project management process, then the systems engineering process, and finally the software process. It would also be possible to develop engineering processes in parallel once the requirements for the top level process are well stabilized.

Lesson 14: Get Support from Change Experts

As mentioned above, surveys were conducted in order to “measure” issues such as culture, implementation history, and team effectiveness. Once the surveys were compiled, we had some indications of organizational strengths and weaknesses. The difficult part was to decide what to do next. As an example, one issue on the survey was risk taking (i.e. people resent taking risks). One possible cause for such behavior could be that people did not want to be blamed for an error. Next we would have to find the cause for this behavior, and so on. It would have been very helpful to have access to someone with expertise in organizational change. This would have saved a lot of long discussions and many wrong answers.

Lesson 15: Tie Process Improvement Activities to Business Objectives

It was observed that software and systems engineering process improvement really picked up momentum when a common focal point was created among management, engineers, and customers. Understanding that process improvement's real benefit lies in improving product quality and reducing time-to-market and cost, leads to improving the organization's ability to better compete. Additionally, a multi-year PIP was a very important tool to illustrate the links between business objectives, project requirements, and process development or improvement. Essentially the PIP illustrated that the engineering of processes was not a paper exercise but an important infrastructure for the successful accomplishment of projects. Being a multi-year plan, the PIP also showed to practitioners the long-term commitment of management to process improvement activities.

Lesson 16. Adopt a Common Vocabulary

To succeed in any project endeavor, a common vocabulary is a basic requirement. As we developed the processes, we realized that different players had different meaning for the same word, or the same word had different meanings, and some words were not well known to some individuals. We mandated one team member as the “glossary keeper.” His role was to collect a vocabulary, propose some clean-up in the terminology, and to gradually build a common glossary for all processes.

Conclusion

We have shown that the development and deployment of engineering and management processes entail technical and management competencies. Five elements are necessary for a successful implementation of organizational changes:

- Management sets a direction and process objectives are linked to business objectives. Without a clear direction, confusion may mislead people from reaching the desired change.
- People are trained to perform new tasks. Without the proper training, anxiety among the organization's staff is likely to slow down the occurrence of change.
- Incentives are provided to facilitate the adoption of changes.
- Resources are estimated and provided. Otherwise, frustration may put an end to the organization's willingness to change.
- An action plan is developed and implemented to avoid false starts.

These years of process improvement activities have demonstrated that constant attention to the people issues is critical to the success of technological changes. We suggest to manage those people issues as risk items and to track them throughout the improvement effort.

Finally, as stated by J. Pfeffer in his book *The Human Equation*, “It is almost impossible to earn above-normal, exceptional economic returns by doing what everyone else is doing ... it is also impossible to achieve some lasting competitive

advantage simply by making purchases in the open market — something that anyone can do” [11]. ♦

About the Authors



Claude Y. Laporte obtained a bachelor of science degree in 1973 from le Collège Militaire Royal de Saint-Jean. In 1980, he obtained a master of science degree in physics at Université de Montréal, and in 1986, a master's degree in applied sciences from the Department of Electrical and Computer Engineering at École Polytechnique de Montréal. He was an officer in the Canadian Armed Forces for 25 years and a professor for more than 10 years. From 1988-92, he was involved in the implementation of the Applied Software Engineering Center. He left the Canadian Forces in 1992 at the rank of major. He joined Oerlikon Aerospace, where he coordinated the development and implementation of software engineering, systems engineering, and projects management processes, methods, and tools. In 1999 he left Oerlikon Aerospace to launch consultation services as a partner of Yortar Technologies. He is the president of the Montréal Software Process Improvement Network.

Yortar Technologies
481 Bissett
Saint-Jean-sur-Richelieu
Québec, Canada J3A 1W6
Voice: 450-349-4088
Fax: 450-349-4620
E-mail: claporte@yortar.com



Sylvie Trudel obtained a bachelor's degree in computer science in 1986 from Laval University in Québec. She worked for more than 10 years in development and implementation of management information systems. She joined Oerlikon Aerospace in 1996 as the process control analyst for the software engineering department. She is a full-time member of the organization's Process Action and Coordination Team. She played a major role in the introduction of formal inspections. She was a team member and site coordinator of the formal software engi-

neering process assessment.

Oerlikon Aerospace
225 Seminaire sud
Saint-Jean-sur-Richelieu
Québec, Canada J3B 8E9
E-mail: strudel@oerlikon.ca

References

1. Paulk, M. et al, Capability Maturity Model for Software, Software Engineering Institute, SEI/CMU-93-TR-24, 1993.
2. Laporte, C., Papiccio, N., "Development and Integration of Engineering Processes at Oerlikon Aerospace," Proceedings of the Seventh International Symposium of the International Council on Systems Engineering, August 3-7, Los Angeles, Calif., 1997.
3. Laporte, C., A. Guay, J. Tousignant, "The Application of a Systems Engineering Process to the Re-Engineering of an Air Defense System," Proceedings of the 8th International Symposium of the International Council on Systems Engineering, July 26-30, Vancouver, Canada, 1998.
4. *A Guide to the Project Management Body of Knowledge*, Project Management Institute, 1996.
5. Scholtes, P., *The Team Handbook*, Joiner Associates Inc, 1996.
6. Popick, P.R., S.A. Shear, "Ten Lessons Learned from Implementing Integrated Product Teams," Proceedings, International Council on Systems Engineering 6th Annual International Symposium, Boston, July 7-11, 1996.
7. Alexander, M., *The Encyclopedia of Team-Development Activities*, edited by J. William Pfeiffer, University Associates, San Diego, Calif., 1991.
8. EIA 731, Electronic Industry Association, Systems Engineering Capability, EIA Standard EIA/IS 731 Part 1: Model, 1998.
9. Block, P., *Flawless Consulting*, Pfeiffer & Co., 1981.
10. Bridges, W., *Managing Transitions*, Addison Wesley, 1991.
11. Pfeiffer, J., *The Human Equation: Building Profits by Putting People First*, Harvard Business School Press, 1998.

What is on the Web?

CrosTalk online!

visit <http://www.stsc.hill.af.mil/Crosstalk/crostalk.html> to find:

- current and past issues (1994-present)
- author guidelines
- form to subscribe or update reader contact information
- theme announcements (editorial calendar)
- send us an e-mail or letter to the editor
- search for articles on many software engineering topics

Rapid Emerging Knowledge Deployment

Kevin Marler
Raytheon Learning Institute

This paper describes how to manage technical knowledge that is transitioning faster than subject matter specialists, human resource analysis, and education deployment trainers can respond through traditional competencies. Radical learning techniques used by self-directed emerging technology study groups can serve as a management tool for gaining and leveraging the knowledge necessary to take advantage of rapidly emerging technology. Reduced education material costs, software licenses, and deployment costs are empowering individuals to drive the speed of technology change, teams to gain emerging technology, and managers to deploy knowledge as a strategic advantage.

Open-Sources Accelerate Change

You no longer need to be managing a sophisticated research lab for your people to be on the brink of technological change. Open-sources [1] are making a radical shift in the cost of emerging technology and, thereby, the cost of gaining rapidly emerging knowledge. Leading edge software advances like Java, Linux, and XML (eXtensible Markup Language) are being provided for only the cost of downloading them through the Internet. Special interest groups are providing no-charge open sources distributions consisting of software listings, compiled versions, and documentation. You can experience, modify, extend, and redistribute open sources for free. Individuals are fostering the transition:

- Java happened during a long weekend hack by Patrick Naughton as a throw-away;
- PERL originated with Larry Wall to produce some reports from a Usenet news-like hierarchy of files for a bug-reporting system for the Net;
- tcl/tk was developed by Dr. John K. Ousterhout as a professor in the Department of Electrical Engineering and Computer Science at the University of California at Berkeley;
- Linux originated with Linus Torvalds in a project to explore the 386 chip and was copyrighted under the terms of the GNU's Not Unix (GNU) General Public License written by the Free Software Foundation.
- The World Wide Web (WWW) was invented by Tim Berners-Lee while working at the Centre Européenne pour la Recherche Nucléaire, the European Laboratory for Particle Physics. Berners-Lee originated the first WWW client, a browser-editor running under NeXTStep, and defined Uniform Resource Locator (URLs), HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML).

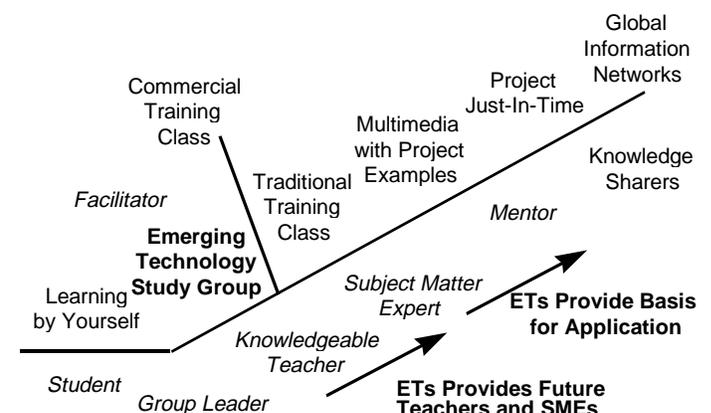
Support is coming from companies eager to get attention on the Internet by giving software away; both major browsers are now free. The World Wide Web Consortium (W3C) was created using seed funding from the Defense Advanced Research Project Agency [2]. The vendor-neutral W3C provides the global community with internet standards for HTML, Standard Generalized Markup Language (SGML), and XML. SGML was

the forerunner to HTML, which is used to format web pages, and XML, which is used to label knowledge assets. Through HTML, anyone can create a web page by hand and with XML one can create his/her own meta-data for sharing knowledge assets. While information remains costly to produce, reproduction has become inexpensive. For example, the cost of compact disc encyclopedias and phone books have dropped from an original asking price of around \$2,000 to significantly less than \$100 [3]. Internet-based dictionaries can be accessed for free.

Managers need to make the paradigm shift from the days when the only way to gain the latest technology was to attend user group conferences, since computer books were rare. Now vendors, interest groups, and individuals host Web sites that often hold more combined information than related publications. Books are published, often before the software releases. When someone asks for knowledge, the right question is, "Do you have Internet access?" Your on-line bookstore can monitor topics and send e-mail with advanced notice of pending publications.

The rate of change exceeds the time to develop subject matter experts, training courses, and human resource interventions. Given how much is free, employees are sometimes gaining more experience at home off the Internet. Managers need to create a learning environment at work that matches their team-based organizations. These teams need to be empowered to rapidly discover, gain ownership, and cross the learning curve.

Figure 1. From individual to global learning.



Rapid Response Knowledge Teams

As waves of technological change are upon us, we are shifting to team-based approaches; teams are struggling to paddle in front of the curl. At Raytheon Systems Co. in Garland, Texas, I have facilitated what I call emerging technology study groups (ETs). I have been sharing a vision that self-directed study groups can gain global advantage, Figure 1.

Object-oriented (OO) methods; Java, tcl/tk, and PERL languages have all moved up this model from individuals studying, through study groups, to formal classes and mentoring. The OO study groups started before UML and Java when there was only one thin Java book available. For organization, I adapted self-directed work teams [4]. They select their own leaders and set their own goals. This feeling of student ownership creates a lively, enthusiastic learning environment in which they train themselves. At times, there have been more active study groups than traditional training classes. The typical study group selects a book, reads a chapter a week, and has team members lead the discussions. Additionally, they were encouraged to select a higher team goal. The Linux study group helped each other install Linux on their home computers. A Java study group built a Java Learning Center, gained Internet access, wrote their own lesson plans, and led self-paced instruction. The successful study groups have used a radical schedule of meeting once or twice a week for an hour. In this way, they differ from cooperative learning, which occurs as part of classroom exercises, see Table 1. Managers can interject work assignments into the classroom as a substitute for cooperative exercises and into project schedules as just-in-time (JIT) study groups.

Employees are experiencing cooperative learning in school. They enter the work force with an expectation that they will be learning in teams, presenting their research findings, and earning part of their evaluations from team-based roles. The first emerging technology study groups were modeled on this expectation. As education is transforming for the Generation-X learner, they are enjoying the shift to interactive hypermedia learning environments. The Internet is becoming the environment for learning, with accompanying CD integration, and online broadcasts and discussion groups. Our study groups have used both our intranet and the Internet to host their materials.

Managing Rapid Learning Expectations

Learning expectations are transitioning from instruction to construction and discovery, from teacher-centered to learner-centered, from absorbing to navigating, from standardized to customized learning, and from teacher as transmitter to facilitator [6]. Training and human resource professionals are adapting

Table 1. *Study group research and cooperative exercises.*

Study Group Research	Cooperative Exercises
Too new for training materials to be prepared	Part of an established training class' exercises
Time allowed between meetings for reading, practicing, and researching	Conducted during class with materials prepared by teacher ahead of time
Group size 10-12, max. 24	Group size 2-3, max. 5

Emerging Technology Study Group (ET)	Strategic Knowledge Team (SKT)
Voluntary self-directed study teams choose their own leaders	Appointed by the knowledge manager. Facilitates the formation of ETs.
Focus on skills desired by the team members	Focus on research assignments from the knowledge manager
Self-motivated	Job-position responsibilities
Based on employee's desire to gain competitive competence	Based on knowledge manager's goals

Table 2. *Self-directed and strategically aligned study groups.*

radical approaches such as accelerated learning environments, adult learning techniques, alternative learning strategies, and knowledge sharing/reuse along with traditional Joint Application Design, Rapid Application Development, and JIT to enable faster learning and deployment. The management role is shifting from making requests of the training department to establishing team-based learning environments. It is deploying trailblazing teams that gain emerging knowledge, skills, and experience, and that facilitate technology dissemination.

Knowledge managers are deploying Strategic Knowledge Teams. They differ from self-directed study groups in their motivation, see Table 2. Study groups are internally motivated; strategic teams, external. The key is that there is usually only one strategic team that can sponsor multiple study groups. Further, the study groups' organizational role can be defined as accountability for commitments to particular learning outcomes, not activities, creating an adaptive sense-and-response team both self-directed and strategically aligned [5].

