2004

Surviving the Quagmire of Process Models, Integrated Models, and Standards

Mark C. Paulk
Carnegie Mellon University

Surviving the Quagmire of Process Models, Integrated Models, and Standards*

Mark C. Paulk
Senior Systems Scientist
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Email: mcp@cs.cmu.edu

Summary

We frequently talk about the “quagmire” of process models and standards: ISO 9001, Software CMM, CMMI, ISO 12207, etc. There are valid reasons for the diversity of models and standards imposed on our software development and maintenance processes, just as there is a legitimate concern that a high overhead is imposed by customers and markets that require multiple models and standards as a condition of doing business. This presentation describes the drivers behind different models and standards, why they are different, two strategies for integrating diverse models and standards from a model/standard developer’s perspective, and some of the strategies that model implementers can use. The business reality is that software organizations need to address a variety of models and standards, now and in the future, but there are mechanisms that can alleviate some of the overhead when thoughtfully used.

Introduction

Several years ago, I was chatting with a colleague who worked for a major aerospace firm. He reported that in the last four months, his organization had undergone two software capability evaluations (SCE), one software development capability/capacity review (SDC/CR), one software development capability evaluation (SDCE), and an ISO 9001 surveillance audit. They were preparing for a CMM-based appraisal for internal process improvement (CBA IPI). His concern was that his 300-person organization was so busy doing assessments, evaluations, and audits that it had no time left for building software.

At about that time, Sheard used the term “quagmire” to refer to the abundance of models and standards that were, and are, being imposed on software organizations [Sheard, 1997; Sheard, 2001]. The term “quagmire” implies a slow descent into the grave (or the nether regions for the less optimistic). While not a unique problem for the software industry, I will focus on systems, software, information technology (IT), and quality frameworks that affect software organizations. This discussion does not exhaust the possibilities, but it does address many of the instances of concern to software engineers.

The Quagmire of Models and Standards

If there is a problem, as implied, has it been addressed in the past few years as “integrated” models and standards “harmonization” has progressed? A number of models and standards exist that are focused on software, IT, or quality topics. The term “framework” refers to models and standards that have a variety of issuing bodies, scopes, architectures, and rating methods. These frameworks include:

- general Total Quality Management (TQM) philosophies such as those of Deming [Deming, 1986; Deming, 1994], Juran [Juran, 1992], and Crosby [Crosby, 1979]
- performance excellence strategies such as Six Sigma [Harry, 2000]
- the criteria for quality awards such as the Malcolm Baldrige National Quality Award in the United States [Baldrige]

Reprinted with permission from **Proceedings of the ASQ Annual Quality Congress**, 24-27 May 2004, Toronto, Canada.

- standards such as
  - ISO 9001 (Quality Management Systems – Requirements) [ISO9001, 2000]
  - Control Objectives for Information and related Technology (CobiT®) [CobiT]
  - ISO/IEC 12207 (Software Life Cycle Processes) [ISO12207, 2002]
  - ISO/IEC 15288 (System Life Cycle Processes) [ISO15288, 2002]
  - ISO/IEC 15504 (Software Process Assessment) [ISO15504-2, 1998]
  - BS 7799 (Information Security Management Systems) [BS7799-2, 2002]
  - BS 15000 (IT Service Management) [BS15000-1, 2002]
- process improvement models such as
  - Capability Maturity Model® (CMM®) for Software [Paulk, 1995]
  - Systems Engineering CMM [Bate, 1995]
  - Software Acquisition CMM [Cooper, 2002]
  - People CMM [Curtis, 2001]
  - eSourcing Capability Model [Hyder, 2004]
  - CMM Integration℠ (CMMI®) [Chrissis, 2003]

As can be readily observed from this lengthy, but hardly comprehensive, list of frameworks, software organizations have an abundance of resources to aid them in quality and process improvement. One survey identified 315 standards, guides, handbooks, and other prescriptive documents maintained by 46 different organizations [Moore, 1999]. The IEEE's Software Engineering Standards Committee has issued a four-volume set of software standards for customer (including systems engineering), process, product, and resource volumes. Some frameworks, such as the Baldrige Award, have a broader scope than software engineering. Some, such as the Software CMM, are focused on software and provide relatively detailed guidance for the discipline. Some, such as BS 7799 (Information Security Management Systems), provide detailed requirements for a specific topic.

