Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements

Carol Woody, PhD

March 2005

Networked Systems Survivability

Technical Note
CMU/SEI-2005-TN-010

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

About this Report iii

Acknowledgements..iv

Executive Summary v

Abstract...vii

1 Introduction 1

1.1 Information Sources 1

1.2 Links to Previous SEI Work 2

2 The Impact of the Organizational Perspective 3

2.1 Recognition of Organizational Risk...3

2.2 Recognition of the Importance of an Enterprise Perspective 5

2.3 Confusion of Terminology 6

3 The Impact of the Project Perspective...8

3.1 Organizational Mechanisms that Constrain Requirements 8

3.2 Software Project Development Mechanisms 9

3.3 Integration is an Emerging Responsibility...10

4 Embedding Quality Requirements Within a Software Development Methodology...12

4.1 Training Concerns About Quality Requirements for Software Designers and Developers 13

4.2 Risk Assessments Must be Explicit and Appropriate...........................14

4.3 Quality Risk Analysis Differs From Operational Risk Analysis.............15

4.4 Establish Links to the Operational Resources 16

5 A Pilot Project for Improved Security in Software Development............ 17

5.1 Approach Used.. 17

5.2 Applying the Approach.. 17

5.3 Links to Other Research Focused on Improved Development Processes... 19

6 Furthering the Research 20

References 21

About this Report

This document addresses a portion of the research covered in the Independent Research and Development Project titled “Eliciting and Analyzing Quality Requirements: A Feasibility Study.” Within the broad area of quality requirements, the research team focused its efforts on security and safety, seeking to identify specific elicitation approaches to enhance the level of organizational consideration. Safety and security quality requirements were selected as the primary focus of the independent research. Broader applicability of the management influences to other quality requirements is assumed but has not been established.

Acknowledgements

I wish to express my thanks to Nancy Mead and Don Firesmith, who participated with me in the Independent Research and Development Project for Eliciting and Analyzing Quality Requirements. Though we could not come to agreement on many points, the discussions were extremely valuable in framing my research ideas.

In addition I wish to acknowledge the following individuals who reviewed this document and provided comments:

- Christopher Alberts
- Audrey Dorofee
- Nancy Mead
- M. Stephen Palmquist

Many thanks to Pamela Curtis for editing this document.

Executive Summary

Quality attributes are influenced by the selection and development of components that make up a system, as well as the organizational environment in which the system is created. The organizational environment for system development can support or reject improved elicitation mechanisms.

The following appear supportive of improved elicitation for quality requirements:

- recognition that poor system quality represents organizational risk, which can result in organizational liability
- recognition of the importance of enterprise planning for effective quality in a system-of-systems environment
- standardization of terminology among business and technology participants in the software development process
- effective policies and procedures to control acquisition and incorporate appropriate consideration of quality requirements in contracting
- inclusion of resources in the elicitation process who are effectively trained in the issues and technology challenges of defining quality requirements to produce quality results
- effective use of independent reviews such as accreditation and certification mechanisms and risk analysis to verify decisions concerning quality requirements made in the development process
- use of a software development methodology that incorporates explicit and appropriate consideration of quality attributes throughout the development process

The following present barriers to improved elicitation:

- failure to recognize the link between organizational liability and software quality attributes
- failure to implement policies and procedures in a sufficiently timely manner to influence software development
- lack of consistent use of terminology across the organization
- lack of appropriately trained resources participating in the elicitation and analysis of quality requirements
- internal organizational barriers, both formal and artificial, that inhibit the effective sharing of information among organizational units responsible for quality results

- lack of explicit inclusion of quality considerations within the system development life cycle

For each of the identified areas, a literature search was used to identify the range of practices in place. Sample development projects were analyzed to identify practices in use and the support or hindrance each provided to quality requirements elicitation. A pilot project was used to identify potential opportunities for improved elicitation within a development environment.

Abstract

Early in the literature review for the Independent Research and Development Project for Eliciting and Analyzing Quality Requirements, the potential conflict of quality efforts (perceived as time consuming) with organizational management direction (driven by time-to-market and cost considerations) was identified. Quality attributes are influenced by the selection and development of components that make up a system, as well as the development environment in which the system is created. A variety of information sources, including conferences, workshops, pilot projects, and technical assessments, was tapped to identify specific management barriers to the adoption of improved elicitation approaches and appropriate organizational behaviors that facilitated the use of improved mechanisms for the elicitation and analysis of quality requirements. This report documents the ways in which the organizational and project management environment for system development can support or reject improved quality requirements elicitation mechanisms. In addition, this report identifies specific activities as promoting improved quality requirements elicitation when they are embedded into the system development life-cycle structure.

1 Introduction

Early in the literature review for the Independent Research and Development Project for Eliciting and Analyzing Quality Requirements, the potential conflict of quality efforts (perceived as time consuming) with organizational management direction (driven by time-to-market and cost considerations) was identified. The importance of researching this conflict as a potential barrier to adoption of improved elicitation techniques was based on consistency with issues identified in the use of risk methodologies within the operational system environment. A portion of the feasibility effort was focused on the identification of specific barriers to the adoption of improved elicitation approaches and appropriate organizational behavior that facilitates the use of effective mechanisms for eliciting and analyzing quality requirements. In keeping with the primary focus of the independent research project, safety and security quality requirements were emphasized, and many of the references are linked specifically to one or both of these attributes as noted in the text. Broader applicability of the management influences to other quality requirements is assumed but has not been established.