Rapid learning techniques are available to the manager to deploy emerging technology as a part of their strategic initiatives and corporate knowledge assets. Global learning teams deployed through the Internet, private government and corporate intranets and extranets, based on open-standards, are facilitating and leveraging strategic alliances. ♦

About the Author



Kevin Marler is a Senior Knowledge Specialist at Raytheon Systems Co., Garland, Texas. He is a subject matter expert for the Raytheon Learning Institute, an internal ISO 9001 auditor, and teaches for Richland College, University of Texas at Austin, and Texas Tech University. He has a master's degree in business administration from the University of Phoenix and a bachelor of science degree from Brigham Young University. His competencies include object oriented, artificial intelligence, and knowledge management with experience leading emerging technology deployment in distributed networks, robotics, real-time graphics, and data warehousing. He spoke at STC '97, "A Use-Case Driven Requirements and Architecture Analysis Process for Systems Engineering," and STC '98, "Training Ourselves in the Global Holodeck, Emerging Technology Study Groups."

Raytheon Systems Company
Raytheon Learning Institute — Garland
P.O. Box 660023
Mail stop DA
Garland, Texas 75266-0023
Voice: 972-205-5325
Fax: 972-205-6540
E-mail: marler@onramp.net
Internet: <http://oorad.com>

References

1. Dibona, Chris ed., Mark Stone ed., Sam Ockman ed., *Open Sources: Voices from the Open Source Revolution*, O'Reilly &

- Associates, 1999.
2. W3C "DARPA Support of the Web" <http://www.w3.org/Consortium/Prospectus/DARPA.html>.
3. Shapiro, Carl, Hal R. Varian, *Information Rules, A Strategic Guide to the Network Economy*, Harvard Business School Press, 1999.
4. Fisher, Kimball, *Self-Directed Work Teams*, McGraw-Hill, 1993.
5. Haeckel, Stephen H., *Adaptive Enterprise, Creating and Leading Sense-and-Response Organizations*, Harvard Business School Press, 1999.
6. Tapscott, Don, *Growing Up Digital, The Rise of the Net Generation*, McGraw-Hill, 1998.

Coming Events

20th IEEE Real-Time Systems Symposium (RTSS)

Dates: Nov. 30, 1999 — Pre-Conference Workshop
Dec. 1-3, 1999 — RTSS '99
Location: Phoenix, Ariz.
Sponsor: IEEE Computer Society
Internet: <http://www.cs.tamu.edu/conferences/rtss/>

24th Annual Software Engineering Workshop Call for Papers

Dates: Dec. 1-2, 1999
Location: Green Belt, Md.
Sponsor: NASA/Goddard Space Flight Center
Software Engineering Laboratory, University of
Maryland, and Computer Sciences Corporation.
Internet: <http://sel.gsfc.nasa.gov/sew.htm>

25th International Conference on Technology Management and Performance Evaluation of Enterprise-Wide Information Systems

Dates: Dec. 5-10, 1999
Location: Reno, Nev.
Sponsor: Computer Measurement Group
Focus: Better computing in the years beyond 2000. Every
one of you has experiences and unique perspectives in
managing enterprise technology assets.
Voice: 609-401-1700
Fax: 609-401-1708
Internet: <http://www.cmg.org/cmg99/>

17th ACM Symposium on Operating Systems Principles

Dates: Dec. 12-15, 1999
Location: Kiawah Island Resort <http://www.kiawahresort.com/>, near Charleston, S.C.
Sponsors: Association for Computing Machinery
<http://www.acm.org/>, Special Interest Group on
Operating Systems <http://www.acm.org/sigops/>
Internet: <http://www.diku.dk/sosp99/>

13th Conference on Software Engineering Education and Training (CSEEs&T)

Dates: March 6-8, 2000
Location: Austin, Texas
Theme: Software Engineering Coming of Age
Topic: Offering guidance, promoting innovation and
collaboration, and stimulating new instructional
approaches to education and training. For further
details visit: <http://www.se.cs.ttu.edu/CSEET2000>
E-mail: mengel@ttu.edu
Voice: 806-742-3527
Fax: 806-742-3519
Internet: <http://www.se.cs.ttu.edu/mengel>

12th Annual Software Technology Conference

Theme: Software and Systems — Managing Risk,
Complexity, Compatibility, and Change
Dates: April 30-May 4, 2000
Location: Salt Lake City, Utah
Co-Sponsors: U.S. Air Force, U.S. Army, U.S. Navy, U.S.
Marine Corps, Defense Information Systems Agency,
Utah State University Extension
Co-hosted by: Ogden Air Logistics Center/CC, Air Force
Software Technology Support Center
Contact: Dana Dovenbarger
Voice: 801-777-7411
Fax: 801-775-4932
E-mail: dana.dovenbarger@hill.af.mil

23rd International Conference on Software Engineering

Dates: May 2001
Location: Toronto, Canada
Sponsors: IEEE Computer Society Technical Council on
Software Engineering, Association for Computing
Machinery, ACM Special Interest Group on Software
Engineering (SIGSOFT)
E-mail: icse2001@csr.uvic.ca

Integrating Knowledge and Processes in the Learning Organization^{1,2}

Linda Levine
Software Engineering Institute

This is the first part of a two-part article. Part one explores the adaptations needed in the process movement and knowledge-creation approaches to achieve the vision of a learning organization. Part two, to be published in the January issue, will look at learning in practice by examining some frameworks and tools that pull together process, knowledge management, and technology to support learning and effective change.

Introduction

When we ask people to change — as we do in improvement or technology adoption efforts — we are asking them to learn. If you pay attention to how people learn, you will be capable of more effective change management. Learning and technology change management reinforce one another. If you are smart about working on how you manage change, you will help to make your workplace a learning organization, and that will pay off in many ways. We assert that learning organizations require not just one or two of these, but all three: process management, knowledge management, and technology.

Technology Change Management and Learning

Technology change management (TCM) is not a single isolated process. In its most robust form, TCM touches many of the socio-technical activities performed in an organization. This picture of technology change management extends beyond systematic, high-control approaches to technology adoption, encompassing the creative exploration and exploitation of technology, knowledges, and processes. The enlarged picture includes business and work processes, and technical systems — as well as processes related to group dynamics and collaboration. To realize technology change management in this far-reaching manner is, in effect, to construct a learning organization.

What About TCM in the CMM?

Rethinking of technology change management can be seen in proposed changes to the Capability Maturity Model® for Software (SW-CMM). Whereas version

1.1 defined the purpose of technology change management as “to identify new technologies (i.e. tools, methods, and processes) and transition them into the organization in an orderly manner,” Draft C of the SW-CMM has enlarged the scope for technology change management with a new name for the activity and greater breadth in the description. Here, in Draft C, the purpose of organization process and technology innovation “is to identify process and technology improvements and innovations that would measurably improve the organization’s software processes and thereby help achieve the organization’s software process improvement goals.” Organization process and technology involves “identifying, selecting, and evaluating new technologies, and incorporating effective technologies into the organization.”

The refinement of technology change management in terms of organization process and technology innovation takes a step toward a more innovative and opportunistic approach. However, much remains to be understood about this complex phenomenon — especially about how these activities are made operational. In this article, we consider some of the ingredients to making TCM a part of daily work, relating to socio-technical activities across the organization. When we use the term “technology change management,” we intend a forward-looking perspective in the spirit of current work on adaptive, flexible, self-organizing systems (an approach most likely to be realized in higher maturity organizations).

Matching Management to the Pace of Change

Learning organizations and work groups

of the 21st century must become expert in managing change in dynamic situations [1]. Older “freezing” and “refreezing” models and metaphors from organization development theory are insufficient to guide us here. Multimedia technologies and practices supporting process change, modeling, simulation, and collaborative and distributed work will be key. Skill sets in the new work force that allow for flexibility, speed, experimentation with rival hypotheses, and collective responsiveness will prevail.³

Overall, TCM represents the fusion of technology innovation and process management as it is fully defined, operationalized, and enacted in a learning organization. Are we ready for technology change management and learning organizations? What is involved? These questions form the basis of our present inquiry.

Two Movements: Process Management and Knowledge Creation

Over the last several decades, organizations that specialize in technological innovation, including technical knowledge and expertise, have experienced two important developments. These developments, emphasizing the roles of processes and knowledges, have evolved in parallel more or less separately. On one hand, the process movement has favored a trend toward establishing fairly formal business and work processes to ensure that highly technical work gets done on time, within budget, and with quality assured and customer satisfaction maintained [2, 3, 4]. In

The Capability Maturity Model for Software and SW-CMM are registered in the U.S. Patent and Trademark Office.

a sense, the emphasis on processes represents a modification and extension of principles of Taylorism and Fordism; and while more flexible, the new approach still remains largely managerial- and control-oriented [5]. We might argue that adaptive organizations need a process focus that balances discipline and innovation, an environment where many voices can be heard and exclusionary interests are resisted.

Defining work processes has made inroads into practice over the last 10 years and extends to process modeling, simulation, and automation, as understood in the context of the organization's larger business processes [3,4,5]. In parallel, a second trend in response to the coming of the knowledge society [6,7], has seen the creation of knowledge-based organizations [8] as being enhanced by integration of quickly evolving information technology [9,10].

Advances in information technology directed toward organizational knowledge and learning — dating to the 1960s — and visions like Doug Engelbart's have provided a revolutionary opportunity for information technology as a medium for facilitating and improving group communication and knowledge creation. Rapid advances have continued [9,10,11,12]. IBM (Lotus Notes), Microsoft and Netscape (intranets and extranets) are capturing a huge projected market based on the use of the World Wide Web. Products that enable conferencing and brainstorming at a distance, as well as multimedia information capture, structuring, visualization, and retrieval are available at fairly minimal costs. Such products extend Engelbart's views of the potential of computer technology even further.

Getting Process Knowledge to Work Together

To be practical, we maintain that local adjustments within both of these movements are necessary. Within the process arena, it is time to counterbalance process formalization with process creation by leveraging individual knowledge through information exchange and by reconciling diverse perspectives. An organization that supports information sharing and knowledge creation amongst its members and is

committed to including and reconciling multiple viewpoints is likely to establish effective and efficient processes as well as improve organizational life.

Within the knowledge creation arena, the challenge is to filter and channel information — without loss — for knowledge-based decision-making. Information technology has in some cases made matters worse by exacerbating information overload in the form of e-mail glut, for example, rather than realizing knowledge-based organizations in practice. Far behind technology's rapid evolution are approaches for its use in enacting knowledge-based and learning organizations.

These necessary adjustments within the process-based and knowledge-based movements pave the way for a more dramatic synthesis: technology change management depends upon integrating technology and process management for innovation and learning. To make learning organizations an everyday reality, the software community can build on its successes at putting information sharing, knowledge creation, and work processes into practice. To date, work flow research has focused on formalized business processes and process-enactment technologies. Unfortunately, this fails to take advantage of the chaotic but potentially rich communication in the workplace. Until process automation taps these information flows, and is aligned with knowledge creation technologies, its potential will be limited to quick fixes, partial solutions, and inadequately informed decision making.

We need to stimulate new communities of practice made up of people and organizations experienced in technology implementation, cooperative work (collaboration technology and practice), organizational learning, and process initiation and improvement.⁴ Most importantly, those familiar with the workings of processes in their local situations must be involved [13,14,15].

What is Organizational Learning?

Approaches to organizational learning starting in the late 1970s [16] have gener-

ated significant interest, but discussion has not yet led to widespread application. Many of these ideas have not been piloted or implemented in everyday work practices of organizations [8,17]. Nor have organizations tapped advances in information technology to create a sense of learning history or corporate memory, except in highly innovative business environments.

Researchers and practitioners have written about this for decades, and yet issues debated in the field 20 or so years ago bear a striking resemblance to those still debated today. For example, we continue to discuss distinctions between adaptive and generative learning [18], between single- and double-loop learning [16,19], between “know-how” and “know why,” and so on.

How is it that organizational learning has persisted, running parallel to the stream of fads in business and management practice? The concept has survived while a whole cadre of consultants and organizational mechanics have paraded by, selling everything from management by objectives, quality circles, total quality management, and management by results to statistical process control, business process re-engineering, business process reinvention, and more recently high performance teams, self-directed teams, empowered teams, and integrated product/process/practice teams.

Is organizational learning just a catch-all, a vessel for goals and related thinking on strategy, productivity, and innovation? Perhaps organizational learning holds true generally and the variation that we see resides in methods, techniques, and practices — the particular means for instantiating a kind of organizational learning. The question remains: Is the staying power of organizational learning in its emptiness or its elasticity? Have we advanced in this area over the last three decades or not, and if so, how or how not?

To add to the fuzziness around the concept of organizational learning, researchers and practitioners also talk about “learning organizations.” For some, the difference is captured between organizational learning, which can be taken to mean learning by individuals and groups

in the organization vs. learning organization, which emphasizes learning by the organization as a total system [15, 20]. By the organization as a total system, we mean there are systemic features to learning beyond the activities of particular individuals who may come and go. This does not mean that people are not important or needed in the learning process; rather, organizational learning is not reducible to individual learning. In this article, we use the terms interchangeably but we are concerned with the second condition — learning by the organization as a total system. We envision a learning organization as one where:

- the organization remembers and learns
- public recording is unobtrusive and useful in the execution of work processes and decision making
- principles and concepts may refer to a group, an organizational unit(s), or a community, suggesting notions of scalability and tailoring
- the notion of learning is different from the additive sum of individual contributions, $a + b + n$ (instead, the whole is more than the sum of the parts)
- learning is applied to produce or modify individual dispositions, policies, processes, and procedures

Can We Get It Into Practice?

We believe that we are poised, more ready than ever, for learning organizations. We are at a watershed — with the potential to get leverage from our intellectual investment in organizational learning, matching our interests with enabling information technology. Current thinking in the discipline of organizational learning offers guidelines for use. However, because of the limitations already noted, we emphasize that constructing learning environments requires that we apply knowledge and capability in related areas, such as processes and systems thinking, group dynamics and performance, education and distance learning, and community memory. Together, these comprise the backbone for communication and cooperative work necessary for a learning organization.

