Standardizing Management of Software Engineering Projects

Roy Rada and John S. Craparo

For a software engineering division of a company, the most important standards are those used for the management of the software engineering projects. While numerous, relevant, *de jure*, software engineering standards exist, national guidelines, such as the Department of Defense's Capability Maturity Model, and corporate standards, such as the Microsoft Solutions Framework, exert a significant influence on the marketplace. A study of the existing standards shows significant similarity across them. Case studies of the use of the standards suggest that a major factor in determining the adoption of one standard over another is the environment of the adopter. A company that depends on Microsoft in important business ways is more likely to adopt the Microsoft Solutions Framework. Likewise a company that is a major customer of the Department of Defense may be inclined to use the Capability Maturity Model. The review of software engineering management standards also reveals the dearth of explicit guidance on the roles and processes to use in upper management. Financial process and audit practices influence the application and need for standards. An examination of the operation of a major manufacturing firm (referred to as Company X)

Roy Rada earned a B.A. in Psychology from Yale University in 1973, an M.D. in General Medicine from Baylor College of Medicine in 1977, and a Ph.D. in Computer Science from University of Illinois at Urbana in 1980. He has served as Editor of Index Medicus at the National Library of Medicine, Professor of Computer Science at the University of Liverpool, Boeing Distinguished Professor of Software Engineering at Washington State University, and now Professor of Information Systems at the University of Maryland at Baltimore County. Roy is a Fellow of the ACM and has authored 10 books and co-authored 150 journal papers. He may be reached at <rada@umbc.edu>.

John S. Craparo spent 10 years at GE Capital Services, where he served as the Senior Vice President & Chief Information Officer for three business units involved in equity investment, leveraged buyout, merger and acquisition, and structured project finance activities. He was also the CIO for corporate systems and global finance operations. He holds a bachelors degree from Iona College and a Master of Science in Technology Management from Polytechnic University (Brooklyn Poly). John sits on the CSIS advisory board of Pace University, where he served as an adjunct Associate Professor of Information Systems for 11 years. John sits on the editorial board of *Continuing Professional Development*. John has authored several commercial software programs, including a popular telecommunications circuit optimization tool. He may reached at <jcraparo@continentalu.com>

Knowledge, Technology, & Policy, Summer 2001, Vol. 14, No. 2, pp. 67-77.

shows again the challenge of managing software projects well and the remarkable success that can accrue from a systematic and standard approach across projects.

Introduction

Gupta and Oasem (1999) say "Companies that wish to survive and compete in this information era are finding that effective use and management of information technology is fast becoming the single biggest contributor to their competitive advantage and thus, also to their bottom line." The management of software development is particularly important because (deMarco, et al, 1999) the rapid pace of change in workplace practices brought about by information technology mandates special attention to management standards. The severe shortage of skilled software engineers also necessitates close attention to their proper management (Rada, 1999). What standards apply to the management of software development projects and what factors determine which standards are adopted by which organizations?

Software Engineering Standards History

One of the first consensus-based standards for software was for the programming language ALGOL in the late 1950s (Bergin and Gibson, 1996). Now numerous, *de jure* standards exist for software. This document focuses on standards for the development or management of software. For the U. S. government, the National Bureau of Standards in 1976 wrote FIPS Publication 38, *Guidelines for Documentation of Computer Programs and Automated Systems,* and organized a software life cycle around ten documents. IBM around this time also had developed and used its own proprietary software engineering standards that it did not share with the rest of the world because IBM considered those standards important to its competitive advantage (Moore and Rada, 1996).

While the 1970s and 1980s were a period of differentiation in software engineering standards, the 1990s were a period of consolidation (Moore, 1997). As a manifestation of the drive to standards with broad reach, the international standards community activity increased. The Software Engineering Subcommittee of Joint Technical Committee 1 of the International Organization for Standardization and the International Electro-technical Commission (ISO/IEC JTC1/SC7) developed numerous standards in the 1990s. The one most relevant to software project management, ISO/IEC 12207, was officially approved in 1995. The IEEE has an extensive library of software engineering standards that have been organized into 4 volumes (SESC, 1998): customer, process, products, and resources.

