Software Quality Strategy Supported by People and Organizations

Charles R. Matthews
The Boeing Company

charles.r.matthews@boeing.com

Abstract

It’s almost axiomatic that software projects will fail to meet targets for cost, schedule, and quality. The body of literature concentrates primarily on technology insertion, methodology, and metrics. In addition to these dimensions, software quality and organizational culture provide important support to programs, ensuring effective communication, requirements management, organizational change management, and integration of senior management vision and goals with program targets. The phenomenological study focused on decisions made at critical points in software programs, drawing from the observations of 25 participants who had worked in all phases of the software development life cycle. From the responses, key lessons were learned that can help in anticipating or avoiding the pitfalls inherent in common government and industry software development practice.

Biography

Charles Matthews is the Manager of the Software Quality Engineering team at The Boeing Company in Mesa, Arizona. His 34-year career has seen activity in all phases of the software life cycle, including work in production environments in Seattle, Washington; Huntsville, Alabama; Washington, DC; and remote desert areas of Arizona. Chuck has worked extensively in support of building Quality Management Systems within Boeing and at suppliers of aircraft parts, ground-based communications, missile defense, and product acceptance systems. Outside of Boeing, Chuck facilitates courses in the software quality body of knowledge and management science.

Charles is a doctoral candidate at the University of Phoenix. He holds a Master’s degree in Information Systems from City University, Seattle, and a Bachelor of Science degree in Management Science from Northern Illinois University.

1 Introduction

Solving the mystery of quality software is not hard; software quality requires planning, discipline and effective communication. Software failures have been blamed for more organizational problems than any other product (Jones 2008). Failure to deliver quality software results in loss of ability to perform the mission, deliver critical services, and execute important initiatives within the government. Failure of the software in turn contributes to loss of reputation for both government agencies and federal contractors. The consequences of failure include loss of funding and jobs. Yet software provides a primary mechanism for commercial and governmental organizations to deliver products, services, and to meet their goals. Therefore, study of the collective decisions made and the result of those decisions on project personnel is important in learning lessons.

Software provides the potential to enhance effectiveness of systems, organizational capability, and to gain immediate benefits from changes (Turk, France, & Rumpe, 2005). However, both organizational and deliverable system change management is complicated by unintended consequences of decisions that introduce change. In a system with the complexity of an aircraft or an organizational management system, one change may be multiplied by the number of changes to legacy systems or programs, data structures, and schema, even while integrating the need for change to the organizational culture. Such a prospect adds difficulty to creation of new business vision or operational concept (Light, 2008). Federal Government practice often does not include assessment of system impact in an effort to align policy and process, which creates risk in terms of government expenditure.

2 Search for best practices

A growing body of research illustrates how software projects fail and how to improve software development and maintenance processes. United States government agencies have not made significant improvement in software development since adoption of the Capability Maturity Model (CMM) in the mid-1990s (Bennatan, 2009; Defense Science Board, 2006). Studies of software projects in 1990 indicated 83.8% of projects were behind schedule, over budget, or cancelled due to failure to deliver required functionality. A 2005 survey indicated 76% of software projects were significantly behind schedule, over budget, and unable to deliver as originally promised. In the nine years since 2005, data indicate similar results. Chronic over budget and behind schedule situations cost taxpayers in government expenditure and in citizens' ability to interact with government agencies (Lipowicz, 2010). The Department of Defense continues to seek improved tools to manage software procurement.

The problem is not new. The Carnegie Mellon Software Engineering Institute (SEI) was established to define best practices for software development, management and delivery of software system products. The SEI has developed and published models and frameworks that are adopted widely by government and industry organizations, agencies, and contractors. In the 1990s and early years of the 21st century, the government flowed down requirements to meet the measurement criteria of the CMM and Capability Maturity Model Integration (CMMI) representations. The CMMI, and complementary models published by the International Organization for Standardization (ISO) and Project Management Institute (PMI) provide guidelines for best practices in project planning, software and system engineering, and business process management. The purpose of each of these models is to provide a path to achieve business goals related to cost, schedule, and quality. Six Sigma, IEEE standards, and balanced scorecards support management system models. However, any process model adopted by the organization also requires cultural alignment and discipline across the organization to adhere to best practices when risks transform to issues.