The issuing bodies for these frameworks include standardization bodies such as the International Organization for Standardization (ISO), which has a joint committee on IT standards with the International Electrotechnical Commission (IEC); professional societies such as the Institute of Electrical and Electronics Engineers (IEEE), which has established a Standards Association to administer its standards-related work, and the American Society for Quality (ASQ); national bodies, such as the American National Standards Institute (ANSI) and the British Standards Institute (BSI); government-sponsored centers, such as the Software Engineering Institute (SEI); industry organizations, such as the Electronic Industries Association (EIA); plus various others. In short, anyone can issue a standard or publish a model, and sometimes it seems like just about everyone has. The number of well-known and generally accepted models is, however, relatively limited.

Perhaps more daunting than the number is the observation that customers may impose a variety of these models and standards on their software suppliers. Even if each of these frameworks provided a significant and unique value-added increment to an organization's capability, the diversity of emphases and perspectives could be counter-productive. The predominant frameworks that software organizations have to deal with are ISO 9001, the Software CMM, CMMI, and ISO/IEC 12207.

ISO 9000 (Quality Management Systems) is a suite of standards dealing with quality management systems that can be used for external quality assurance purposes. ISO 9001, which contains the requirements for a quality management system, is the standard of specific interest to the software community, but it is much broader in scope than just software. Although formerly biased towards a manufacturing environment, the latest release (2000) removed much of the manufacturing bias. Sector-specific variants, such as QS9000 and TL9000, provide recommendations for adopting ISO 9001 in specific environments, e.g., the automotive and telecommunications industries. Software-specific guidance for interpreting ISO 9001 is available in ISO 9000-3, which is currently being revised to reflect the 2000 release.

® COBIT is registered in the U.S. Patent and Trademark Office by the Information Systems Audit and Control Association.

® Capability Maturity Model, CMM, and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

℠ CMM Integration is a service mark of Carnegie Mellon University.

The Software CMM is a five-level model that describes the process maturity of software organizations and provides a roadmap for software process improvement. The five levels describe stages of organizational transformation, thus it is referred to as a staged model. Developed by the SEI at the request of the U.S. Department of Defense (DoD) to help identify the capability of software contractors, its use for improving the software process has spread far beyond the DoD community. The SEI distinguishes between two forms of appraisal against the CMM: assessments for process improvement and evaluations for source selection and contract monitoring. The Software CMM is arguably the best known and most widely used model for software process appraisal and improvement today, but the DoD has directed that it be retired in favor of CMMI.

CMMI integrates systems engineering, software engineering, integrated product and process development (IPPD), and acquisition into a single model. It supports two different architectures (continuous, similar to ISO/IEC 15504, and staged, similar to Software CMM). It is intended to be both comprehensive and easily extensible. Proponents of the Software CMM emphasize the importance of focusing on the software discipline. Proponents of CMMI emphasize the need for an integrated approach to appraising capability and improving processes. The SEI's sponsor in the DoD has chosen the integrated approach. Guidance on how to interpret CMMI for IT and software-only organizations is being developed [CMMI Guidance].

ISO/IEC 12207 (Software Life Cycle Processes) establishes a common framework for software throughout its life cycle from conception through retirement. It addresses the organizational context of those software processes both from the system's technical viewpoint and from the enterprise's business viewpoint. It is defined at the process rather than the procedure level, with three categories of processes: primary (business and technical), supporting, and organizational. The IEEE has adopted ISO/IEC 12207 with some extensions. IEEE/EIA 12207.0 is a "sandwich" that wraps adds a foreword and some annexes around the full text of ISO/IEC 12207. IEEE/EIA 12207.1 provides guidance on the data produced by the life cycle processes. It describes 84 different information items but not documents.

Many models and standards for software process improvement have been developed. This proliferation led to the development of ISO/IEC 15504, a suite of standards for software process assessment. Popularly known as SPICE (Software Process Improvement and Capability dEtermination), ISO/IEC 15504 provides a framework for harmonizing different approaches to assessing and improving the software process. The process capability dimension in ISO 15504 is characterized by a series of process attributes, applicable to any process, which represent measurable characteristics necessary to manage a process and improve its capability to perform. This capability dimension can be applied, in principle, to any process, e.g., the processes in ISO/IEC 12207 or ISO/IEC 15288 (System Life Cycle Processes). It defines the continuous architecture that has been adopted in CMMI. ISO published technical reports on ISO/IEC 15504 in 1998 as a stage in developing an international standard. The standard has not been released at this writing, and it is unclear how significant an impact it may have. When used for harmonization, it is primarily a set of reference requirements, but an exemplar model and appraisal method are planned.