National governments recognize the critical character of software development and have subsidized particular organizations to develop and support guidelines. For example, the British Central Computer and Telecommunications Agency (CCTA) develops guidelines for the information

technology industry. Two CCTA products are the "IT Infrastructure Library" and "Projects in Controlled Environments." The primary objective of the "IT Infrastructure Library" is to establish best practices and a standard of IT service quality that customers should demand and providers should seek to supply? (CCTA, 2000). "Projects in Controlled Environments? PRINCE plans and controls a project primarily through a Product Breakdown Structure.

The Software Engineering Institute was founded by the United States DoD and has developed the Capability Maturity Model (CMM) to assess the capability of a software engineering organization. By considering what the objectives are as indicated by CMM, an organization can work backward to a definition of the roles and schedules that it needs to achieve a high level of capability.

While standards can be viewed from the perspective of history, other perspectives are also useful. One perspective is by topic, such as software quality, life cycle, or software project management. Another view is by originator of the standard, such as:

- developer of *de jure* standards,
- government-funded developer of guidelines, or
- corporate developer of proprietary standards.

Practically speaking a standard is simply what people use (Rada, 2000). The emphasis in this document is on software management standards.

Corporate Standards

When a private company that develops software for its internal use looks for a software project management standard, its main concern might not be whether the standard is *de jure*, de facto, or otherwise. Rather the concern is with the appropriateness of the standard to the corporate needs and perhaps the extent to which a complete solution is available. Software engineering standards from Microsoft are considered next.

Originally based on best practices within Microsoft product development, the Microsoft Solutions Framework (MSF) (Microsoft, 2000) was created in 1994 to promote consistency and effectiveness within the Microsoft Consulting Services organization. Research and customer feedback has contributed to the refinement of the MSF. Additionally, Microsoft offers training courses in MSF, and over 25,000 people have taken MSF courses.

MSF revolves around a team with a process. The team has well-defined roles played by a small number of people who interact on a peer-peer basis. The process is iterative and milestone-based and supports rapid prototyping.

Each MSF project team has exactly six roles: product manager, program manager, development, testing, user education, logistics planning. Each role may be assumed by a handful of different people, but what is important for the team is that the role discharges its responsibilities.

The MSF process model describes the phases, milestones, activities and deliverables of a project and their relationship to the team model. The process model's underlying practices and principles include: using versioned releases, scheduling for an uncertain future, managing trade-offs, managing risk, maintaining a fixed ship-date mindset, breaking large projects into manageable parts, performing daily builds, and using bottom-up estimating.

Tying together the roles and the processes are documents that are delivered by roles at precise points in the iterative process. MSF provides detailed document templates. For instance, the "vision and scope" document for a project has to be delivered at the beginning of a project with the input of all the roles and under the responsibility of the product manager. The "vision and scope" document template has over a dozen headings and guidance as to what should be provided under each heading.

Other Factors Influencing Standards

Failed software projects like those described by Glass (1997) have led several companies to adjust their accounting methods. The adjustments, in part, have come in the wake of requirements by accounting standards bodies like the American Institute of Certified Public Accountants (AICPA). Standard Operating Procedure 98-1 was issued in order to provide more specific guidance to firms about how to account for the cost of software developed for internal use (Munter, 1999). The guide has led many firms to shorten the financial life of software and increase the threshold for the capitalization of software projects—as we found at Company X discussed later in this paper. Both of these changes keep bad software off the company balance sheet. In the former case, failed software projects that are considered impaired assets could naturally be removed from the balance sheet at the earliest date possible. In the latter case, a threshold of $500,000 creates an environment that limits the number of projects that can make their way to the balance sheet in the first place. Possible financial impairment becomes moot. These projects are expensed immediately.

Complexity is created as firms begin to choose various parts of standards in an à la carte manner. This menu approach to corporate standards may help reflect the culture of an organization; however, it makes the task of control far more complicated. Jones, et. al. (1996) has attempted to reduce complexity by introducing activity based modeling tools based upon those used in cost accounting. Measuring time on task in a granular way may lead to greater control when using standards created by different bodies. The Information Systems Audit and Control Foundation has a different approach. While this body is mainly concerned with often-competing security and control standards, it offers the COBIT framework (ISACF, 1996) that can be used to build eclectic standards into an organized set of generally applicable and accepted practices.