3 Changing environments

The aerospace industry, as with many other industries, faces rapid changes in technology, increasingly intense competition for revenue and information, demand new functionality and modernization, and a faster rate of obsolescence as systems age. This environment puts pressure on procurement organizations across the government and prime contractors who must manage and control increasingly complex supply chains. Plug and play concepts and Agile development appear to increase the probability of project failure (Mosquera, 2008). A recent article in the *Software Practitioner* journal highlighted how fundamental misunderstanding of how Agile complements the traditional software life cycle process, instead of replacing it, contributed to the initial failure of the Affordable Care Act websites (“Was it about Waterfall vs. Agile?” 2014). Experience indicates that leaders and practitioners in an organization must understand the basics of software engineering and software project management when challenges present themselves, or crisis management occurs as leaders depart from early plans in attempts to meet business goals.

The body of available research points to a number of process areas as root cause for failure, such as project and program management, quality management, requirements management, and configuration management. Quantitative assessments have suggested a relationship between failure to adhere to quality software practices within one or more of the process areas identified in the CMMI. Within each process area, a collection of best practices have been developed that contribute to measurement and certification of maturity in a structured assessment (Paulk, Weber, Curtis, & Chrissis, 2006). The integration of best practices represent the most probable path to success for software projects and provide a foundation for a disciplined organizational culture that provides quality software products (Humphrey, 2004). Integration of best practices across organizational disciplines is necessary in addition to software and system engineering. Best practices are also needed in project management, quality, financial management, and operations.

3.1 The social organization

Many social reasons for project challenges and failures relate directly to adherence to best practices in organizational and project processes, including issues arising from organizational culture and the relative maturity of practices across projects. Increasingly, software engineers and project managers have engaged in understanding earned value within the software project. Earned value management increases awareness of cause and effect with respect to schedule deviation and how decisions affect suppliers, developers, and customers. Increased knowledge of cost, schedule, and quality metrics and the failure to meet targets leads to frustrated developers in addition to potential loss of revenue and corporate reputation (Bennatan, 2009). Misunderstanding of qualitative factors that balance quantitative factors represented by earned value contribute to failed software projects. Government Accountability Office (GAO) studies and academic researchers cite a growing list of software project failures for loss of confidence in the ability for government contractors and project managers to deliver systems capabilities required for government operational and management system needs. A review of government reports and peer-reviewed literature indicates causes that include lack of adequate training, too much oversight, too little oversight, unclear requirements, aggressive technology insertion, untried methods, obsolescence, and management practices across supply chains.

The GAO (2009) summarized the problem as follows: U.S. Government efforts fail to deliver software products as promised in spite of industry certifications for software development organizations. While government leaders and industry executives establish and exemplify an organization’s culture and values, they fail to effectively manage expectations for delivery of systems solutions.

A software development organization exemplifies a techno-social system that combines intellectual activity with technology elements to solve a problem. The human and technology elements of this system are mutually dependent and are in turn highly dependent on organizational management and decision making of the human element, rather than the technical aspects of the system (Bennetan, 2009). A common element of organizations is the need to remain viable through achievement and maintenance of competitive advantage. Therefore, technology and innovation are vital factors in organizational development.

3.2 Social structure and organizational goals

Organization theory addressing techno-social theory has matured to account for organizational governance systems and systems that contribute to effective operation of the organization. Rapid advancement of technology and its effect on software development inspired foundational theories that matured into diffusion of innovation theory for information systems (Park & Yoon, 2005). Diffusion of innovation suggested the observable phenomenon whereby technology adoption and acceptance grows at an increasing rate around network architecture, wireless tools, and electronic communication toward a world where all systems and devices are connected.

