Learning: The Engine for Technology Change Management

Linda Levine
Software Engineering Institute

This is the second part of a two-part article. Part one, published in November, explored the adaptations needed in the process movement and knowledge-creation approaches to achieve the vision of a learning organization. Part two looks at learning in practice by examining some frameworks and tools that pull together process, knowledge management, and technology to support learning and effective change.

This author considers the role learning typically plays in technology adoption through lessons learned and training, extending more recently to the use of repositories. This is contrasted with models that support learning and successful change more actively—through improvement frameworks and tools—as illustrated by the examples of IDEALSM, an established model for software process improvement,[1] and the IDEALSM-based New Technology Rollout (INTRo), a web-based process guide for technology change management.[2]

Learning is the fuel for technology change management. In turn, the creative exploration and exploitation of technology, knowledge, and processes are critical ingredients for realizing learning organizations. Once we recognize these interdependencies and the need for knowledge integration, we will be able to move away from education and training-by-firehose instruction toward the next generation of approaches and mechanisms to support communication, coordination, and collaboration.

Why is Learning Important?
Technology change management is not an isolated activity but a process that touches many of the socio-technical activities at work in an organization. This bigger picture of technology change management includes business and work processes, and technical systems, as well as processes related to group dynamics and collaboration.

The connection is clear. When we ask people to change how they do their work, 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 how you manage change, you will help make your workplace a learning organization, and that will pay off in many ways.

Learning is the keystone for dealing with the high number of failed change efforts, the rapid rate of change in information technology, and the need for new organizational and management constructs. Why is learning important?

- Approximately 70 percent of business process re-engineering efforts, or redesigns, fail [1,2]. The number and degree of failures would decline if we paid attention to connections between technology adoption and learning. Resistance is typically trivialized and resistors are seen as people that need handling. In fact, resistance can anticipate and surface flaws in intent, design, and implementation, and be a predictor of problematic and high-risk endeavors [3].
- Work groups of the 21st century will manage change in dynamic situations. Older freezing and refreezing metaphors from organizational development are inadequate [4]. Multimedia technologies and practices supporting process change, modeling, simulation, and collaborative and distributed work are key. Skill sets in the new work force that allow for flexibility, speed, and experimentation will be prized. We must learn how to learn.
- Traditional management constructs are outdated. Managers and practitioners express different kinds of fearfulness about adopting certain innovations, specifically process automation, workflow technologies, and groupware [5,6]. We must address the incompatibilities between collaborative technologies and organizational hierarchies and bureaucracies if we are to ease the transition to newer forms of work groups, including high-performance teams, self-directed, autonomous teams, and integrated product (process or practice) teams.

What About TCM in the CMM?
We can already see a rethinking of technology change management in proposed changes to the Capability Maturity Model for Software (SW-CMM v1.1.) Draft C of the SW-CMM, enlarges the scope for technology change management: the purpose of Organization Process & Technology Innovation "is to identify process and technology improvements and innovations that would measurably improve the organization's software processes, thereby helping achieve the organization's software process improvement goals."

The transformation of Technology Change Management into Organization Process and Technology Innovation is a step toward a more robust and innovative approach. However, much remains to be understood about these activities, especially about how they are operationalized.

Are we ready for technology change management and learning organizations? Yes, more than ever. We are at a watershed with the potential to get leverage from experience in process improvement, our intellectual investment in organizational learning, matching our interests with enabling information technology. But first, how are we learning now?

Learning: The Current State
If we were to describe how learning takes place in our organizations, chances are we would say it occurs through:
- training

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

- professional development
- lessons learned documents
- war stories
- process asset libraries (PALs) or repositories

Training and professional development are usually associated with individual learning. Project learning can be captured when lessons are documented; war stories are sometimes shared informally with others. Such stories convey valuable, undocumented learning (that may be imprudent to record) gained during the life of a project. When and where are these lessons applied? A good question, and one most organizations fail to address. Process asset libraries represent our best current attempt at information sharing and organizational learning, where artifacts (process and method descriptions, plans, standards, policies, templates, forms, etc.) are made available for adaptation and reuse through repositories. Too many of these libraries, however, degrade to online file cabinets or archives. The repository's role as coordinating mechanism—a forum and space for information sharing and exchange—in the larger learning environment is never realized.