Silver Bullets and Basics

Too often, we observe a premature inclination to jump to a technological solution without paying attention to the basics. For example, development teams may be over-eager to automate processes, which have not been fully defined or used in manual operations [5]. Similarly, doing computer-supported cooperative work does not guarantee that contributors are collaborating, in the best sense of the word, or working productively as a team. These tendencies reveal wishful thinking that adding technological support will magically allow users to bypass a host of needs and constraints. The technology is seen, naively, as a silver bullet.

No technology can compensate for bad practice, nor can it substitute for an understanding of basics or fundamentals. However, experimenting with and piloting new technologies can help co-evolve fundamentals and technologies. For all of these reasons, we underscore the importance of related knowledge in several disciplines and in local practice. Initially, one may focus on the technology and thinking about systems and processes. In the end, a learning organization must reckon with good practice in teaming, education, sharing information and archiving lessons, and corporate memory — recording and analyzing decision-making and related history [21] — for recurring and problematic themes, and all in a manner that is coherent, yet streamlined and accessible.

All organizations inevitably enter this problem space from somewhere — with prior knowledge and experience in many of the areas identified. Local understanding may be fragmented or isolated but it represents organizational learning, even if that learning is somewhat sparse or dislocated. In many organizations, learning efforts have simply not been strategically conceived, or with any intent for integration. This may have occurred for any number of reasons, including the relative immaturity of the technologies in question and risks associated with the same.⁵

Working on Learning

A learning organization establishes the capability to understand its environment and culture, including its current activi-

ties and work processes to evaluate what is understood, and to initiate improvements where necessary. These metaprocesses centered on the capability to learn are both independent of and dependent on the people in an organization. This capability enables decision making and affects outcomes, representing the combined experience, expertise, and knowledge of all participants involved in a group activity. Group activities may pertain to a team, project, department, or program. In such an effort, a strong leader may be best able to summarize or express the voice of the group, but what is being expressed is the product of uninhibited information flow, analysis, and negotiation. Our goal in this article is to support organizational dialogue. All organizations are capable of learning; it remains for the organization to find its own medium and voice.

Organizations are *independent* of their members because work processes — along with associated policies, values, mechanisms, and techniques — may exist long after people have left the organization or before new people have come on board. Moreover, viable and effective processes are not dependent on extraordinary individuals to carry them out. Organizations with a strong process focus have an increased potential for democracy. By mobilizing multiple perspectives, experiences, and expertise from across an organization, and channeling these for decision making, the organization, as a whole, can monitor relevant market conditions, continuously adapting its processes to satisfy changing technical and business needs. By the same token, an organization's culture and the identity of its members is derived, in part, from these articulated processes. They determine the quality of life and the loyalty of members, and must be adjusted continuously to gain and keep the commitment of the organization's members.

Organizations are *dependent* on their members and in the free flow of ideas. These interactions form the creative source for organizational learning and are necessary conditions for the ongoing viability of the processes that are created. A number of researchers have pointed to the importance of talk and interaction as

a basis for mutual understanding [22], for narrative exchanges as the basis of learning-in-working and innovation [14], and of records, documents, visualizations, and artifacts to accomplish work and to engender shared ways of viewing the world [23,24,25,26]. Talk, stories, and documents serve a dual role — information bearing and social bonding — in single exchanges and multiple, connected instances. Most organizations will have to undergo structural and cultural changes to reap the potential benefits of talk, where members of different projects or programs contribute to the same discussion or branched threads. These changes cannot happen overnight. This is just as well, for it is unwise for organizations to attempt a unilateral shift all at once.

Tools, alone, will not create the organizational cultures, structures, and conditions that are needed for their best use. Rather, the necessary mutual adaptation of the technology and organizational processes and forms must come through trial and experimental use [27, 28].

In part two of this article, we look at technology change management in practice — by considering the role that learning plays in process models and improvement frameworks. We illustrate with several examples, including IDEALSM, a model for software process improvement, and the IDEALSM-Based New Technology Rollout (INTRo).

Technology change management can contribute to building a learning organization. But for technology change management to become part of daily work, these activities must be integrated with evolving socio-technical and business needs and processes. New communities of practice must be nurtured, including people and organizations experienced and knowledgeable in technology investigation, change management, and collaboration practice. ♦

About the Author

Linda Levine leads the effort on the IDEALSM transition framework development, aimed at extending improvement models into structured processes for technology adoption and rollout at the Software Engineering Institute (SEI). She is a process developer for INTRo, a web-based adapt-

able process guide with proven practices for introducing new technology into software-intensive organizations, co-developed by the SEI and Platinum Technology. In addition to working on the diffusion and transfer of software technologies and change management, she researches technology suppression, reasoning and communication, design disciplines, and the relationships between organizational learning and the use of collaboration technology. Dr. Levine holds a doctorate in rhetoric (communication) from Carnegie Mellon. She publishes widely and is the co-founder of the Working Group (8.6) on Transfer and Diffusion of Information Technology, part of the International Federation for Information Processing.

Software Engineering Institute,
Carnegie Mellon University,
Pittsburgh, Pa., 15213
Voice: Tel. 412-268-3893
Fax: 412-268-5758
E-mail: ll@sei.cmu.edu

References

- Levine, L. An Ecology of Resistance. In T. McMaster, E. Mumford, E. B. Swanson, B. Warboys, and D. Wastell (Eds.). *Facilitating Technology transfer through Partnership: Learning from practice and research*. IFIP TC8 WG8.6 International Working Conference on Diffusion, Adoption and Implementation of Information Technology (pp. 163-174). Ambleside, Cumbria UK, London: Chapman & Hall, 1997.
- Humphrey, W. *Managing Technical People: Innovation, teamwork, and the software process*. Reading, Mass.: Addison-Wesley, 1997.
- Heineman, G. T., Botsford, J. E., Caldiera, G., Kaiser, G. E., Kellner, M. I., Madhavji, N. H. "Emerging technologies that support a software process life cycle." *IBM Systems Journal* 33(3), 501-529, 1994.
- Curtis, B., M. Kellner, and J. Over, Process modeling. *Communications of the ACM* 35(9), 75-90, 1992.
- Christie, A. M., Levine, L., Morris, E. J., Zubrow, D., Belton, T., Proctor, L., Cordelle, D., & Ferotin, J-E. Software process automation: Experiences from the trenches. (SEI Technical Report SEI-96-TR-013). Pittsburgh, Pa.: Software Engineering Institute, 1996.
- Drucker, P. F. Post-Capitalist society. New York, N.Y.: HarperBusiness a division of Harper Collins Publishers, 1993.
- Drucker, P. F. "The Age of Social Transformation." *The Atlantic Monthly*, November 1994.
- Nonaka & Takeuchi, *The knowledge-creating company*. New York, N.Y.: Oxford University Press, 1995.
- Seybold, P. B., Office Computing Group. Doug Englebart's design for knowledge-based organizations, Part 1: Required technology: Open hyperdocument systems. *Paradigm Shift: Guide to the Information Revolution* 3(8) 1-9, 1992a.
- Seybold, P. B. Doug Englebart's design for knowledge-based organizations, Part 2: Coevolution of Organizations and Technology. *Paradigm Shift: Guide to the Information Revolution* 3(9), 1-15.1, 1992b.
- Englebart, D. & Lehtman, H. "Working Together." *Byte*, December, 245-252, 1988.
- Englebart, D. C., "Toward high-performance organizations: A strategic role for groupware." In D. Coleman (Ed.), *Proceedings of the Groupware '92 Conference* (pp. 1-24). San Jose, Calif.: Morgan Kaufmann Publishers, Inc., 1992.
- Brown, J. S. & Gray, E. S., "After re-engineering: The people are the company." *Fast Company*, Premiere issue, 78-81, 1995.
- Brown, J. S. & Duguid, P. Organizational Learning and Communities of Practice: Toward a unified view of working, learning, and innovation. *Organization Science* 2(1), 58-82, 1991.
- Schein, E., "Organizational Learning: What is New?" Working paper, MIT Sloan School of Management, 1997. On the internet at <http://learning.mit.edu/res/wp/index.html>
- Argyris, C. & Schon, D. A., *Organizational Learning II*, Reading, Mass., Addison-Wesley, 1996.
- Simon, H., Bounded rationality and organizational learning. *Organization Science* 2(1), 125-134, 1991.
- Senge, P., *The Fifth Discipline*. New York, Doubleday, 1990.
- Argyris, C., Single and double loop

- models in research on decision making. *Administrative Science Quarterly*, 21(3), 363-375, 1976.
20. Lundberg, C. C., Creating and Managing a Vanguard Organization: Design and Human Resources Lessons from Jossey-Bass. *Human Resource Management*, 30(1), 89-112, 1991.
 21. Roth, G. L., "Learning Histories: Using Documentation to Assess and Facilitate Organizational Learning." Working paper, MIT Sloan School of Management, 1997. On the Internet at <http://learning.mit.edu/res/wp/index.html>.
 22. Bannon, L. J. and K. Kuutti, "Shifting Perspectives on Organizational Memory: From storage to active remembering." In J. Nunmaker and R. Sprague (Eds.), *Proceedings of the 29th Annual Hawaii International Conference on System Sciences*, (pp. 156-167), 1996. Maui, Hawaii: IEEE.
 23. Suchman, L., *Plans and Situated Actions*. Cambridge: Cambridge University Press, 1987.
 24. Brown, J. S. and Duguid. The social life of documents. *First Monday: Peer-Reviewed Journal on the Internet*, 1996. <http://www.firstmonday.dk/issues/issue1/documents/index.html>.
 25. Schein, E., On dialogue, culture, and organizational learning. *E.M.R.* Spring, 23-29, 1995. (reprinted from *Organizational Dynamics*, Autumn 1993).
 26. Schein, E., "Three cultures of management: The key to organizational learning," *Sloan Management Review*, Fall, 9-20, 1996.
 27. Leonard-Barton, D., "Implementation as Mutual Adaptation of Technology and Organizations." *Research Policy* 17,(5) (Oct.), 102-110, 1988a.
 28. Leonard-Barton, D., "Implementation Characteristics of Organizational Innovations." *Communication Research* 15, 603-631, 1988b.

Notes

1. This work is sponsored by the Department of Defense. Some of the ideas expressed here can be found in an earlier work: see Levine, L. and Monarch, I. (1998). Collaborative technology in the learning organization: Integrating Process with Information flow, access, and interpretation. In J.F. Nunamaker, Jr. (Ed.), *Collaboration Systems and Technology Track*, vol 1., Proceedings of the 31th Annual Hawaii International Conference on System Sciences, (pp. 444-459). Mauna Lani, Hawaii: IEEE Computer Society Press, Los Alamitos, Calif.
2. The Software Engineering Institute is a federally funded research and development center sponsored by the Department of Defense.
3. In "This Organization is Dis-organization," Polly Lebarre describes the transformation at Oricon Holding A/S, a company that makes hearing aids in Denmark. Lars Kolind, CEO, reflects on how he began to share his vision for reinventing the company, as drafted in his four-page (New Year's 1990) memo. It amounted to a declaration of

disorganization. Oricon needed breakthroughs, Kolind wrote, and breakthroughs require the combination of technology with audiology, psychology, and imagination. The ability to "think the unthinkable" and make it happen. In organizations of the future, he continued, "staff would be liberated to grow, personally and professionally, to become more creative, action-oriented, and efficient." What was the enemy of these new organizations? The organization itself (*Fast Company*, June/July 1996, p. 80; italics Lebarre).

4. When we use the phrase "collaborative technology," we mean an alignment between process and information-sharing technologies.
5. At the SEI, for example, we have experimented with an empowered team and built practices for chartering teams. We have conducted experiments with groupware to characterize knowledge integration in software development. The institute's education program and master's program in software engineering employed its media studio to offer courses by satellite. Two CD-ROMs have been developed for just-in-time learning. There is interest in an environment that might incorporate the notion of shared virtual space, a library of process assets, various repositories, and support for asynchronous collaboration. Like others, we need to realize our intellectual investments and our organizational learning, if we are to become an organization that learns.

The Managing Software Innovation and Technology Change Workshop

Eileen Forrester, SEI
Priscilla Fowler, SEI
Sharon Guenterberg, Litton PRC

The Managing Software Innovation and Technology Change (MSITC) Workshop was held in June as a joint effort of the Software Engineering Institute (SEI) and Litton PRC. It was organized to determine what actions that leading organizations are taking to meet the challenges of integrating technological change strategically as well as operationally within their organizations.

Background and Objectives

The workshop convened representatives of organizations with experience in managing software-related innovation and technology change. The workshop objective was to share ideas, tested strategies, practical approaches, and analyses of lessons learned for adopting innovations and new technologies. Through working with their peers as well as invited experts, participants were to take away enhanced understanding of how to manage changes related to acquisition, adoption, and implementation of new information technologies for software-intensive products and systems.

We distributed an invitation to submit position statements for the workshop to several hundred experienced individuals. Acceptance was based on review of these position papers. All the workshop position statements are available as part of the final program. These are available online (<http://www.sei.cmu.edu/programs/te/tech-changewkshop.html>).

For the workshop's formal sections, 12 participants were asked to make short presentations for discussion during the workshop. In addition, we invited a panel of senior technical managers to address how technology change management (TCM) looks from their perspective, and asked a professor from the University of Pittsburgh, known for his expertise in managing innovation and technology, to provide input and commentary on the workshop findings. Other participants included those submitting position statements; members of the program committee; and management of the sponsoring organizations, Litton PRC and the SEI. There were 25 participants, each of whom had several years of diverse experience with TCM. The remaining time during the workshop—about half the total—was spent in working sessions. The remainder of this article gives highlights from the workshop formal program and working sessions and our plans to follow up.

Highlights from the Workshop

The formal program consisted of a senior managers' panel, position paper presentations, and commentary from the invited expert.

Senior Managers' Panel

An invited panel of five senior managers from Litton PRC, Boeing, the SEI, and the Office for Science and Technology of the Spanish Presidency responded to the following two questions that addressed how the workshop looks from a senior technical manager's perspective:

- How do you strategically manage technology?

- How do you encourage and reward innovation?