1.1 Information Sources

In addition to a literature review, a variety of information sources was tapped to support the identification and analysis of management influences. Conference presentations, workshops, and panel discussions were used to reach a broad range of organizations involved in eliciting and analyzing quality requirements to initiate further discussion on the management issues. They include the following:

- Presentation: “Considering Operational Security Risks During Systems Development,” SEPG 2004, Orlando, Florida, March 9, 2004
- Presentation: “Considering Operational Security Risks During Systems Development,” European SEPG, London, England, June 17, 2004
- Panel Discussion: “Can Secure Systems Be Built Using Today’s Development Processes?” European SEPG, London, England, June 17, 2004
- Workshop: “Considering Security Risks During the System Development Life Cycle,” Liberty University, Lynchburg, Virginia, August 6-8, 2004
- Presentation: “Embedding Security into a Software Development Methodology,” PSQT/PSTT 2004 North, Plymouth, Minnesota, October 25-29, 2004

Feedback from a self-selected pilot organization provided many of the observations in this document. Experiences from Independent Technical Assessments of major software development projects conducted for large federal agencies added to the content.

1.2 Links to Previous SEI Work

Based on the catalog of practices within the CERT® Operationally Critical Threat, Asset, and Vulnerability Evaluation[SM] (OCTAVE®) method, a subset of operational security risk was identified that can and should be addressed earlier in the life cycle. Decisions are made within the acquisition and development stages of software development that directly impact the security risk that must be addressed when the software is fielded into operations.

As part of three Independent Technical Assessments (ITAs), security was identified as a critical quality requirement. This quality attribute was analyzed using the Software Engineering Institute (SEI) Architecture Tradeoff Analysis Method® (ATAM®) evaluation process and further explored in follow-on security workshops. A number of managerial issues were identified from these efforts. Confidentiality constraints limit the direct use of information obtained during an ITA; only general observations are included in this report.

[SM] Operationally Critical Threat, Asset, and Vulnerability Evaluation is a service mark of Carnegie Mellon University.

® CERT, OCTAVE, Architecture Tradeoff Analysis Method, and ATAM are registered with the U.S. Patent and Trademark Office by Carnegie Mellon University.

2 The Impact of the Organizational Perspective

Safety and security are emergent properties and as such do not have absolute requirements. The organizational perspective on safety and security heavily influences the consideration given to these quality areas during requirements elicitation and analysis. Most organizations have recognized a need for security and safety and have delegated the responsibility for addressing safety and security to specific units whose influence is determined by their position in the organizational structure and level of funding. A subset of the organizations in our research sample have recognized, usually through the impact of regulatory requirements or the impact of failures, that the responsibility for security and safety is a broadly shared role that permeates all levels of the organization. To be effective, the actions of this diverse group of participants must be appropriately coordinated and managed.

It is much easier to identify the failure of security or safety than to specifically identify its success. Variations in understanding what determines security and safety contribute to the challenge. Limitations in sharing of information about security and safety problems that have occurred within the organization or at peer organizations contributes to a perception that these quality requirements are outside of the range of criticality for a software project. When the perceived risk of a security or safety failure is low, the organizational focus shifts to areas of perceived higher importance. There is never sufficient time and resources to address all project requirements, and organizational importance factors heavily in the selection process.

The level of risk awareness of the system users is critical to effective elicitation and analysis of quality requirements such as security and safety. Users are focused on functionality and ensuring that the system will help them perform specific functions. Quality requirements such as performance and usability are more visible within that limited perspective. External organizational checks on the project team are more likely to enforce an appropriate organizational consideration of safety and security risk.

2.1 Recognition of Organizational Risk

Security and safety failures can result in organizational liability. The recognition of this possibility and the level of importance placed on the potential liability vary greatly among organizations. The level of consideration given to security and safety risk appears linked to the level of organization concern and the mechanisms used by an organization to analyze risk. Assessments that apply inappropriate techniques can result in complacency and lack of appropriate consideration of safety [Leveson 95] and security [Verdon 04] requirements.

Greater importance is placed on quality requirements for safety and security if a certification or risk assessment is required by organizational policy prior to system deployment. In addition, greater importance is placed on establishing and meeting safety and security requirements when the organizational units tasked with enforcing standards and regulations report to senior-level management. Widely used certification processes such as DITSCAP (DoD Information Technology Security Certification and Accreditation Process) and NIST 800-26 are aimed at the protection and control mechanisms applied to electronic information sources within the operational environment. If the organization's only interest in certification and accreditation is to perform the process to meet an external mandate, there is a tendency to focus on minimizing the control mechanisms without consideration of the requirements that would drive the need for the controls.

If the development effort is delivered through multiple releases, certification and accreditation must be addressed for each implementation. Complete reliance on certification to drive the security requirements can lead to a short-term focus and narrow requirements for each release. As a result of this approach to security requirements, one organization was forced to fund extensive rework of controls at each release as the application matured.

The management practice of separation of duties and segmentation of responsibility, which is designed to ensure checks and balances, can make efforts to improve security and safety requirements more difficult. Effective consideration of safety and security within a software development effort requires coordination across a broad group of participants to ensure the existence of policies, user awareness, and organizational enforcement mechanisms. Many levels and units of the organization are involved in the development and implementation of these controlling factors. When responsibility for safety or security is assigned to a specific unit, the quality of security and safety requirements considered within the software development can depend on the involvement and understanding of the assigned group with the organization as a whole. In addition, the inclusion of security and safety requirements can depend on the interactions and influence of this assigned unit with each individual project development team. If responsibility is divided among many groups, the level of coordination among these groups to ensure a complete solution for the organization as a whole will determine success. In one organization the responsibility was assigned to a policing unit whose primary focus was physical security and vetting individuals for organizational participation. Organizational consideration of safety and security was limited to hardware vulnerability protection and specific point-to-point data sharing agreements where information left the organization's environment. Authorization and authentication mechanisms were left to each application development team.

A long-range approach that should prove more robust over time is emerging from organizations that link the physical and technology aspects of security and safety within a single responsible unit. As access control, telephony, and facility controls for building infrastructure move onto the network, the recognition of the increased level of organizational dependency on the network raises the level of concern for effective security and safety of all systems running there [Sarkar 04].