Many other frameworks are likely to affect software organizations. A partial list is contained in Table 1, but a comprehensive list is infeasible in a paper, although the IEEE Standards Association and the ASQ periodically report on the status of standards work [Duncan, 2004], and frameworks are continually being revised and replaced, so the life span of any static list is limited.

Table 1 Partial List of Models and Standards Affecting Software Organizations

Framework	**Focus**	**Source**	**Status**
Baldrige	Malcolm Baldrige National Quality Award	US Department of Commerce	Revised annually
CBA IPI	*CMM-Based Appraisal for Internal Process Improvement method*	*SEI*	*Superseded by SCAMPI*
CMMI	Capability Maturity Model Integration: systems engineering, software engineering, integrated product and process development, and acquisition	Government/industry consortium with SEI	v1.1 release in 2002
CobiT	Control Objectives for Information and related Technology: corporate governance	Information Systems Audit and Control Association	v3 released in 2000

Framework	**Focus**	**Source**	**Status**
DOD-STD-2167A	*Software development*	*DOD*	*Superseded by MIL-STD-498*
DOD-STD-2168	*Software quality*	*DOD*	*Superseded by MIL-STD-498*
DOD-STD-7935A	*Documentation*	*DOD*	*Superseded by MIL-STD-498*
EIA 632	Systems engineering	EIA and INCOSE	Released in 1999
EIA 731	*Systems Engineering Capability Model*	*EIA*	*Superseded by CMMI*
eSCM-SP	Service providers of IT-enabled services	Carnegie Mellon University	v2 released in 2004
FAA iCMM	Systems engineering, software engineering, and acquisition	Federal Aviation Administration	v2 released in 2001
IEEE 1220	Systems engineering	IEEE	Released in 1998
IEEE/EIA 12207	US adaptation of ISO/IEC 12207	IEEE and EIA	- - -
IPD-CMM	*Integrated Product Development CMM*	*Industry consortium with SEI*	*Superseded by CMMI*
ISO 15939	Information Technology—Software Measurement Process	ISO	Released in 2002
ISO 9000	Quality management systems	ISO	Released in 2000
ISO/IEC 12207	Software life cycle processes	ISO and IEC	Released in 1995, amended in 2002
ISO/IEC 15288	System life cycle processes	ISO and IEC	Released in 2002
ISO/IEC TR 15504	Software Process Improvement Capability dEtermination (SPICE)	ISO	TR released in 1998
J-STD-016	*Software development, quality, documentation*	*IEEE and EIA*	*Superseded by ISO/IEC 12207*
MIL-STD-498	*Software development, quality, documentation*	*DOD*	*Superseded by J-STD-016*
MIL-STD-499B	*Systems engineering*	*DOD*	*Superseded by commercial standards*
People CMM	Human resource management	SEI	v2 released in 2001
PSM	Practical Systems and Software Measurement	DOD and US Army	v4 released in 2001
PSP	Personal Software Process: individual programmers	SEI	Released in 1995
Q9000	US version of ISO 9000	ASQ	- - -
QS 9000	Automotive industry adaptation of ISO 9000	General Motors, Chrysler, and Ford	- - -
RTCA DO-178B	Aviation software systems safety	Radio Technical Commission for Aeronautics and FAA	Released in 1992
SA-CMM	Software acquisition organizations	Industry consortium with SEI	v1.03 released in 2002
SCAMPI	Standard CMMI Assessment Method for Process Improvement	SEI	v1.1 released in 2001
SCE	*Software Capability Evaluation (source selection and contract monitoring)*	*SEI*	*Superseded by SCAMPI*
SDC/CR	*Software Development Capability/Capacity Review (source selection)*	*US Air Force*	*Superseded by SDCE*
SDCE	Software Development Capability Evaluation (source selection)	US Air Force	Released in 1995

