The Software Engineer: Skills for Change

Dr. Stephen E. Cross and Dr. Caroline P. Graettinger
Software Engineering Institute (SEI)

The rapid pace of change in software engineering means that software-intensive organizations must develop a core competency for proactive change management. While most software engineers are concerned with doing their jobs and getting a product out, others need to take on the role of technology change-agents for their organizations. We suggest that a career path in change management will be a critical need and opportunity for the 21st century software engineer.

During the past 15 years, the Defense Science Board (DSB, www.acq.osd.mil/dsb) has made more than 130 recommendations intended to improve the ability of software engineering organizations to produce high-quality software on time and at cost, yet these organizations continue to find it difficult to make these changes. If improvement were simply a matter of purchasing a new software tool, change would not be so difficult. But to realize the improvements called out in the DSB reports and others like them requires changes in the day-to-day practices of software engineers and their managers. Achieving these kinds of changes requires more skills than the ability to purchase a new tool—skills for change management become important.

Unfortunately, too few of the software-intensive organizations are proactive about managing change. A recent article by Boehm and Basili shows that poor software engineering practices are still a major contributor to software defects [1]. While some organizations adopt new and better software engineering practices and technologies because they recognize their strategic value, most organizations are more reactive than proactive.

Reactive organizations are at the mercy of change led by others, and generally attempt radical improvements in their software engineering disciplines only after customer pressures or disaster. By then, this catch-up game is costly and is often implemented by pick-up teams of individuals who may be well intentioned, but lack the change management skills to be successful. The most common result is money spent with little change to show for it.

Change: What Past Studies Say

Let us look in more detail at some of the past recommendations for change in the software engineering community. The DSB conducted task forces on defense software in 1987, 1994, and 2000 [2, 3, 4]. Each report advocated management and technical practices aimed at helping organizations improve the quality of their software as well as their productivity, cycle time, and cost effectiveness. For example, each report offered recommendations in the areas of acquisition policy, contracting, work force, and technology to encourage the Department of Defense (DoD) and its industry base to adopt spiral development and evolutionary acquisition techniques, and to exploit evolving commercial technology. Each report observed with concern the shortage of software engineers in the government and defense industry base and the need for better training and professional development.

However, each report proposed slightly different approaches to realize the recommendations, depending on the technology, organizational trends, and opportunities prevalent at that time. For example, the 1987 report stressed the role of high-order languages, most notably Ada and computer-aided software engineering tools in support of Ada implementations.

The 1994 report encouraged use of commercial products and processes and a greater focus on the design phase, most notably on software architecture and product line practice. Both were considered fundamental to a successful strategy for migrating software engineering from a line-by-line endeavor to one in which systems were assembled from commercial components.

The 2000 report provided six recommendations:
1. Stress past performance and process maturity.
2. Restructure contract incentives.
3. Collect, disseminate, and employ best practices.
4. Initiate independent expert reviews.
5. Improve the software skills of acquisition and program management.
6. Strengthen and stabilize the technology base.

In essence, the 2000 DSB study summarized the important recommendations from past studies. These recommendations say that change is required just to gain the advantages of what is known to be effective software engineering practices.

However, there are other sources of change as well. DoD software requirements are expanding, including demand for increased functionality and flexibility of software within any one system. Human resources shortages are continuing, as observed in studies during the past 15 years, with severe shortages in the government and the defense industry base; productivity improvements are essential. Lastly, security issues are now a major concern. These trends all suggest additional changes that organizations and their software engineers will need to deal with in their day-to-day practices.

Skills for Change

What are the implications for the 21st century software engineer? While most software engineers are concerned with doing their job and getting a product out, others need to take on the role of software technology change-agents. Such engineers need to develop the following skills:

• Identifying and evaluating new software engineering methods and tools.
• Understanding how individuals react to and commit to new innovations/technology.
• Understanding how organizational culture influences the adoption of new innovations/technology.
• Utilizing effective mechanisms for institutionalizing and sustaining an adoption.
To apply this knowledge successfully requires engineering competency coupled with competencies in management and psychology. Unsuccessful attempts at change can often be traced to failure in one or more of these competencies. It is important then for organizations to develop these skills.

One approach we have seen recently is the creation of a career path in software engineering change management. Engineers who elect this path are provided with training, mentors, and a certification process designed to develop their change management skills. There is a promotion path with a top position at the executive level. Change management is treated as a key success factor in these organizations, with a career path as part of the infrastructure to support that competency.

As two practitioners in the field of change management have stated [5], “No technology will remain in first place forever. The trick is to pick technologies likely to remain useful long enough for the firm to recover its investment as well as to place itself in a position to leverage the next big technological advancement. This is a difficult task.”

Even with skills for change management, technologies can still come along that are so revolutionary, it is not possible or practical to leverage the existing technologies [6]. Who will be more able to recognize and respond to this change: those with skills for change or those without? Obviously, those with change management skills will be the winners.

### Change: A Medical Analogy

Let us conclude with a quick look at the field of medicine as an analogy to demonstrate the challenges of adopting new practices and why knowledgeable, experienced change-agents are so vital. While this, like most analogies, is an oversimplification, and there is always a danger of carrying it too far, we nevertheless offer it as a way of relating to some of the skills needed in change management.