2.2 Recognition of the Importance of an Enterprise Perspective

Organizations have attempted to achieve technology consistency across a wide range of initiatives through the use of enterprise architecture planning. The recognition of the need for shared services among applications and the resulting impact of individual systems on each other within an operational environment allows for consideration of a system-of-systems approach. An effective strategy for improved consideration of security and safety within the enterprise planning process requires extensive participation of knowledgeable experts responsible for organizational safety and security. This enterprise process must be led by individuals with experience in system-of-systems environments who understand the challenges and problems of a widely interconnected environment. The resulting architecture includes consideration of organization-wide mechanisms to address safety and security needs that cross many projects. Individual projects can apply the available mechanisms instead of building or buying their own, making it more cost effective to participate.

Each development effort must have access to the resources familiar with the enterprise perspective. In one organization, security and safety experts working with the enterprise architecture were assigned to key projects to support the generation of requirements, and improved elicitation approaches were readily incorporated into the normal project workflow.

In another organization, each project assembled a separate set of security requirements independently. When project requirements were assembled to create the shared global mechanisms for inclusion in the enterprise architecture, the following problems delayed the availability of organization-wide mechanisms beyond the time frame usable by each project:

- inconsistencies in the level of development of security and safety requirements at the project level
- confusion in the varying uses of terms
- lack of leadership experienced in developing and implementing system-of-systems architectures

Use of the enterprise solution for this organization required twice the effort at the application level to first generate an interim solution to meet the project deadlines and then apply the enterprise option through a future maintenance effort when it became available. There was also the risk with this “bottom up approach” that the organization would over-invest in a wide range of mechanisms to address every conceivable issue without a business rationale for the expenditure.

For all of the projects considered in this research, there was insufficient recognition of the impact of the new application on the operational environment. Participants in the requirements elicitation process must include those knowledgeable in the “end state” if the development effort is seeking to modernize technology platforms, institute business re-engineering, or migrate to different technology architecture. As an example, an organization shifting from a primarily batch processing environment to a highly interactive Web-based environment, a common transition for many government and commercial organizations at this time, cannot rely on existing users and operational staff to project the safety and security requirements without experts familiar with the types of risk inherent in the target environment. Based on the study of actual projects, the current operational safety and security requirements are too frequently used as the quality requirements for the developed product without considering how the development effort will change the operational environment. In a similar manner, the designers of an application that will institute business process reengineering cannot rely exclusively on the existing users to project the issues and concerns that will be part of the new processes, since they have no context on which to base their decisions.

2.3 Confusion of Terminology

Understanding of the meaning of security and safety varies widely. This confusion contributed to inappropriate coordination efforts among business units sharing the responsibility for security and safety in several of the organizations in the research sample.

The SEI Software Technology Roadmap (STR) Glossary defines safety as “a measure of the absence of unsafe software conditions. The absence of catastrophic consequences to the environment” [SEI 04]. Firesmith defines safety within the context of usage of an application or component as “the degree to which accidental harm is prevented, reduced, and properly reacted to” [Firesmith 03]. In Firesmith’s analysis of requirements, he includes safety as a quality factor of dependability. Leveson, a widely referenced author on safety, uses the definition of “freedom from accidents or losses” [Leveson 95]. This author points out that safety is frequently merged under reliability or security or dependability, but should be considered a distinct quality handled directly within tradeoff decisions and not hidden within other attributes. Requirements gathered in one quality area may be applied to several, but differences are sufficient that important issues will be lost if the safety quality is not specifically evaluated independently [Leveson 95].

Security is defined in the SEI STR as “the ability of a system to manage, protect, and distribute sensitive information” [SEI 04]. The OCTAVE methodology includes a definition of security requirements as “outlining the qualities of information assets that are important to an organization. Typical security requirements are confidentiality, integrity, and availability” [Alberts 01]. Another source defines information security as “to apply any technical methods and managerial processes on the information resources (hardware, software and data) in order to keep organizational assets and personal privacy protected” [Hong 03].

Verdon and McGraw note that application security has come to mean the protection of software after it is already built based on general usage by practitioners. “Application security is based primarily on finding and fixing known security problems after they’ve been exploited in fielded systems” [Verdon 04]. The IT Governance Institute states the “objective of information security is protecting the interests of those relying on information and the systems and communications that deliver the information from harm resulting from failures of availability, confidentiality and integrity" [ITGI 04]. Anderson, author of a widely used text on security, devotes several pages to terms frequently used in describing security, such as asset, system, subject, identity, person, role, confidentiality, privacy, integrity, anonymity, secrecy, authenticity, vulnerability, trust, and trustworthy, that require clarification and are easily misunderstood [Anderson 01]. Security is often merged under dependability [SEI 03].

The definition for dependability frequently referenced at the SEI comes from writings by Laprie. "Dependability is that property of a computer system such that reliance can be justifiably placed on the service it delivers. The service delivered by the system is its behavior as it is perceived by its user(s); a user is another system (physical, human) which interacts with the former" [Laprie 95].

Though the differences among definitions are subtle, they are sufficient to cause problems in developing good requirements for safety and security. Definition concerns were discussed in all of the security workshops conducted during this research project, and participants pointed out that confusion was frequently encountered among participants with varying technology backgrounds, including computer operations staff, data administrators, architects, systems engineers, business analysts, and programmers. Too frequently the technology-trained groups focused totally on the technology solution without consideration of the human user perspective, which applied a different definition of dependability than espoused by Laprie.

To address the challenge of terminology confusion, which was identified as critical, a step was added to establish agreed definitions at the beginning of the requirements elicitation process for use throughout the proposed elicitation process [Mead 04]. All stakeholders participating in the elicitation process must participate in this definition step to establish appropriate consistency. To ensure a level of consistency with good practice, definitions from IEEE and other standards bodies would be used as candidates. Once definitions are established, agreement on appropriate grouping of quality requirements can be applied. If the definition of dependability appropriately addresses safety and security, the elicitation process can consider grouping them. However, the caution pointed out by Leveson should not be ignored—it is likely that more appropriate analysis and tradeoff choices will be made if each attribute is elicited and analyzed as a separate quality.