Our understanding of the organization as a structure, as building and edifice, drives how we tackle learning and work. We build complementary forms in the same image as the organization—whether these forms are products or projects or teams—in order to handle problems. The challenge awaits to envision and internalize a newer understanding of the structure of the organization as a permeable, flexible, virtually networked locus of activity. If we accept this challenge and the shift it represents, gradually we will gravitate to solutions that emphasize transaction and interaction, and serve as vehicles for communication and collaboration. As our experience base grows, so will our ability to work in solution spaces where learning is valued.

Active and Interactive Learning: The Desired State

"The lecture is the most inefficient method of diffusing culture. It became obsolete with the invention of printing. It survives only in our universities and their lay imitators, and a few other backward institutions…. Why don't you just hand print lectures to your students? Yes I know. Because they won't read them. A fine institution it is that must solve that problem with platform chicanery" [7].

The models and mechanisms for organizations of the 21st century will support learning by doing—communication, coordination, and collaboration—in an interactive mode. Now, we primarily learn from discrete events in which we are involved, and we have scant understanding of the work and learning of others. Consequently, we fail at lessons applied on a triple score.

1. Few incentives explicitly encourage lessons applied and information sharing;
2. Even fewer mechanisms transfer knowledge across teams and parts of an organization;
3. We rarely reflect on the learning process. Chris Argyris offers a valuable characterization.

Argyris distinguishes between single-loop learning, which asks a "one-dimensional question to elicit a one-dimensional answer." He offers the example of how a thermostat measures temperature against a standard setting and turns the heat on or off accordingly. He contrasts this with double-loop learning, which "takes an additional step or, more often than not, several additional steps. It turns the question back on the questioner. . . . In the case of the thermostat, for instance, double-loop learning would wonder whether the current setting were actually the most effective temperature at which to keep the room and, if so, whether the present heat source were the most effective means of achieving it. A double-loop process might also ask why the current setting was chosen in the first place. In other words, double-loop learn-ing asks questions not only about objective facts but also about the reasons and motives behind those facts" [8].[3]

Double-loop learning describes the active learning we have been emphasizing. The benefits for knowledge integration and transfer across organizations and communities are also evident; this was the topic for Part One of this article. The practical question remains: What early examples do we have of frameworks or tools that try to pull together process, knowledge management, and technology to support organizational learning? Two such examples are IDEAL (Initialing, Diagnosing, Establishing, Acting, Leveraging), a high-level model for software process improvement, and the IDEAL-Based New Technology Rollout (INTRo), a web-based process guide for technology change management. INTRo was designed with the principles of organizational learning and knowledge management in mind. We consider these examples and the role that learning plays in each.

The IDEAL Model

The IDEAL model was conceived as a life cycle model for software process improvement based upon the CMM for software. The model provides a disciplined engineering approach for improvement, focuses on managing the improvement program, and establishes the foundation for a long-term improvement strategy [9]. Following the phases, activities, and principles of the IDEAL model has proven beneficial in many process improvement efforts. The model consists of five phases listed below.

Development of IDEAL was prompted by requests from people who were engaged in Software Process Improvement (SPI) initiatives based upon the SW-CMM. The original model incorporated results of fieldwork the Software Engineering Institute (SEI) had conducted with early SPI adopters. Designed to satisfy specific needs, its focus was initially