There is a subtle but powerful influence that support organizations like Finance and Audit have on software engineering standards. The external

influence is not necessarily driven by culture, but by a need to control complexity and predict the return-on-investment of a software asset.

Case Studies with Corporate Standard

Best software engineering practices prove often to not be widely adopted in industry (Dutta et al, 1998). What does one find in looking at the adoption of the software project management methodologies of Microsoft? Descriptions of the use of MSF by an anonymous Company X, by Damgaard, and by Navision are presented.

Company X

An informal case study of software management standards at a major, global manufacturing company is next presented. The company prefers to remain anonymous and will be called here Company X. Company X manufacturers to a global audience, incorporates Microsoft products in its own product line, and extensively uses Microsoft products internally. The Company intends to do the majority of its customer transactions on the Internet and already is a world-leader in that regard. Company X employs about 40,000 people of whom about 2,500 work in the Information Technology (IT) Division that services internal IT needs.

The IT Division develops the information technology infrastructure of Company X in collaboration with the business units of X that use this infrastructure. The approach to standardization of method began with a strict division-wide approach but evolved in the early 1990s into a loose approach. The loose approach hindered productivity, and in 1998 a division-wide standardization of team methodology was initiated. The evolution from one approach to another was consistent with the model of internal standardization change espoused by Monteiro and Hanseth (1999).

The IT Division reviewed available standards that it could adopt that would be consistent with its principles. Its principles are that teams should work closely with business units and teams should act quickly and creatively. To support such teams the IT Division chose the Microsoft Solutions Framework (MSF).

In early 1999, all the IT project managers were trained in the use of MSF. The *IT Policy* document states that every new project must comply with the MSF. To audit the success of the software project management standardization, a survey was conducted in late 1999. Individuals from every major unit—a total of approximately 40 individuals—were surveyed. The survey included about thirty short-answer questions and was followed-up as appropriate with in-depth interviews. The response rate for the survey was one hundred percent.

The survey of the impact of MSF on the division behavior showed that typical project teams were performing similarly to how they had before the introduction of MSF and that the diffusion of MSF throughout the organization had not proceeded as far as intended. A couple of units had gone

fully into MSF mode. Those units reported that they were very comfortable with MSF and that the company standardization effort was helping. However, several units were annoyed at the intervention by the corporate headquarters in the way of working of the unit and felt that working to MSF as a standard would be a bureaucratic nuisance rather than a productivity enhancer. Those who were opposed to working with MSF understood the corporate plan but were able to proceed with management methods largely as prior to the corporate standardization effort. No formal compliance organization per se existed, and there were few repercussions for not following the directive to adopt the MSF team and process model.

The MSF describes in great detail how to run a project but does not say how to run that part of the organization that needs to coordinate across many MSF projects. The Company X IT Division has, of course, a current way of running the projects and extensive documentation to support that management. This documentation constitutes an "Organizational Manual" that is being refined to take advantage of the impetus provided by the MSF for improved performance of individual projects. To support coordination across projects the following four management processes have been identified as useful:

1. Standards,
2. Training,
3. Quality Control, and
4. Consulting.

This approach is consistent with that taken by other corporations (see for instance, Haley et al, 1995) but is not prescribed per se by MSF. In further detail:

- The IT Division's "Standards" process addresses the continual refinement of MSF to suit the Company's particular situation and the development of new standards to address coordination across projects. For instance, one standard might say that whenever three or more defects have been identified in a delivered product, then the project team that developed the product is penalized "Training" assesses the competencies of employees and provides training to address competency gaps.
- "Quality Control" monitors the performance of projects to determine compliance with the standards. Projects that are identified as non-compliant are referred to Consulting.
- "Consulting" helps projects achieve compliance.

Each of these four processes is associated with a team that implements the process, and the four teams work closely with one another in a loop from what is defined by the Standards team, to what is taught by the Training team, to what is monitored by the Quality Control team, to what is remedied by the Consulting team.