Framework	Focus	Source	Status
SECAM	*Systems Engineering Capability Assessment Model: Systems engineering*	*International Council on Systems Engineering (INCOSE)*	*Superseded by CMMI*
SE-CMM	*Systems Engineering Capability Maturity Model: Systems engineering*	*Industry consortium with SEI*	*Superseded by CMMI*
SW-CMM	*Software CMM: Software development and maintenance*	*SEI*	*Superseded by CMMI*
TL 9000	Telecommunications adaptation of ISO 9000	QuEST forum (Quality Excellence for Suppliers of Telecommunications Leadership)	- - -
TSP	Team Software Process: programming teams	SEI	Released in 1999

In principle, many of the frameworks listed in Table 1 should have been retired. For example, all of the U.S. military standards have been retired in favor of commercial standards, e.g., ISO 12207. The practical reality, however, is that many long-term projects may still have to comply with frameworks that have been retired, and organizations may still choose to use frameworks, such as the Software CMM, that are no longer officially supported. This complicates an already complex picture, but the set of software-specific frameworks should, in principle, have become simpler in recent years. I suspect that reality and principle have not yet converged for many software organizations.

Drivers Behind the Frameworks

If customers were satisfied with the products and services that they received, most of these frameworks would probably not exist. The term "software crisis" is, however, well-known to buyers and users of software [Gibbs, 1994]. In a 2000 study, the Standish Group identified 23% of software projects as failures and 49% as challenged in meeting cost and schedule targets [Standish, 2000]. This is an improvement over their 1994 study, where the percentages were 31% and 53% respectively, but it is still unsatisfactory.

Models such as the Software CMM can help an organization systematically grow its capability to meet commitments and build software effectively and efficiently. Standards can provide assistance and can protect the buyer by [Moore, 1999]:

- providing a vocabulary for communication between the buyer and seller
- providing objective criteria for otherwise vague claims regarding the product's nature
- defining methods for characterizing elusive characteristics such as reliability
- assuring the seller that specific quality assurance practices were applied

Numerous studies have been published reporting the effectiveness of various frameworks in achieving these goals [Clark, 2000; Harter, 2000]. Some frameworks may be more effective than others in some contexts. For example, one study found that both ISO 9001 and the Software CMM improved product attributes and return on quality for Indian software organizations, but CMM highly rated firms were better than ISO 9000 certified firms [Isaac, 2003].

Different issuing bodies have focused on different aspects of the software problem. Enterprise-level frameworks, such as the Baldrige Award, address business and general process and quality issues. Systems-oriented frameworks, such as CMMI, address problems at the hardware/software systems level. Software-oriented frameworks, such as the Software CMM, focus on software management and engineering issues. A myriad of standards and models address specific processes, work products, and "ilities" for software. Each provides value from its perspective, but coordinating and integrating the variety of frameworks to realize a positive benefit is a non-trivial task.

Integrating Multiple Frameworks

One way of dealing with the quagmire of models and standards is to select the "best" framework and stick with it. I would recommend the Software CMM, but that is clearly a biased recommendation on my part since I led the Software CMM team. "Best" is likely to depend on where you stand. For IT and software-only organizations, the Software CMM may well be the best single framework, even though the SEI is no longer supporting it. For many hardware/software systems builders, CMMI may be an appropriate choice. Even for organizations that have reasonably full control of their engineering environment, it may make sense to adopt ISO 9001 initially, after certification use the Software CMM, and, upon reaching the higher maturity levels, implement a measurement-based strategy, such as Six Sigma. This strategy has been successfully used by many software organizations that considered it crucial to be ISO certified, but found that the Software CMM guidance was inadequate at Levels 4 and 5.

"Picking the best" is moot if your customers mandate that particular frameworks be used. If a potential customer uses a framework such as the Software CMM or CMMI in selecting and/or managing its suppliers, it seems reasonable to use the same framework in defining and improving your processes. Some frameworks, e.g., ISO 9001, may be expected in some markets. If you have multiple customers using a variety of frameworks, then integration becomes a concern.

When using multiple frameworks, an organization faces a number of challenges. First, the scope of different frameworks is likely to differ, with some amount of overlap that must be addressed. Interpreting the framework with the broader scope from the perspective of the framework with the narrower scope is usually appropriate. For example, an organization using the Software CMM and Six Sigma should interpret the Six Sigma principles and practices from the perspective of a software organization (since presumably it is a software organization that must integrate these two frameworks).