Jude Franklin, Chief Technical Officer of Litton PRC, described the TCM program and development of the Strategic Technology Plan that advises senior management on where to invest their scarce overhead funding in terms of technology. This program includes tracking and evaluating new technologies, including those that emerge from within the organization, and developing tools that help technical personnel keep abreast of these technologies. Cora Carmody, General Manager of Internal Information Systems and Chief Information Officer at Litton PRC, spoke about the need for organizations to strategically manage Information Technology (IT) and to use that as the basis for identifying transformations and establishing an organizational structure that supports the identified goals. Goals include knowledge management, IT asset management, and electronic commerce.

Steve Cross, Director of the SEI, described SEI members of the technical staff as change agents, technology scouts, and “impact amplifiers.” SEI is organized into technical initiatives, the leaders of which are like commando unit leaders. Work is focused around expediting the technology maturation process; the model for that process is based on Geoffrey Moore’s notion of “crossing the chasm.” This includes addressing the specialized needs of early technology adopters and the broad needs of mainstream markets.

Gonzalo Leon, professor of the Universidad Politecnica de Madrid, and Director of the Office for Science and Technology, Presidency of the Spanish Government, represented an international, public-sector perspective. He reflected on the need to manage technology at the level of a country. His focus was on encouraging innovation and technology change in small- to medium-size enterprises (SMEs), which constitute 97 percent of Spain’s industrial base. He stressed that research and development must be funded and carried out in the context of technical and other innovation.

John Vu, Boeing Technical Fellow and Chief Engineer of the Boeing Company, described their MSITC efforts. These include technology evaluation, based on tracking trends and forecasting in software engineering, for the entire Boeing Co. Evaluation also includes trying out the technologies in testbeds, followed by recommendations to the Software Council, and communications—making sure the organization knows that particular technologies are being investigated. Besides evaluation, Boeing uses the IDEALSM model and the Capability Maturity Model® (CMM®) as the basis for action plans, which he said are “where the technology transition happens.” Regarding how Boeing rewards innovation, Vu noted that it is re-evaluating skill codes to better reflect the importance and range of skills Boeing needs in software engineering.

In the discussion among participants and panelists that followed, three key points were made:

- SMEs are “most of the world,” while workshop participants primarily represented large organizations and those who worked as consultants or provided other services to large organizations.
- While creating an infrastructure to support SMEs and others attempting to implement MSITC was important, it was even more important to create motivation and mindset so that organizations would take advantage of the infrastructure.
- The importance of identifying MSITC functionality, independent of the size of the group or organization performing it.

In summary, the panelists found the biggest challenge in the MSITC includes selling a common vision to management, addressing the compressed time cycle of change and new technology, making technology management an accepted role, communicating how MSITC

helps an organization accomplish its mission and business goals, and fostering a culture of change in the organization itself.

Summary of Some Position Statements

Three software tools for facilitating TCM were presented.

Linda Levine of SEI on “TCM: Integrating Knowledge and Processes in the Learning Organization.” This is the first of a two-part article, which can be found in the November issue of CrossTalk. She says that the interplay of technology, knowledge, and process is integral to innovation or change. She maintains that organizations doing TCM must be learning organizations, and describes an example tool, INTRo. INTRo, based on the IDEAL model and other SEI and LBMS work, attempts to join elements of technology, process, and knowledge management.

Ron Kohler and Stan Przybylinski (Center for Electronic Commerce, Environmental Research Institute of Michigan) contributed “Tools for Knowledge Management: Fieldwork in Evaluating RAPTR, An Intelligent Groupware System.” Readiness Assessment and Planning Tool Research (RAPTR) was described as a system to support change teams in planning and executing reengineering projects—or an electronic tutor for change agents. The paper also presented the results of using anthropological techniques to learn about intended users and operators of RAPTR.

Mike Lefler of Litton PRC presented a description of a third tool in his paper, “Tek*Aware: A Litton PRC Project for Technology Monitoring.” Tek*Aware, still in the experimental stages, is used by Litton PRC technical staff and engineers to monitor technology, gather and filter promising technologies, and help match technology information to known user needs.

Papers by Suzanne Garcia of aimware (“Managing Technology and Innovation”) and Sharon Guenterberg of Litton PRC (“Litton PRC’s TCM Program: The Continuing Quest for Aligning People, Technology and Strategy”) reflect the experiences of a very small and a very large organization in practicing what they preach. Both organizations have had success in using TCM methods with their customers that were originally developed for internal use. Both papers have lessons learned on techniques for fostering innovation and a learning environment.

Donald V. Dortenzo and Michele Nimerick submitted “Technology Innovation at Higher Levels of Process Maturity.” It reflects their work at the Software Productivity Consortium to help member organizations to be successful with TCM at higher CMM levels. They noted the need for quantifiable goals driven by business needs and reported on pilot efforts to test TCM guidance in an operational environment.

Stan Rifkin, Mark Paulk, Mac Patrick, and Lewis Gray challenged our assumptions about TCM. Rifkin (Master Systems) contributed “Discipline of Market Leaders and Other Impediments to Implementing Software Process Improvement.” He asserts that organizations need to determine their priority with respect to market strategy: are they focused on operational excellence, customer intimacy, or product innovation? He points out that TCM efforts must be aligned to the chosen market strategy, just as process improvement must align with business goals.

Paulk, of SEI, submitted “Analysis Tools for Different Perspectives on Process and Technology Change Management, which can be found in the November issue of CrossTalk. He offers three perspectives that may be of value when considering change management:

- internally driven change vs. externally driven change
- change directed at products and services vs. those directed at design and production processes, and

- incremental vs. revolutionary change. His paper highlighted work from Geoffrey Moore's *Crossing the Chasm*, Fichman and Kemerer's assimilation gap, and the Daghfous and White's innovation analysis model.

Patrick, of Visa International and Process Advantage Technology, said that "new technology alters our 'sensorium'—the ratio among our senses" and that people adopt technology "when they are ready and the technology is ready, or 'whole'—in their terms."

Gray, of Abelia Corp., suggests that we need requirements for technology change, and the free concurrence of the intended adopters of change, rather than simply setting criteria that allow us to declare success independent of the experience of intended users of technology.

Reed Augliere, of The Future Research Co., submitted a paper on using simulation technology to quantify the benefits and mitigate the risks of proposed technology changes. The technique was used to help a banking consortium evaluate and choose among alternative technologies to support a new electronic commerce system. It is a real-world example of the challenges associated with IT infrastructure planning and an example of one practical approach to TCM.

While there was collective breadth in the position statements, there were some obvious gaps in the total TCM picture they represented. Little on managing individuals or teams for innovation was presented, although it is implicit in the work of several participants. In addition, most of the contributions were strictly focused on the software-intensive industry and failed to take account of existing work in other disciplines and industries on managing innovation and TCM. A valuable exception was the participation of an anthropologist in the development of RAPTR.

Despite these omissions, the group of papers represents an attempt to wrestle with a complex set of factors, conditions, and inputs to change and innovation. No silver bullets were proposed; the working sessions and follow-on actions to the workshop were the participants' commitment to identifying and sharing good TCM practice.

Dr. George White's Role as Commentator

Dr. George White, of the University of Pittsburgh, was served as an expert witness to the workshop activities. Based on his response to the presentations, working sessions, and other discussions, White offered commentary and informal critiques throughout the workshop and formal critiques during the closing session. He cited a position paper submitted by Rich Bland of Litton PRC when he stressed the importance of solving the problems associated with updating existing legacy systems that have been in place for the last 50 years with newer, more supportable technology. He also cautioned that the devil is in the details in terms of managing software innovation and technology change, as cited by Priscilla Fowler of the SEI in her position paper. Finally, he underscored the importance of learning as encapsulated in Levine's paper, and emphasized the value of having true social scientists, such as anthropologist Kohler, to give software practitioners fresh insight on TCM.

Highlights from the Working Group Sessions

There were six working group sessions, some in small groups, some as plenary sessions.

Session Commenting on TCM-related CMMI Key Process Areas

This working session reviewed key source materials related to TCM for CMMI project -- particularly, the SW-CMM V2 Draft C materials for Organization and Process Technology Innovation (OPTI) and Organizational Improvement Management (OIM). The goal was to apply

the insights of active researchers and practitioners in the area of managing technology and innovation to the expression of technology and innovation management concepts and practices in CMMs. Change requests based on this session have been forwarded to the CMMI architects. The findings focused on the distinctions between managing large, discontinuous innovations and incremental improvements, clarifying terminology, and considering the architecture of the process areas for greatest effectiveness in managing innovation and technology change.

Session Drafting a TCM Position Statement

This working group represented a diversity of disciplinary perspectives, including anthropology, rhetoric and communications, computer science, software engineering, telematics, marketing, and machine learning. Each individual had extensive experience in TCM work, and had approached it from his or her disciplinary perspective. This group created a position statement on TCM to begin to define the problem the workshop grappled with. The statement follows:

TCM is essential to the emerging world technological order, but planned technological change will fail if the people who have to live with it reject it. For TCM to succeed, TCM practitioners need to take a holistic—or “system of systems”—approach to TCM. In particular, they need to recognize that different usage domains produce different instantiations of TCM. It should be the business of the workshop to develop the domains of usage and domain-specific risks to the successful instantiation of TCM.

Participants indicated that by “domain” they meant a specific technical area, such as xerigraphy, and by “usage domain” they meant this technical area particularized for a common culture or similar set of users, such as those in the United States or Latin America.

Session Drafting a High-Level Procedural Approach to TCM

This group’s goal was to identify factors to consider in implementing a new technology. The factors they identified were placed in a quasi-procedural framework. Key factors identified were:

1. noting the existence of triggers for consideration of new technologies, such as senior management special interests or customer requirements
2. assigning ownership for implementation, so that the leader and the sponsor of the effort is clearly identified and initial resources are provided
3. identifying the stakeholders for the new technology, and the need for using a defined methodology to accomplish this, such as simulation of the outcome, quality function deployment, or an anthropological approach
4. scoping and validating the assumptions behind the technology implementation, to be sure that the technology should be implemented, and if so where and how broadly within the organization
5. performing a detailed analysis prior to implementing the technology in specific situations, attending to the maturity of the technology, trade studies, and results from piloting
6. selecting a TCM model for implementing the technology, such as IDEAL, simulation, piloting, or evolutionary spiral process
7. developing a deployment plan, based on considerations of “whole product” issues, and deployment and transition mechanisms such as training and education, intraorganizational communications, and web material
8. getting senior management approval of the specific deployment plan and allocation of resources needed. Note that this is different from obtaining initial sponsorship, which supports and encourages work preparatory to deployment.
9. deploying the technology according to the plan developed, and revising the plan as results and feedback require

10. closing out the deployment, including “acceptance testing” of the new technology—
determining the success of the implementation (the TCM process) and the technology

***Session Considering Requirements, Next Steps
to Improve Understanding and Practice of TCM***

As with many such workshops, workshop participants were concerned that both the enthusiasm and findings would be carried beyond the workshop. They brainstormed ideas for how to accomplish this and move progress in TCM forward. Their ideas were classified into five categories:

1. Building a community of practice extending TCM
Examples: use of a collaborative work site, panel and birds-of-a-feather sessions at upcoming events, active information sharing, and email dialogue
2. Short-term action items
Examples: document results of workshop, draft a case study of good TCM practice such as Litton PRC
3. Basic research (information gathering such as from the library)
Examples: catalogue competing models and relative merits, articulation of TCM assumptions
4. Applied research and development (field work, with results analyzed quantitatively)
Examples: develop and pilot application criteria for TCM models, develop domains of use
5. CMM-related
Examples: describe TCM actions for lower levels of maturity, develop metrics for TCM effectiveness

Session Creating a “Mind Map” for TCM

A session was conducted to identify, graphically document, and begin to categorize the many activities involved in successful TCM practice.

Session Creating a Graphical Metaphor (Mural) for TCM

Many metaphors surfaced during the “mind map” session on what it was like to implement software innovation or technology change within an organization. These metaphors were captured and depicted in a TCM mural.

Followup to the Workshop

Ongoing Participation

Participants agreed that further dialogue was important. In the closing session, the suggestions of the “next steps” working group were endorsed. One immediate task -- already completed -- was establishing an online, web-based working group using the shareware tool Basic Support for Cooperative Work (BSCW). All workshop participants have been incorporated as members under BSCW, and can contribute articles and bibliographies, start and participate in discussion threads, and invite nonworkshop participants as members to expand the group.

***Plans to Publicize the Workshop Results,
Recruit Broader Participation***

Immediate plans for publicizing the MSITC workshop findings more broadly include organizing a panel discussion at the SEPG 2000 Conference. An SEI technical report will expand on this

paper and include all the position statements as well as more details of working group findings. This technical report also will be created as a web site, with additional materials and the opportunity for broader dialogue with others interested in TCM related to software and IT. More workshops will be scheduled once this initial work is well under way. Meanwhile, we solicit your feedback on this paper, on the findings it reports, and on general or specific TCM issues. Write to ecf@sei.cmu.edu, pjf@sei.cmu.edu, or guenterberg_sharon@prc.com; also let us know if you are interested in participating as this work progresses.

computer systems has its own set of computers that provide it critical data.

The faults that propagate may not be easy to identify or overcome. Should a program fail by crashing or producing obviously incorrect data, most computer systems will survive — albeit in a degraded state. With this type of failure it will be obvious that there is a problem and contingency plans can be executed. The program may eventually become unusable because its data has aged, but this can be overcome during recovery operations. However, should the failure be subtler, recovery may be impossible. Inaccurate data can get into the database of an information system and spread to other computer systems. Recovery by backing out the incorrect information will not be possible after a relatively short time.

Figure 1 illustrates a simple example by representing a system of computer systems as a graph. A mission-critical computer system (MC) is in the center of the diagram. The interfaces are indicated by the edges. The original fault occurs in the node with the diagonal lines. The defect propagates through the interfaces represented by solid lines until it reaches the mission-critical computer system causing a failure. The original failure occurs so far away from the mission critical system that it is not easy to identify the risk this posed to the mission-critical system.