In early 2000, it became clear to Company X that enhancements to this model had to be made in two areas. "Standards" and "Quality Control" were

not consistent across application development teams or across their corresponding geographic regions. The Organizational Manual and the MSF process did not insure that projects were completed. In March 2000 the organization made an assessment of all strategic software development projects underway in the organization. Strategic projects were defined as those which create income, enhance manufacturing, or increase customer satisfaction. Two other project categories were defined: Tactical and Baseline. The former represent prerequisite work needed to insure the success of a strategic project. An example would be the CPU upgrade of the company's mainframe computer to support a new three-tier order management system. Baseline projects are defined as work needed to "keep the lights on." An example is the reorganization of the internal end-user helpdesk in order to lower company operating expenses.

As part of the assessment, a scorecard was created that measured the success of IT development across several measures over the previous twenty-four months. The results of the scoring exercise surprised the IT management organization. During the previous two years, 260 Strategic software development projects were started and three were completed. In addition, the full time equivalent (FTE) Software Developer population was 650. Of that number only 30 percent or 195 of the FTE population was dedicated to developing or database modeling. The rest were dedicated to business analysis, administration, and MSF training. The ratio did not make sense to management.

These facts led to a decision to make several changes to the Company X software development approach.

- Strategic Projects would be limited to 15 at any moment in time.
- FTE would be realigned so that 75 percent of personnel would be dedicated to development activities.
- All Strategic Projects would require a Cost Benefit Analysis and Business Sponsor.
- Each Strategic Project would undergo a detailed review by the senior CIO staff every 6 weeks.
- Each Project MSF Phase would require signoff by a newly formed Technical Review Team.
- The Organizational Manual would be reconstituted as a 3 year Enterprise Architecture Plan based upon Capability Maturity Model criteria.
- Standards related to the number of projects that could be practically worked toward a successful end, closer scrutiny of goals, and greater specificity created in the Enterprise Architecture plan are bearing fruit. Through August 2001, the organization has completed 12 of the first 15 Strategic Projects on time, on target and within budget.

Damgaard and Navision

Damgaard develops, sells, and services enterprise-resource planning software and employs about 500 people. To quote Damgaard (2000): "Damgaard's products are developed solely with Microsoft technologies in

mind."Damgaard has adopted completely the MSF as a way of managing its project teams (Microsoft, 1999). In the switch to MSF a few years ago, staff members were moved from the previous setup of departments defined by function into teams aligned by product. The entire Development Division was re-organized into MSF teams with roles and responsibilities from across all disciplines represented within each team, including developers, testers, sales and service. An important part of the project was physically putting each team into shared office space. Damgaard implemented quality control and consulting teams to look into errors and to act upon them. Statistics from this quality control and consulting activity are an integral part of Damgaard's release procedure.

Navision has 550 employees and over 30,000 clients. It products run primarily on a Microsoft infrastructure. Navision has also committed to the MSF (Microsoft, 1999b). The ability to implement MSF incrementally suits Navision's way of handling growth. The MSF Team model has provided a shared reference for procedures and terminology. Communication within and between teams is facilitated. MSF program manager Peter F. Jørgensen comments,"With MSF, we don't need to discuss organization and responsibilities every time we start up a new project. Everyone on the team knows their role and the process we are going through. This improves both the development process and communication in general."

Daamgaard is an example of a firm that has embraced the software management theory of its captive sales-channel software platform associate Microsoft. Navison favors standardization which helps control growth and enhances speed. Cultural influence, strategic partnering, strong management control, and perhaps fear of the unknown limit these organizations from straying too far from out-of-the-box MSF. On the other hand, we see Company X highly customizing its software development process to the point that MSF has become the spirit and not the law.

De Jure and National Guideline Cases

Three cases are presented of organizations using *de jure* standards or national guidelines. The Boeing Company follows various *de jure* standards, as dictated by its customers. AlliedSignal Aerospace and Raytheon are presented for their adherence to the CMM quality assessment method.

The Boeing Company <www.boeing.com> manufactures aerospace products, and Boeing's software engineering projects employ thousands of people. However, Boeing does not use one software development method across all projects. A significant portion of Boeing's business is for various agencies of the US Federal government, including DoD. While these agencies strive to follow international standards, they continue to have agency-specific standards for specialized systems development. When these government agencies purchase systems from a vendor, they may well require that the systems are development in accord with the specific standards advocated by the agency. Thus different software development projects within Boeing may be required to use different development standards.