| Table 1: *The Five Phases of IDEAL*	
I Initiating	Laying the groundwork for a successful improvement effort
D Diagnosing	Determining where you are relative to where you want to be
E Establishing	Planning the specifics of how you will reach your destination
A Acting	Doing the work according to the plan
L Learning	Learning from the experience and improving your ability to adopt new technologies in the future
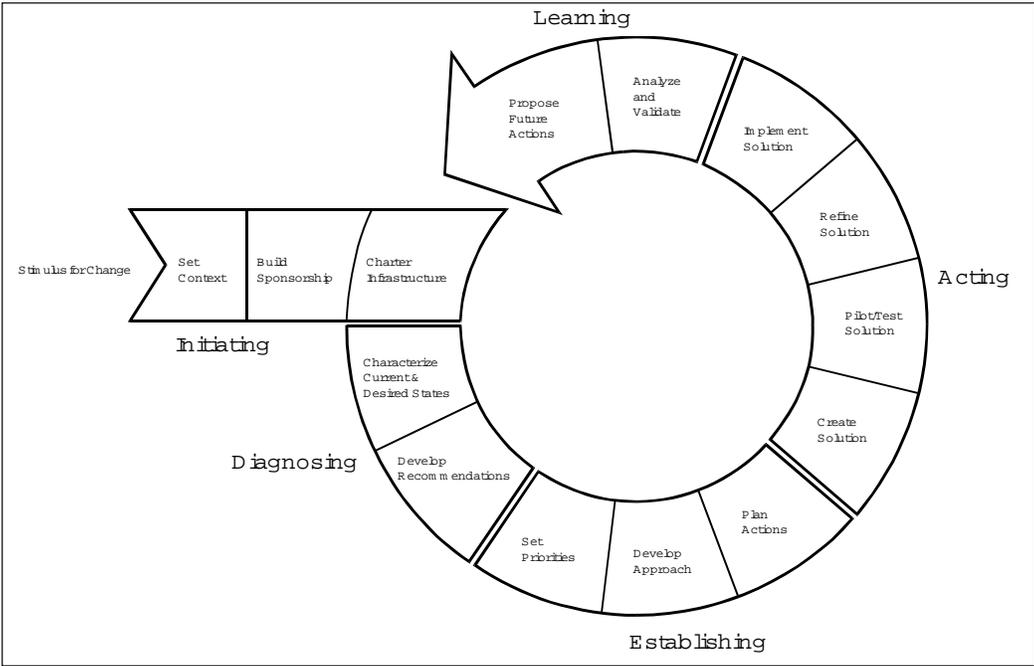
Figure 1: *The IDEAL Model*

restricted in terms of breadth and depth. In terms of breadth, the language used was process-specific; in terms of depth, it was seen as a life cycle model for larger, long-term efforts, such as moving an organization from one maturity level to another. The potential for extending the model's application to other technology domains and to efforts of virtually any size soon became clear, and the SEI launched an effort to reframe the model in a way that would facilitate its broader application. This effort resulted in the representation shown as Figure 1.

The more generic language used also opened the door for adopting the IDEAL Model as a standard life cycle that can be applied at all levels of activity within any major change effort (for example, in SPI terms, a maturity level, a key process area, a key process area component, and an activity).[4] Some find the multiple levels of applicability helpful.

In the learning phase of IDEAL, the adoption or improvement experience is reviewed to determine what was accomplished, whether the effort met the intended goals, and how the organization can more effectively or efficiently implement change in the future. Addressing any and all of these concerns represents active or double-loop learning.

As a whole, the model reinforces learning through the concept of continuous process improvement, but clearly

IDEAL locates specific learning activities in the learning phase. Some have observed that learning should not reside in one phase (as after-the-fact reflection). Instead, learning needs to be recognized as an interwoven process thread. However, threading has not been expressly called out in the existing model.
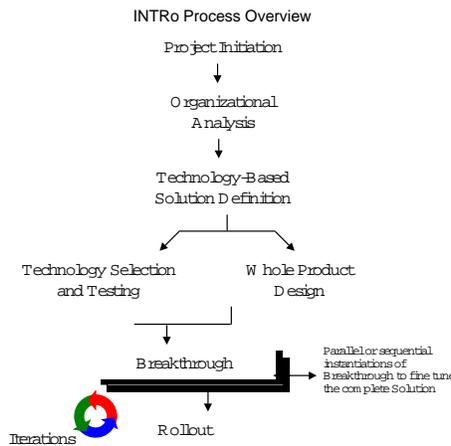