One way of addressing these scoping differences is by mapping the requirements in one framework to the requirements in the other. Usually, the narrower-scope framework will provide more detailed requirements and guidance on implementing parts of the broader framework. Although there may be some specific points in the broader framework that are neglected in the narrower, satisfying the more focused frameworks requirements can be considered *prima facie* evidence that the broader framework's equivalent requirements are satisfied.

While this may be acceptable in implementing a framework, it is more problematic when trying to integrate assessment/audit results. Assessment/evaluation/audit methods may have quite different rules of evidence for determining satisfaction. Even if the requirements in the different frameworks are compatible and complementary, differences in rigor may mean that the results of one method may not be acceptable to assessors and auditors using another method. Since much of the overhead and disturbance associated with multiple frameworks arises because of multiple appraisals in a narrow window of time, method integration must be considered as well as model integration.

If the issuing/authorizing body for a framework decides to integrate assessment/audit results from another framework, then a software organization can leverage that decision. I am unaware of any method that incorporates results from another framework's assessment or audit in a systematic fashion, although there are currently discussions on doing so for the eSourcing Capability Model for Service Providers (eSCM-SP).

Cross-certification is another possibility: an organization can hire assessors/auditors who are qualified and authorized to use multiple frameworks/methods. I am aware of a handful of professionals, for example, who are authorized Lead Assessors/Evaluators and/or auditors for CBA IPI, SCE, SCAMPI, ISO 9001, and/or eSCM-SP. (Note that certification is a misnomer for CBA IPI, SCE, and SCAMPI, since the SEI does not support certification, although the term is commonly used.) The burden of integrating the varying framework and method requirements and resolving any conflicts falls on the assessor/auditor; the burden of identifying appropriately qualified assessors/auditors for the relevant frameworks falls on the software organization.

Software organizations have only limited control over integrating assessments/audits. The customer or other external bodies may be the sponsors of the appraisal. While this may eliminate the direct cost of multiple appraisals, it does not affect the overhead of participating in them – and an appraisal may last up to three weeks.

For internal process improvement, the organization controls both the appraisal and the improvement program. Cross-certification is a viable possibility, but mapping between frameworks is desirable to leverage their commonality. Halvorsen and Conradi identify two mapping strategies: bilateral and needs based [Halvorsen, 2001]. In a bilateral comparison, two frameworks are compared textually [Paulk, 1995a; CMMI Mappings]. In a needs mapping, framework requirements are mapped to organizational and environmental needs. A third alternative is to map the different frameworks to a reference model; ISO/IEC 15504 is intended to be such a harmonization framework. There may be a concern that the reference framework is not sufficiently comprehensive to include all of the possible requirements in all of the frameworks that need to be considered.

Conclusions

Software organizations have to deal with a variety of models and standards. Some organizations may be able to control their use of the frameworks that they have to address; many have to deal with frameworks imposed on them by customers and external bodies, such as regulatory agencies.

Cross-certification can help, but issuing/authorizing bodies that systematically incorporate the results of complementary and supplementary appraisals into their methods would be of broader value. Bilateral mappings are relatively common. Mappings to a reference framework are potentially useful, and many different frameworks have been mapped to ISO/IEC 15504, including Software CMM and CMMI, as well as frameworks that have not been mentioned in this report, such as Trillium and Bootstrap.

References

[Baldrige] Malcolm Baldrige National Quality Award, <URL: http://www.quality.nist.gov/>.

[Bate, 1995] Roger Bate, et al., "A Systems Engineering Capability Maturity Model, Version 1.1," CMU/SEI-95-MM-003, Carnegie Mellon University, Software Engineering Institute, November 1995.

[BS15000-1, 2002] BS 15000-1:2002, "IT Service Management: Part 1: Specification for Service Management," British Standards Institute, 2002.

[Chrissis, 2003] Mary Beth Chrissis, Mike Konrad, and Sandy Shrum, CMMI: Guidelines for Process Integration and Product Improvement, Addison-Wesley, Boston, MA, 2003.

[Clark, 2000] Bradford K. Clark, "Quantifying the Effects on Effort of Software Process Maturity," IEEE Software, Vol. 17, No. 6, November/December 2000, pp. 65-70.

[CMMI Guidance] CMMI Interpretive Guidance Project, <URL: http://www.sei.cmu.edu/cmmi/adoption/interpretiveguidance.html>.

[CMMI Mappings] CMMI Mappings and Comparisons, <URL: http://www.sei.cmu.edu/cmmi/adoption/comparisons.html>.