Observation of software development organizations suggests that introduction of the various standards mentioned before – that is, CMMI, ISO 9001, earned value management, Malcolm Baldridge, Six Sigma, Lean Manufacturing – provide conflict among organizational paradigms, and both customers and senior organizational leadership contribute to this condition. Academic models suggest that competing elements across an organization are basic and stable, and individuals define themselves within the prescribed boundaries of accepted practices consistent with the organization’s goals and objectives.

4 Study of common themes

A phenomenological study of 25 people who represented functional practice in seven process areas across the software development life cycle provided a view of organizational decisions and project performance. Participants were members of the prime contractors’ respective organizations, representing process areas in each phase of the software development life cycle. The goal of the study was to explore organizational culture and decision-making approaches along the project timeline. Interviews focused on points in each program where difficult decisions were required. Five core themes emerged as presented in Table 1. Two supplemental themes emerged through observation of program management practices and root cause analysis.

Table 1. Percentage of projects observed within identified themes.

Emerging themes in empirical study	**Occurrence of theme**
1. Quality culture	91%
2. Internal decisions/deviations causing stressors	100%
3. External decisions/deviations causing stressors	89%
4. Delivery schedule as number one stressor	100%
4a. Multiple subcontractors contributing schedule impact	86%
5. Communication issues degrading project execution	100%

One supplemental theme emerging from the study data focused on the starting point of a program involving both government customers and contractors. Analysts who wrote the request for proposal and the corresponding proposal were seldom the people assigned to the program management office at the program start. Many contractors maintained professional staffs who captured business for the contractor; conversely, project risk seemed reduced in those organizations in which the proposal staff work engaged with the professionals whose job it is to execute the proposal. This approach to implementation prevents the pitfalls that cause risks to emerge by using the combined skills and experience of the proposal and implementation staffs.

A second supplemental theme suggested that U.S. government regulations promote unneeded bureaucracy that promotes inter-organizational conflict and additional overhead. The government organizational culture became misaligned with the software engineering culture in all cases observed. Observation further revealed that when third-party consultants were engaged to oversee a contractor, the third party aligned with the government organization, thus reinforcing the organizational incongruence. The existence and effect of organizational culture emerged as a common theme through the study.

4.1 Quality culture

During the project start-up phase, a quality process was agreed upon in all projects studied. Roles and responsibilities were defined, and each participant understood the organizational culture, policies, roles, and responsibilities. Participants were trained in the quality software process, organizational policies, and project standards. All project management participants, and 92% of participants in other functions (requirements, design, quality assurance, and configuration management) asserted that initial experience provide that both government and contractor program managers were interested in process discipline, most often using the CMMI and ISO 9001 based quality management systems. Leaders in the government and contractor organizations agreed to the process architecture, and senior management universally undertook training in processes. Meetings were held to establish reporting channels, quality oversight, and management for contractor organizations and government customers.

Software development leaders, typically software engineers, provided outlier responses. Follow-up interviews revealed most managers perceived that software design and development "just happens." Since software code is not tangible, enforcement of process discipline was not considered a science: "Code was treated as a black box," according to one participant. While measures of complexity were used in estimating the project, management did not understand them, and subject matter experts were not asked to provide insight. Therefore, senior management was not able to determine with any certainty why schedules slipped, why more labor power was needed, and why application of more resources did not resolve schedule issues.

4.2 Internal decisions causing stressors

A practice observed in some government agencies requires the proposal team leader to become the program manager. When a program faces immediate organizational change at program start, an element of risk is established at the outset of the program that challenges effective teaming. Use of a capture leader to get the business tends to drive aggressive behavior which encourages undercutting accurate cost estimates of the project. The proposal team and the project team represent different skill sets and create organizational subcultures early in the project that must be addressed later.