The U.S. Army Software Development Center-Washington (SDC-W) is responsible for 29 Army computer systems. We have developed a two-step process to help assess the risk of propagating Y2K defects to support SDC-W's Y2K. The first step is to model the system by following a definable, repeatable process. The second step is to feed the model into a discrete event simulator. This results in a better understanding of the risk posed by the interconnected nature of computer systems.

The process is iterative. As we analyze the results, we identify computer systems and interfaces that present the most risk. This risk is usually associated with a lack of knowledge. By putting more effort into understanding the Y2K behavior of those systems we can sometimes reduce the risk associated with those computer systems. In other cases we can increase the repair effort, or as a last resort we can begin contingency planning. We continually update our model as we learn more about the computer systems involved.

MODELING THE SYSTEM

The initial step of the risk assessment methodology is to model the system. The main concept is to break down the system into smaller parts. We use a definable, repeatable process to assess the relative risk associated with those parts. Because we have some consistency in the risk assessment of the parts, we can recompose the risk and have a meaningful understanding of the relative defect propagation risk.

Each computer system is considered a *black box*. We are interested in whether or not it will fail. We are not concerned with failure modes, or failure times. In our abstraction, a computer system can fail in two ways: on its own, or due to a defect passed to it through an interface.

A program is modeled by three primary factors: an overall probability of failure, the impact of a failure, and the criticality of the system to the Army. The probability of failure is not an absolute assessment, but an assessment relative to the other computer systems. An initial assessment of the probability for the computer system is adjusted by factors derived from an assessment of risk factors common to other systems. The methodology explains the process through which we derive these numbers. The impact concerns the impact on that specific system. Given that there is a failure, this indicates the expected severity of the fault. Conversely, the criticality factor concerns the impact on the system. This factor models the importance of this program to the correct functioning of the system of computer systems being assessed. We use the methodology described in Risk Assessment Methodology described below to determine the probability of failure, impact of failure on the computer system, and impact on the Army's mission.

Each individual computer system has interfaces with other computer systems. We model these interfaces simplistically. Each interface is unidirectional and includes all information and defects passed from the source to the sink computer system. Again we follow the methodology, described below, to determine the probability that given the source computer system fails, it also will cause the sink to fail. This probability is relative as it is for the computer system assessment.

The output of this activity is a directed graph. The nodes of the graph represent computer systems while the edges represent interfaces. For simplicity edges are unidirectional. Two edges are used to model the interface for cases where two computer systems pass information to each other.

Risk Assessment Methodology

Our risk assessment methodology consists of three steps which are performed concurrently. While working on one computer system we discover information about another and include it in all appropriate assessments.

1.1 Identifying Computer Systems

The first step is to identify the computer systems to model. As a minimum, all the computer systems that share direct interfaces with mission-critical systems must be modeled. Determining how many links away from the mission critical system is a trade-off between cost and knowledge. It can take days to collect and assess all the information concerning a computer system that is outside the control of the SDC-W, so weighing the effort against the benefit is necessary.

A necessary part of identifying the computer systems is identifying the interfaces to model for each computer system. Some computer systems have hundreds of interfaces and once again decisions have to be made concerning cost vs. the benefit of analyzing an interface.

1.2 Analyzing Computer Systems

The risk assessment of computer systems includes a base determination of the relative failure probability and four adjustment factors. After the base is adjusted, the result is a final relative probability of failure. The amount of information that we have on the computer systems we model varies from very detailed (for the computer systems the SDC-W manages) to very general. Our use of a definable and repeatable process gives some consistency to the results. Our model allows us to incorporate this wide variance in detail into single abstraction.

1.2.1 Base Probability of Failure

The analyst looks at many factors to generate an assessment of the system's base probability of failure. These factors include organizational, process and product factors.

The support of the functional proponent for Y2K work is critical. Functional proponents who prioritize new work over Y2K repairs add risk to a project. The reputation of the developer also plays a factor. An organization that has trouble producing quality code will also probably have trouble making Y2K repairs.

A key process factor is the Y2K project progress, with respect to where it should be at any given time. In addition, the types of tests run give a very good indication of the risk. Standalone tests with synthetic data are useful, but live end-to-end tests provide a high degree of assurance that computer systems are Y2K compliant. Since there is a great deal of product uncertainty until January 1, 2000, process plays a major part of the risk assessment.

Looking at the product itself is difficult. In most cases we do not own the product or have access to it. One indirect measure that we use is whether or not the developer has access to the complete source code and documentation. Proper assessment, testing, and repair requires an understanding of the software that is difficult without source code and documentation. Engineers must work much less precisely in a black-box environment. We also look at the repair, both the technique used and the extent of the repairs. A system written from the start to be Y2K compliant carries far less risk than one which undergoes extensive repairs. We have also determined that there is a different risk associated with computer systems that are repaired by altering their internal logic when compared with computer systems that employ wrappers or similar techniques.

1.2.2 Adjustment Factors

These factors are used to adjust the base probability of failure if there is a risk to the system identified in this category. Items that fall into these categories play a part in the assessment of numerous systems and are kept in a central database.

Commercial-Off-the-Shelf (COTS) Software

This includes all the software components that the computer system relies on in its run-time environment. The most obvious and universal is the operating system. Different versions and patches are tracked in terms of their relative Y2K risk. Database management systems and other system software that comprise the computer system are included in this assessment.

COTS Hardware

This category is similar to the COTS Software category, except it includes the hardware components of a computer system. The actual box or boxes on which the software executes is the primary example. However, we also include the peripherals that the computer system requires to perform its function. Many computer systems rely on a large set of devices, such as uninterruptable power supplies (UPS), and automated tape back-up systems. The risk associated with these devices is

incorporated here.

Infrastructure

This includes the critical parts of the infrastructure that must be available for the computer system to function. The most obvious examples are the networks the systems rely on to transfer data. This category can also be used to delve into great detail about the infrastructure. Items such as the power grid and building security access systems also pose a risk to a program's function. We have considered this level of detail, but did not use it in our assessments because the increased fidelity does not increase the accuracy.

Application Portfolio

This category includes components of the application software that are common to more than one computer system. Programming languages and libraries are part of this category.

1.2.3 Impact

To determine impact we look at the importance of data-related calculations to the functions of the computer system. Our definition of impact is confined to the computer system we are analyzing, and not its interfaces to other systems. We take a simple view of the impact of a failure on the system. We describe the impact as low, medium, or high. We arrived at this abstraction after many experiments with more detailed descriptions of the impact. We found that higher fidelity descriptions of impact did not provide us better quality risk assessments.

1.2.4 Criticality Factor

In many risk assessment models the criticality factor would be rolled into the impact. However, since we are specifically looking at the risk associated with the interconnected nature of the computer systems we broke the category out of impact. We determine criticality factor by comparing the Army's functional area core business processes with the business processes performed by the computer system. The criticality factor considers the importance of the computer system to the Army's mission.

1.3 Assessing Interfaces

The third part of our assessment is examining the interfaces between computer systems. After experimenting with much more detailed models we found that we could maintain an acceptable level of accuracy by describing an interface with one number. That number is a relative probability that given a failure of the source computer system the sink computer system will also fail. We base this assessment on the existence and quality of the System Interface Agreements(SIA)/Memorandums of Agreement(MOA) between the two responsible parties. The SIA/MOAs also tell us the type of information passed across the interface. We use this to assess the sensitivity of the interface to data-related errors. We also include the level of testing of the interface. A tested interface is better than one that has not been exercised. Actual data is better than synthetic data, but nothing tells us more than a live test.

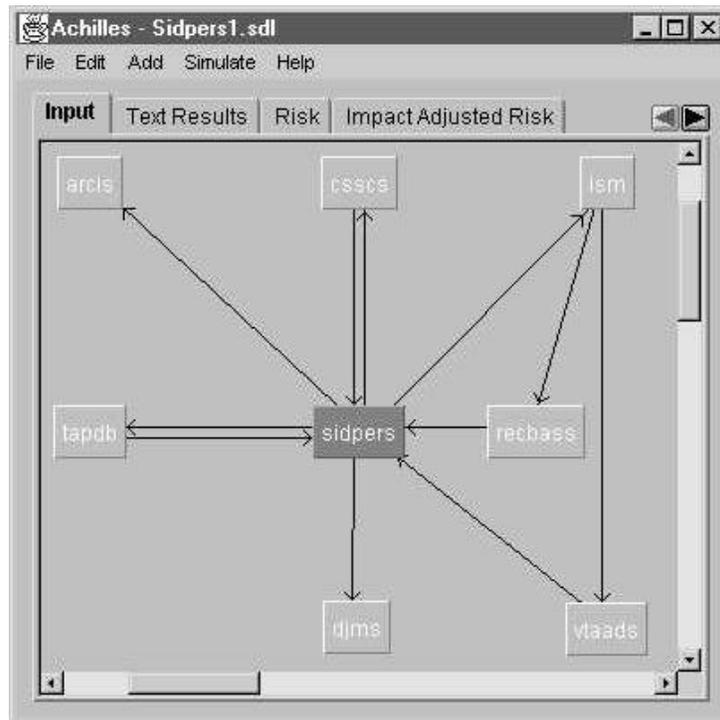


Figure 2: Achilles Input Screen for a simple example.

ACHILLES

Achilles is the discrete event simulator used to combine the behaviors of the system components. It uses a Graphical User Interface (GUI) to collect the assessment results and create a model. We then use Achilles to simulate the system interactions and analyze the results. We can look at the results both graphically and textually.

2.1 Creating the Model

Computer systems are added to the model by "pointing and clicking." The user chooses the location to minimize crossing interfaces and maximize clarity in the final model. Figure 2 shows the input GUI with a simple notional system model.

The screenshot shows a "Program Dialog" window for the system "sidpers". It contains the following fields and options:

- Name: sidpers
- Primary:
- Impact: High, Medium, Low
- Criticality Factor: 8
- Inventory Results:

0000 System	0.001
1000 COTS SW	0.000
2000 COTS HW	0.000
3000 Infrastructure	0.100
6000 Application Portfolio	0.000
Adjusted Probability	0.001
- OK button

Figure 3: Computer System Model Input Dialog Notional Example

When a user identifies the computer system location, he enters or modifies the modeling data through a dialog box (Figure 3). The dialog box allows the designation of the computer system name. This is where the user designates the system as mission critical (primary), enters the criticality factor and the impact. The base failure probability and adjustment factors also go here.

Adding interfaces to analyze is done in a similar way. The user indicates the source and sink computer systems using the mouse. A dialog box for entry of failure probability is presented in Figure 4. Once the model is complete the simulation is executed and the results can be analyzed.

2.2 Results

Achilles provides both textual and graphical output. The textual output offers detailed insight into the behavior of the system, while the graphical output gives a high-level portrait of the defect propagation risk.

2.2.1 Graphical Results

The graphical results take the same form as the input, the difference being that they are color coded to better illustrate the risk (Figure 4). The SDC-Washington codes Y2K computer system risk using a simple "red," "yellow," and "green" status which we use for Achilles output. Achilles displays the computer systems and interfaces using this same scheme.

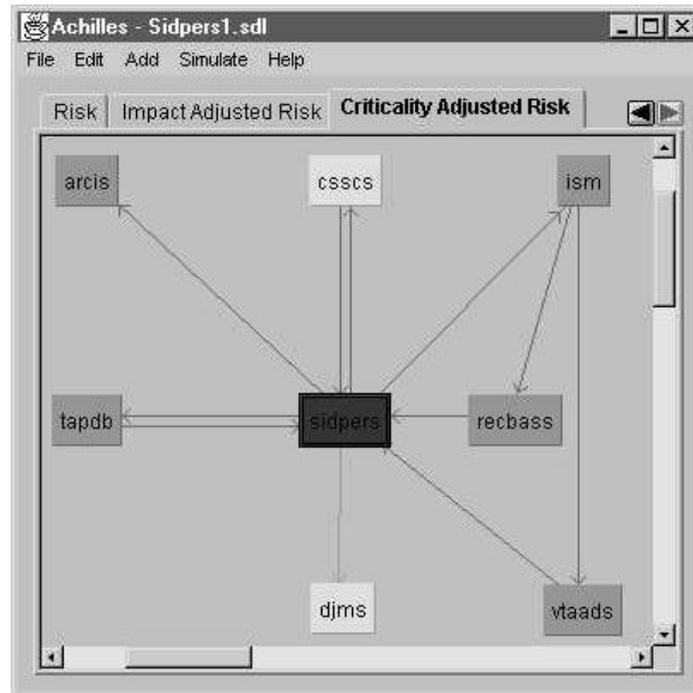


Figure 4: Example Notional Graphical Risk Results

2.2.2 Text Results

Figure 5 shows an example of the output provided for each computer system in the model. After the computer system identification the relative, impact-adjusted and criticality adjusted risk of a Y2K failure is shown. The rest of the output shows a breakdown of the relative risk from each adjustment factor as well as the probability that a failure will be caused by a defect passed through an interface.

```

Program ARCIS
    0.691 risk
    0.691 impact adjusted risk
    0.469 critical adjusted risk
5000 Interfaced System failure rate:      0.423
1000 COTS H/W failure rate:                0.000
2000 COTS S/W failure rate:                0.000
5000 Infrastructure failure rate:          0.041
6000 Application Portfolio failure rate:   0.000

```

Propagated Defect failure rate:	0.227
---------------------------------	-------

Interface: SIDPERS->ARCIS 0.227 risk
Interface: SIDPERS ->CSSCS 0.275 risk

Figure 5: Example Notional Detailed Risk Results

For each interface in the model, output similar to that shown in Figure 6 is presented. This indicates which interfaces have the most potential to pass defects that cause failures.

CONCLUSIONS

The Y2K problem for an individual computer system is not a technically difficult problem. The tough issues in implementing single system solutions are mostly project management dilemmas. As difficult as those problems may be for computer systems in isolation, the difficulty increases dramatically as the number of interfaces increases. DoD systems typically have interfaces that number in the hundreds and estimating the risk posed by those interfaces is a daunting task.