[CobiT] Control Objectives for Information and related Technology (COBIT 3rd Edition, 2000), <URL: http://www.isaca.org/cobit.htm>.

[Cooper, 2002] Jack Cooper and Matt Fisher, "Software Acquisition Capability Maturity Model (SA-CMM), Version 1.03," Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2002-TR-010, March 2002.

[Crosby, 1979] P.B. Crosby, Quality is Free, McGraw-Hill, New York, NY, 1979.

[Curtis, 2001] Bill Curtis, William E. Hefley, and Sally A. Miller, People Capability Maturity Model, Addison-Wesley, Boston, MA, 2001.

[Deming, 1986] W. Edwards Deming, Out of the Crisis, MIT Center for Advanced Engineering Study, Cambridge, MA, 1986.

[Deming, 1994] W. Edwards Deming, The New Economics for Industry, Government, Education, Second Edition, MIT Center for Advanced Educational Services, Cambridge, MA, 1994.

[Duncan, 2004] Scott Duncan, "Standards Chair Report," ASQ Software Quality, Winter 2003-2004, pp. 12-17.

[Gibbs, 1994] W. Wayt Gibbs, "Software's Chronic Crisis," Scientific American, September 1994, pp. 86-95.

[Halvorsen, 2001] Christian Printzell Halvorsen and Reidar Conradi, “A Taxonomy to Compare SPI Frameworks,” Proceedings of the 8th European Workshop on Software Process Technology, Witten, Germany, 19-21 June 2001, pp. 217-235.

[Harry, 2000] Mikel Harry and Richard Schroeder, Six Sigma: The Breakthrough Management Strategy Revolutionizing the World’s Top Corporations, Doubleday, New York, NY, 2000.

[Harter, 2000] Donald E. Harter, Mayuram S. Krishnan, and Sandra A. Slaughter, “Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development,” Management Science, Vol. 46, No. 4, April 2000, pp. 451-466.

[Hyder, 2004] Elaine B. Hyder, Keith Heston, and Mark C. Paulk, “eSourcing Capability Model for IT-enabled Service Providers v2,” Carnegie Mellon University, Pittsburgh, PA, 2004.

[Issac, 2003] George Issac, Chandrasekharan Rajendran, and R.N. Anantharaman, “Do Quality Certifications Improve the Software Industry’s Operational Performance?” ASQ Software Quality Professional, Vol. 6, No. 1, December 2003, pp. 30-37.

[ISO9001, 2000] ISO 9001:2000, “Quality Management Systems – Requirements,” International Organization for Standardization, 2000.

[ISO12207, 2002] ISO/IEC 12207:1995/Amendment 1:2002, “Information technology – Software life cycle processes,” International Organization for Standardization and International Electrotechnical Commission, 2002.

[ISO15288, 2002] ISO/IEC 15288:2002, “Systems engineering – System life cycle processes,” International Organization for Standardization and International Electrotechnical Commission, 2002.

[ISO15504-2, 1998] ISO/IEC TR 15504-2:1998, “Information technology – Software process assessment – Part 2: A reference model for processes and process capability,” International Organization for Standardization and International Electrotechnical Commission, 1998.

[Juran, 1992] J.M. Juran, Juran on Quality By Design, The Free Press, New York, NY, 1992.

[Moore, 1999] James W. Moore, “An Integrated Collection of Software Engineering Standards,” IEEE Software, November/December 1999, pp. 51-57.

[Paulk, 1995] Mark C. Paulk, Charles V. Weber, Bill Curtis, and Mary Beth Chrissis, The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley Publishing Company, Reading, MA, 1995.

[Paulk, 1995a] Mark C. Paulk, “How ISO 9001 Compares With the CMM,” IEEE Software, Vol. 12, No. 1, January 1995, pp. 74-83.

[Sheard, 1997] Sarah A. Sheard, “The Frameworks Quagmire,” Crosstalk: The Journal of Defense Software Engineering, Vol. 10, No. 9, September 1997.

[Sheard, 2001] Sarah A. Sheard, “Evolution of the Frameworks Quagmire,” IEEE Computer, Vol. 34, No. 7, July 2001, pp. 96-98.

[Standish, 2001] “Extreme Chaos,” The Standish Group International, 2001 <URL http://www.pm2go.com/sample_research/ >.