Such issues relating to misunderstanding of complexity and quality requirements led to the first program decision by the project manager, which was to not report to the customer deviations from plan as risks. Project and program management believed that early design decisions based on complexity analysis and early schedule pressure could be fixed by the software engineers. Internal decisions based on this belief did not require reporting to the government customer, and one project manager suggested, "The messenger bearing bad news is often shot." This belief creates a culture where risk is not discussed openly.

4.3 External decisions causing stressors

When schedule and cost deviations were reported to the government customer, key project personnel were replaced or shuffled within the organization. Participants in the study interpreted these organizational decisions as *failure*, and no other explanation was provided openly so that appropriate corrective action and realignment could be taken internally. All study participants commented on lack of clear communication about personnel changes creating a change of approach in each functional area, which led to diffused collaboration as each function retreated into its own discipline. The perception of schedule as the number one driver to the project was reinforced.

A government practice of hiring third-party consultants when risks emerged reinforced organizational tension and further eroded trust. By contrast, establishing a relationship early in the project between the contractor and the third party provides less tension within the project team and between the contractor and government customer. The best dynamic is one where the contractor seeks to inform the third party assessor.

In 89% of responses, participants reported that government oversight from a third party resulted from reporting process and schedule deviation to the government. In this study, the third party was provided by a Federally Funded Research and Development Center (FFRDC). The evaluations provided by the third party cited other government-funded projects reported by the Government Accountability Office (GAO), Congressional testimony, and government publications to support their findings and recommendations. Recommendations included process and organization changes which were challenged by contractor program management, in turn creating an adversarial relationship between the contractor, the government, and the third party. Project members believed themselves caught in a no-win position and perceived a doomed project. Project participants concluded that senior management and other stakeholders had determined they knew how to deliver the software product better than the project participants. This perception was reinforced each time an organizational change occurred or a program directive was announced.

4.4 Delivery schedule causing stressors

The projects within the study planned for an iterative spiral-development model for software development, modified by tailored prototyping. The development model worked well in the beginning, but when projects came under stress for behind-schedule deliveries each project was forced to “deliver now and fix it later.” Study participants described how requirements were redefined to meet the deadline. Each participant admitted that quality was affected by the redefined requirements management practice, which bypassed procedures (shortcutting) and deviated from the planned software life cycle model. The customer’s confidence in the delivered product suffered when heavy emphasis on schedule led to minimizing the timeline for requirements verification, testing, and software qualification. Two participants noted program management direction to talk with the customer outside of the configuration management process to capture information and to generate software code resulting from the conversation.

4.5 Communication channels degrading execution

Each project continued to face challenges, and decisions made by the program management office were not communicated clearly or effectively. Early in each project, participants noted open and effective communication, but as organizational changes occurred, schedule pressure mounted, and deviations were reported to the customer, less information was passed between program management and project personnel. In three of the projects studied, organizational decisions were made on a Friday, new management was in place on Monday, with no reason for the turnover provided and no formal communication of the change.

Participants revealed that decisions not to involve personnel who would execute the project was a significant factor in later challenges and risks. While generation of the proposal and statement of work were significant activities, the proposal phase was not considered within the scope of the software development life cycle. Participants who listed the proposal phase as a major problem area suggested that since the government and contractors provide professional acquisition personnel to generate the request for proposal, few people who have experience in managing the mission or capability of the system are involved, and communication is limited from the outset of the project. Critical information was not shared with the project team until they had a *need to know*.

The second supplemental theme suggesting that organizational culture is a significant factor provides support for the fifth major theme as well. Inflexibility between government and contractor organizational interfaces reinforced a “dichotomy between the government and [contractor] skill sets and became unbalanced” as projects encountered increasing cost and schedule pressure. Trust issues emerged when government program leaders attempted to manage technology insertion and requirements were not fully understood or communicated. When the executing team began to slide schedules due to poor requirements and lack of definition, trust eroded until the emphasis for both organizations became to deliver the product at the expense of satisfying initial program goals. Effective knowledge management, processes, and practices were challenged in third-party assessment reports, which commonly recommended that the prime contractor develop new processes, methods, standards, and management processes which were better (and most) understood by the assessors.

Misalignment of organization and culture between contractor teams and government organizations illuminated the difference in quality practices, the requirements of which were not contradictory. However, the specific practices became a distraction for contractors to overcome through communication with the government customer. Late recommendations to alter practices created a negative impact to product delivery in terms of schedule, cost and requirements. In turn, cultural clashes damage the institutional trust between contractor and customer organizations.

5 Conclusions

Organizational decision making both reflects and drives organizational values, the culture, and performance of the systems that operate a company. Common misunderstandings created when difficult decisions are made affect perceptions and subsequent actions of project team members. One key conclusion projected that dynamics and practices within the organizational culture are dependent more on social dynamics than the formal process architecture. The social contract to follow defined processes must be reinforced through consistent and effective communication.

Schedule becomes the most common driver for a project as risks emerge and deviations follow. The pressure to deliver becomes the primary criteria for each organizational decision. Introducing new requirements or redefining requirements to give schedule relief creates additional risk as the team attempts to react to shifting priorities. Redefining processes, shortcutting, or waiving planned activities, lowers confidence – and therefore trust – in the project team, both from within the organization and the customer. The study suggested that the senior management directing the proposal team often does not understand the complexity of many software system development efforts and how the organization's management systems work together to implement the organization's vision and goals. Yet, the program manager, often a senior manager, attempts to manage the program/project to a predefined baseline not developed or understood by the team executing the project.

Effective communication is the most important factor in leading any software development effort, especially as challenges and risks emerge. Changes in program management, the program management office, or team leadership required communication of situation and context in terms of organizational definition, requirements management, expectations, and delivery. Organizational and program priorities must be communicated, reestablished, and reinforced. When the need for early and clear communication is ignored, the project team is drawn away from requirements management and compliance to process architecture and toward crisis management and delivering the product at all costs. In terms of the CMMI, heroes are created.

Further study is needed to determine the extent to which social structures react to project risk. What behaviors emerge as senior management decisions are made? Anecdotal evidence indicates that many software development organization avoid sharing qualitative data needed to understand organizational culture, and a gap in understanding of organization theory in software and system development organizations exists in formal study. Leaders in government and industry benefit from understanding the evolving social dynamics in software engineering beyond technology considerations. Understanding how people work together in and across organizations can reduce rework and cost and increase return on software development investment.

References

Bennatan, E. M. 2009. “Project failures: Ignoring the warning signs.” *Advanced Project Solutions.* Retrieved from http://www.advancedprojectsolutions.net/

Defense Science Board. 2006. *Transformation: Process assessment*. Washington, D.C.: Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics. Report 2006-04-DSB_SS.

Government Accountability Office. 2009. *High risk series: An update.* (GAO-09-271). Washington, D.C.: Author.

Jones, C. 2008. *Software quality in 2008: A survey in the state of the art*. Software Productivity Research, LLC.

Light, P. C. 2008. *A government ill executed: The decline of the federal service and how to reverse it.* Cambridge, MA: Fellows of Harvard College.

Lipowicz, A. 2010. “Boeing’s SBInet project under fire again.” *Washington Technology* (January).

Mosquera, M. 2008. “GAO says costs, schedule hamper IRS modernization.” *Government Computing News, 22*(17): 22-25.

Paulk, M. C., Weber, C. V., Curtis, B., & Chrissis, M. B. 2006. *The capability maturity model: Guidelines for improving the software process.* (2nd ed.). Boston: Addison Wesley.

Turk, D. E., France, R. B., & Rumpe, B. 2005. “Assumption underlying agile software development processes.” *Journal of Database Management, 16*(4): 62-87.

Was it about Waterfall vs. Agile? (2014). *Software Practitioner*, *24*(1): 6-7.