3 The Impact of the Project Perspective

Effective elicitation and analysis of safety and security requires cross-functional participation. The expertise to identify and evaluate organizational risk generated by and impacting a software project is scattered throughout the organization. The channels available to connect this knowledge to the project for participation in the requirements elicitation process are limited by the structures controlling the project and the IT environment within the organization. Three primary functions must be met: identification of critical risks that require project attention, identification of appropriate mitigation responses that may already exist within the infrastructure, and assignment of resources to develop and/or incorporate mitigation responses within the development effort. If the appropriate resources are not available to the project and decision-making roles do not include mitigation actions when needed, limitations in the requirements elicitation effort will carry through all the phases of the software development life cycle, since requirements form the basis of the remaining steps in the life cycle. Ways in which the organization exerts control on a software development project can directly and indirectly influence quality requirements.

3.1 Organizational Mechanisms that Constrain Requirements

Organizational policies and regulatory compliance procedures are a major resource for security and safety requirements. When policies are out-of-date or tied up in endless review cycles, the input available to a project from this source becomes questionable. We encountered both of these conditions in separate projects in our research sample, along with requirements problems resulting from them.

Policies and procedures also control the acquisition mechanisms used for software purchases and contracting for software development. In one organization a major security policy had been under review for over a year pending approval. In the meantime, the old policy’s lax security requirements were provided to a contractor. It took some time to realize the old requirements were responsible for an implementation plagued with security problems.

Documented operational procedures provide another major resource for security and safety requirements. These convey the mechanisms by which an organization is addressing its policies and procedures. However, the recognition that these documents must be maintained with consistency is not always a priority. In one organization operational changes due to upgrades and expansions were not reflected in the documentation of the operational security requirements, but an outsourced software development team was legally required to meet the documented requirements based on provisions automatically included by procurement policy in the contract. The delivered system had to be adjusted at implementation by a contract amendment and additional funding to change the mechanisms at an added cost and implementation delay so the software could function in the actual operational environment.

The level of technical expertise available within the Project Management Office (PMO) is critical to providing an appropriate level of consideration of quality requirements elicitation. Many projects include a range of applications and components that compose a system-of-systems that are highly interdependent. Quality requirements must be consistently identified at the highest levels and assigned to the appropriate components across all segments. If the PMO is staffed for acquisition, contracting, and cost monitoring, but does not include enterprise architects and senior systems engineering expertise to provide a top-down perspective and influence decisions made within each subproject, the appropriate level of oversight will not be available and decisions made at the subproject level will be too narrow.

3.2 Software Project Development Mechanisms

When the stakeholders include individuals knowledgeable in safety and security, the potential for effective requirements elicitation is greatly improved. Participation of these experts is only useful when they can frame their input in a form that subject matter experts (SMEs) will understand and recognize as valuable. As an example, one organization had participation of a wide range of experts, but the quality requirements for security were identified as costly and time-consuming by the users providing the funding and were removed in order to meet budget constraints.

The structure of the software development methodology can impact the level of consideration for quality requirements. System designers and developers are usually too close to the workings of the application to identify critical risks [Hope 04]. Risk assessment steps, certification requirements, and mandated authority-to-operate reviews imposed on each project through the life-cycle methodology provide forcing mechanisms for approval by experts external to the project and knowledgeable in safety and security. The importance of these mechanisms is linked to the organizational visibility of the review participants and level of authority granted to the reviewing participants to slow or block the project timetable. In observed organizations with less formal project control, development teams concentrated on satisfying the needs of the users sponsoring the project, and the level of consideration for quality requirements was dependent on the knowledge level of the user.

The development approach to testing provides another controlling mechanism on quality requirements. When the design of testing is part of the requirements elicitation effort, there is a greater opportunity for better requirements [Graham 02]. Test analysis forces consideration of completeness and testability of outcomes.

For validation of safety and security requirements, only limited checks can be performed within a laboratory environment. Definition of the target implementation environment is a key to identification of inherited risks that need to be considered within the requirements elicitation process. Safety and security failures are “typically consequences of unanticipated scenarios and Byzantine subsystem interactions" [Johnson 03]. Interactions can involve automated linkages between components, between the system and its environment, human and computer interfaces, and management levels. Performing a business function may involve the linkage of steps across multiple applications, multiple users, and multiple locations within the infrastructure. Consider the medical example of a physician requesting a patient's blood sample. The users include patient admissions, the physician, a phlebotomist, a lab technician, and a payment provider. The applications could include admissions, billing, accounts receivable, patient records, order entry, laboratory control, and results reporting. Consider how different the security and safety requirements are if all of the pieces are confined within a single hospital setting, distributed among a series of inter-linked medical facilities, or scattered geographically among multiple medical businesses. Many of these steps may involve integration with existing design flows and implementations. Formal requirements specification is needed to address this situation. Formal verification and validation mechanisms, especially for systems-of-systems and embedded systems, are lacking [Johnson 03].

If the validation mechanisms consider only confirmation of process completion and do not include analysis of how the processes are executed or implemented, critical safety and security flaws are missed [Leveson 04]. One example is the crash of an American Airlines DC 10 at Chicago's O'Hare Airport in 1979. The linkage between manual procedures, process verification steps, and performance of the plane's flight control system violated a safety constraint, but flawed control actions were not identified in quality assurance based on the confirmation mechanisms.

Another related example is the failure of a security mechanism designed to protect sensitive documents that stopped working when operating features on which the process depended were dropped in a system upgrade [May 98]. Implementations of security and safety mechanisms should not enforce mandated versions of infrastructure components. Limitations on the ability to upgrade the infrastructure can cause organizations to miss applying important patches that leave the infrastructure vulnerable to known high risks.

