



High-Leverage Best Practices – What Hot Companies are Doing to Stay Ahead and How DoD Programs Can Benefit

Norm Brown

Software Program Managers Network

Performance-based management based on 16 industrial-strength, commercial best practices is now within the reach of all software intensive programs. Program managers in government and industry now have a choice: whether or not to utilize these commercial best practices in controlling and managing their large-scale software-intensive projects. Commercial companies' survival is critically dependent on developing and fielding software in a predictable and disciplined manner. Performance-based management is critical to their success, and such management is typically dependent on effective deployment of a small set of 16 Critical Software Practices™ that address three key areas: product integrity, product construction, and project management. The Software Program Managers Network (SPMN) has captured and described these practices for potential use by defense programs.

Cutting-Edge Companies vs. Potemkin Organizations

Extensive benchmarking of commercial and industrial large-scale software development projects leads to a remarkable conclusion: commercial companies are employing high-leverage practices most Department of Defense (DoD) projects do not even think about. Although some of these commercial projects rival the exceedingly high size and complexity of many DoD projects, most do not. The use of these practices becomes even more important and critical as project size and complexity increase.

In the course of the Software Acquisition Best Practices Initiative, the

SPMN conducted, and still conducts, extensive industry best practice benchmarking. SPMN's benchmarking revealed that successful commercial companies are bottom-line driven and focus improvement activities on the big cost and schedule drivers. They are continually looking to identify specific ways to make dramatic improvements and to track their improvement progress in these high leverage areas. Can anyone imagine the Chrysler Corporation not knowing its scrap rate, not knowing the principal cost drivers for automobile manufacturing, or worse, knowing that it has a 42 percent scrap rate but doing nothing about it?

Cutting-edge companies, like

Motorola's Cellular Infrastructure Division in Cork, Ireland, have figured it out. These companies use best practices to control and manage their software development, not because they are particularly concerned about cost, but because they are obsessed with minimizing time-to-market — success in the software business. These practices also reduce cost as a byproduct of reducing development time.

Cutting-edge companies have learned to identify what drives bottom-line issues of schedule, cost, predictability, and customer satisfaction, and how to drive them in the desired direction. Over the past five years, the SPMN, an Army/Navy/Air Force software development support

SPMN Best Practices Background

The Department of Defense (DoD) has, for the last few years, been stressing the importance of deploying and leveraging commercial best practices in defense programs. In furtherance of this policy, the DoD in 1994 initiated the Software Acquisition Best Practices Initiative, and asked the SPMN to lead it. The Initiative's mission, built on work the SPMN had already accomplished, was to identify and convey these software acquisition best practices to service programs.

Among the Initiative's outputs was a set of nine fundamental best practices, that were recommended to the service acquisition executives in a memo from Noel Longuemare, Under Secretary of Defense (Acquisition and Technology) and Emmett Paige, Assistant Secretary of Defense (Command, Control,

Communications, and Intelligence) in 1995*.

That memo also provided for continuing best practice benchmarking and identification, and for support to programs desiring assistance in implementing these practices. Four years of continuing best practice research efforts have led the SPMN to integrate the nine fundamental practices with other essential best practices into a robust synergistic set of 16 critical practices.

The SPMN supports more than 200 programs, many of which express a fundamental need to improve how software is developed. As part of that support, SPMN consultants work with both the service program offices and their development contractors.

*Memorandum for Secretaries of the Military Departments Attn: Service Acquisition Executives, Vice Chairman, Joint Chiefs of Staff, Software Acquisition Best Practices Initiative, July 8, 1994.

organization, has benchmarked successful and not-so-successful software development activities across industry sectors in its quest to catalogue what does and does not work in the real world. The schedule and financial consequences of effectively using best practices should be obvious areas of attention for DoD software program managers.

Who is Doing What

For companies that develop software in the commercial sector, being first in the market says a lot about who will and will not survive in the global economy. It is not surprising that the commercial sector leads the way to adopting and effectively using best practices.

Motorola Iridium, in Scottsdale, Ariz., used best practices to cut defects by a factor of three, cut test time by a factor of four, and reduce overall development time by a factor of 10 — all while the software it was building grew in complexity by a factor of three. Although Motorola may have challenges with Iridium, these problems are not of the software variety. World-class companies cannot afford to allow their competitors to beat them to the marketplace. Survival of the fittest is the first law of the market.

In the defense sector, where about \$42 billion [1] is spent annually on software development and maintenance, there are serious financial disincentives to cutting cost and schedule:

- lower revenue stream
- lower profit
- reduced bonuses

In the past, the defense sector's view of the industry was the more waste you produce, the more you are paid, and that includes more profit. The only reason to adopt best practices in this sector is when the market is shrinking, as it is today, but the share going to software has substantially increased, not only as a percent of the total but in absolute dollars. For example, Raytheon's Electronic Systems Division, in Sudbury, Mass., cut rework by 81 percent, tripled productivity, and substantially increased predictability.

Best practices are a way of life at Lockheed Martin where implementation of best practices is targeted to save the corporation \$2.6 billion by 2003. At

Lockheed Martin's Ocean, Radar, and Sensor System, productivity has increased by a factor of six and errors have been reduced by a factor of 25 through the implementation of best practices.