Figure 2: *IDEAL-Based New Technology Rollout*

INTRo

INTRo is a web-based process guide focused on making connections among business problems, value propositions, technology solutions, and their implementation.[5] INTRo was designed to help organizations adopt and implement new tools or technology. Since many such efforts are complex, and the effects so far

reaching, a structured approach is required. Successful technology change management also requires comprehensive knowledge and skills that often do not reside in a single organization, individual, or team. Typical practitioners do not have the full range of skills needed to manage technology adoption. INTRo helps to fill the skill gap.

All approaches to technology change involve assumptions and expectations about what is central in a change effort. Many see "new technologies dominating the improvement effort; others focus on changes in process or business practices; some perceive organizational change. Very few, however, perceive change in multiple facets of the organization (process, technology, people, culture, organization); they commonly recognize change in no more than one or two dimensions [10]. INTRo was designed to recognize the multiple dimensions of change, including multiple subsystems in the organizational system, and to integrate these perspectives.

IDEAL provides a usable, understandable approach to continuing improvement by outlining the steps necessary to establish a successful improvement program. INTRo goes further, embodying the detailed how-to information needed to manage the introduction of a new technology, organized into a work breakdown structure of stages, steps, and tasks. Tips, checklists, guidelines, and tutorials accompany process descriptions. Related work products, such as a project schedule, a configuration baseline, and a training strategy, are also included in template form.

INTRo consists of seven stages, as summarized in Table 2.

Learning in INTRo

INTRo departs structurally from IDEAL and learning is not treated separately. Tasks that facilitate knowledge integration are interwoven. Learning is built into the process, rather than accounted for at the end in a lessons learned or post-mortem activity. The approach to knowledge integration is accomplished a number ways, and each of these is discussed briefly.

- stage management and stage end assessment activities
- process and product improvement activities
- knowledge and skills transfer mechanisms and strategy
- just in time: techniques, tutorials, related kernals
- process threads
- coordinating mechanisms

Stage Management, Stage-End Assessment Activities

After project initiation, all of the stages begin with a stage management step and end with a stage-end assessment step. Stage management sets the stage and ensures the readiness to begin, with tasks that include kickoff, monitoring of project progress, issue identification and resolution, and management of exceptions. Stage-end assessment, at the end of a stage, involves reviewing and baselining project/deliverables and preparing for the next stage.

Process and Product Improvement

Stage-end assessments sometimes include process review activities. These reviews verify that all planned processes have been completed according to the standards of the stage. Based on the results, problem reports or change requests may be generated—on the process being audited—to initiate process improvement activities. In rollout, the final stage of INTRo, extensive product and process improvement activities are performed. These tasks involve review and baseline of project/deliverables, collection of project feedback and metrics, metrics analysis, analysis of product quality, and completion of process reviews.

Knowledge and Skills Transfer Mechanisms and Strategy

Key activities highlight learning. For example, knowledge and skills transfer focuses on determining the strategy, including roles and responsibilities, and mechanisms for knowledge transfer and support of the new technology after rollout when in maintenance mode. The strategy covers mechanisms that will take