Knowing the risk to mission-critical computer systems is important to three critical areas of work now on-going: assessment, contingency planning and recovery. DoD computer systems cannot be assessed in isolation. Due to the uncertainty involving other systems, risk management techniques that take into account the interconnected nature of the computer systems must be used. Simulation is a powerful tool in this respect.

To deal with the Y2K problem we needed a definable, repeatable process to assess Y2K risk and use risk in Y2K project planning. The driving consideration to using simulation as opposed to numerical analysis was the effort required to assess the risk. Because of the dynamic nature of the Y2K risk understanding, we also needed a model that was adaptable. We found our first attempts to create a model too detailed. Our higher fidelity model was not showing a corresponding higher accuracy result and the cost increase was large. We found that the simplified model presented in this paper to be much more practical.

ACKNOWLEDGEMENTS

We would like to thank the Y2K Project Team at USA CECOM Software Development Center-Washington for assistance and funding.

ACHILLES TOOL AVAILABILITY

The tool is available to anyone by contacting LTC Welch at dd2354@usma.edu

ABOUT THE AUTHORS

Lt. Col. Welch is an associate professor of computer science at the U.S. Military Academy. He teaches software engineering, and has been an Army software engineer. His military assignments include Infantry and Special Missions units. His current research interests include dynamic reconfiguration, software engineering, managing the risk from Y2K failures, and distributed simulation. He received his bachelor of science degree from USMA, a master's degree in computer science from California Polytechnic State University, and his doctorate from the University of Maryland.

Col. Kevin J. Greaney is the Commander of the U.S. Army Software Development Center - Washington. He has extensive experience in information technology and systems acquisition from the tactical to the strategic level. He has served in information technology positions with Special Operations Command, 1st Special Forces Operation Detachment - Delta, the U. S. Army War College and the 82nd Airborne Division. He is Acquisition Corps Level III certified in Program Management and Computers-Communications.

REFERENCES

Directorate of Information Systems Command Control Communications and Computing. "U.S. Army Year 2000 (Y2K) Action Plan, Revision II, Proactive Actions Supporting Force XXI and the Department of the Army C4I Technical Architecture." 1998.

Directorate of Information Systems Command Control Communications and Computing. "Army Year 2000 (Y2K) Action Compliance Certification Checklist, Revision II." January 1998.

General Accounting Office, "Year 2000 Computing Crisis: Business Continuity and Contingency Planning." (August). Available at <http://www.gao.gov/y2kr/GAO/AIMD-10.1.19.htm> 1998.

General Accounting Office, "Year 2000 Computing Crisis: An Assessment Guide." (September). Available at <http://www.gao.gov/y2kr/GAO/AIMD-10.1.14>, 1998.

General Accounting Office, "Report to Congressional Requesters: Defense Computers Year 2000 Problems Threaten DoD Operations." (April). Available at <http://www.gao.gov/y2kr/AIMD-98-72.htm>, 1998.

Evan, James R. and David L. Olson. *Introduction to Simulation and Risk Analysis*. Prentice-Hall, New York. 1998.

Kappelman, Leon, Darlu Fent, Kellie Keeling and Victor Prybutok. "Calculating the Cost of Year 2000 – Compliance." *Communications of the ACM*, Vol 41, No. 2, February 1998.

Khoshgoftaar, Taghi, Edward Allen, Robert Haltreed, Gary Trio, and Ronald Flass. "Using Project History to Predict Software Quality." *IEEE Computer*, Vol. 31. No. 4, April 1998.

Perry, William E. "Effective Methods for Testing Year 2000 Compliance." *Crosstalk*, Vol 12, No. 1 January 1999: 16-18.

Pressman, Roger, *Software Engineering A Practitioner's Approach*. McGraw-Hill. New York. 1995.

Rubin, Howard. "Bracing for the Millennium." *IEEE Computer*, Vol. 32, no. 1 January 1999:51-56.

About the Authors

Donald J. Welch

Department of Electrical Engineering and Computer Science

United States Military Academy

West Point, N.Y. 10996

Voice: +1 914 938-2858

E-mail: Donald-Welch@usma.edu

Kevin Greaney

U.S. Army Information Systems Software Development Center – Washington

4035 Ridgetop Road

Fairfax, Va.

Voice: +1 703 275-9552

E-mail: greaneyk@fairfax-emh1.army.mil



Determining the Completed Effort of Adapting Existing Software

Ronald Cobb, Linda Smeraglinolo, and Dave Wood
Harris Corporation/GCSD

This paper describes a method for validating the estimated equivalent number of delivered source instructions determined using the Boehm adaptation adjustment factor.

THE METHOD FOR ESTIMATING effort to adapt existing software using the adaptation adjustment factor will become more prevalent as software developers turn increasingly to reuse as a way to cut schedule and development costs. To determine the accuracy of our estimates, we must develop a method to assess completed projects, which includes reused, modified, and new source code, and compare this to our original estimation. This paper proposes a simple method for such an assessment.

Due to the nature of modified code, the calculations in this paper are made in physical, rather than logical source, instructions.

Background

Determining the effort and schedule of proposed software work generally involves estimating the total source lines of code to be developed, then using a cost estimation tool such as Constructive Cost Mode (COCOMO) or REVIC. Estimating software development costs with reuse, as proposed by Barry Boehm [1], involves arriving at an equivalent number of delivered source instructions (EDSI). EDSI is the amount of source instructions which could be developed in the time required to adapt the reused software to meet current requirements. The formula uses the percent of design modification (DM), the percent of code modification (CM), and the percent of integration expected (IM). The EDSI is calculated by using an adaptation adjustment factor (AAF) as follows:

$$AAF = 0.4(DM) + 0.3(CM) + 0.3(IM)$$

$$EDSI = (\text{reused code}) * (AAF/100) + (\text{new code})$$

For example: Reusing 1,000 lines of existing code with 50 percent design modification, 50 percent code modification, and 50 percent integration with 1,000 new lines of code would be equivalent to developing 1,500 new lines of code.

Assessment Method

The method we use for determining the EDSI when a project is completed requires four data points. The first two values are easily determined with a code counter. They are the amount of the original code (OC) and the amount of the delivered code (DC). The next two values are determined by running a UNIX diff command on each file in the original code with the same file in the delivered code. Both files must be processed to remove comments. The output from the diff command is fil-

tered for lines removed and lines added and piped to a saved file. For example:

```
diff /old/main.c /new/main.c | grep '<' > /removed/main.c
diff /old/main.c /new/main.c | grep '>' > /added/main.c
```

The files in the 'removed' directory contain the lines of code not in the new files (either deleted or changed), and the files in the 'added' directory contain the lines of code not in the old files (either added or changed). The total number of lines in all the files in the 'removed' directory (REM) and the total number of lines in all the files in the 'added' directory (ADD) are used to determine the AAF as follows:

- The actual new code is the difference between the delivered code and the original code: $ACT_NEW = DC - OC$
- The actual modified code is the difference between the 'added' directory and actual new code:
 $ACT_MOD = ADD - ACT_NEW$
- The actual modified code is the difference between the 'removed' directory and actual changed code:
 $ACT_REMOVED = REM - ACT_MOD$
- The percent of code modification is the actual modified code plus the actual removed code, divided by the amount of the original code: $CM = (ACT_MOD + ACT_REMOVED) / OC$
- Since we have no accurate method for determining the amount of design modification, we set it equal to the amount of code modification: $DM = CM$
- The amount of integration is set to the percentage of original test cases rerun: $IM = (\text{rerun tests} / \text{total original tests})$.

The AAF is then applied to determine the EDSI:

- The actual reused code is a product of the AAF and the original code: $ACT_REUSED = AAF * OC$
- The EDSI is the sum of the actual new code and the actual reused code. $EDSI = ACT_NEW + ACT_REUSED$

For example: A software system contains 45,000 lines of code before enhancements (OC) and 63,000 after (DC). The total source lines in the 'removed' directory is 16,000 (REM), and the total source lines in the 'added' directory is 28,000 (ADD). If all of the original test cases are rerun (that is $IM = 100$ percent), then:
 $ACT_NEW = DC - OC = 63,000 - 45,000 = 18,000$

$$\text{ACT_MOD} = 28,000 - 18,000 = 10,000$$

$$\text{ACT_REMOVED} = 16,000 - 10,000 = 6,000$$

$$\text{CM} = (10,000 + 6,000) / 45,000 = 35.5 \text{ percent}$$

$$\text{DM} = 35.5 \text{ percent } \{\text{same value as CM}\}$$

$$\text{IM} = 100 \text{ percent}$$

$$\text{AAF} = 0.4(\text{DM}) + 0.3(\text{CM}) + 0.3(\text{IM}) = 0.4(35.5) + 0.3(35.5) + 0.3(100) = 54.85$$

{0.4,0.3,0.3 are the coefficients suggested by Boehm for DM, CM, and IM}

$$\text{ACT_REUSED} = (\text{AAF} / 100) * \text{OC} = 24,682 = (54.85/100) * 45,000 = 24,682$$

$$\text{EDSI} = \text{ACT_NEW} + \text{ACT_REUSED} = 18,000 + 24,682 = 42,682$$

Conclusion

Using a method such as this allows software project engineers to verify the accuracy of their estimates by comparing the planned effort against actual performance. As stated in the introduction, these calculations are for physical source instructions. If metrics are kept in logical source instructions, a ratio of physical to logical source instructions can be provided by most code counters. This ratio can then be used as a multiplier on the physical source instructions to determine the logical. ♦

About the Authors



Ronald Cobb is a software engineer with the Harris Corporation Government Communication Systems Division (GCSD). He has more than 14 years experience in software development, software project leadership, and software process improvement. Cobb was formerly the manager of the GCSD software engineering process group, a position he held for three years. He has a bachelor's degree in computer science from the University of Colorado, Boulder, and a master's degree in business administration with a technical leadership focus from Florida Tech.

Harris Corp.
PO Box 9100 (MS: 5-5872)
Melbourne, Fla. 32902
Voice: 407-726-1239
Fax: 407-727-4555
E-mail: rcobb01@harris.com



Linda Smeraglinolo is a member of the engineering process group (EPG) at Harris Corp. She has 17 years of software experience, including three years in process improvement. She has held a variety of positions on numerous software projects prior to joining the EPG. She has a bachelor's degree in computer science from Rollins College and a master's degree in business administration from Florida Tech.

Harris Corp.
PO Box 9100 (MS: 5-5872)
Melbourne, Fla. 32902
Voice: 407-727-6926
Fax: 407-727-4555
E-mail: lsmeragl@harris.com



Dave Wood is a software engineer with the Harris Corporation Government Communication Systems Division (GCSD). He has 12 years experience as a software developer, software tester, and software instructor in object-oriented and C++ development. He has a bachelor of science degree in computer science from the University of North Florida, and a master of science degree in computer science with emphasis in Artificial Intelligence from Florida Tech.

Harris Corp.
P.O. Box 9100 (MS: 5-5872)
Melbourne, Fla. 32902
Voice: 407-729-7885
Fax: 407-729-7325
E-mail: dwood@harris.com

Reference

1. Boehm, B. *Software Engineering Economics*, Prentice-Hall, 1981.

Validating Software Requirements

John K. Sharp
Sharp Informatics

Requirements validation must convincingly establish that the design meets actual requirements of the process being supported. This type of validation ensures that the software meets the form, fit, and function to solve the problem that is known to the subject area experts. Although validation of requirements must be considered a technical effort, the individuals with the real world knowledge (i.e. subject matter experts) are seldom skilled enough to read, much less approve, the technical models that specify the design. The challenge for requirements validation is to explain the technical specification in such a way as to make the subject matter experts capable of precisely validating the design so that they can be held accountable for the resulting application. This article will present an approach for validating the requirements contained in an information model.

GRAPHICAL INFORMATION MODEL DIAGRAMS contain a condensed form of the subject area knowledge. This presentation of knowledge in a formal syntax of knowledge benefits the communication among analysts and implementers, but it usually creates a barrier between the information technology (IT) professionals and the subject matter experts. This barrier is unique in the world of engineering because it is one of the few times that the person with the knowledge (e.g. subject matter expert) is not responsible for the resulting product. Would management allow the manufacturing of a prototype part without having the most knowledgeable person (subject matter expert who understands form, fit, and function) being accountable for the resulting part?

Current Problems in Validating Software Requirements

All too often, validation of requirements is left to the analyst, who by learning about the subject area/technology under investigation becomes the de-facto expert. This absolute reliance on the analyst also allows the implementer (who may also be the analyst) to depart from the design as the system is built and modified because the subject matter expert is only interested in the result and not the design. The resultant product may function correctly, but have an out-of-date design specification. This complicates maintenance.

Software tools support the implementation side of application development. They document the results of the analysis and significantly reduce the time required for coding. These tools improve productivity but they assume that the specified design meets the requirements. If this assumption is not valid, the software tools speed up the production of nonfunctional software.

Studies of the software life cycle have found that 45 to 60 percent of all errors in resulting applications come from errors in the design [1]. Rework during the implementation phase of the life cycle is also impacted because of discovered design errors. Errors in the resulting application lead to an unsuccessful project. Maintenance takes over and the application must be reworked or abandoned. Furthermore, a graphical design may not be complete and the implementer is forced to talk to the

subject matter expert as well as the analyst and others to determine what should be delivered. Without requirements validation being included in the software life cycle, all of the participants could have done the right job and the project could still be unsuccessful. A design containing errors can be wonderfully implemented in an impressive application without being of any value to the user. All of this state-of-the-art technical work is useless because the real world process needs are not supported by the application. The conversion of the incorrect application into one that supports the process encourages short cuts and fixes that hinder future maintenance. This may also result in an incomplete and undocumented built design. Establishing the validity of the design should be a required step.

In attempts to get better requirements, I know of cases where the subject matter experts have learned to be information systems (IS) analysts because the IS analysts could not do the required job. At other times, the IS analysts have become the subject matter experts by default. The amount of wasted corporate resources is significant in all such efforts. With this history it is understandable that management has grown to distrust the IS capabilities of their IT organizations.