However, Boeing's IT Division does closely follow a high-level quality standard and structured methodology that emphasizes teamwork, performing to target, and training employees.

AlliedSignal Aerospace is another large American company with a substantial business with DoD. DoD strongly encourages its vendors to demonstrate a high-level of maturity according to the Capability Maturity Model (CMM is funded by DoD). Not surprisingly, AlliedSignal Aerospace committed itself to corporate-wide software process improvement in 1992 by using the CMM. The improvement initiative covered all software development and maintenance activities in product development (Buchman, 1996). The improvement effort was managed centrally by a Software Technology Council and was implemented and funded locally in order to maximize the improvement potential. The results were impressive as some sites went from Level 1 to Level 3 on the CMM scale within three years.

Raytheon is yet another large American company doing much of its business for the DoD. Raytheon also chose the path of process improvement, guided by the CMM. Raytheon has been able to demonstrate sustained, significant, and measurable improvements to its software engineering process (Haley, et al, 1995). Raytheon organized its initiative into an executive committee responsible for steering and oversight, and into four Working Groups, each responsible for a major area in the initiative. The Policy and Procedures Group initially captured and documented best practices so that they could be applied across all projects. The Training Group elevated the importance of training from *ad hoc* "on the job"learning to a full commitment of the software organization to ensure that each project had its engineers fully trained before beginning work. The Tools and Methods Group developed the technologies (CASE tools, workstations) and the methods (Ada, object-oriented). The Process Database Group developed the process and quality metrics and statistical process control to assess the performance of both projects and the process. These working groups tailored the process to be effective within the Raytheon culture. The projects and the process worked together to achieve increases in predictability, productivity, and quality.

Conclusion

Companies must choose standards that fit with their internal corporate culture. West (1999) has shown that choices of standards are often highly influenced by organizational factors. Companies that already work closely with Microsoft Corporation are naturally inclined to consider a software development methodology supported by Microsoft. Companies for whom the DoD is a major client are inclined to follow methods supported by the DoD, such as CMM. The case studies in this document support that tendency. The companies that adopted MSF were committed to Microsoft in various business ways, and the companies that followed CMM were major customers of the DoD.

If an organization has many software engineering teams, then some standardization of methods across teams is important. Deployment of a corpo-

rate software engineering organizational standard with consistent roles and schedules across projects can lead to various benefits, including:

- Better planning, both short-term and long-term, leading to more predictable outcomes.
- Increased staffing flexibility (decreased sensitivity to employee turn-over) as well-defined roles and processes repeated across project teams facilitate reuse of experience.
- Organizational learning which drives a higher quality product, at lower cost, and in less time.
- Financially viable information assets with little chance of becoming an impairment on the firm?s balance sheet.

These benefits might not be seen in the life of a given project easily but will be obvious across an organization and over the lives of many projects.

A wide range of standards for managing software projects is available. These standards focus on the software life cycle and on quality assurance for software projects. The enforcing of consistent standards across many projects is, however, perhaps the least well understood problem in software management. How to manage many simultaneous projects depends on many issues of which the proper choice of tools and further levels of management are prominent.

CCTA has advanced broad strategies for corporate-wide management of software projects and emphasizes setting direction, implementing plans, and managing assets. More specifics are needed. To the extent that a software development methodology for a project embraces various principles of operation, one might hope that the management across projects would be consistent with the management within projects. For example, with MSF one might imagine generalizations of the six project roles so that they would correspond to senior management roles. Continuing this example, a company might want at the upper levels of management a product manager (marketing vice-president) and a program manager (chief operations officer).

For a software engineering division of a company, the most important standards are those used for the roles and processes of their software engineering teams. Some standards are developed internally by a company for its sole use, some standards are products developed and marketed by for-profit companies to other companies, some standards are the result of government funding to national centers, and some are the product of formal standards development organizations. The issue is less that one standard is better or worse than another, and more that the company choose a particular standard and consistently work to it.

References

Bergin, Thomas and Gibson, Richard (1996). *History of Programming Languages*, Addison-Wesley Press.