3.3 Integration is an Emerging Responsibility

The emergence of standards and techniques such as Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP) for loosely coupling systems from a broad range of platforms and locations into a federated infrastructure has heightened the challenges for safety and security. The footprint of direct participants is not always visible to any single project group. The role of an integration specialist is emerging [Girard 04]. Through the formation of Integration Competency Centers, organizations are breaking down the traditional project-to-project connectivity approach and forming enterprise approaches, but methods for this level of development are largely informal and reactive.

Organizations that have formally launched enterprise teams that include architects, business analysts, application experts, and technical expertise are gaining short-term benefits, but the enterprise standards to support mechanisms for integrated security such as Security Assertion Markup Language (SAML) and Web Services Security (WSS) are still emerging [Gebel 03].

Projects to institute global security mechanisms such as role-based access control (RBAC) are underway in several of the organizations observed for this research. These organizations are pushing software development teams to design to the mechanisms using RBAC because of the promised ease of implementation and simplification for the users that interact with many systems. Unfortunately the mechanisms are proving difficult to establish and maintain across the organization due to the dynamic nature of organizational structures and assignment of responsibilities [Gebel 03]. The use of RBAC is becoming a high risk to each project implementation due to the complexity of the RBAC effort and the lack of coordination with the application development projects, which are usually controlled by different groups within the organization. In order to apply RBAC within the application project, the entire user population for the application must be using RBAC. Otherwise, the application must implement multiple types of authentication with varying levels of control and consistency or plan to convert to RBAC when it is fully deployed and reauthorize and retrain the user population for the application at that point in time.

4 Embedding Quality Requirements Within a Software Development Methodology

Methods, tools, and techniques for the identification and analysis of quality requirements require specialized knowledge and may require controlled environments for prototyping early in the development process [Tryfonas 01]. As a result, these are seen as costly and time consuming, which limits their use. However, experience clearly shows that this cost is more than recouped in the long term. Existing methodologies and tools need to be tailored to include appropriate mechanisms for quality requirements at elicitation and throughout the development life cycle to ensure appropriate quality results. Five lessons identified in an analysis of security and usability for a project designed to increase the level of user control over the security of their portable devices [Balfanz 04] can be applied to many of the quality requirements issues within the development life cycle:

- Neither usability or security [or any other quality attribute] can be added after the fact with any degree of effectiveness.
- Tool selection [PKI and SSL are tools, not solutions] is only a piece of the end result required for a complete solution, and the requirements for the complete solution must be clearly identified before the tools are selected to ensure appropriate choices.
- Security mechanisms within an application must be compatible with what the user needs to accomplish [or the user will figure out ways to break them].
- Designers and developers must gain typical user input about usability and security; also they must recognize that they are not typical users.
- Plan and act locally; global solutions cannot be ensured to solve a local problem [both usability and security are, at least partially, local issues].

Without enforcement mechanisms that act as review and gating steps within the development life cycle, the limitations of resources and time constraints that plague every development effort can drive participants to bypass consideration of quality requirements until problems arise. A lax review process will hinder the ability of the organization to enforce standards or to address difficult-to-resolve incompatibilities of quality requirements such as safety and security with policy. For policy to be used to consistently apply organizational needs across all projects, it must be established and communicated throughout the organization in a timely manner. One organization within the research sample developed policy based on actual implementations to ensure point-in-time compatibility for regulatory compliance, but this mandated a continuous update of policy when new development projects changed the installed base. This organization abandoned the use of policy as a controlling mechanism for software development.

4.1 Training Concerns About Quality Requirements for Software Designers and Developers

Safety engineering books to address system accidents became available in the 1990s. Much research resulted from major software malfunctions linked to industrial accidents such as the 1984 Union Carbide chemical plant toxic chemical release in Bhopal, India. Between 1985 and 1987, six people died from massive overdoses of radiation administered by computer-based radiation therapy machines. In 1979 Three Mile Island and in 1986 the Chernobyl nuclear plant gained world attention [Leveson 95].

Operational concerns and the need for educational materials for security were not seriously addressed until a security incident halted the Internet in 1988. The CERT Coordination Center was established in 1988 through a collaborative agreement between Carnegie Mellon and the U.S. Department of Defense. This agreement was established in response to the Morris worm, a self-replicating program released by William Morris that crippled the Internet earlier the same year and impacted worldwide communications [Pethia 01]. Training programs for information security, such as the information assurance program at the Navel Postgraduate School and the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University, were offering classes as early as 1990.

While programs are available to train experts in safety and security engineering, references to these knowledge areas are not always available in general software development course materials. A review of selected textbooks written for teaching systems analysis and design was conducted to sample the level of exposure to quality requirements and related issues provided to college students being trained in design and development. Textbooks were selected informally based on known classroom use reported by university instructional colleagues. Within each text, the following information was identified:

- references to quality requirements
- references to risk
- references to security and safety
- references to use cases and software error handling
- references to operational environment
- references to standards

The following concerns and gaps were noted from this preliminary review. An instructor would need to augment course material to address these issues:

- Recognition of a risk environment for software implementation with consideration of abuse cases and potential failure handling was missing.
- There were no references to any types of quality requirements.
- Recognition of standards and best practices (key information sources for safety and security) was missing.

This review is by no means comprehensive, but it should alert organizations to consider the need for ensuring sufficient training and experience levels of individuals participating in quality requirements elicitation. Organizations cannot assume that experienced software development participants are sufficiently exposed to safety and security issues to ensure appropriate consideration of these qualities within a software development project.