Requirements Validation Using Natural Language Modeling

The Natural Language Modeling (NLM) analysis procedure for model validation consists of a set of deterministic steps that generate questions about simple sentences that are based on the graphical design. The precision contained in the graphical model is maintained through the application of the NLM procedure. This procedure focuses on simple sentences that are understood by the expert. The expert's "yes" and "no" answers to the NLM questions are compared to the answers required by the graphical design. If the expert's answers and the design do not agree, then the involved rule is investigated and the correct rule is determined. The model is validated when all of the answers from the expert and the model are the same.

Every "yes" and "no" answer pattern results in one of three outcomes:

1. validating a portion of the model
2. developing new questions that lead toward validation
3. detecting an inconsistency that must be resolved

In many cases the clarity of the NLM questions allow the expert to fully understand the implication of the requirements. The analyst is able to assist in the clarification of subject area rules that may be incomplete or inconsistent. This results in better requirements and in applications that are produced with less rework during implementation and less maintenance after release. The subject matter experts become accountable for the final design. The NLM procedure also allows other subject matter experts who have not been involved in the initial validation to collaborate in the independent validation of the design. None of these subject matter experts are required to understand any aspects of the NLM modeling procedure or to understand a graphical presentation of the modeling results. The analyst understands both the graphical model and the NLM procedure, but functions only as a facilitator for extracting the information required for the validation of the model. The analyst does not personally validate any of the subject area knowledge in the model.

Validation Examples

The validation of a graphical model using the NLM procedure is possible because of the model's unique yes/no answer pattern for the NLM questions and the subject area expert's ability to independently answer the same questions. A simple example will be presented here to show how the procedure identifies an error in a particular type of graphical model called an IDEF1x model. The type of graphical model is not important because the underlying knowledge rules will be the same for a subject area no matter what graphical form is used to present the design. An object-oriented model [2] example can be validated using this same technique.

Because the NLM validation procedure is deterministic, the analyst's knowledge of the subject area is not required to produce or validate the model. This allows two approaches for validation. The first approach uses the real world objects and sentences from the subject area and the second uses variables in place of the real world constructs. Models have always been constructed by educating the analyst about the subject area and then having him/her develop the required model. The precision of the NLM procedure allows the analyst to produce models without knowledge of the subject area being addressed. Although the variable approach may only have direct use with proprietary or security sensitive applications, it can also allow an independent analyst to lead a totally unbiased validation effort. Since the independent analyst could not apply any existing real world knowledge about the subject area, the validation will come only from the involved subject matter experts. Furthermore, the results will be precise and will possibly lead to an implementation that is correct the first time.

Real World Validation Example

The example problem will be a movie marquee that presented the movies being shown in a theater. Although this example is simple because of the shared real world knowledge among all movie goers, it highlights the common problem that the analyst assumes that his/her knowledge is sufficient or needed for pro-

Monday Movie Presentation

Session	Theater 1	Theater 2	Theater 3
10 a.m.	Jaws	Snow White	Invisible Man
noon	Jaws	Mad Max	Invisible Man
3 p.m.	Mad Max	Fantasia	Invisible Man
6 p.m.	Jaws	Fantasia	Invisible Man

Figure 1. Movie marquee.

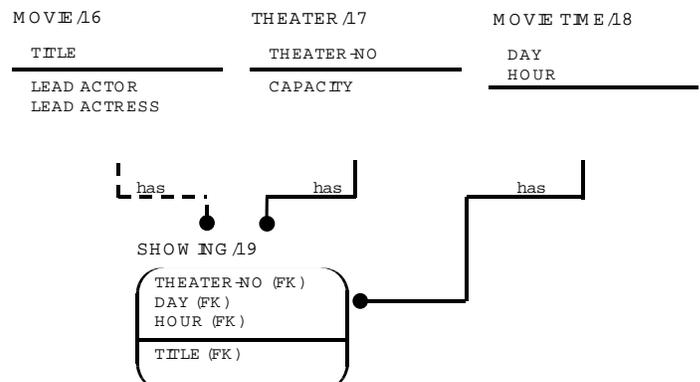
ducing the model.

The analyst would interview the theater manager and review appropriate examples, such as the marquee in Figure 1. The model in Figure 2 reflects the results of the modeling effort. The model presented is based on the constructs specified in the IDEF1x standard [3]. The manager can understand at least some of the knowledge in the diagram when it is explained by the analyst. The manager can agree with what was said, but he/she cannot be held accountable for the contents of the graphical model. Implementation should be based on the graphical model, so errors in the resulting system due to errors in the graphical model will put the analyst and the expert in the position of blaming each other for the result. No accountability is possible in this situation.

The model in Figure 2 documents the results of analysis for the movie marquee problem. Some of the efforts in reviewing this graphical model deals with the decision to make "Movie Time" an entity as opposed to having it be part of "Showing." Although this could be of interest, the main effort should be to determine if this model contains the correct rules for the subject area. The only way to validate the rules is to involve the subject matter expert in a process that will make him/her accountable for the resulting model. The NLM procedure has the expert create sentences based on the model that contain instances from examples used in the modeling process or independently supplied by the expert. The validation is accomplished by asking questions about the sentences.

Figure 2 is a standard IDEF1x model. The child entity (rounded corners) has allowed populations (foreign keys - FK) that come from the parent entities (square corners). Using a

Figure 2. Example of IDEF1x model.



population instance from Figure 1 and the “Showing” entity in Figure 2 the subject matter expert could generate the following sentence:

1. In theater **1** on **Monday** at **1 p.m.** the movie *Jaws* is showing.

The instances in sentence one from Figure 1 are shown in bold type. Each one of these instances can be replaced with an appropriate label as shown in sentence two to form a type sentence. These two sentences, one instance and one type, are now used in the NLM procedure to determine if the associated portion of the model is correctly specified.

2. In theater <Theater-No> on <Day>at <Hour> the movie <Title> is showing.

The NLM analysis procedure consists of a set of questions that the expert answers about the involved sentences. The first question deals with the ability to replace one instance value in sentence one and have both the original sentence and the modified sentence exist at the same time. The formal statement of question one in the NLM procedure is:

Q1.1. Given that fact “In theater **1** on **Monday** at **1 p.m.** the movie *Jaws* is showing.” is true, is it allowed for another valid Theater Number [for example “2”] to exist such that the fact “In theater **2** on **Monday** at **1 p.m.** the movie *Jaws* is showing.” is also true?

The subject matter expert would answer this question “no.” This answer is placed at the end of the first row of the matrix in Figure 3. The Q1 is asked in sequence for each individual placeholder. The matrix presentation of the answer vector for subject matter expert’s Q1 answers is shown in Figure 3.

For model validation the NLM procedure now references the graphical model to answer the same questions. A skilled analyst can obtain these answers directly from the IDEF1x model in Figure 2. The answer vector for the IDEF1x model is shown in Figure 4.

The NLM procedure specifies the next analysis step based on the yes/no answer vector. For model validation the NLM procedure now references the graphical model to answer the same questions. A skilled analyst can obtain these answers directly from the IDEF1x model in Figure 2. The answer vector for the IDEF1x model is:

2. In theater <Theater-No> on <Day>at <Hour> the movie <Title> is showing.

The model is validated when the subject matter expert’s answers are the same as the model’s answers. For this example

Figure 3. Expert’s answers to Question 1.

1	Monday	1 p.m.	Jaws	
				Allowed
another	Monday	1 p.m.	Jaws	no
1	another	1 p.m.	Jaws	yes
1	Monday	another	Jaws	yes
1	Monday	1 p.m.	another	no

1	Monday	1 p.m.	Jaws	
				Allowed
another	Monday	1 p.m.	Jaws	yes
1	another	1 p.m.	Jaws	yes
1	Monday	another	Jaws	yes
1	Monday	1 p.m.	another	no

Figure 4. Graphical model’s answers to Question 1.

three of the four answers are the same. The first answer is different. This difference must be investigated. A dependency exists for every “no” answer. The independent portion of the sentence that creates the dependency must be established. The NLM procedure does this through generating a set of questions that will establish all rules that apply to this sentence. In this example two rules are identified.

The last “no” answer is dependent on the identifier for “Showing” in the model in Figure 2. The real world situation for this dependency is that an individual theater can show only one movie at a time. Showing two or more movies at one time in the same theater would be chaotic. The analyst as well as all moviegoers understands this rule. The first “no” answer is dependent on the movie and movie time. The real world situation for this dependency is that only one copy of a particular movie is available at one time. This rule is due to the flat fee that the theater must pay for each copy of a movie and the manager’s decision to rent only one copy of a movie at a time. This second rule is only uncovered if the manager provides this information directly or the analyst asks the necessary questions. In most cases subject matter experts do not know what information is of value to the analyst and critical information may not be given. In too many cases the analyst uses his/her knowledge and may not ask the expert. Model quality may also depend on the amount of time spent with the manager. This often results in an extended amount of training for the analyst in certain aspects of the process and requires a significant amount of the manager’s time. It may be more efficient for the analyst to follow a procedure that requests the analyst to ask the needed questions.

Obviously, all of the correct data could be stored in an application developed using the first model, but data that violates a business rule could also be stored in this application. The model shown in Figure 5 enforces both of the rules and does not allow bad data to appear in the application. The new rule is expressed as the alternate key (AK) in the “Showing” entity.

All of the rules within the model can be properly established using a series of questions that are asked of the subject matter expert. The analyst’s independent knowledge about the subject area is not needed. Because of this, real world instances in the sentences can be replaced with variables so that the validation procedure could still be completed without exposing confidential information to the analyst.

Variable Validation Example

For validation with variables, each instance in a sentence is

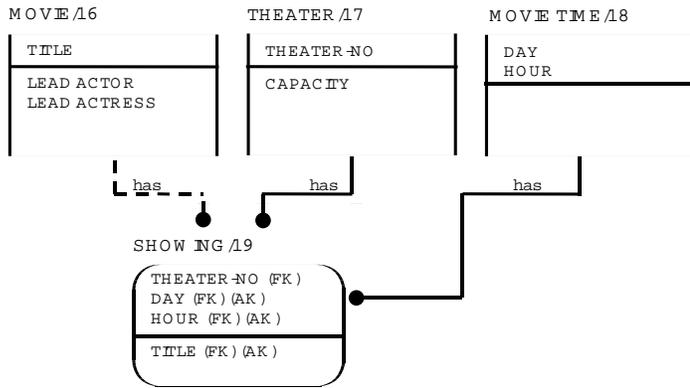


Figure 5. Corrected IDEF1x model.

replaced by a variable and the text segments in the sentence are replaced with constant text segments. The expert would know the meaning or semantics of the sentence and the objects. The “yes/no” answer pattern would be the same because the expert answers the same questions. An analyst who is knowledgeable of the subject area can produce the yes/no answer pattern for the graphical model. The validation would proceed by comparing the yes/no answer vectors as previously discussed. Using this approach, an analyst that was not associated with the generation of the model could independently validate the model while being prevented from knowing the technical content of the model.

- 3. Text1 a1 text2 b1 text3 c1 text4 d1 text5.
 - 4. Text1 <A#> text2 <B#> text3 <C#> text4 <D#> text5.
- Sentences three and four have been created from sentences one and two with:

Text1= In theater A#=Theater-No. a1=1
 Text2= on B#=Day b1=Monday
 Text3= at C#=Hour c1=1 p.m.
 Text4= hr the movie D#=Title d1=Jaws
 Text5= is showing

The subject matter expert knows the meanings of the variables so the Q1 answer vector in Figure 6 is the same as the one in Figure 3.

The analyst who developed the graphical model would create the answer vector in Figure 4. The comparison of the vectors would proceed as before and two rules would be found. The analyst validating the model without knowledge of the subject area would know the two rules that exist, while the knowledge-

Figure 6. Expert's answers to Question 1 using variables.

a1	b1	c1	d1	
				Allowed
another	b1	c1	d1	no
a1	another	c1	d1	yes
a1	b1	another	d1	yes
a1	b1	c1	another	no

able analyst and the subject matter expert could also attach the real world meaning to the rules. The analyst validating the model has no knowledge of what is being validated, but by following the NLM procedure he/she establishes whether or not the model precisely reflects the expert's knowledge of the subject area.

Conclusion

The result of model validation using the NLM procedure is that the subject matter expert is accountable for the contents of the model, even though the expert cannot read the developed graphical model. This type of validation is critical in the development of high reliability/high risk systems. By requiring that the analyst follow a deterministic validation procedure, the results of the validation effort depends on the expert's knowledge of the subject area and not on the analyst's knowledge of the subject area. The integrity of the resulting application is improved because the subject matter expert is accountable.

Validating graphical models using NLM is one way to minimize some unnecessary costs of IS efforts. Correcting the initial design also allows for an understanding of failures in the original modeling process. Problems can be identified and internal procedures can be modified as required. The need for staff training can also be assessed. ♦

About the Author



John K. Sharp is the founder and principal consultant for Sharp Informatics. Before starting Sharp Informatics in 1997, he was employed by Sandia National Laboratories in Albuquerque, N.M. for 18 years. While at Sandia he held staff and management positions in all areas of information technology, including analysis, design, implementation, maintenance, information architecture, data administration, and information technology research. Dr. Sharp is the developer of a mathematically precise information analysis procedure. Sharp was the editor of the international standard for conceptual schemas. He has co-chaired two international conferences on Natural Language Modeling and he has presented numerous papers and seminars at professional conferences.

Sharp Informatics
 Albuquerque, N.M.
 Voice: 505-243-1498
 Internet: www.sharp-informatics.com

References

1. Jones, Capers, Software Quality Forum, Sandia National Laboratories, April 1-3, 1997, Albuquerque, N.M.
2. Sharp, John, "Validating an Object Oriented Model," *Journal of Conceptual Modeling*, December 1998, www.inconcept.com
3. IDEF1x Data Modeling Standard, Federal Information Processing Standards Publications, FIPS PUB 184, December 1993.