The medical field offers some useful analogies to software engineering. While we think of DoD weapon systems as being highly complex, we think it safe to say that the system that physicians work on (the human body) is even more complex. In both fields however, the practices and tools of the practitioners are changing rapidly. A software engineering change-agent (the equivalent of the physician in our medical analogy) should be aware of the changes in the field and have the competencies to evaluate and select the appropriate changes for their organizations (where the organization is equivalent to the patient in our medical analogy).

We can carry this analogy further by relating the various medical specialists and diagnosticians to the various change-agent roles and skills. For example, sometimes a patient is willing to make whatever changes are necessary to achieve improvement and possibly a complete cure. They follow their doctor’s advice and even take it upon themselves to learn more about their condition and treatment options. Such patients are willing adopters of the change.

On the other hand, when the changes require a complete lifestyle overhaul, this can be a much bigger challenge to the patient. When that happens, the patient may require other support in the form of psychologists, therapists, etc. that will address the various dimensions of the treatment.

Likewise, the software engineering change-agent must be prepared to address the multiple dimensions of the change – business, technical, cultural, and political. Whether these skills are embodied in one or more change-agent roles will be determined as more organizations explore the realm of software engineering change management. It is probably safe to say that organizations will have a spectrum of adopters who respond and commit differently to changes in their day-to-day software engineering practices. Hence, we believe there is a need for skilled practitioners with the know-how to address these issues.

### Conclusion

Change in the field of software engineering provides organizations and software engineers with both opportunities and challenges. The opportunities are in the adoption of new practices and supporting technologies that will improve the quality of their products while at the same time enabling more predictability in their development costs and schedules.

The challenges are in recognizing that no technology or practice will remain in place forever, and the organization must be prepared to continually improve their software engineering discipline and technologies.

We propose that organizations with core competencies in change management have these advantages:

- Better able to continually identify, evaluate, and implement real improvements.
- Less likely to waste resources and time on inappropriate choices and poor implementation. Organizations need the skills to identify, select, and institutionalize new practices and technologies that will provide an acceptable return on investment and put them in a position to leverage the next big advancement. We believe these are the skills that the software engineering change-agent needs now and in years to come.

### Acknowledgements

The authors would like to thank Dr. John Goodenough and William Pollak, both of the Software Engineering Institute, for their valuable contributions to this article.

### References


### Notes

1. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.
Department of Defense, www.nap.edu/catalog/5463.html, influenced a DoD acquisition policy change to refrain from mandating ADA in DoD software-intensive systems.

3. In a series of interviews conducted last year by the SEI with several DoD industry organizations, the continued migration of software engineers to the commercial sector was attributed to the perception of more rewarding positions in the commercial sector.

About the Authors

Dr. Stephen E. Cross, Ph.D., is director and CEO of the Software Engineering Institute at Carnegie Mellon University. He is also principal research scientist in the Robotics Institute and the Institute for Software Research International in the university’s School of Computer Science. Currently he is chairman of the Information Science and Technology panel. Cross has a doctorate from the University of Illinois, a master’s in electrical engineering from the Air Force Institute of Technology, and a bachelor’s in electrical engineering from the University of Cincinnati. He is a graduate of the U.S. Air Force Test Pilot School, the Air War College, and the National Defense University. He retired from the Air Force in 1994.

Dr. Caroline P. Graetinger, Ph.D., leads the Accelerating Software Technology Adoption initiative at the Software Engineering Institute. She manages R&D activities to identify, develop, mature, and codify recommended practices for technology transition and change management. She also helps organizations develop capabilities for technology transition and change management. Graetinger has a doctorate, master’s, and bachelor’s degrees in electrical and computer engineering from Carnegie Mellon University.

Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213
Phone: 412-268-6109
E-mail: cpg@sei.cmu.edu

SOFTWARE ENGINEERING TECHNOLOGY

Get Your CROSS TALK
Free Subscription

Fill out and send us this form.

OO-ALC/TISE
7278 Fourth Street
Hill AFB, UT 84056
Fax: 801-777-8069 DSN: 777-8069
Voice: 801-775-5555 DSN: 775-5555

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

NAME:_____________________________

RANK/GRADE:_____________________

POSITION/TITLE:___________________

ORGANIZATION:___________________

ADDRESS:_________________________

BASE/CITY:_________________________

STATE:______ ZIP:__________________

VOICE:____________________________

FAX:______________________________

E-MAIL:__________________________

CHECK box(es) to request Back issues:

apr 1999____ aim for quality
May 1999____ climb to the top
Dec 1999____ evolution of software
Jan 2000____ lessons learned
Feb 2000____ risk management
Apr 2000____ cost estimation
May 2000____ into the future
June 2000____ keeping time
Nov 2000____ software acquisition
Dec 2000____ project management
Jan 2001____ modeling/simulation
Feb 2001____ measurement
Mar 2001____ process improvement
Apr 2001____ web-based apps
May 2001____ software odyssey

Does Your Software Project Rank as One of the Government’s Top Five?

The Department of Defense and CrossTalk are currently accepting nominations for the top five software projects in the government.

Outstanding performance of software teams will be recognized and best practices promoted.

Nominations will be accepted through July 20, 2001 at:
www.stsc.hill.af.mil/CrossTalk

Entrants will be selected based on adherence to quality and to the contracted cost, schedule, and requirements.

To be eligible, projects must have performed under a government contract (internal government contracts also eligible) and provided a software deliverable during the period of January 2000 through June 2001. The deliverable may be a completed project or an incremental deliverable.