4.2 Risk Assessments Must be Explicit and Appropriate

The need for considering risk within the context of a software project is not well understood based on approaches used by organizations included in this research project. The processes and procedures used by the project management team to address project risks are good indicators of the level of skill for risk management available within the project. For example, one organization engaged in a multi-year acquisition would only consider short-term (90 days or less) risks; anything longer than that was ignored. In addition, because of the poor risk process, only a handful of even this narrow time frame of risks were identified. In another organization, one project delegated the identification and mitigation of project risks to a contractor and refused to consider internal risks that might impact the project. Another project team in this same organization insisted that risks should not be identified because of the negative connotation they gave to the project, which reflected badly on the dedication of the project participants to meeting the project goals.

Risk management is a key component to the effective elicitation of safety and security requirements. Safety and security requirements address incidents that have the potential to occur. The criticality of each requirement is based on the risk associated with it by the stakeholders. Based on an expected limitation of risk management capability within a project team, mandated assessments by impartial, trained resources that are required at specified points within the development life cycle will be needed to enforce consideration of appropriate risks. This will not guarantee effective risk management, however. Risk analysis is a subjective process that is highly dependent on the perspective of those performing the analysis. An understanding of the threat environment and potential ramifications to the organization if threats are realized is required for an accurate analysis. This cannot be an informal, ad hoc process [Alberts 03]. It must be consistent, structured, carefully documented, and maintained through the collection of lessons learned to provide a means for an organization to improve over time [McGraw 04]. Open communication among users, developers, and operational support at all levels of the organization is needed to ensure effective risk identification.

Formal assessments should be performed at several stages of the development life cycle to ensure that mitigation is appropriately handled and that new risks introduced by changes within the project and the organization are included in the mitigation analysis. McGraw suggests risk analysis during requirements analysis, design, and testing [McGraw 04]. Our research identified risk analysis as a key step for quality requirement trade-off considerations, and a risk assessment is a step in the requirements elicitation framework [Mead 04] developed from this research project.

4.3 Quality Risk Analysis Differs From Operational Risk Analysis

Not all operational practices for safety and security are relevant to a development effort. In addition, the development effort will inherit a group of safety and security risks from the operational environment based on the standards and procedures and infrastructure into which the new implementation will connect.

The assessment cannot be focused on a specific implementation and specific components, since these remain to be determined. Two techniques have been identified for scoping an assessment for security analysis and should be applicable to other quality requirements. One provides a component view and the other a business view. Each approach will identify gaps that need to be analyzed to determine whether requirements are adequate.

A component view strategy for considering operational security risk during system development was developed based on the OCTAVE Methodology [Alberts 04a]. Consideration of the inherited risk from commercial off-the-shelf (COTS) components and existing implementation interfaces is a key issue. This strategy involves five steps:

1. Define the target system.
    - software applications under development (processing modules, interface modules, database modules, etc.)
    - software applications in place (COTS modules, existing systems, existing interfaces, etc.)
    - hardware components
    - operational environment
2. Determine security attributes for the target system.
3. Identify threats (ways the security attributes could be compromised) to the target system.
4. Identify risks (impacts from the threats) to the organization.
5. Develop a protection plan to address critical risks through component, infrastructure, and organizational protection mechanisms.

A second strategy that focuses on critical business processes required for the stakeholders to complete business objectives is under development [Alberts 04b]. Consideration of business steps that cross development project, application, and database boundaries is a key aspect of this approach. Processes that cross multiple operational sites and responsible organizations are also considered, as well as the linkage between technology and procedural process steps. For each selected business process, the following steps are performed to determine whether appropriate controls are in place to ensure effective completion of the business process:

- Identify specific steps by role.
- For each step,
    - identify key software components and data store and
    - identify security requirements for each component and data store.

- Identify steps of high risk to the business process assurance.
- Analyze assigned security mechanisms for each high-risk step to confirm an acceptable level of assurance that security requirements are being met.
- Identify excessive security and gaps.
- Define mitigation plans to address gaps.

4.4 Establish Links to the Operational Resources

An understanding of the target operational environment is critical to the elicitation and analysis of safety and security requirements. The operational environment may already be addressing many of the safety and security risks important to a development project with mechanisms that can be readily adapted to meet project quality requirements. Conversely, the operating environment may be introducing risks that are unacceptable for a project, requiring operational changes or additional mitigation efforts within the development project.

The project approach to implementation planning, also referenced as transition to operations (T2O), is an indicator of the viability of the link to operational resources. Many projects are cocooned within the development environment and operational aspects are not considered until just prior to implementation. Contracted development efforts are frequently structured in this manner, and the deliverable is tested code, not a working implementation. In organizations in which operational support has been outsourced, involvement of operational resources in requirements elicitation is limited by contractual arrangements that may have been in place for many years. Participation of operational expertise may be limited to documented requirements and procedures. The utility of these will depend on the level of consistency between the target operational environment for the project and the existing environment.

If implementation planning is initiated early in the life cycle, the links to operational resources can be established for inclusion of these participants in the requirements elicitation process. This improves the chances for effective consideration of safety and security requirements. Security and safety expertise is highly specialized, costly, and difficult to obtain. Operational support staff can be a resource for this expertise to the project team.

5 A Pilot Project for Improved Security in Software Development

Through a series of workshops conducted in a collaborative effort with a director of programming services and the development staff at a medium size university, an approach to inserting security considerations into a software development methodology was devised. A development project has been selected as a trial for the enhanced methodology to identify specific improvements and refine the approach based on usage.

5.1 Approach Used

While the university participants were aware of the need for security, there was no understanding of what this would mean to the actual development process and how changes would impact the development staff. The following were needed before the specific task of enhancing the software development methodology could begin:

- characterization of the role software development and technology has within the organization (information assets and stakeholders)
- characterization of the threats considered relevant to the organization
- characterization of the potential impacts to the organization if threats were realized (risk environment)
- characterization of the current organizational understanding (stakeholder perceptions) of the role of technology, threat environment, and impact potentials

With the information from the above tasks, the following areas of work can be specified:

- goals of the improvements to be gained with an enhanced methodology
- training and resources needed by developers and project managers using an enhanced methodology
- specific steps within the methodology that are insufficient or incomplete for secure development
- organizational support needed to effectively apply an enhanced methodology

5.2 Applying the Approach

A series of workshops and discussion groups was organized over a period of three days to provide a forum for concentrated information gathering and sharing. The first workshop included the director of programming services and all of the development staff. Through a facilitated discussion led by the SEI, the security concerns, challenges, and organizational environment were characterized. Feedback from participants indicated an increased appreciation for the complexity of security and the impact of development decisions on the security of the installed product. While the group as a whole knew a great deal about software security and the organizational needs, this information was widely disbursed and had never been formally assembled into a coherent perspective.