Outsourcing Acquisition and Procurement Shops

J. Michael Brower

Department of Justice, Immigration, and Naturalization

When it comes to information technology (IT), procurement and acquisition shops have often suggested to program managers (PMs) that a function be competitively sourced. Now, procurement and acquisition shops are themselves being competitively sourced. Competitive sourcing euphemistically refers to outsourcing, or the emigration of in-house functions to an outside provider. Contracting officers (COs), often considered the rainmakers in both public and private organizations, survive by the award and administration of pacts with service providers. Consequently, procurement chiefs are quick to remind PMs that limited permanent staff and scant indigenous IT talent connotes an outsourcing solution. For the federal government, Congress authorizes a set number of employees, or full-time equivalents, making contracting-out indispensable for mission accomplishment.

THE AGE OF PUBLIC sector outsourcing has given redoubled importance to statements of work, the procurement shop, and to contracting officers and their on-site representatives. This has been particularly the case in acquiring IT services since the mid-1990s. The Office of Management and Budget's A-76 guidelines for agencies to evaluate whether to keep noncore functions in-house or outsource them has been applied to all elements of IT. Of particular interest to military planners has been outsourcing hardware and software configuration, help desk support, visual information, records and mail management, telecommunications, and all installation and maintenance of computer wherewithal. With online procurement increasingly in vogue and Department of Defense aggressively planning to complete \$2 billion worth of IT functions during the next three years according to McLean, Va.-based Federal Sources Inc., what must be considered by the procurement and the funding program when outsourcing?

If We Must Outsource IT Consider These Factors

Create a statement of work so it is clear what — not how — work will be done. Provide incentives, and plan for disincentives with an appropriate deduction schedule.

Demand value-based pricing and benchmarking — the contractor's price must be readjusted on occasion to the market price or renegotiate rates on a reg-

ular basis. Build that into the contract language.

The following costs must be transparent to the government IT executive as the outsourcer's customer:

1. Cost of telecommunication line — contractor reports cost per hours, per distances, per line, per switch, per gateway device, etc.
2. Per-person cost measurement — costs incurred by contract employees have to be reported by the contractor for agency comparison purposes.
3. Operational cost measurement — contractor must report costs in terms of cost per unit hours, storage costs, total cost per hour, fixed costs, and variable costs.

In the private sector, the maxim of cutting costs to increase profits is observed in good times or bad. Hence, the CO almost always encourages the employment of a lower cost, more readily hired/fired, less union-organized workforce in exchange for the permanent workforce. Procurement shops have played the part of docent in the museum of outsourcing — however, some of the head honchos of contracting-out have become the exhibits rather than the tour guides!

The paradigm ending indigenous contracting functions is known as business-process outsourcing. IT advances have made outsourcing of procurement and acquisition functions not only financially viable, but managerially trendy. For instance, both IBM and Electronic Data Systems offer contracting services. IBM's

Global Services division promotes its Business Process Service using Lotus Notes-based, Internet-accessible procurement software and a vast supply network to offer outsourcing of contracting functions [1]. Global Services, based in Somers, N.Y., had revenues of \$26 billion in 1997 [2]. IBM will take over procurement tasks for Hartford, Conn.-based United Technologies Corp. (UTC), a \$25 billion manufacturer of aerospace, automotive, and building products. UTC's goal is to cut \$750 million in purchasing costs by 2000.

While administrative lead times for government procurements often are measured in months, commercial companies like Boeing and Texas Instruments can measure their lead times in days, hours, and even minutes. The pattern of outsourcing of procurement shops first took shape in industry rather than reinvention-ridden government. Long lead times are the principle contributor to outside sources replaced by in-house contracting functions.

The Financial Rub of Outsourcing

Outsourcing is the divestiture of a core or noncore in-house function to an outside provider.

Privatization, often confused with outsourcing, refers to the transfer of strictly public work, and often the wherewithal to perform that work, to the private sector.

Contracting-out is not necessarily outsourcing or privatization. For exam-

ple, contracting-out for IT services that have never been performed in-house is not outsourcing.

IT labor costs associated with outsourcing, privatization, and contracting-out can and do vary. Outsourcing-oriented contracts can be written between government or private entities. According to the Outsourcing Institute, the second largest share (30 percent) of the estimated \$100 billion worth of U.S. corporate outsourcing ventures in 1996 was in administrative and similar services [3]. It does not take an encyclopedic grasp of things financial to anticipate that the outsourcing of procurement shops — the prime mover of most, and certainly the administrator of all, contracting-out — was predictable.

Why is this kind of outsourcing profitable? Just as outsourcing-type contracts have mainly financial motivations, outsourcing the contracting personnel and their work is based on specialization, which efficaciously lowers costs of employees, and shortens the associated lead times. COs may resent the latest nostrum from corporate cube-farm idea hamsters — outsourcing contracting — but they must be the first to respect the financially sound logic underpinning this strategy. PMs are poised to ask whether they can get existing procurement services for a reduced price at a break-even quality level. Replying in the affirmative, companies such as IBM offer to immediately tighten overhead costs as they rip into the

expense of fringe benefits for in-housers. The procurement process resembles to busy executives the chaotic, annual ritual of the running of the bulls in Pamplona, Spain. PMs are tired of being gored by the expense and lost time associated with the current method of contracting-out.

The systematic and aggressive use of IT outsourcers who tap into reduced-cost labor pools in low-cost geographic areas quickly undermines the economic viability of in-house procurement shops [4]. It is a return to Adam Smith's pin factory with a vengeance: specialization ruthlessly reduces IT costs. Many more in-house procurement shops will soon find themselves the victims of service shedding. Outsourcing has attached itself lamprey-like to the soul of every cutback-stricken organization — and the contracting shop is increasingly coming under the penetrating gaze of fiscal watchdogs. It is ironic — and to those last, least, and left-out during outsourcing's heyday — pleasurable to witness the champions of expropriation themselves expropriated. ♦

About the Author



J. Michael Brower is a Vermont writer. His previous assignment was at the Pentagon with the Office of the Assistant Secretary of the Army (Financial Management and Comptroller), Army Business Practices

Directorate. He is currently at the Department of Justice, Immigration, and Naturalization Service.

Dept. of Justice-INS
EOR-DDP, Rm. 321
70 Kimball Ave.
So. Burlington, Vt. 05403-6813
Voice: 802-660-5010 ext. 5452
Fax: 802-660-5148
E-mail: john.m.brower@usdoj.gov or
jmichael@together.net
Internet: <http://www.geocities.com/capitolhill/lobby/2985/>

References

1. LaMonica, Martin, "IT outsourcing extends its reach into mainline business functions" *Infoworld*, July 13, 1998, pg. 61.
2. Madden, John, "UTC Outsources Procurement," *PCWeek*, July 13, 1998, pg. 76.
3. Pritchett, Price, *Outsourced: The Employee Handbook*, Pritchett & Associates, Inc., 1997, pg. 8.
4. Applegate, Lynda M. et. al., *Corporate Information Systems Management*, Times Mirror Book, 1996, pg. 248-249.

The opinions expressed in this article are those of the author and not necessarily those of the Department of Justice.



CROSSTALK

SPONSOR	<i>Lt. Col. Joe Jarzombek</i>
PUBLISHER	<i>Reuel S. Alder</i>
ASSOCIATE PUBLISHER	<i>Tracy Stauder</i>
MANAGING EDITOR	<i>Kathy Gurchiek</i>
ASSOCIATE EDITOR/LAYOUT	<i>Heather L. King</i>
ASSOCIATE EDITOR/FEATURES	<i>Heather Winward</i>
VOICE	801-775-5555 DSN 775-5555
FX	801-777-8069 DSN 777-8069
E-MAIL	crosstalk.staff@hill.af.mil
STSC ONLINE	http://www.stsc.hill.af.mil
CROSSTALK ONLINE	http://www.stsc.hill.af.mil/ Crosstalk/crosstalk.html
ESIP ONLINE	http://www.esip.hill.af.mil

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address:

Ogden ALC/TISE
7278 Fourth Street
Hill AFB, Utah 84056-5205
E-mail: stsc.custserv@hill.af.mil
Voice: 801-775-5555
Fax: 801-777-8069 DSN 777-8069

Editorial Matters: Correspondence concerning Letters to the Editor or other editorial matters should be sent to the same address listed above to the attention of *CROSSTALK* Editor.

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 *Guidelines for CROSSTALK Authors*, available upon request. We do not pay for submissions. Articles published in *CROSSTALK* remain the property of the authors and may be submitted to other publications.

Reprints and Permissions: Requests for reprints must be requested from the author or the copyright holder. Please coordinate your request with *CROSSTALK*.

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

Coming Events: We often list conferences, seminars, symposiums, etc., that are of interest to our readers. There is no fee for this service, but we must receive the information at least 90 days before registration. Send an announcement to the *CROSSTALK* Editorial Department.

STSC Online Services: This can be reached on the Internet. World Wide Web access is at <http://www.stsc.hill.af.mil>. Call 801-777-7026 or DSN 777-7026 for assistance, or e-mail to schreiflr@software.hill.af.mil.

Back Issues Available: The STSC sometimes has extra copies of back issues of *CROSSTALK* available free of charge. If you would like a copy of the printed edition of this or another issue of *CROSSTALK*, or would like to subscribe, please contact the customer service address listed above.

The Software Technology Support Center was established at Ogden Air Logistics Center (AFMC) by Headquarters U.S. Air Force to help Air Force software organizations identify, evaluate, and adopt technologies that will improve the quality of their software products, their efficiency in producing them, and their ability to accurately predict the cost and schedule of their delivery. *CROSSTALK* is assembled, printed, and distributed by the Defense Automated Printing Service, Hill AFB, Utah 84056. *CROSSTALK* is distributed without charge to individuals actively involved in the defense software development process.

What is the Buzz about Change?

You endured long hours of study, tough labs, and hard exams as a software engineer student. Now you are in the real world and rising to a higher economic status. Nice car, comfortable house, good neighborhood, and an invitation to the neighborhood party. This is where you will meet your new neighbors — the banker, the lawyer, maybe a dentist or an architect.

Within the first hour you realize your invitation was based on your economic qualifications. The next hour you discover your continued membership is hinging on social acceptance. The final hour reveals that your association with innovative technology is the only idiosyncrasy that separates you from the parvenus in silence on the couch across the room. If the neighbors knew how passive your profession is on innovation you would be past the couch and out the door.

Look at this month's theme, "Managing Change In Your Culture." We are at the heart of a revolution in the way the world does business. How do we address change? We talk of adapting, managing, structuring, and surviving. Boring.

Most software books, journals, and consultants preach the Capability Maturity Model and continuous process improvement. Although useful and necessary, this is not innovation. It is house cleaning, putting in place what should have been there in the first place. At best we are remodeling, but no one is talking about designing a new software house that is more efficient and effective.

If there are innovators out there they are drowned out by the flood of change clones trying to organize, structure, and help you cope with change. It is all becoming repetitive — the IDEAL Model, successful steps to change, managing technology change, technology change management. Today, if you want to be a change management consultant, you simply head to Kinko's for some business cards, develop a model or two, and declare yourself open for business. It is as if all anyone wants to do is polish the already-chiseled Mount Rushmore. Let us move on to Stone Mountain or Crazy Horse!

Where are the innovative changes, the ideas that create value, new customers, and new markets rather than new IDEF models? Where are the innovations that impress customers, not just the boss?

The recipe for innovation requires a cup of creativity and a dash of discipline. It appears we have swapped the formula. We are fixated on the discipline or management aspect with little consideration for creativity. Harvard professor John P. Kotter, author of *Leading Change*, points out that, "the engine that drives change is leadership." He cautions that, "a pure managerial mindset inevitably fails [to implement change], regardless of the quality of people."

Could it be most of our executives, while good managers, lack leadership skills? Many were promoted based on software engineering skills that parallel management rather than leadership skills. They can keep the status quo running smoothly but may lack the skills to define the future and align people to realize that future.

Perhaps we should look to the bee. The beehive is a model of organization, structure, and industry yet a hive's survival depends on maverick bees known as scouts. Scout bees forego the traditional process-oriented roles that a queen, worker, or drone fulfills. They leave the crowded hive and wander around the fields finding new sources of nectar that assure the hive's continued success.

Are you leading change or managing it? Are your engineering scouts searching for innovations that assure your long-term survival or merely improving the status quo?

"The major advances in civilization are processes which all but wreck the society in which they occur," observed mathematician and philosopher Alfred North Whitehead. Once you have captured that process in IDEF0 and achieved your desired maturity level, start destroying it. Enjoy the party.

— Gary Petersen, TRI-COR Industries

Got an idea for *BACKTALK*? Send an e-mail to backtalk@stsc1.hill.af.mil

Technology Change Management

Structured Approaches to Managing Change

An overview of three perspectives of technology change management.

Mark C. Paulk 4

Addressing People Issues when Developing and Implementing Engineering Processes

Follow the process one defense contractor used to make people-friendly decisions.

Claude Y. Laporte 8
Sylvie Trudel

Rapid Emerging Knowledge Deployment

Using radical learning techniques to manage and keep pace with technology change.

Kevin Marler 14

Integrating Knowledge and Processes in the Learning Organization

Part one of a two-part series explores the adaptations needed to become a learning organization.

Linda Levine 17

Web Additions

The Managing Software Innovation and Technology Change Workshop

What are leading organizations doing to meet the challenges of technology change?

Eileen Forrester 22
Priscilla Fowler
Sharon Guenterberg

Y2K Defect Propagation Risk Assessment Using Achilles

A process and tools to assess the defect propagation risk of interconnected systems.

Donald J. Welch 22
Kevin Greaney

Software Engineering Technology

Determining the Completed Effort of Adapting Existing Software

A method for using the Boehm adaptation adjustment factor.

Ronald Cobb 23
Linda Smeraglinolo
Dave Wood

Validating Software Requirements

An approach for validating the requirements contained in any information system model.

John K. Sharp 25

Open Forum

Outsourcing Acquisition and Procurement Shops

Who and what is being contracted-out.

J. Michael Brower 29

Departments and Announcements

2 From the Publisher: CMMI Offers an Enterprise Focus for Technology Change Management

7 Call for Articles

31 BackTALK: What is the Buzz about Change?

CrossTalk Update 3

Coming Events 16

CROSSTALK

Ogden ALC/TISE

7278 Fourth Street

Hill AFB, UT 84056-5205

BULK RATE
US POSTAGE PAID
Permit No. 481
Cedarburg, WI