Stages and Steps	Tasks: Key Activities
Project Initiation Project Kickoff	§ Identify business drivers and objectives; Define scope; develop overall approach
	§ Establish team; Determine costs
Project Organization	§ Identify key products, Define configuration management, product reviews, product baselines, reuse strategy
Product/object Management Procedures Scheduling, Budget, and Control	§ Develop schedule; produce budget; establish project control procedures; Perform cost/benefit and risk analysis
Organizational Analysis Requirements Definition	§ Define high-level requirements; Collect previous change effort information; Draft high-level sketch of the solution
Current State Baseline	§ Review existing processes and assets; Develop a business process model of the current state; Review technical architecture requirements; Gather data about culture
Desired State	§ Create list of improvements; Investigate technology ideas; Design new business process model; Identify levers of change, Identify sources of resistance
Technology-Based Solution Definition Technology-Based Solutions/Options	§ Package solution options; Perform analysis on multiple solutions; Make recommendations; Determine increments for solution; Develop plan for acquisition; Identify products for evaluation
Approach for Breakthrough and Rollout	§ Develop high-level plan for Breakthrough; Determine iterations for breakthrough and rollout; Develop resistance mitigation strategy; Conduct stakeholder briefings
Technology Selection and Testing Architecture/Components Identification & Selection	§ Evaluate and select the technical infrastructure; Include performance testing
Technology Procurement	§ Review contract agreements; Obtain required signatures; Generate purchase orders
Architecture Installation & Testing	§ Install components for testing; Setup user access rights; Test multi-user access and component configurations
New Technology Test Scenarios	§ Review test results and baseline architecture
Whole Product Design Social Design, Policies & Standards,	§ Review resistance mitigation strategy; Revise reward, incentive, and compensation programs; Restructure organization if needed
	§ Establish policies to support the technology; define standards
Support Mechanisms, Training Preparation	§ Determine help desk & technical support; Develop documentation; Determine communication mechanisms
Knowledge Skills & Transfer Mechanisms	§ Identify training requirements for all roles/levels; Develop training strategy; Design or produce training materials; Pilot the training on those who will support the new technology
	§ Knowledge & skills transfer to ensure success after rollout; Identify user group members and develop charter for group
Breakthrough	§ Detail breakthrough plan; Create post installation test scripts; Conduct training to breakthrough team
Breakthrough Kickoff	§ Install infrastructure components; Customize or configure components; Migrate data; Perform acceptance testing; Test whole-product solution
Installation & Solution Testing Breakthrough Monitoring	§ Monitor and manage resistance; Review solution testing results; Implement critical changes; Schedule other changes before rollout begins
Rollout Iterative Rollout Planning	§ Detail the plan for rollout; Create post installation test scripts
	§ Conduct briefing with rollout users; Conduct training
Rollout Launch	§ Install the infrastructure components; Customize/configure the technology; Migrate data; Perform acceptance testing; Enable support mechanisms & other whole-product solution elements
Installation & Customization	§ Monitor and manage resistance; Review effectiveness of solution; Implement critical Change Requests and schedule remaining CRs before next rollout
Iterative Rollout Review	§ Baseline business process and technical architecture models; Review metrics and make recommendations for process improvement; Make product improvement recommendations

Table 2: *Stage, Step and Task Summaries for INTRo*

the metrics analysis data, knowledge gathered from technical support, and help-desk personnel into account. A user group for process feedback is created. Information flow is planned among this group, technical support, and the help-desk team.

Just in Time Learning: Techniques, Tutorials, Related Kernels

INTRo offers techniques and related kernels of information, in context, in the web-based guide. For example, the requirements definition step includes a technique on critical requirements analysis; the current state baseline step includes techniques on use case modeling and business process modeling. Other techniques are provided for performing personal and walkthrough reviews, etc.[6]

Process Threads

INTRo takes a first step in building process threads for tasks that are related but conducted at different points in the process. For example, early in the organizational analysis stage, we begin a thread related to organizational culture. Cultural analysis is important: the information gathered will tell about previous change efforts that were more and less successful. The organization's culture also will influence our approach for breakthrough and rollout. The thread is woven into subsequent stages, through development of the resistance mitigation strategy, and the design of the whole-product solution. Process threading allows us to build on previous learning in a topic/area and emphasizes the iterative nature of solution development. The so-called answer on the culture of an organization is not gained in a single task but through a series of relevant, related activities performed over time. Initial activities in culture process thread are summarized in Table 3.

Coordinating Mechanisms

These make up the engine for knowledge management. All of the learning features already described support coordination and integration. These mechanisms relate to both the structure and content of INTRo, and range from

- activities, such as stage management and stage-end assessment; process and product reviews
- associated activities: use of process threads
- artifacts: knowledge and transfer strategy, resistance mitigation strategy, solution definition, whole-product design, etc.
- agents: project team, breakthrough teams, user groups, learning center

Still Learning

INTRo is undergoing end-to-end testing and the development team is gathering comments and requirements from reviews and pilot testing; the next version of INTRo will incorporate this input. Version 1.0 established a baseline that may now be refined and enhanced.

Colleagues working in different cities and in different organizations co-developed INTRo. Organizations adopting new technology also often have the need to collaborate among distributed locations—INTRo requires the use of collaborative techniques. We used collaborative tools to enable our work, and as we were developing, we captured data on how the remote collaboration was working. We are analyzing this data and may encapsulate practices to be included in future versions of INTRo that support cooperative work on virtual or distributed teams.

In many ways, INTRo represents an intersection among process management, knowledge management, and collaboration tools and techniques—all crucial aspects of organizational improvement and organizational learning. Future versions will make more deliberate connections among these disciplines.

Conclusion

In part one, Integrating Knowledge and Processes in the Learning Organization, we showed that local adjustments were necessary within the process and knowledge-creation movements. Within the process arena, it remains for us to balance process formalization with process creation by leveraging individual knowledge and diverse perspectives through information exchange. Within the knowledge-creation arena, the challenge to filter and channel information for decision-making awaits. Organizations that support information sharing and knowledge creation are much more likely to establish effective and efficient processes and to improve organizational life [11,12].

In practice, technology change management represents the fusion of technology innovation and process and knowledge management as it is fully defined, operationalized, and enacted in a learning organization [13]. In part two, we looked at frameworks and tools that pull together process, knowledge management, and technology to support learning and successful change. We illustrated this with the IDEAL[SM] model for software process improvement, and INTRo, a web-based process guide for technology change management. To different degrees, and levels of detail, both support the premises of active learning. INTRo places greater emphasis on organizational learning and enabling technology.

As we approach the 21st century, it is clear that our understanding of organizations, learning, and work is still unclear, changing, and likely to keep changing. We have yet to envision the future of organizations—as adaptive, virtual networks of activity. If we accept this challenge and the shift that it represents, we will begin to effect solutions that reflect this vision and foster communication, coordination, and collaboration. As our experience base grows, so will our ability to create spaces where active and interactive learning routinely occur.◆

About the Author

Linda Levine leads the effort at SEI on IDEAL[SM] transition framework development, aimed at extending improvement models into structured processes for tech-

Organizational Analysis	
Requirements Definition Current State Baseline Desired State	– Collect previous change effort information – Gather data about culture – Identify levers of change, Identify sources of resistance

Table 3: *Initial Activities in Culture Process Thread*

nology adoption and rollout. She is a process developer for INTRo. She also researches technology suppression, reasoning and communication, design disciplines, and the relationships between organizational learning and the use of collaboration technology. Dr. Levine hs her doctorate from Carnegie Mellon. She publishes widely and is the co-founder of the Working Group (8.6) on Diffusion, Transfer, and Implementation of Information Technology, part of the federation for Information Technology for Information Technology (IFIP).