From the first workshop the following conclusions were identified relative to security considerations within the university development projects:

- Broader user education in security awareness is needed to help seed better development requirements.
- Developers must recognize that security issues are more than viruses and worms.
- The current development methodology is limited in addressing security.
- Security considerations are needed in every step of the development effort to ensure the end result.
- Linking security into the project risk assessment is important.

Using information gathered from the first workshop, a subteam of development staff responsible for the definition of enhancements and changes to the development methodology participated in two subsequent workshops. The first workshop focused on the requirements elicitation portion of the methodology to identify appropriate means for including the elicitation framework developed from this independent research project [Mead 04]. The second workshop focused on the remainder of the development methodology to identify steps and artifacts needed to appropriately incorporate security into all of the steps of the methodology. Key needs that required support outside of development were identified, including system user understanding of the importance of security, better coordination with the operational environment, and the need for stronger security expertise available to the development team at key points in the development cycle.

Two communication vehicles are needed in the development process to document the security discussions of the development team. The first is a high-level view of the security plan, which represents an agreement between the developers and the users defining the level of technology support that will be provided within the system for security considerations. This agreement will be the result of a security risk assessment and will include the planned mitigation steps to be applied. The second is a detailed view that includes the physical segments of the security implementation as the application interacts with the operational environment to provide an agreement between developers and operations as to the details of the implementation of security for the project. A review function for both documents will be provided by the security officer to ensure that an appropriate level of security based on overall university planning is incorporated. This resource will provide the outside expertise and “additional set of eyes” to ensure that development is not missing something obvious because of limited knowledge or limited focus on security.

The identification of metrics for monitoring security compliance to ensure that the results meet the plan will be included as part of the development. These metrics will also be part of the two security documents. In addition, the development team will be identifying ways to capture process metrics to identify ways to justify to the user the anticipated increase of time spent on requirements and security analysis.

5.3 Links to Other Research Focused on Improved Development Processes

Based on work as a member of Microsoft's central security team, Howard has identified some of the same issues that came out of the pilot study described above [Howard 04]. The following are key components that must be addressed for improved security:

- Participants in the development process must be educated in the issues of security and how to address them within the systems development life cycle.
- Understanding the threat environment of the target system is critical to developing appropriate risk reduction designs.
- External review of the architecture, designs, code, and testing at timely points of the development cycle is critical to effective security consideration.

Other sources emphasize the need for including security within each step of the system development life cycle. McGraw provides a suite of best practices to be applied across the system development life cycle to improve the emergent property of security within a developed system [McGraw 04]. These practices include

- explicit security requirements and abuse cases
- risk analysis at strategic points within the system development life cycle
- external reviews by security experts
- penetration testing to stress the security

Analysis of project abandonment [Ewusi-Mensah 03] points to the criticality of expert participation at the point where alternatives for satisfying functional specifications and trade-off decisions are made. This is identified as the point where the technical competence of project participants represents a significant risk factor to project outcome.

6 Furthering the Research

The best frameworks, methods, and tools will not gain broad usage if organizations cannot identify means for inserting them into the existing environments. Identification of organizational roadblocks and conduits for adoption of improved requirements for quality elicitation is only a starting point. Additional case studies for a range of domains, development approaches, and organizational structures are needed to confirm the accuracy and completeness of the characterizations provided within this document. Since safety and security quality attributes were the main focus of this independent study, the management influences need to be validated as influential against other quality requirements for broad consistency.

Consistency in terminology would greatly improve how quality attributes are identified, described, and analyzed. This consistency needs to carry among the many technology domains of architecture, design, development, implementation, and operational support. Standard terminology for communicating about quality attributes between technology specialists and non-technical stakeholders is another area of need. If decision-makers cannot understand what is needed and why with a sufficient level of confidence to apply resources, quality attributes will continue to be inappropriately ignored. Consistency of terminology could be useful when regulations are needed to enforce consideration of quality attributes such as security, privacy, and safety. The current range of definitions provides room for inappropriate decision making.

Training materials are needed for self-study and use with architecture, systems design, and development courses to promote the need for quality requirements and to expose students to the framework for elicitation of safety and security requirements. This research effort used workshops as the communication vehicle to provide an open forum for information exchange, but materials with greater structure will be needed for broad adoption of the requirements elicitation mechanisms and management approaches to improved quality requirements.

Enhancing the risk management capabilities for a project and an organization to include the broad range of risks associated with failures of safety and security is needed. Research to strengthen the importance of links between the various types of risk management applied with the software development and support in an organization is needed.

The pilot for embedding security into a software development methodology shows promise. Further pilot work with different domains, development methodologies, development organizations, and types of development will be needed to assemble sufficient data to define a structured approach.

References

URLs are valid as of the publication date of this document.

[Alberts 01] Alberts, C. & Dorofee, A. *OCTAVE Method Implementation Guide v2.0*. http://www.cert.org/octave/omig.html (2001).

[Alberts 03] Alberts, C. & Dorofee, A. *Managing Information Security Risks*. Boston, MA: Addison-Wesley, 2003.

[Alberts 04a] Alberts, C.; Dorofee, A.; & Woody C. “Considering Operational Security Risks During Systems Development.” *SEPG 2004* (CD-ROM). Pittsburgh, PA: Software Engineering Institute, 2004.