Boeing is another success story that benefits from best practices. They carefully monitor their rework metrics and rework drivers, and have managed to bring their rework substantially below industry norms. That translates to substantially improved time-to-fielding and associated cost savings.

Dark Clouds on the Horizon for Software Waste

What will it take to make the rest of the defense sector take notice of the rampant waste that exists in software development? In the commercial sector, it is not inconceivable that stockholder derivative suits will be the driving influence on medium and large companies that develop software to support their operational mission and not as a commercial endeavor to adopt best practices. These suits will put corporate directors on notice as the ones responsible for excessive software development costs.

The congressional sector has become active on the taxpayer's behalf, and has become concerned and quite interested in programs using best practices. The House Armed Services Committee in its year 2000 defense bill directed, "mandate[ing] the use of identified best practices for software development and management for all acquisition programs" [2]. The Senate Armed Services Committee (SASC) in its year 2000 defense bill requested a report from each of the services on best practices implementation [3]. SASC is concerned that "...not enough has been done to adopt management best practices to the acquisition, development, and maintenance of software defense-wide" [3]. The SASC report requests that "...the [Defense] Department report to Congress by February 1, 2000 on its efforts to identify and adopt best practices in software development." It also requires including six specific metrics in the report.

If the department implements this direction, great improvement in cost and schedule will come to defense projects and to the entire defense industry.

A Fundamental Difference in Approach: Practices vs. Processes

Best practices and process improvement aim to achieve improvements in how software is developed. While process improvement serves as a floodlight on what can be done, offering a rich spectrum to choose from, best practices are the laser beam, pinpointing high-leverage activities directly coupled to the bottom-line. The bottom-line is improved by focusing on related implementation detail. The focus should not be merely on generic process improvement, but on what really counts. And what really counts are the underlying cost and schedule drivers — critical best practices that attack these cost and schedule drivers.

An important aspect of critical best practices is that they can be immediately applied. They are independent of the Software Engineering Institute (SEI) maturity level an organization may be at today — although you will have to tailor the practices to the circumstances of your specific project¹. Another aspect of critical best practices for organizations interested in moving-up the SEI Capability Maturity Model® (CMM) ladder is that these practices constitute a set of tactical disciplines that move organizations about 80 percent of the way to CMM Level 3, and a rapid, early implementation of high-payoff practices. The practices provide the tactical detail that the strategically oriented CMM does not address. Consultants in CMM improvement have indicated that SPMN's 16-Point Plan is an effective template for substantially reducing the 18-24 months it typically takes to create a CMM improvement plan from scratch.

Three Areas of Attack

The 16 Critical Software Practices have emerged as a product of the Airlie Software Council from its initial work in 1994-95 in identifying nine essential best practices. These form the core of the 16 Critical Best Practices, augmented with additional understanding of commercial practices by continued benchmarking and

"High Leverage Best Practices..." continued on page 14.

Fax: 407-242-4118
E-mail: lbearden@ge-harris.com

References

1. Paulk, Mark, et al., "Capability Maturity Model for Software, Version 1.1," Technical report CMU/SEI-93-TR-24, Software Engineering Institute, Pittsburgh, Pa., 1993.
2. Daskalantonakis, M.K., "Achieving Higher SEI Levels," IEEE Software, July 1994, Vol. 11, No. 4, pp. 17-24. Reprinted in *CROSS TALK*, September 1994, Vol. 7, No. 9, pp. 12-18.
3. Dunaway, Donna K. and Masters, Steve, *CMM Appraisal for Internal Process Improvement (CBA IPI): Method Description* (CMU/SEI-96-TR-007). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1996.
4. Dunaway, Donna K. *CMM-Based Appraisal for Internal Process Improvement (CBA IPI) Lead Assessor's Guide* (CMU/SEI-96-HB-003). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., 1996.

"High Leverage Best Practices..." continued from page 5.

project consulting experience. They have all been successfully tested in the crucible of successful large-scale software projects.

The Airlie Software Council identified three major areas of software development the 16 Critical Practices address:

- product integrity
- product construction
- project control

These areas and subsumed practices can be found developed further in this journal in Jane T. Lochner's article on page 6. The practices are useful for controlling complexity inherent in all large-scale software projects — and keeping it from spinning into uncontrollable chaos. Each practice makes a high-leverage contribution and are "high-leverage" practices because of the relatively low cost, quick implementation, and dramatic effect on the bottom line.

Where the Rubber Meets the Road

The critical practices and related implementation both defined in the 16-Point Plan were selected to deliver maximum leverage to programs wanting to dramatically improve their bottom-line and to expedite progress in organizations desirous of moving to the SEI CMM Level 3.

The CMM serves as a meaningful strategic framework for process improvement; the 16 Critical Practices constitute a tactical infrastructure that enables software development organizations to effectively address many of the CMM's Key Process Areas (KPAs).

Although these 16 Critical Practices serve this infrastructural role to CMM KPAs, their fundamental role is independent of this relationship to the CMM — they focus, at their essence, specifically on addressing improvements to the bottom-line — enabling significantly reduced time-to-field and related cost reduction and quality improvements. Although many of the CMM KPAs have similar, if not identical titles as critical process, they are largely two sides of the same coin.

This plan was devised to enable and facilitate an effective and straightforward implementation of critical best practices. Discussions with numerous consultants who assist organizations with CMM improvements make it clear that companies typically take between 18 and 24 months to design an improvement plan for getting from CMM Level 2 to Level 3; and during this time much of the process improvement momentum dissipates and

management support wanes. The 16-Point Plan can serve well as a template for reaching Level 3. Since CMM Level 2 has a significant focus on improvement in project management and Level 3 has a key focus on team effectiveness, the critical best practices address both of these key improvement areas.

What You Can Do

1. First determine whether or not your project has a detailed plan of all activities needed to achieve the next milestones, together with or including the personnel resources and time allocations necessary for this completion.

Although obvious, many programs lack this detailed planning. Without it, tracking by earned value will be meaningless, schedule compression cannot be completed, critical path and near-critical path cannot be identified through statistical schedule verification, tools cannot be employed, risk identification capabilities will be diminished, and you will not be able to use schedule automated control and authorization tools. If such a detailed plan does not exist, have one made.

2. Ensure that the effective structured peer reviews trend of a Fagan-like variety are being conducted to all detailed task products; that such reviews constitute task completion criteria for earned value and configuration management purposes; and that architectures are being modeled and simulated.
3. Ensure that a "bottom-up" risk management process is in place — one that has risk identification facilitated among front-line developers with management involvement; risk mitigation planning for high impact, high probability risks that a risk officer can manage and focus the process; and a culture that rewards risk identification — not punishes it. Be sure the likelihood of key development personnel suddenly leaving the project is considered as a major risk. If the project is planning a heavy reliance on reuse, then ensure that this is noted as a major risk as well.
4. Consider the 16 Best Practices and prioritize them in accordance with the needs of your particular program.

More About Best Practices

SPMN has also developed a template plan for large-scale defense projects: the 16-Point Plan for Performance Based

"High Leverage Best Practices..." continued on page 27.

The Team Handbook, Joiner Associates, Madison, Wis., 1996.

- Weinberg, Gerald, *Quality Software Management: Volume 4: Anticipating Change*, Dorset House Publishing, New York, 1997.

Notes

- Companies who have the need for parallel development of multiple releases use this concept. Although Sun has implemented this differently, the

release train idea described in http://solaris.license.virginia.edu/sun_microsystems/workshop4.2_docs/teamware/solutions_guide/casestudy.doc.html No. 8868 is similar in concept.

- The hole-in-the-floor model of change: Some set of people upstairs develops the perfect system. The change plan consists of drilling a hole in the floor. The system is dropped through to the people below.

Supposedly people instantly change to the new system. Unfortunately, people generally cannot change without integration and practice.

- Affinity grouping is the activity of creating sets of similar ideas together under one theme. In this case, we wrote each problem on a sticky note, silently organized the sticky notes into groups, and then named each group.

"High Leverage Best Practices..." continued from page 14.

Management. This identifies and describes in detail the 16 Critical Software Best Practices. SPMN, in coordination with the Airlie Council, also developed a related implementation handbook and compact disc, that is in beta testing, and that is being enriched through a review process by a tri-service group of program managers. These materials, along with related briefings, video clips, and other material are available without charge from the Software Program Managers Network at www.spmn.com. ♦

About the Author



Norm Brown is the change agent for more than 200 Army, Navy, Air Force, and other defense programs. He founded the SPMN, which conducts benchmarking of commercial best management and technical practices, and provides technical and management consulting support to these programs, and which has some 10,000 members. He founded the Airlie Software

Council, led the Defense Department's Software Acquisition Best Practices Initiative, and was a member of the DoD Software Management Review Board. He chaired the Joint Logistics Commanders Group on Technical Data and Computer Software and the group that drafted the Federal Acquisition regulations on

Technical Data and Computer Software, and numerous other groups and committees. He was responsible for the review of more than 300 defense software acquisition projects. He served in numerous acquisition selection boards and advisory positions to the Secretary of the Navy, Deputy Secretary of Defense, and other defense officials. Brown was the software manager for a number of real-time Navy weapons systems and was a commercial software developer.

Software Program Managers Network
P.O. Box 2523
Arlington, Va. 22202
Voice: 703-521-5231
Fax: 703-521-2603
E-mail: spmn@aol.com

References

- News Release, Office of Assistant Secretary of Defense (Public Affairs), Washington, D.C., July 13, 1994.
- House Armed Services Committee Report 106-162, pp. 198.
- Senate Armed Services Committee Report 106-50, pp. 340-341, May 17, 1999.

Note

- The Software Engineering Institute has developed the Capability Maturity Model, which identifies where a company lies along a continuum of software development maturity.

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@stsc1.hill.af.mil or mail to

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

Please limit letters to less than 250 words. Include your name, phone number, and e-mail address with any letter. We will withhold your name if you desire.