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

1. Wellins, Richard S. and Murphy, Julie Schulz. (1995). Re-engineering: Plug into the Human Factor. *Training and Development,* pp. 49, 33-37.
2. Kock, Ned. (1999). *Process improvement and organizational learning: The role of collaboration technologies.* Hershey, Pa.: Idea Group Publishing.
3. Levine, L. (1997). 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.
4. Schein, E. (1996). Three Cultures of Management: The Key to Organizational Learning, *Sloan Management Review,* Fall, pp. 9-20.
5. Christie, A.M..; Levine, L.; Morris, E. J.; Zubrow, D.; Belton, T.; Proctor, L.; Cordelle, D., and Ferotin, J-E. (1996). Software process automation: Experiences from the trenches. (*SEI Technical Report* SEI-96-TR-013). Pittsburgh, Pa.: Software Engineering Institute.
6. Grudin, J. (1995). Groupware and cooperative work: Problems and prospects. In R.M. Baecker, J. Grudin, W.A.S. Buxton and S. Greenberg, (Eds.), *Readings in Human Computer Interaction: Toward the Year 2000* (pp. 97-105). San Mateo, Calif.: Morgan Kauffman.
7. Skinner, B. F. (1948). *Walden Two.* New York: The Macmillan Co.
8. Argyris, C., Good Communication that Blocks Learning, *Harvard Business Review,* July-August 1994
9. Gremba, J. and Myers, C. (1997). *The IDEAL^SM Model: A practical guide for improvement.* Bridge, Pittsburgh, Pa.: Software Engineering Institute. Also at www.sei.cmu.edu/ideal/ideal.bridge.html
10. Price Waterhouse Change Integration Team. (1995). *Better change: Best practices for transforming your organization.* Chicago: Irwin Professional Publishing.
11. Brown, J. S. and Duguid, P. (1991). Organizational learning and communities of practice: Toward a unified view of working, learning, and innovation. *Organization Science 2(1),* 58-82
12. Brown, J.S. and Gray, E.S. (1995). After reengineering: The people are the company. *Fast Company, Premiere Issue,* pp. 78-81.
13. Lundberg, C. C. (1991). Creating and Managing a Vanguard Organization: Design and Human Resources Lessons from Jossey-Bass. *Human Resource Management,* 30(1), 89-112.

Acknowledgements

Notes

1. The Software Engineering Institute is a federally funded research and develop ment center sponsored by the Department of Defense. IDEAL^SM is a service mark of Carnegie Mellon University. For more information visit www.sci.cmu.edu/IDEAL.html
2. INTRo is a collaborative effort between SEI and Computer Associates (formerly Platinum technology, Inc).
3. For more of Argyris' insight, refer to Argyris, C. (1976). Single- and double-loop models in research on decision making. *Administrative Science Quarterly,* 21(3), 363-375. ; and Argyris, C. and Schon, D. A. (1996). *Organizational Learning II,* Reading, Mass.. Addison-Wesley, et. al.
4. IDEAL is similar to other improvement models, (e.g PDCA) but differs in subtle and important ways (Myers, 1998). PDCA is an acronym for Plan, Do, Check, Act: four steps in an improve ment cycle that is widely used in Total Quality Management (TQM). This cycle is described by W. Edwards Deming's *Out of the Crisis.* Deming is well known as the father of PDCA.
5. INTRo is a co-development effort between the SEI and Computer Assoc. (formerly Platinum technology Inc.)
6. These techniques and related kernals are made available by Computer Associates (formerly Platinum technology Inc.) as part of the INTRo co-development effort. They are process assets in the Process Library in Platinum's Process Engineer tool set.

Suggested Additional Readings

Bannon, L. J. and Kuutti, K. (1996).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). Maui, Hawaii: IEEE.

Roth, G. L. (1997). Learning Histories: Using Documentation to Assess and Facilitate Organizational Learning, Working paper, MIT Sloan School of Management. At http://learning.mit.edu/res/wp/index.html

Schein, E. (1995). Kurt Lewin's Change Theory in the Field and in the classroom: Notes Toward a Model of Managed Learning. Working paper, MIT Sloan School of Management. On the web at http://learning.mit.edu/res/wp/index.html

Schein, E. (1995). "On Dialogue, Culture , and Organizational Learning," E.M.R. Spring, pp. 23-29. (reprinted from *Organizational Dynamics,* Autumn/1993)

Schein, E. (1997). Organizational Learning: What is New? Working paper, MIT Sloan School of Management. On the web at http://learning.mit.edu/res/wp/index.html

Senge, P. (1990). *The Fifth Discipline.* N.Y.: Doubleday.