Buchman, Caroline, D. (1996) "Software process improvement at AlliedSignal Aerospace" *29th Hawaii International Conference on System Sciences,* ed. H. El-Rewini, IEEE Comput. Soc. Press: Los Alamitos, CA.

CCTA (2000) IT Infrastructure Library Online, <http://www.ccta.gov.uk/itil/index.htm>.
Damgaard (2000) Damgaard and Microsoft: A Strategic Alliance Full of Benefits? <http://www.damgaard.com/Damgaard/Damgaard.nsf/All/CORPORATEPartners-Microsoft>.
DeMarco, David A., Jonathan D. Rabson, and Michael Seebacher (1999) The Human Characteristics for Innovation in the Development of Information Technology? *1st IEEE Conference on Standardisation and Innovation in Information Technology*, Aachen, Germany, Sept. 15-17, edited by Kai Jakobs and R Williams, IEEE Press (also available at <http://www-i4.informatik.rwth-aachen.de/~jakobs/siit99/Proceedings.html>).
Dutta, S, Lee, M. A., Van Wassenhove, L. (1998) Adoption levels of software management best practices in European organizations? *European Conference on Software Process Improvement*, p. 450-456, Meeting Management: Farnham, UK.
Glass, R. L. (1997) *Software Runaways*. Prentice Hall: NJ.
Gupta, T. and Qasem, A. (1999) IT diffusion: national best practices? *Information Resources Management Association International Conference*, edited M. Khosrowpour, Publisher Idea Group Publishing: Hershey, PA.
Haley, T. B. Ireland, E. Wojtaszek, D. Nash, and R. Dion, "Raytheon Electronic Systems Experience in Software Process Improvement," Carnegie Mellon University, Software Engineering Institute, CMU/SEI-95-TR-017, November 1995.
ISACF (1996) *COBIT: Control objectives for information and related technology*. Information Systems Audit and Control Foundation: Rolling Meadows: IL.
Jones, Capers. and Jones, T. Capers. (1996) *Applied Software Measurement: assuring productivity and quality*. McGraw Hill: NY.
Microsoft (1999) Microsoft Energizes Damgaard Developers? <http://www.microsoft.com/msf/downloads/MSF_at_Damgaard.doc>.
Microsoft (1999b) Navision Software Development Done Right? <http://www.microsoft.com/msf/downloads/MSF_at_Navision.doc>.
Microsoft (2000) Microsoft Solutions Framework: Turning Technology Projects into Business Solutions? <www.microsoft.com/msf/>.
Monteiro, Eric and Ole Hanseth (1999) Developing corporate infrastructure: implications for international standardization? *1st IEEE Conference on Standardisation and Innovation in Information Technology*, Aachen, Germany, Sept. 15-17, edited by Kai Jakobs and R Williams, IEEE Press. (also available at <http://www-i4.informatik.rwth-aachen.de/~jakobs/siit99/Proceedings.html>).
Moore, James and Roy Rada, Organizational Badge Collecting? *Communications of the ACM*, 39, 8, pp 17-21, August 1996.
Moore, James W (1997) *Software Engineering Standards: A User's Road Map*, IEEE Computer Society Press: Los Alamitos, CA.
Munter, Paul (1999), Accounting for Software Development Costs? The CPA Journal, February. http://www.nysscpa.org/cpajournal/1999/0299/Features/F420299H.HTM
Rada, Roy "Sharing standards: IT Skills" *Communications of the ACM*, 42, 4, pp 21-26, April 1999.
Rada, Roy (2000) "Consensus versus Speed" *Information Technology Standards and Standardization: A Global Perspective* edited by Kai Jakobs, Idea Group Publishing: Hershey, PA, pp 19-34.
SESC (1998) Software Engineering Standards Committee Annual Report, Standards Making Organization of the IEEE Computer Society Technical Council, <http://www.computer.org/standards/sesc/annual98.html#AttachmentB>.
West, Joel (1999) Organizational Decisions for I.T. Standards Adoption: Antecedents and Consequences? *1st IEEE Conference on Standardisation and Innovation in Information Technology*, Aachen, Germany, Sept. 15-17, edited by Kai Jakobs and R Williams, IEEE Press (also available at <http://www-i4.informatik.rwth-aachen.de/~jakobs/siit99/Proceedings.html>).