[Alberts 04b] Alberts, C. & Dorofee, A. “Security Incident Response: Rethinking Risk Management.” *Proceedings of the 18th International Congress and Exhibition on Computer Assisted Radiology and Surgery*. Chicago, IL, June 23-27, 2004. Amsterdam, The Netherlands: Elsevier, 2004.

[Anderson 01] Anderson, R. *Security Engineering A Guide to Building Dependable Distributed Systems*. New York, NY: Wiley Computer Publishing, 2001.

[Balfanz 04] Balfanz, D.; Durfee, G.; & Smetters, D. K. “Search of Usable Security: Five Lessons from the Field.” *IEEE Security & Privacy 2*, 5 (September/October 2004): 19-24.

[Ewusi-Mensah 03] Ewusi-Mensah, K. *Software Development Failures*. Cambridge, MA: The MIT Press, 2003.

[Firesmith 03] Firesmith, D. *Common Concepts Underlying Safety, Security, and Survivability Engineering* (CMU/SEI-2003-TN-033, ADA421683). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents/03.reports /03tn033.html.

[Gebel 03] Gebel, G. "Roles and Access Management: Seeking a Balance Between Roles and Rules." *Directory and Security Strategies, Research Overview, 1*. Midvale, Utah: Burton Group, 2003.

[Girard 04] Girard, K. "The Fab Five." *CIO 18*, 3 (November 1, 2004).

[Graham 02] Graham, D. "Requirements and Testing: Seven Missing-Link Myths." *IEEE Software 19*, 5 (September/October 2002): 15-17.

[Hong 03] Hong, K.; Chi, Y.; Chao, L.; & Tang, J. "An Integrated System Theory of Information Security Management." http://www.emeraldinsight.com/researchregister/.

[Hope 04] Hope, P.; McGraw, G.; & Anton, A. "Misuse and Abuse Cases: Getting Past the Positive." *IEEE Security & Privacy 2*, 3 (May/June 2004): 90-92.

[Howard 04] Howard, M. "Building More Secure Software with Improved Development Processes." *IEEE Security & Privacy 2*, 6 (November/December 2004): 63-65.

[ITGI 04] IT Governance Institute. *COBIT Security Baseline*. http://www.itgi.org/.

[Johnson 03] Johnson, S. "Formal Methods in Embedded Design." *Computer 36*, 11 (November 2003): 104-106.

[Laprie 95] Laprie, J. "Dependability—Its Attributes, Impairments and Means." *Predictably Dependable Computing Systems*. Edited by B. Randell et al. Berlin: Springer, 1995.

[Leveson 95] Leveson, N. *Safeware System Safety and Computers*. Boston, MA: Addison-Wesley Publishing Company, 1995.

[Leveson 04] Leveson, N. "A Systems-Theoretic Approach to Safety in Software-Intensive Systems." *IEEE Transactions on Dependable and Secure Computing 1*, 1 (January-March 2004): 66-86.

[May 98] May, L. "Major Causes of Software Project Failures." *CrossTalk*. http://www.stsc.hill.af.mil/crosstalk/1998/07/index.html.

[McGraw 04] McGraw, G. "Software Security." *IEEE Security & Privacy 2*, 2 (March/April 2004): 80-83.

[Mead 04] Mead, N. R. "Requirements Elicitation and Analysis Processes for Safety & Security Requirements." *Proceedings of the Third International Workshop on Requirements for High Assurance Systems (RHAS 2004)*. Kyoto, Japan, Sept. 6, 2004. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. http://www.sei.cmu.edu/community/rhas-workshop/rhas04-proceedings.pdf.

[Pethia 01] Pethia, R. "Information Technology—Essential but Vulnerable: How Prepared Are We for Attacks?" Testimony before the House Committee on Government Reform, Subcommittee on Government Efficiency, Financial Management and Intergovernmental Relations, September 26, 2001. http://www.cert.org/congressional_testimony /Pethia_testimony_Sep26.html.

[Sarkar 04] Sarkar, D. "Two Converging Worlds: Cyber and Physical Security." FCW.COM. http://www.fcw.com/fcw/articles/2004/1213 /tec-convergsec-12-13-04.asp (2004).

[SEI 03] SEI. *Quality Measures Taxonomy*. Software Technology Roadmap. http://www.sei.cmu.edu/str/taxonomies/qm_tax.html.

[SEI 04] SEI. *Glossary*. Software Technology Roadmap. http://www.sei.cmu.edu/str/indexes/glossary/.

[Tryfonas 01] Tryfonas, T.; Kiountouzis, E.; & Poulymenakou, A. "Embedding Security Practices in Contemporary Information Systems Development Approaches." *Information Management & Computer Security 9*, 4 (August 2001): 183-197.

[Verdon 04] Verdon, D. & McGraw, G. "Risk Analysis in Software Design." *IEEE Security & Privacy 2*, 4 (July/August 2004): 79-84.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
March 2005

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
Carol Woody, PhD

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2005-TN-010

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

Early in the literature review for the Independent Research and Development Project for Eliciting and Analyzing Quality Requirements, the potential conflict of quality efforts (perceived as time consuming) with organizational management direction (driven by time-to-market and cost considerations) was identified. Quality attributes are influenced by the selection and development of components that make up a system, as well as the development environment in which the system is created. A variety of information sources, including conferences, workshops, pilot projects, and technical assessments, was tapped to identify specific management barriers to the adoption of improved elicitation approaches and appropriate organizational behaviors that facilitated the use of improved mechanisms for the elicitation and analysis of quality requirements. This report documents the ways in which the organizational and project management environment for system development can support or reject improved quality requirements elicitation mechanisms. In addition, this report identifies specific activities as promoting improved quality requirements elicitation when they are embedded into the system development life-cycle structure.

14. SUBJECT TERMS
software quality, quality requirements elicitation, software assurance, software management

15. NUMBER OF PAGES
34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL