Communications of the Association for Information Systems

Volume 34 Article 71

1-2014

Web Engineering Security (WES) Methodology

William Bradley Glisson
University of South Alabama, Mobile, bglisson@southalbama.edu

Ray Welland
School of Computing Science University of Glasgow, Glasgow

Recommended Citation
Glisson, William Bradley and Welland, Ray (2014) "Web Engineering Security (WES) Methodology," *Communications of the Association for Information Systems*: Vol. 34 , Article 71.
DOI: 10.17705/1CAIS.03471
Available at: https://aisel.aisnet.org/cais/vol34/iss1/71

Communications of the Association for Information Systems

CAIS

Web Engineering Security (WES) Methodology

William Bradley Glisson
School of Computing
University of South Alabama, Mobile
bglisson@southalbama.edu

Ray Welland
School of Computing Science
University of Glasgow, Glasgow

Abstract:

The impact of the World Wide Web on basic operational economical components in global information-rich civilizations is significant. The repercussions force organizations to provide justification for security from a business-case perspective and to focus on security from a Web application development environment standpoint. The need for clarity promoted an investigation through the acquisition of empirical evidence from a high level Web survey and a more detailed industry survey to analyze security in the Web application development environment ultimately contributing to the proposal of the Essential Elements (EE) and the Security Criteria for Web Application Development (SCWAD). The synthesis of information provided was used to develop the Web Engineering Security (WES) methodology. WES is a proactive, flexible, process neutral security methodology with customizable components that is based on empirical evidence and used to explicitly integrate security throughout an organization's chosen application development process.

Keywords: industry, organization, method, case study, security, information systems

Volume 34, Article 71, pp. 1359-1396, January 2014

I. INTRODUCTION

The World Wide Web (WWW) instigated radical paradigm transformations in today's global information-rich civilizations. Many of society's basic operational economical components such as healthcare, government agencies, and financial services depend on Web-enabled systems in order to support daily commercial activities. E-commerce achieved global acceptance as a valid channel for conducting business. The total U.S. e-commerce spending in 2010 reached $227.6 billion [comScore, 2011]. U.S. e-commerce sales for the second quarter of 2011 alone were $47.5 billion, which is an increase of 3 percent from the first quarter of 2011 [U.S. Department of Commerce, 2011]. The money spent on e-commerce applications to support this revenue stream is in the billions. PricewaterhouseCoopers' (PWC) [2011] survey indicates that organizations are reducing information security budgets or deferring security initiatives. Even though Deloitte's [2010] survey indicates that budgets are not as large of an obstacle as in the past, it is still a major impediment for many organizations. The latest Computer Security Institute (CSI)/FBI survey [2010/2011] puts forth the idea that IT budgets may have been cut overall, but security expenditures were considered mandatory from an investment perspective. Complicating matters, Deloitte's [2010] survey indicates that less than a third of their respondents "have established information security metrics aligned (with) business value(s) and report on a scheduled basis." The economic, legal, and societal interest in the growth of e-business is creating a demand for a more secure Web-enabled business environment. Despite the critical role of security, in the potential growth of e-commerce, reports are continually produced by CSI/FBI [Computer Security Institute, 2009, 2010/2011] illuminating the fact that security breaches continue to cost organizations millions of dollars annually.

Over the past several years, security developed into a focal point of interest in industry. This is evident through the results of industry surveys and press releases focusing on security from large corporations like Microsoft [Bradley, 2011] and IBM [2011]. The 2011 survey by Frost and Sullivan indicates that "application vulnerabilities represent the number one threat to organizations" [Ayoub, 2011]. They go on to indicate that 20 percent of the information security professionals in their survey were involved in software development.

The synergy of this information indicates that organizations need to approach security issues from a calculated and business-aligned perspective. This places additional pressures on Web Engineering applications to successfully integrate security into the development process. Web Engineering is:

> *… the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web-based applications* [Deshpande, 2004, Deshpande, Murugesan, Ginige, Hansen, et al., 2002].

It is important to recognize that "Vanilla—Off the Shelf" Web Engineering methodologies do not inherently make any direct references to security. Consequently, today's Web applications face increased susceptibility to major security problems.

Increasing academic and commercial discussions highlight the need for security integration into the software development lifecycle. This battle cry, echoed by many in the industry, generally fails to detail how this integration can be effectively achieved. The market is producing economic support for an idea, as quoted by Steven R. Rakitin, that W. Edwards Deming put forth several years ago stating that "[t]he quality of a product is directly related to the quality of the process used to create it" [Rakitin, 1997]. A major difference between Web application development and conventional software development is a greater emphasis on security [Deshpande et al., 2002]. Hence, the increase in costs associated with security issues should raise concerns over the way security is addressed in Web application development processes.

The research proposition is that an impartial security methodology applicable to different Web Engineering development processes will strengthen security. The WES methodology incorporates ideas from academic research, Web surveys, and qualitative open-ended industrial surveys. Prior to developing the methodology, a survey of professionals in a financial services company was carried out, which clearly identified some major problems with existing security processes and some lessons that could be learned. The results of this survey are included in the appendix to this article. WES supports the concept that software development methodologies must integrate security as a specific objective within all stages of the development process [Glisson, 2008]. WES also builds on existing research by addressing the criteria for fifth-generation security methodologies [Siponen, 2005a]. There are a wide variety of published software development methods [Ramsin and Paige, 2008], and many organizations customize methods based on business need or use hybrid methods. Therefore, it would not be productive to define another completely new software development process; hence, WES is designed to be "process neutral." The result is a practical, impartial security methodology that integrates security into an organization's existing Web Engineering development methodology. This methodology was partially implemented in the same Fortune 500 financial organization in which the initial survey of professionals was conducted.

This article presents the Web Engineering Security (WES) methodology in its entirety which is described in detail in a Ph.D. dissertation [Glisson, 2008]. Previous publications have discussed specific aspects of the methodology but do not examine the entire process. Section II discusses relevant work. Section III covers the research methodology and previous work related to the development of the methodology. Section IV explains the Web Engineering Security environment. Section V explains the WES core principles. Section VI covers the WES lifecycle and Section VII discusses stakeholders and deliverables. Section VIII discusses WES goals, including industrial practicality, and Section IX discusses a practical application of WES. Section X provides a summary and describes future work.

II. RELEVANT WORK

In order to appreciate the current state of the security methodology research, it is necessary to acknowledge previous research in the field of information security design methods. Baskerville's [1993] analysis separated numerous system methods into three generations. The first generation consisted of checklists and risk analysis. This stage focused on actual physical systems' specifications. The second generation engineering methods focused on complex customization through the use of engineering concepts and mechanistic procedures that relied heavily on functional requirements. Baskerville cites Waters in his explanation of mechanistic engineering methods; stating that mechanistic engineering methods:

> *… focus on the production of mechanical specification of input, storage, and output formats, along with details of procedures needed to transform input or storage into outputs* [Baskerville, 1993].

Baskerville goes on to indicate that common tools implemented with these methodologies include system and program flowcharts, record layouts and print charts. Baskerville notes that the waterfall methodology [Royce, 1987/1970] is an example of a mechanistic engineering application development approach. The second generation security development methods are summed up by Baskerville as top-down engineering, rapid prototyping system, and logical flowchart methods. This summary would include solutions like Fisher's approach, Parkers' security diagram, and the U.K. Government's Central Computing and Telecommunications Agency's (CCTA) Risk Analysis Management Method (CRAMM) [Baskerville, 1993]. The third generation of security methods are model-driven. Baskerville cited Structured Systems Analysis and Design Methods (SSADM) and the Logical Controls Design method as examples of third-generation security models. Even though Baskerville's analysis of the security design methods did not directly examine the applicability of the security methodologies to the Web development, he did make an important point that is applicable to Web Engineering application development. Baskerville's analysis did suggest that:

> *… systems methods will neither be trustworthy nor successful unless the general research regarding systems methodology incorporates security analysis design as an explicit objective* [Baskerville, 1993].

Baskerville [1992] pointed out in a previous paper that security must be practical and feasible while mitigating, what he terms as conflict development duality. Baskerville described conflict development duality as the tension between developing a secure system versus a functional system.

Siponen [2005a] updates and expands on Baskerville's analysis of information security development approaches declaring that there are five information system security generational classifications. Siponen arrives at his conclusion after an examination of the contributing research disciplines and an evaluation of seventeen modern information system security methodologies. Security is a highly diverse research subject that is an area of interest for a variety of disciplines. Siponen identifies four research communities as contributors to information security research, including Management Information Systems (MIS), computer science, software engineering, and mathematics. According to Siponen's research, MIS accounts for the social and the organizational aspects of a problem. Computer science has a "positivist" [Hirschheim, 1985; Siponen, 2005a] orientation, which is understood to be the application of scientific methods, to solving computing problems. Software engineering uses both a positivist and an interpretive approach, while mathematics takes a quantitative approach to solve problems. An interpretive approach, in this context, is read to mean that the researcher is attempting to understand the data and the results generally within the social context and the context of the information system [Klein and Myers, 1999; Siponen, 2005a]. The reality is that research from any of the contributing disciplines can be classified as interpretive or positivist depending on the specifics of the research. The evaluation of seventeen modern information systems contributed to the creation of the two additional security methodology generations.

Siponen's first three generations correspond with Baskerville's generational classifications. Siponen [2006] explains that the first and second generations include checklist, management criteria, and maturity criteria. Checklist attempts to solve security problems through the identification and implementation of countermeasures via a list [Siponen, 2006]. An example of a checklist is the Security Audit and Field Evaluation (SAFE) for Computer Facilities and Information Systems [Siponen, 2005b]. According to Siponen, the idea of standards evolved from checklists into recommendations that the organization should implement. Siponen [2006] explains that by meeting specific standards and/or achieving certifications, organizations are able to display a level of management and trustworthiness to business partners and customers. Some well-known standards in use today include the International Organization for Standardization and the International Electrotechnical Commission standard (ISO/IEC) 17799/27002 [27000.org Directory, 2014; ISO, 2012; Siponen, 2006], the Systems Security Engineering—Capability Maturity Model (SSE-CCM) [2003], and the Common Criteria (CC) [Common Criteria, 2012].

ISO/IEC 17799/27002 attempts to provide fairly comprehensive information security management recommendations in regards to initiating, implementing and maintaining systems that are concerned with information security [27000.org Directory, 2014]. ISO/IEC 17799 consists of several sections that contain information on everything from security policies to asset management, to human resource security, to business continuity management [27000.org Directory, 2014]. ISO/IEC 17799 does define information security in terms of confidentiality, integrity, and availability [27000.org Directory, 2014]. It should be noted that ISO/IEC 12207:2008 specifically addresses systems and software engineering lifecycle processes [International Organization for Standardization]. The standard tries to address a broader scope by examining aspects of the following areas: Project initiation and planning, Functional requirements, System design specifications, Build and document, Acceptance, Transition to production, Operations and maintenance support, and Revision and system replacement [Hansche, Berti, and Hare, 2004].

SSE-CCM [2003] presents a document-intensive, best practices, highly structured model solution designed to support statistical process control to all forms of software engineering. The SSE-CCM version of the lifecycle includes concept, development, production, utilization, support, and retirement stages. This all-inclusive approach is composed of twenty-two processes. The first eleven process areas focus on security and the last eleven focus on "project and organizational activities" [SSE-CMM, 2003]. The process areas that focus on security provide a high level initiative telling organizations what to address. For example, under Coordinate Security, they indicate that:

> *… all members of the project team are aware of and involved with security engineering activities to the extent necessary to perform their functions* [SSE-CMM, 2003].

This statement focuses on the team, not the methodology being used. The CC attempts to fuse an assortment of international standards into a set of evaluation criteria to be utilized against information technology products [Rothke, 2004].

Siponen defined the third generation as consisting of structural and object-oriented security methods, information modelling methods, and stepwise security methods. He also indicates that the third generation is focusing on modelling information system security requirements. Third generation security models would include approaches like the Spiral Approach, the Logical Approach, Data Flow Diagrams (DFD), and Entity Relationship (ER) modelling. According to Siponen, the fourth generation builds on the third generation by addressing the social and socio-technical aspects of the methods. The term socio-technical was originally coined by Bostrom and Heinen [1977] in a paper in which they were examining Management Information Systems (MIS) project failures. They described an organization work system as being comprised of two components: the social and the technical. Bostrom and Heine went on to explain that the technical aspects focused on the task, the processes, and the technology. While the social side of the system is focused on people attributes, relationships, reward systems, and authority structures. Basically, the social component is concerned with the management aspect of the business. Bostrom and Heinen [1977] defined the socio-technical perspective as an intermediate position between the two extremes. Siponen gives the Survivable IS approach as an example of a fourth generation methodology.

The fifth generation, of security methodologies, that Siponen [2005a] discusses is really the next generation of methodologies. This implies that the fifth generation security methodologies do not currently exist, a point which he also articulates in a later article [Siponen, 2006]. Siponen describes four criteria that the fifth generation security methodologies should strive to achieve. These criteria are as follows:

- Integration with all types of software development methodologies
- Painless adaptability of security methods with practitioners
- Provide empirical evidence of their usefulness [Siponen, 2006].
- Use of social ideas and techniques ensuring congruent design and user expectations

Siponen, Baskerville, and Heikka [2006] did propose, develop, and test a Meta-Notation solution for secure information design methods. In this research, they defined six meta-requirements as a way to analyze secure information systems. However, it has been argued that security is more than notation and modelling. Secure application development is both technical and social [Mouratidis, Jürjens, and Fox, 2006]. Secure application development does not end when testing is completed. It needs to take into account implementation into production and end-user feedback. Siponen and Heikka [2008] provide a critique of several secure information systems design methods from a modelling perspective. The ability to model data from a security standpoint is advantageous. However, the approach to modelling should be based on the needs and the capabilities within individual organizations. Siponen's points, regarding the fifth generation, brings us to the heart of the security problem. There have been few industrial attempts to comprehensively address user-focused aspects; methodology integration; practitioner malleability, and employment of Web Engineering Security throughout the Web-based application development process via the establishment of a comprehensive security methodology.

The need for information security is attributed to several factors ranging from the enormous interconnection of assorted and distributed systems, the existence and availability of sensitive information, computer crime anonymity, the lack of geographic boundaries, and the absence of forensic evidence [Joshi, Aref, Ghafoor, and Spafford, 2001]. Baskerville [1993] noted the lack of security in development methods. Baskerville [1993] recognized that third-generation information systems development methodologies lacked security considerations. This problem still exists today. There is a lack of security methodologies that are compatible with existing application development methodologies. Siponen's [2005a] analysis only found three approaches that could be smoothly integrated in information systems development methodologies. According to Siponen [2005a], these methodologies are Baskerville's logical approach [1989], Booysen and Eloff's spiral approach [1995], and McDermott and Fox's abuse case methodology [1999]. Siponen and Baskerville point out that Agile methodologies "have few features specifically addressing security risk" [Siponen, Baskerville, and Kuivalainen, 2005]. Baskerville, Ramesh, Levine, Pries-Heje, et al. [2003] discussed the implications of Internet-speed development for software management which does not specifically mention security. They suggest that "... development speed is paramount," hence, "Quality becomes negotiable..." [Baskerville et al., 2003]. However, security should be negotiable based only on an in-depth risk analysis.

Security is inherently not a part of "Vanilla—Off the Shelf" Web engineering development processes, and this inherent lack of security encourages environments that are susceptible to exploitation via potential breaches. Web Engineering methodologies do not make any direct references to security, consequently today's Web applications face major security problems. Therefore, an improved definition of Web Engineering Security modifies Deshpande's explanation of Web Engineering [Deshpande, 2004, Deshpande et al., 2002] as follows:

> *Web Engineering Security is the systematic, disciplined, and quantifiable amalgamation of security with a Web-based application development process.*

A specific Web Engineering Security methodology provides a road map for developers and management to follow during a Web-based application development project. A methodology attempts to provide guidance for all of the various aspects of security during the individual stages of the application development process. In order to allow organizations and individuals to preserve and capitalize on existing Web application development capabilities, and possible market advantages, a process neutral approach was explored. The phrase "a process neutral approach" is chosen to convey the idea that the design of the security methodology endeavors to seamlessly integrate with a variety of existing Web application development methodologies.

A process neutral approach to the implementation of security is based on the fact that organizations use a variety of methodologies during their Web application development projects. This variety ranges from the traditional Waterfall approach, or some variant thereof, to agile approaches in order to support Web application development. A process neutral approach provides an organization with the opportunity to support its existing Web application development methodology regardless of the style of the methodology. It also complies with Siponen's recommendation that new methodologies should strive to integrate with all types of software development methodologies [Siponen, 2005a].

The process neutral approach provides a roadmap for organizations that are using a more traditional methodology for Web application development from a deliverable perspective. The number of deliverables that an organization will require depends on the culture of the organization, the methodology that the culture is comfortable with implementing and, to a large extent, the regulatory impact on the business. Businesses that are more conservative in nature, and under a large amount of regulation, such as a large financial institution, are going to require deliverables at every stage of the development process. On the other hand, smaller businesses are more inclined to be agile in nature and require fewer deliverables during each stage of the development process. A process neutral approach allows a methodology greater flexibility to support agile methodologies. The Agile community's manifesto states that:

> *We are uncovering better ways of developing software by doing it and helping others do it. We value:*
>
> - *Individuals and interactions over processes and tools.*
> - *Working software over comprehensive documentation.*
> - *Customer collaboration over contract negotiation.*
> - *Responding to change over following a plan.*
>
> *That is, while there is value in the items on the right, we value the items on the left more* [Agile Alliance Organization, 2007; Fowler and Highsmith, 2001].

In order to support the agile community's manifesto, a new security methodology needs to be flexible enough to integrate with existing Web application development methodologies in order to meet the needs of specific organizations. At the same time, it needs to encourage interaction among project members. This increased interaction among all of the individuals involved in the project, over security issues, raises the overall security visibility of the Web application while supporting software deliverables. The security methodology needs to encourage customer input into the security aspect of the Web application. The flexibility of the methodology provides the implementing organization the freedom to decide on the amount of documentation that is appropriate for the Web application being developed and the culture of the organization. The overall flexibility of the methodology allows the implementing organization the capability to decide the rigidity of the methodology. Pursuing a process neutral approach attempts to support the ideals of the agile manifesto along with providing the flexibility to integrate into traditional application development methodologies.

Direct contributions of individuals involved in Web development projects provide the fundamental ingredients for a project's ultimate success or failure. This is especially true in the security arena. The methodology should support the individuals involved in the development process by providing guidance so that the end product is a secure Web application, while meeting the needs of the customer and business. This necessity for versatility supports research into a process neutral approach in order to allow appropriate customization while meeting the needs of the individual stakeholder groups.

III. PREVIOUS WORK RELATED TO THE DEVELOPMENT OF THE WEB ENGINEERING SECURITY METHODOLOGY (WES)

This section presents the research methodology and summarizes the authors' previous published work that motivated and informed the development of WES. The research presented in this article was conducted from a pragmatic perspective. As Walsham [2006] noted, "... action research represents a highly involved type of research study." This involvement made it an ideal approach to investigate building security into large-scale application development projects. Action research is characterised by focusing on practical issues, placing an emphasis on change, interactive cycles, collaborating with practitioners, and encouraging multiple data generation approaches along with practical and academic outcomes [Oates, 2006].Four data collection methods were used in this research: (1) relevant literature review, (2) Web survey, (3) interviews, and (4) participatory observation. Relevant literature was gathered at the start of the research to help understand the overall problem. A Web survey was conducted to gain a boarder perspective of the issue. In-depth one-to-one interviews were conducted to understand the issues within a large financial organization. The researcher was embedded in the organization for a year and a half to complete the project. This immersion into the organization helped the researcher understand how the organization worked from a pragmatic perspective. The research strives to discover an effective solution to the integration of security into large-scale development projects based on the means available in the organization, the constraints of a multi-faceted environment, and the fluidity of a working development environment. The WES methodology was developed, implemented, and evaluated in the organization. The results of the research were presented to both industrial partner management and academic audiences.

Previous work by McDonald and Welland on Agile Web Engineering gave us some insight into the problems of developing commercial Web-based applications [McDonald, 2004; McDonald and Welland, 2005]. We developed our initial ideas on the structure of WES based on a literature survey of security problems in software systems [Glisson and Welland, 2005]. To improve our understanding of specific problems in Web Engineering Security, we carried out two surveys that provided further evidence to inform the refinement of the WES process.

The first of these was a Web-based survey that identified several elements that organizations were failing to address when considering a Security Improvement Initiative (SII) for Web development projects. The detailed results of this survey were reported in the Web Survey Technical Report [Glisson and Welland, 2007b] and summarized in a previous paper [Glisson and Welland, 2007a].

The five essential elements indentified in this survey were:

EE1. **Web Application Development Methodology.** Before security can be addressed in an organization's Web application development process, there needs to be an application development methodology in use within the organization. The work on AWE addressed this issue [McDonald, 2004; McDonald and Welland, 2005].

EE2. **Web Security Process Definition.** There needs to be a clear organizational definition of security and its impact on the business, Web applications, and the development process. This informs the definition of a development process that integrates appropriate security measures into the existing development process in order to produce a more secure end-product. This motivated our work on WES.

EE3. **End -users Feedback.** In many organizations there is a lack of end-user feedback in the internet, intranet and extranet development processes. McDonald and Welland [2005] previously advocated strong support for end-user participation in Web application development. McDonald [2004] carried out an evaluation of an existing internet banking user interface using a random sample of members of the public. This revealed a significant number of simple problems that were addressed within a week and re-evaluated with a different random sample, which showed that the interface had been greatly improved. No attempt had previously been made to test the interface with real end-users.

EE4. **Implement and Test Disaster Recovery Plans.** Our survey revealed that about half of the respondents' organizations had a disaster recovery plan, but only about half of these had been tested in the last twelve months. The growing importance and continued vulnerability of Web-based applications mean that organizations need to consider risk management and business continuity through disaster recovery planning. Organizations have to make hard decisions on exactly how much risk they are willing to accept and exactly how much money they are willing to spend to achieve the agreed upon level of security [Glisson and Welland, 2006a].

EE5. **Job-related Impact.** The survey revealed that the majority of the organizations do not have a job-related impact for not following the security development process. There needs to be a job-related impact associated with security process compliancy. Employees need to understand that there is a job-related impact for not following organizational processes. This becomes even more important when considering security.

The second survey was based on one-on-one interviews with a range of employees within a large financial services organization. The details of this survey and the interviews are given in Appendix 1. This survey was conducted in the same organization as the research for the Agile Web Engineering (AWE) process, and the new results from the application development part of the survey support previous findings [McDonald, 2004]. This survey sample consisted of various employees representing a variety of roles with a diversity of work experiences within the technical side of the organization. The detailed results of this survey were reported in the WES Application Survey Technical Report [Glisson and Welland, 2006] and analyzed in a previous paper [Glisson, McDonald, and Welland, 2006b].

The survey identified six Security Criteria for Web Application Development (SCWAD):

SC1. **Active organizational support for security in the Web development process.** Active managerial support for security in the Web development process is critical. Without the support of management, there is no hope for effective integration of security within the development process.

SC2. **Proper controls in the development environment.** The term *proper controls* is a very broad term that encompasses policies, knowledge, technology, and processes. These controls are a necessity, and they help to provide structure to the development environment.

SC3. **Security visibility throughout all areas of the development process.** Security must be visible throughout all areas of the development process. The survey findings indicated that there was a problem with visibility due to the fact that, after design approval, there is no verification that the implemented design matches the approved design.

SC4. **Delivery of a cohesive system, integrating business requirements, software, and security.** The goal of any development process should be to deliver a cohesive system, integrating business requirements (needs), software, and security. This means that the security requirements of the business need to be identified as early as possible in the development process so that they can be incorporated into the design and the construction in order to produce secure software.

SC5. **Prompt, rigorous testing and evaluation.** The development process must include rigorous end-user relevance testing and evaluation. Testing is critical to the success of an application. Testing should be conducted from a design and programming perspective. Testing must take into consideration how much is realistically achievable with available resources and within tight timescales.

SC6. **Trust and Accountability.** The development process should encourage the development and maintainability of trust and accountability within the organization. Trust and accountability really make up the heart and soul of security.

A Web Engineering Security methodology needs to exist within an organizational framework that actively supports security both within the development process and more widely across the whole organization. Therefore, before discussing the core principles of a Web Engineering Security methodology, the organizational environment within which the methodology exists needs to be considered.

IV. WEB ENGINEERING SECURITY ENVIRONMENT

A Web Engineering Security process exists within an organizational structure, and it is essential that the culture of the organization embraces all aspects of security. A Web Engineering Security process, like any other development process, is of little use if it is not embedded in a supporting environment.

The essential features of this environment include good communication, security education, and cultural support. These principles are interdependent and need to work in concert in order to achieve and maximize the desired effect from a security perspective. This concept is illustrated in Figure 1—The Security Environment.

The implementation of these concepts within a specific organization can be achieved through the execution of individual or a combination of theories. How a particular organization chooses to achieve these goals is dependent on the culture of that company. As Ahmad Al-Omari et al. [2012] note, several theories have been proposed in the area of information security awareness. D'Arcy and Hovar [2009] explored the effects of counter-measures in terms of general deterrence theory, Bulgurcu, Cavusoglu, and Benbasat [2010] investigate factors that impact employee compliance with information security policies, along with the impact of information security awareness and employee attitudes. The result from their study indicate that an employee's attitudes, beliefs, and self-efficacy impact their intention to comply with information security policies. Research in the area of security training has been conducted by several researchers, including Puhakainen and Siponen [2010], Karjalainen and Siponen [2011], and D'Arcy, Hovav, and Galletta [2009]. While these papers propose different solutions to address security training, education, and behavior in organizations, all of the papers agree that security policy noncompliance is a major threat to an organization's security. Puhakainen and Siponen's [2010] findings also "indicate that visible support of IS security by top management is necessary to ensure that users comply with IS security policies."

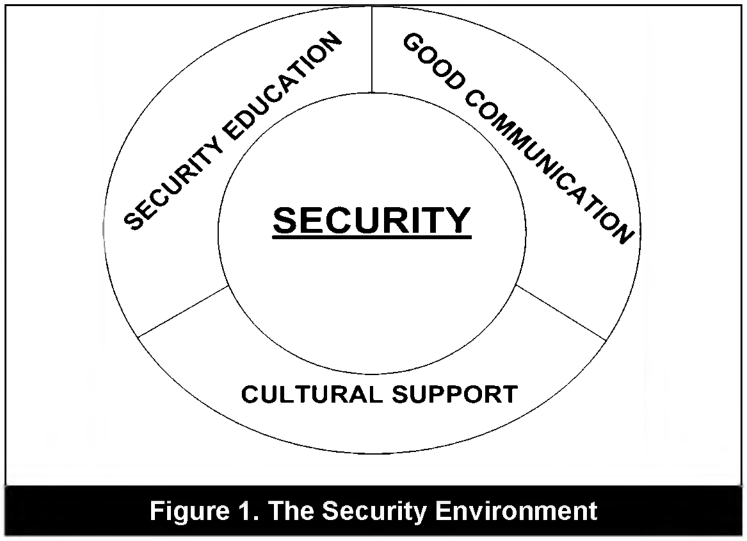


Figure 1. The Security Environment

Security Education

The Organization for Internet Safety (OIS) publishes *Guidelines for Security Vulnerabilities Reporting and Response* [2004]. These guidelines highlight the fact that any flaws in the system design or application coding can potentially lead to security vulnerabilities. This means that security education should cover an array of issues, including knowledge transfer, coding practices, technical attacks, social engineering attacks, security processes, everyday activities, and potential impact analysis methods.

This problem is emphasized due to the availability and accessibility of Web applications. Hence, designers and developers must be educated on common development flaws, best coding practices, and the implementation of practical development solutions. Security should not be left to the acquisition of the functional and non-functional security requirements. Security is more than a technical issue; it is a people, a process and an educational issue that must be addressed in its entirety. Organizations need to encourage knowledge transfer among employees and provide for proper training. As Niekerk and Solms [2004] noted, “Employees must know why information security is important and why a specific policy or control is in place.” They go on to break information security education into three areas: awareness, training, and education. Awareness highlights information security to focus an employee’s attention. Training is used to impart necessary competencies and skills while education integrates competencies, skills, and specialities into a universal collection of knowledge.

Education is an important area of the security process. Security education should not only include raising awareness of the different types of technical attacks and social engineering attacks [Mitnick, 2002], but it should also include information about the current environment. Expanding on Siponen’s [2000] idea that security guidelines must be justified, along with understanding relevance, employees should know with whom they should discuss security, how it fits into their everyday work environment (i.e., their development process), and the potential impact security imposes on the Web application solution that they are implementing.

Good Communication

Good communication is a critical component of the environment, as it is needed to assure solution cohesiveness within the development team and within the organization. In order to cut down on possible confusion and to ensure that everyone is communicating properly, organizations should define:

- What security means to the business
- What it means to a Web application
- What it means in the development process
- What a Web Engineering Security development process entails

Good communication helps to provide the foundation for security visibility throughout the entire application development methodology. Hence, good communication should encourage security visibility through the development process, an auditable process, a clear understanding of the defined metrics, the delivery of a cohesive system, and the dissemination of the importance of the integration of development and security methodologies.

In order to achieve these goals, there needs to be good communication within the organization. This includes good communication between management and the development team, among members of the development team, between the development team members and the stakeholders from the business unit, and between the development team and end-users. The communication between management and the development team is needed, due to the fact that management is responsible for setting the policies, standards, and procedures to which the development team must adhere.

Development Team Communication

Communication should be encouraged and fostered in the technical side of the organization. The organization’s management, in concert with the infrastructure architects, need to provide a security vision for the future. This can be communicated in the present through the creation of current and future standards. This support and integration with communication is a critical component for the purpose of driving future security initiatives in an organization. The marketing and dissemination of this information is necessary to effectively implement this initiative. If your employees do not know about the tools that are available, they will not use them. It is also true that if the tools and/or methods are not effective in completing the job, are too complicated to use effectively, or are just not user friendly, then employees are likely to avoid using them.

If the tools or the methods are not productive for various reasons, then the individual members of the development team should be encouraged to suggest alternative tools or methods to be evaluated. A channel for communicating feedback to management for both positive and negative communication needs to be established in the organization. If this channel is not established, then developers will inevitably use their own tools to complete the job. Their decisions, realistically, could range from using off-the-shelf solutions to open source software. Off-the-shelf software could put the organization in jeopardy from a legal perspective. This idea is supported by investigations into the integration of legal requirements into methodologies [Compagna, Khoury, Krausová, Massacci, et al., 2009]. Then there are legal implications when software is sold for personal use and used in a commercial environment. On the other hand, open source software could introduce potential security breaches into the organization.

The interaction between management and the developers helps to introduce and sustain flexibility in the Web application process. More importantly, it gives the development team a sense of ownership in reference to the methods and the tools that are used in the development process. Along with this interaction, all the tools and the methodologies that are used in the development process need to be reviewed frequently. This review helps to ensure that the tools and the methodologies are achieving the desired goals. Baskerville et al. [2003] advocate the practice of "tailoring the methodology daily" when doing Internet speed development and also discuss the use of different methodologies for different releases of a system. One of the twelve principles of Agile Software in the Agile Manifesto [Beck, 2001] is: "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly." For example, the Scrum process [Schwaber and Sutherland, 2011] includes a review meeting at the end of every iteration, called the *Sprint Retrospective,* which includes encouraging the Scrum Team to "improve its development processes and practices." Developers should also be encouraged to share technical knowledge with each other. This distribution of information encourages debate on technical solutions and distributes application knowledge through the group. This distribution of knowledge helps keep a balance in the group, thereby reducing dependency on individual employees.

Stakeholder Communication

Communication with the business unit stakeholders is needed to acquire the appropriate application requirements. Security solutions must also be confirmed with these stakeholders. This communication directly impacts the potential effectiveness of a security solution. The security that is implemented should meet the needs of the organization so that it adds value to the end product and to the overall business process. Essential elements that contribute to good communication include a clearly defined application development methodology and a clearly defined Web security development process. Business stakeholders must be explicitly included throughout the security development process.

Baskerville et al. [2003] observed that customers are "implanted" in all phases of the development environment in Internet-speed development in order to deal with fuzzy and rapidly changing requirements rather than relying on formalized requirements management processes. One of the twelve principles identified in the Agile Manifesto [Beck, 2001] states that "Business people and developers must work together throughout the project." For example, XP [Beck and Andres, 2004] uses the concept of the "on-site" customer as part of the development team.

End-user Communication

Balfanz, Durfee, Smetters, and Grinter [2004] state "Usability must be designed into systems from the beginning of the development process and this is particularly true for usability of security." They also warn against the security experts' belief that security is more important than any other users' needs, including whether the user can actually use the security mechanisms. This means that system designers and security specialists must be aware of end-users' capabilities and perceptions. In an intranet or extranet system development, where the user community is homogenous, it is possible to identify representative groups of users who can be consulted about end-user needs and perceptions of security during system development. These groups can also be involved in system testing and evaluation of the delivered product. However, it is more difficult to identify such representative samples of end-users for an Internet application that includes the general public.

Cultural Support

Cultural support should drive the efforts in security and education along with the efforts in good communication. Cultural support for security should embrace confidentiality, integrity, and availability throughout the management structure. Active organizational support for security in the Web development process is critical. Without the support of management, there is no hope for effective integration of security within the development process. Managerial support for security needs to be both proactive and reactive. Management needs to be proactive by supporting employees, hence, giving them the necessary tools and developing the necessary policies so that employees can be successful in their endeavors. This would include proper controls for the development environment such as software versioning controls, providing up-to-date code libraries, setting the policies for testing code and for establishing trust and accountability within and outside of the organization.

Likewise, management needs to be reactive by stating and enforcing job repercussions if employees do not follow security practices within the development process or the development process in general upon which the security process depends. One solution would be to provide positive and negative reinforcement. The idea is to reward individuals who adhere to the security process. An example would be to provide monetary rewards to programmers based on the amount of secure code they produce, not the total amount of code that they generate. On the other side of this issue, there needs to be repercussions for individuals who do not follow the organization's security development process. Another idea is to tie security to the employees' yearly evaluation [Wylder, 2004].

Security Synergy

The environment that is most conducive for fostering security in the Web application development environment is the intersection of all three principles. The intersection of security education and practicing good communication should help build confidence in the overall security of the organization, the general security knowledge of the employees and encourage compliance with organizational policies. The distribution of security information and how that impacts the daily activities of employees helps to provide practical solutions to security issues. This approach helps to propagate the concept that security needs to be viewed in the application development process as "everybody's problem" [Graff and van Wyk, 2003]. Integrating security responsibilities and security education into the development process increases employee confidence in addressing security issues and sends the signal to the development group that security is an important issue.

A metric system should also be developed that helps the organization determine the success of the development process security initiative. This should include issues such as general security education, training, monitoring, and tracking all development bugs. This will help the organization determine if it is actually delivering a cohesive system that integrates the business, the software, and the security perspectives.

V. WEB ENGINEERING SECURITY (WES) CORE PRINCIPLES

The Web Engineering Security (WES) methodology was designed to complement Web software development through customer communications, short development cycles, and practical security solutions to business problems [Agile Alliance Organization]. WES attempts to achieve this by stressing core principles while providing a general outline with customizable subcomponents. These core principles have been distilled from our earlier work and relevant literature.

Process Neutrality

WES is designed to be process neutral, which means that it can be integrated with an existing process for Web application development. The objective is to ensure that security considerations are integrated into every phase of the development process. The WES process strengthens security in a Web application development environment by implementing a security process that integrates seamlessly into an organization's development process capitalizing on existing synergies. In practice, organizations have invested, to varying degrees, money, time, and staff training into their own Web application development methodologies. The variety of application development processes found within organizations ranges from traditional Waterfall to completely Agile, or some hybrid of these approaches. A customizable security neutral methodology provides an organization with the opportunity to support its existing investment in its chosen Web application development methodology regardless of the implementation details. This seamless integration places the responsibility for defining the process stakeholders and the process deliverables within individual organizations implementing the WES process.

Security Visibility

Security should be visible in all steps of the development process if it is to be implemented with any success. This implies that the development process needs to be security focused. The term *security focused* translates into the use of effective and efficient designs, good coding practices, addressing security issues such as authentication and authorization issues, having specific security testing criteria, and acquiring feedback from the end-user that is security specific. This means that the process encourages secure practices such as acquiring specific security requirements, infrastructure re-use, re-usable components, coding standards, coding practices, end-to-end data security, secure designs, and takes into account security policies, procedures, and standards.

Delivery

The goal of any development process should be to deliver a cohesive system, integrating business requirements/ needs, software, and security. This means that the security requirements of the business need to be identified as early as possible in the development process so that they can be incorporated into the design and the construction in order to produce secure software. As Balfanz et al. [2004] noted: "The security community has long argued that security must be designed into systems from the ground up; it can't be 'bolted on' to an existing system at the last minute."

The incorporation of security into the development process should be as seamless as possible. The security that is implemented should meet the needs of the organization so that it adds value to the end product and to the overall business process. The application development must be effective when considering time to market, rapid application deployment needs, introduction of new technology, and efficiency.

Prompt, Rigorous Testing and Evaluation

Testing is critical to the success of an application. Testing should be conducted from a design and programming perspective using both automated and manual scripts; design and code reviews; black, white, and grey box testing; and end-user testing. Testing should also take into consideration as much as is realistically achievable. This could include penetration testing and possibly bringing in external testers (ethical hackers) to validate application security, when the risk is deemed appropriate for such an action. End-user testing translates into the process of being accountable for the security requirements, the environment, and the practicality of the solution from the end-user's perspective. Actual end-users, not surrogate end-users, need to be used in the testing and evaluation of the application. End-users will perform operations, submit data, and interpret instructions in ways that the development team, the business team, or the technical staff within an organization could never dream! This is also true from a security perspective. The end-users of a Web application may be restricted to employees (intranet) or employees and trusted third parties (extranet) or include the general public (Internet).

Observing employees potentially reveals security issues and application problems that could be manipulated into contributing to a security breach. It could be argued that employees are not always forthcoming with information, especially if the lack of security or the potential security vulnerability either does not directly affect their duties or actually helps them to accomplish their assigned tasks. Therefore, a multiple stream approach consisting of involvement in testing, observation, and consultation is recommended when working with systems that will be used by employees. Similar considerations apply to extranet systems that include trusted third-party employees.

There is extensive general guidance on designing usable Web applications [Travis, 2009; U.S. Department of Health and Human Services, 2006] and ensuring accessibility [W3C, 2008], and this applies to the security aspects of an interface. Testing the security aspects of applications designed for use by the general public is much more difficult than intranet applications, as it is necessary to balance usability for the genuine customer, probably with limited computing expertise, and the security requirements to prevent malicious access (hacking). The concept of considering end-users in the security aspect of the application development process is not new. Saltzer and Schroeder [1975] identified eight "useful principles that can guide the design and contribute to an implementation without security flaws." One of these was "Psychological Acceptability: It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanism he must use, mistakes will be minimized." Saltzer and Schroeder's viewpoint is from the perspective of minimizing mistakes through the human interface design, which is a valid point, but it does not specifically address end-user involvement in testing or evaluation.

Balfanz et al. [2004] identified five general lessons for usable, secure systems design. One of these is "Keep your customer happy." They state that the target audience must be involved in testing and evaluating the system and that a small population of subjects will provide useful input. However, they do not discuss how to choose these subjects. This is a major problem for Internet applications. It would be possible to seek existing customers to participate in testing and evaluation, perhaps offering some incentive to take part. However, such a group may not be representative of the general customer base; they will be more committed and probably more comfortable with technology than the majority of end-users. Another possibility is to use a random sample of people, not necessarily business customers, offering a small incentive to take part in a one-off evaluation. Such a sample provides a spread of technical abilities, but few of the subjects, if any, will have any knowledge of the business context. End-user testing could be contracted out to a third party. There are commercial organizations that offer usability testing of Web applications, including security aspects, and will recruit a panel of general users for testing, in addition to providing expert opinion about usability issues.

End-user Feedback

If a development process does not attempt to acquire feedback from the end-users, this could signal potentially large problems with the development process alignment with the needs of the business. Lack of feedback directly impacts the potential effectiveness of a security solution. Hertzum, Jørgensen, and Nørgaard [2004] carried out a usability study on six e-banking systems in Denmark. They identified a major problem with the installation of e-banking software by end-users and suggested that most end-users had no "real understanding of the involved security issues." Therefore, they simply followed a sequence of instructions without being aware of potential pitfalls. For example, three of the e-banking systems did not explain nor enforce the use of strong passwords.

End-users should be observed and consulted for information on the effectiveness of the implemented security solution. Useful feedback can be collected by automatically gathering and analyzing data from support functions (e.g., online technical support or help desk) regarding security problems. This can be used to identify changes required in future releases of the system and provide development staff with a general understanding of end-user capabilities. Balfanz et al. [2004] state, “Support questions provide another window into what users find difficult or unintuitive, and all of this information can feed back into redesign iterations and implementation and interface refinement.” They also point out that collecting this type of data will provide useful input for the design of future systems; it could be treated as ‘Lessons Learned’ within the system development process. Existing research [Balfanz et al., 2004; Saltzer and Schroeder, 1975] coupled with the results of our previous work strengthens the case for an organization to seek end-user feedback from a security perspective.

Implement and Test Disaster Recovery Plans

Security is really a risk management game in today’s society [Viega and McGraw, 2005]. In today’s Web-enabled environment disruptions are measured in minutes, not hours [IBM Global Services, 1999].

The logical progression, once the risk and cost decisions have been made, is to address the need for a disaster recovery plan. There are a multitude of reasons for developing and implementing a disaster recovery plan. These reasons not only include the obvious technical attacks on an organization’s website, as reported by The Open Web Application Security Project (OWASP) [2004], but also natural disasters, terrorist attacks, and information security breaches. Possibilities that have been blatantly exhibited recently include Stuxnet [Fildes, 2010]; Floods in Australia, Sri Lanka, and the Philippines [Kroeger, 2011]; and the Japan earthquake which triggered a Tsunami [BBC, 2011].

These events stress the need for organizations to have and test a disaster recovery plan. If the organization does not have a disaster recovery plan, then it is difficult to develop a cost effective secure design solution for a Web application.

Trust and Accountability

The development process should encourage the development and maintainability of trust and accountability within the organization. Trust and accountability really make up the heart and soul of security.

Trust

Trust can be defined as “Firm reliance on the integrity, ability, or character of a person or thing” [Dictionary.com, 2005]. It is the foundation for a good relationship because it realistically adds value to the communication that takes place in the relationship [Kaplan, 2004]. Hence, Kaplan’s reference to Gerick’s explanation of trust is that “trust is not transitive, distributive, associative, or symmetric except in certain instances that are very narrowly defined” [Kaplan, 2004]. This information is of key importance to understanding the overall concept of trust. Establishing trust is the heart of security, for without trust you cannot rely on the information that is presented. A major component in gaining trust is to manage risk and then implement appropriate controls, educate employees, and monitor effectiveness [Kaplan, 2004]. A tried and true approach to identifying risk is a risk assessment initiative. Hence, trust should be identified in the risk assessment and mitigated in the design to establish and maintain trust. Since nothing is truly risk free, the goal is to mitigate the risk so that it is at an acceptable level. Hence, it is imperative that the development process takes risk into consideration. This is typically done via a risk analysis. The earlier this is completed in the development process, the better.

Accountability

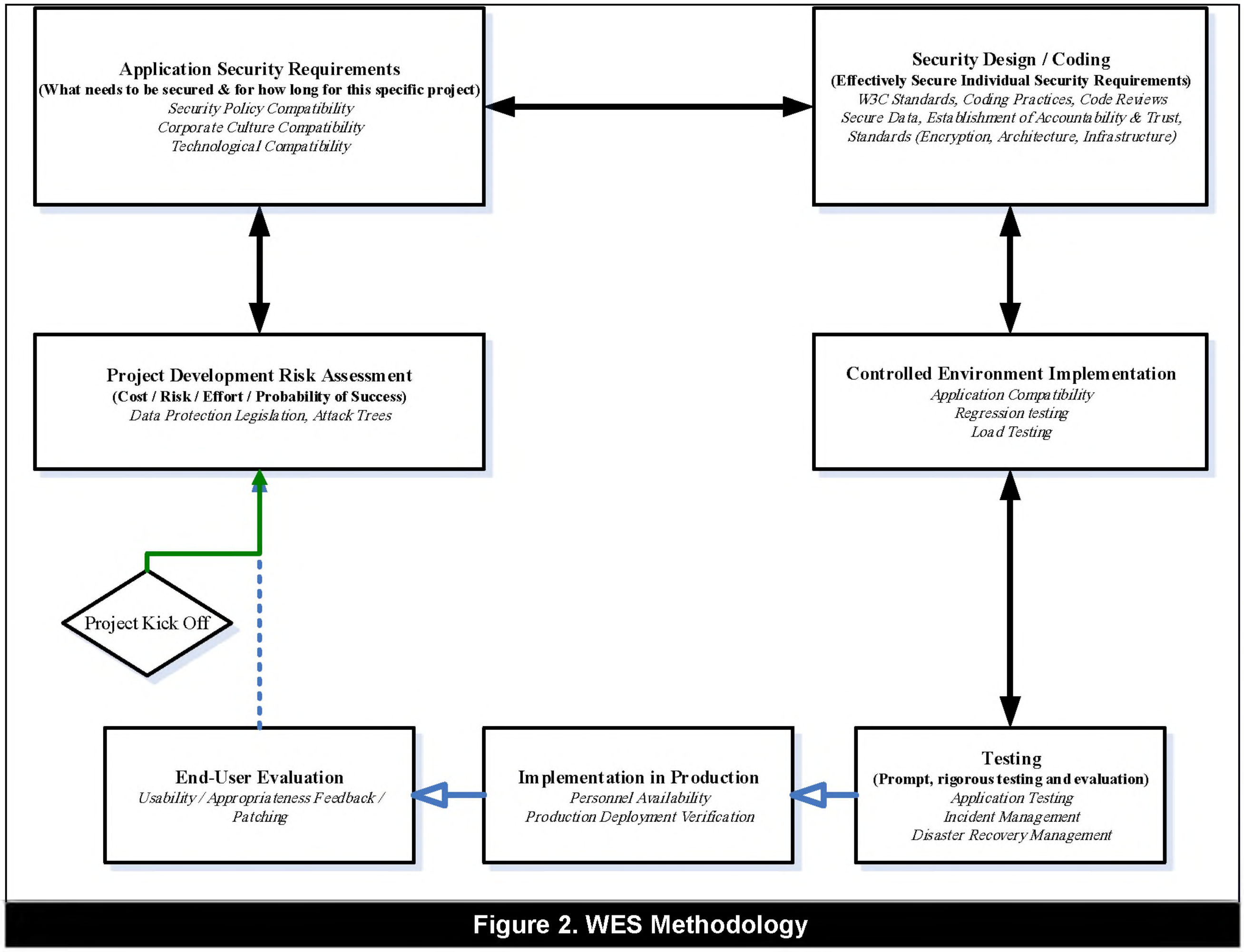
If the heart of security is trust, then the soul is accountability. Without accountability in a system, there is no security, just as, in life, without a soul there is no person. Accountability is critical to the enforcement of security. Individuals have to be successfully identified and authenticated in order to be held accountable for their actions through the use of logs and the effective implementation of access methodologies. The effective establishment of trust and realistic implementation of accountability controls should be visible within the organization’s security policy, the application’s design, coding practices, coding standards, application testing, and project feedback, as a project progresses through the application development lifecycle.

VI. WES LIFECYCLE

The Web Engineering Security (WES) methodology, as shown in Figure 2—WES Methodology, starts with a Project Development Risk Assessment. The Project Development Risk Assessment is the initial phase and it examines the security risk associated with the implementation of a project. The Application Security Requirements phase examines the requirements from the customer’s perspective within the framework of organizational compatibility. Security Design/Coding examines the architecture, the solution design, and the coding practices that are implemented to meet the requirements. A Controlled Environment Implementation scrutinizes the application's interactions with the entire environment before specific aspects of the application are examined.

Testing is critical to the success of many applications. This hypothesis holds true in the area of security as well. Testing not only includes the examination of code but incident management and disaster recovery as well. Implementation of the application in a production environment should take place only after successfully completing testing. End-user evaluation is used to establish the success of the application's security features and for security maintenance.

The WES methodology implicitly supports the concept of separation of duty between everything that happens before testing and everything that happens after testing. This is demonstrated through the color of the line, the line style, and the directional arrows displayed in Figure 2—Wes Methodology. The ideal situation is that the developers and the testers who work on the project are not the same individuals who implement the project into production. Depending on the size of the organization, this may not be possible. Regardless, once code migrates from the test environment to the production environment, it should not be allowed to return to testing without going through another iteration of the process.



Application Security Requirements
(What needs to be secured & for how long for this specific project)
Security Policy Compatibility
Corporate Culture Compatibility
Technological Compatibility
Security Design / Coding
(Effectively Secure Individual Security Requirements)
W3C Standards, Coding Practices, Code Reviews
Secure Data, Establishment of Accountability & Trust,
Standards (Encryption, Architecture, Infrastructure)
Project Development Risk Assessment
(Cost / Risk / Effort / Probability of Success)
Data Protection Legislation, Attack Trees
Controlled Environment Implementation
Application Compatibility
Regression testing
Load Testing
Project Kick Off
End-User Evaluation
Usability / Appropriateness Feedback / Patching
Implementation in Production
Personnel Availability
Production Deployment Verification
Testing
(Prompt, rigorous testing and evaluation)
Application Testing
Incident Management
Disaster Recovery Management

Figure 2. WES Methodology

After Implementation in Production, end-users should be consulted in an attempt to determine the usability of the security solution, suitability of the security solution, and to help identify any security issues that need to be resolved. Once this information has been attained, the process should start the next iteration of the WES development process. Ideally, the iterations in the process should be concise. Succinct iterations encourage smaller, frequent code releases which, by nature, means that less code is introduced into a system at a single point in time. Injecting a smaller quantity of code into an existing system, in theory, denotes that smaller chunks of code are being tested at a single point in time. This potentially allows testers to focus in detail on smaller amounts of code and hopefully improve security test results.

Project Development Risk Assessment

The purpose of the risk assessment is to identify any risk associated with the development of the proposed application functionality. An excellent definition of risk is:

> *… risk is a measure of the loss of what you consider valuable, the impact of losing it, the threats to those assets, and how often those threats could be successful* [Tiller, 2004].

This would include examining appropriate data protection legislation that might apply to your organization's application. There are several tools and suggested practices available in the market for conducting risk analysis. These tools include Cobra [C&A Systems Security Limited, 2012], the Facilitated Risk Analysis Process (FRAP) [Peltier Associates, 2005], and the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) developed at Carnegie Mellon [Alberts, 2003]. The National Institute of Standards and Technology (NIST) publishes recommendations for conducting companywide risk analysis on its website [Peltier, 2000].

If an organization-wide risk analysis is conducted periodically, then the information in the analysis can be used as a starting point for the application risk analysis. The reverse is also true. Information from the individual application analysis can be used as an initial guide to organizational analysis. The risk assessment piece of the methodology can be customized to work in conjunction with an organization's existing risk analysis processes. The basic idea is to:

- Detail critical functions
- Determine the necessary service levels in doing so, identify possible threats, and outline their motivating factors
- Estimate the probability of an attack
- Estimate the probability of a successful attack
- Outline the cost of providing protection [Ellis and Speed, 2001; Pfleeger and Pfleeger, 2003; Phaltankar, 2000]

The answers generated from researching the statements above should help answer the following questions reproduced from Ozier's work: "What could happen? (What is the threat?); How bad could it be? (What is the impact or consequence?); How often might it happen? (What is the frequency?); and How certain are the answers to the first three questions? (What is the degree of confidence?) The key element among these is the issue of uncertainty captured in the fourth question. If there is no uncertainty, there is no 'risk', per se" [Ozier, 2004].

Application threats can cover a wide range of possibilities including human errors in coding, user errors, external attack, fraudulent individuals, technical sabotage, acts of God, and disgruntled employees—all of which should be accounted for in the risk assessment [Ellis and Speed, 2001]. After conducting the risk assessment, the specific application security requirements need to be determined through in-depth conversations with the stakeholders and evaluation of organizational compatibility. Organizational compatibility determines how well security requirements fit into the frame work of an organization. The general areas that make up this category include security policy compatibility, corporate culture compatibility, and technical compatibility.

By conducting a Project Development Risk Assessment, the business and the information technology group can analyze each stage of the development by identifying the associated risks. This would include determining the states of the application and how they can be used or misused as the case may be. This step provides an opportunity for the organization's development team to understand the application from a risk point of view and helps to generate appropriate questions to address the application security requirements phase. Depending on the size of the organization and the market requirements, both the governmental and commercial perspectives, the risk analysis can be used to help identify known risks, point out new risks, and ensure that these risks are acceptable. Depending on the needs of the organization, this can be either a very formal process or a very informal process. If it is a formal process, then the advantage for management is that it presents a clear understanding of the risks before a substantial investment is made in the development of the Web application. The disadvantage of a highly formalized process is that it can slow down the development process. In reality, there will be a lot of cross-over communication between the Project Development Risk Assessment stage and the Application Security Requirements stage. Informal processes tend to be faster but introduce more risk through a potential lack of environmental and risk understanding. The deliverables that could possibly be generated at this stage include a formal project risk analysis document, a risk analysis document used to gather end-user requirements, and a document detailing high-level issues for design and testing.

Application Security Requirements

Specific application security requirements have to be acquired from the stakeholders. The project risk analysis should be used to help gather the security requirements by generating a series of questions and responses that filter the desires of the stakeholders into a list of detailed needs. The Application Security Requirements phase allows the development team to make a specific effort to acquire the security requirements through effective communication with the stakeholders. Hence, the stakeholders involved in this stage would probably include the business unit and the technical staff. They should coordinate these requirements with the organization's security compatibility constraints. The security compatibility constraints encompass several important issues that include security policies, standards, baselines, procedures, guidelines, the corporate culture, and existing technology. For the purposes of this article, the terms listed in Table 1: Terms have been taken directly from *The Security Policy Life Cycle: Functions and Responsibilities* by Patrick D. Howard [2004]. Once these requirements have been captured, they should be examined against the organization's security policy, the corporate culture, and technical compatibility.

Security Policy

"The goal of an information security policy is to maintain the integrity, confidentiality and availability of information resources" [Hare, 2004]. Policies, standards, baselines, procedures, and guidelines can assist in large organizations to provide cohesiveness within the organization. In smaller organizations, where it is not mandatory through regulation, they can be implicit to the organization. The policy provides the "what" and the standards, baselines, procedures, and guidelines provide the "how" [Hansche et al., 2004]. They can work in concert to support the organization from a security perspective. The security policy encompasses all business interactions providing overall guidance to protecting resources [Premkumar and Devanbu, 2000]. This includes acceptable computing practices, all interactions with the network, Internet, messaging, and business specific applications or services [Ellis and Speed, 2001]. Companies may need to meet security policy standards requirements like the ones put out by the International Standards Organization (ISO) [ISO, 2012]. In the context of Web development, the main area of concentration, with regards to the security policy, would be application compatibility within the corporation. However, all areas would need to be addressed to ensure overall compatibility. The security policy should be a living document and updated as new architectures and applications are developed [Symantec, 2005]. If a security policy does not exist at project inception, then the organization may need to investigate the validity of creating the appropriate document.

Table 1: Terms [Howard, 2004] _ENREF_53

Policy:	A broad statement of principle that presents management's position for a defined control area
Standards:	Rules that specify a particular course of action or response to a given situation
Baseline:	A platform-specific security rule that is accepted across the industry as providing the most effective approach to a specific security implementation
Procedures:	Define specifically how policies, standards, baselines and guidelines will be implemented in a given situation. Procedures support policies, standards, and baselines.
Guidelines:	A general statement used to recommend or suggest an approach to implementation of policies, standards, and baselines

From a legal perspective, it is important to recognize that a company's policies, standards, baselines, procedures, and guidelines should be compliant with relevant legal obligations. Cyber-crime is a reality that cannot be ignored in today's global business environment. The ramifications from a financial perspective and a legal perspective are potentially enormous. Web application security needs to be incorporated into the entire development methodology. This includes upfront acknowledgement of the potential legal implications involved with the development and deployment of the Web applications. Effective security resolutions need to acknowledge the legal ramifications that the application introduces to the company and the attendant risks need to be mitigated to the organization's satisfaction. For this reason, a checklist of relevant legislation could be helpful. The purpose behind the checklist is not to introduce a debate over the legislative or the legal enforcement challenges that computer crime presents. Nor is it to discuss the effectiveness of the current legislation or potential conflicts between legislation enacted in different countries. The point is to acknowledge the increasing global legislation that is developing due to the growing impact of the World Wide Web on everyday life, on business economical environments, and on national importance. The legislative list provides a snapshot in time of current relevant legislation. Due to the dynamic nature of legislation, it is understood that the list will continue to change over time as the Web integrates into the global environment. Economies continue to integrate with the Web to produce and/or provide goods and services. Societies continue to increase dependence on the Web to help provide basic operational economical components. This increasing dependency introduces potential national security risks. Therefore, societies are demanding a more secure World Wide Web which leads to the continued creation of new, and the refinement of existing, security legislation. The ripple effects and the overall impact of security legislation on the global economy are unknown.

Corporate Culture

Corporate culture needs to take everything into account, ranging from employee security awareness programs, to employee education on social engineering attacks (discussed below), to recognition of organizational norms. Corporations need to educate the application end-user employees and their development staff in terms of security. They also need to remind employees periodically about security policies, standards, baselines, procedures, and guidelines. One approach to this is to make the issue important to the employee by integrating it into their annual evaluation [Wylder, 2004]. This will not solve all of an organization's security problems; however, it does provide an avenue for encouraging good security practices [Wylder, 2004].

Corporate culture needs to be examined from several different perspectives that include managerial acceptance of the importance of security, the threat of social engineering, employee perception of security and security habits, and technological acceptance of cultural norms. Managerial acceptance and habits, from a cultural standpoint, are critical to the success of security within an organization. Large businesses, looking to strengthen security in their corporate cultures, need to have the highest possible ranking champion promoting the change. In small organizations, the change should be introduced by the owner. If management takes security seriously and encourages a secure environment through their actions, then the odds of this having a positive trickledown effect to employees within the organization are good.

Technology Compatibility

Existing technology needs to be examined from two viewpoints; a compatibility point of view and a value-added point of view. When an application is being proposed, the solution needs to be compatible with the existing infrastructure in the organization. Does the technical expertise exist in the organization to write the application in the proposed language? Does the hardware infrastructure support the new applications? Is the existing code repository compatible with the development of the new application? There are both hard and soft costs associated with these types of questions that need to be taken into consideration when considering any new application development.

Technology needs to be examined from a value added point of view. Whether or not you subscribe to the individual aspects of the "value configuration(s)" [Afuah and Tucci, 2003] which include the value chain, the value shop, and the value network, one of the goals of the organization is to provide added value regardless of the product or service that is being offered [Afuah and Tucci, 2003]. Technology is a major contributor to this goal in today's marketplace. Hence, when examining the validity in developing a new application, the organization should be asking how this will help them add value to their organization.

In general, the area of technological compatibility deals with an organization's existing applications, software compatibility, legacy systems, and the acquisition of new software and technology [Boman, 1997]. When considering the technical compatibility of a system, it is necessary to consider the existing employee skill set within the company. To implement a technical solution, does the necessary skill set exist within the company, can it be acquired easily through employee training, or will it require the company to acquire the necessary skills though outsourcing? To answer these questions, an in-depth analysis will need to be conducted and compared with the solution's requirements. Technological compatibility, from a security standpoint, needs to examine the application to see if it is compatible with existing security solutions already in production. An example would be a new application that is not compatible with the company's existing single sign-on solution. If a solution requires new technologies, the organization should rate the security capabilities of the new technologies and determine if they meet the company's security standards out of the box. If they do not, can they be brought up to speed and at what cost?

This does not mean that these are the only areas that can contribute to this category or that they all have to be present within this section to ensure compatibility. There are environments that may choose not to implement a security policy or to investigate corporate culture due to the size of the company. For instance, a large financial institution will probably have all three categories (security policy compatibility, corporate culture compatibility, and technical compatibility) documented to some extent. However, a small family-run business, like a local restaurant, probably will not have a security policy and the culture in that business will be implicit. However, more than likely, they will have technical compatibility issues that they will need to address.

Security and the Human Element

Technical solutions alone will not provide protection against the human element. They will not provide protection against an end-user who reveals his/her passwords, users who circumvent security to complete a specific task, or insider attacks [Ellis and Speed, 2001]. When it comes to information security "the human factor is truly security's weakest link" [Mitnick, 2002]. This fact spawned an area of warfare in the business world known as *social engineering.*

Social engineering attacks take place when an outsider or insider observes an organization, gathers information, and makes necessary business contacts under the premise of a legitimate purpose in order to gather information [Mitnick, 2002]. This information is then used to acquire more information until the intruder acquires something of value [Mitnick, 2002]. The same tactics can be used by a current employee to gain unauthorized privileges. Company employees need to be educated on the existence of social engineering attacks and how to identify and prevent these attacks from occurring [Mitnick, 2002].

The perception of security, and its importance to the business, needs to be effectively communicated at an employee level. If the employees do not place a great deal of importance on security and they regularly post passwords on screens or in accessible areas, trade passwords with colleagues, or grant system access to outside vendors, then they are creating a security risk for the company.

Technological acceptance of corporate norms occurs when an implemented solution becomes accepted and then becomes expected. The implementation of single sign-on is an easy example. The implementation of a single sign-on solution for several existing applications could reasonably influence employee expectations for future applications. The justification for going against expectations needs to be examined and justified to the employees. Otherwise, employees could start to circumvent security when it suits their needs.

Security Design/Coding

Once the application security requirements have been determined, the next issue that needs to be addressed is security design. The design of the application needs to consider the overall architecture, the application design, and good design principles.

This information then allows the technical architect, in the Security Design/Coding phase, to pick the most appropriate technical controls from a design, risk, and cost perspective. Once the high-level design decisions have been made, then the coding takes place. The programmers should take into consideration coding standards, good coding practices, code reviews, and appropriate security measures. Encouraging programmers to adhere to coding standards and to pursue good coding practices will increase the code readability which will inherently improve software maintenance. This improvement should be felt in both enhancement maintenance and patch maintenance. Estimations indicate that maintenance accounts for an average of 60 percent of an application's software expense [Glass, 2003]. In reality, "better software engineering development leads to more maintenance, not less" [Glass, 2003]. If an application meets the needs of a particular market, then the application will be enhanced through the addition of new features and improved functionality. It should be noted that this is considered new development in a lot of organizations. Patch maintenance is another area that is critical to defending against cyber vulnerabilities [Dacey, 2003]. Any improvement in an organization's software maintenance capabilities translates into long-term savings.

Code reviews ensure that the code is doing what it is supposed to do, decrease errors in the code and ensure that more than one person understands the application. The implementation of the type of code review is up to the individual organization. Code reviews can encompass everything from pair programming, to design reviews, to manual reviews of written code. It is up to the organization to decide the best avenue for implementation so that the organization is not dependent on a single employee for modifications and support for a specific application. Applying appropriate security measures will help ensure data security and security consistency throughout the application.

The architecture needs to fit into the existing organizational environment. There are several issues that need to be addressed within the realm of architecture. Some of those issues are:

- Application layers [Fernandez, 1999]
- Application maintainability [Graff and van Wyk, 2003]
- Information compatibility from a data transfer standpoint
- How strongly typed the language needs to be [Lipner, 2004]
- Approach to privileges, i.e., role-based or inheritance
- The approach to default privileges from the application and the user's standpoint [Pfleeger and Pfleeger, 2003]
- Security in-depth-use passwords and another mechanism, such as an encrypted key of some sort, for determining object access [Pfleeger and Pfleeger, 2003].

The design of the application needs to address:

- The language that will be used [Lipner, 2004]
- Ease of use—the easier security solutions are to use, the less likely that they will be circumvented [Pfleeger and Pfleeger, 2003]
- Authorization techniques
- The use of encryption algorithms
- The establishment of trust
- The establishment of accountability

It should be noted that the establishment of trust should link back to the project risk assessment. The amount of trust that is designed into an application is directly related to the amount of risk that an organization is willing to tolerate and the total cost that they are willing to absorb. Accountability, through the implementation of appropriate mechanisms, is an essential ingredient to security.

The design needs to examine the code from common attack standpoints and implement the appropriate controls to ensure secure data. A professional code-management system should be used by the development team to ensure accountability, within the team, and provide a means of roll back [Foster, 2004]. After design selection, the solution is coded. During coding, the developer should be cognizant of the World Wide Web Consortium (W3C) coding standards and pursue secure coding practices [W3C, 2005]. One idea that a designer should keep in mind, when designing a secure solution, is to balance the need for a secure application with the need for a particular functionality.

Another idea that a designer should strive to attain is the creation of simple design solutions that solve specific problems and fit into the applications global architecture. The design will depend on the level of security the customer is willing to accept from a risk and/or cost standpoint.

Controlled Environment Implementation

Depending on the needs of the organization, the Controlled Environment Implementation can be as complex as implementing it into an environment that mirrors the production environment, or it can be as simple as running the application on a desktop. In the latter case, both the desktop application installation and the server mirroring environment can be used to test the security controls. The point here is to release the code in a secure environment that simulates the production environment for compatibility testing before the application is made available to the general public. The goal of the environment is to minimize surprises. Basically, this phase allows the developers to test the application's compatibility with the operating system and interfacing programs before application testing and a production release.

The controlled environment implementation should also take into consideration application compatibility, load testing and regression testing. The new application compatibility with the native operating system and other pre-existing applications is important. Compatibility also needs to be verified with applications on the same server and applications that live offsite (internal to the organization or external to the organization) where data is being exchanged.

Security Testing

Testing takes place from both the developer and the end-user perspective. Developers should be running their own battery of tests when the code is conceived. Again, it should be stressed that the methodology is designed to work in conjunction with existing organizational tools and processes. If the organization already utilizes automated testing tools, they should be used in this stage to augment the testing process.

Actual end-users should be incorporated into the testing campaign whenever possible. The stakeholders should be writing test scripts and actively interfacing with the application to ensure that the program is performing accordingly. End-users' participation in the security testing of the Web application holds the process and the solution accountable from a practicality perspective.

The National Institute of Standards and Technology (NIST) estimates that "93% of reported vulnerabilities are software vulnerabilities" [Ounce Labs, 2004]. The Organization for Internet Safety (OIS) publishes Guidelines for Security Vulnerabilities Reporting and Response. In this document, they define a security vulnerability as:

> *… a flaw within a software system that can cause it to work contrary to its documented design and could be exploited to cause the system to violate its documented security policy* [Organization for Internet Safety, 2004].

Hence, any flaws in the system design or application coding can potentially lead to security vulnerabilities. The Open Web Application Security Project (OWASP) provides an excellent listing of the top ten vulnerabilities in Web Applications. The top ten vulnerabilities are taken directly from the OWASP report and include Un-validated Input, Broken Access Control, Broken Authentication and Session Management, Cross Site Scripting (XSS) Flaws, Buffer Overflows, Injection Flaws, Improper Error Handling, Insecure Storage, Denial of Service, and Insecure Configuration Management [The Open Web Application Security Project, 2004].

This list complements information discussed in the previous articles, the “Top Web application security problems identified” and “The Bugs Stop Here” which were published in 2003 [Berinato, 2003; Mimoso, 2003]. Only when testing requirements are satisfied and the vulnerabilities have been addressed to the satisfaction of the organization should the application be moved into production.

Developers need to examine their code independently and the program as a complete entity in order to determine possible misuse from a functional standpoint. That is, programs should do precisely what they are designed to accomplish.

> *Vulnerabilities can stem from the rapidly evolving use of software, in which programs meant for a limited purpose are applied in ways not anticipated by their developers* [Williams, Shimeall, and Dunlevy, 2002].

Thus, can a program be manipulated in a manner that might create problems, and can this be stopped or mitigated? A primary example is an email server that is used to propagate a virus or used in a denial of service attack.

Testing is critical to the success of any application. Testing should cover application testing, incident management, and disaster recovery plans. Application testing includes validation errors, program behavior testing, and code analysis. This will involve implementing appropriate programs to test static and runtime code, penetration, and application scanning. Automation, where possible, of the testing process will help provide stability. Testing should also involve executing scripts from both the developer and the end-users to test the application. An important part of the testing phase should be to decide appropriate action plans for incidences. When there is an issue, what are the procedures that need to be implemented to resolve the situation? This should also include amending the disaster recovery plan where appropriate. If the organization does not have a disaster recovery plan, then, they should investigate the creation of a plan. The disaster recovery plan on the organizational level should be a living document. The disaster recovery plan for the application should be flexible enough to allow for the addition of a new functionality. The disaster recovery plan should be tested after the initial creation and each amendment. Testing is where everything should come together in the development process. Hence, testing should:

- Contain a requirements check against the application to ensure initial and continued fulfilment.
- Ensure sufficient mitigation of threats identified in the risk analysis.
- Be as prompt as is reasonably possible so that an organization is competitive in the Web application development market.
- Involve actual end-users, not surrogate end-users.
- Be as comprehensive as possible. This will be determined based on the amount of risk the application presents to the organization’s reputation and the organization’s core business.
- Be tailored for security.
- Take advantage of an organization’s existing testing infrastructure.
- Should include external testing to verify application security where the risk warrants the expense.
- Implement a matrix to measure the success of the testing and effectively track bugs.

Implementation into Production

After testing is complete, then and only then is the application prepared for Implementation into Production. The introduction of the application into the production environment needs to be completed with the involvement of the appropriate security personnel. Appropriate personnel need to be present to ensure proper application deployment into the production environment. If possible, this allows for immediate issues resolution at the time of implementation. If issues are discovered after the application deployment into production, then the application must go through the process again and be re-implemented into production.

End-user Feedback

End-user Evaluation is critical from the standpoint of security. Whenever it is possible, actual end-users should be used in the security evaluation of a Web application. End-users are the ultimate variable in the execution of an application. If end-users are circumventing the application's security in order to make their lives easier or perform their jobs in a timely manner, then these issues need to be investigated and resolved [Glisson and Welland, 2005].

An efficient and effective response to application security breaches is mandatory to Web-based business survival. If the application is compromised due to a flaw in the design or the code, then the security issue needs to be addressed, realistically, as rapidly as possible. If the application is not secure, businesses run the possibility that the application will be abused, corporate credibility lost, and financial consequences incurred.

End-user Evaluation involves both communicating with the user to determine the success of the application's security and security maintenance. This can range from informal communication, to surveys, to structured interviews with the end-user. Security needs to find a balance between usability and providing a secure environment.

Security maintenance focuses on discovering vulnerabilities after a production release. As new technologies emerge from the viewpoint of development and maintenance, new vulnerabilities will be created, uncovered, and these issues will have to be addressed to maintain application security [Lipner, 2004; Pfleeger and Pfleeger, 2003]. Patches will need to be tested to ensure that they resolved the newly discovered issue and to ensure that they do not create new security vulnerabilities in the application.

VII. WES STAKEHOLDERS AND DELIVERABLES

The stakeholders who are involved in a specific project obviously depend on several criteria ranging from resources to project visibility, to project risk, to funding. A large organization is more likely to have the resources available to assign different people from different areas to the project. A small company, on the other hand, may have employees conducting multiple job functions.

Project visibility is a factor when considering the amount of resources that will be assigned to a specific project. High profile projects affect many people within and/or outside of the organization. Hence, they will probably receive more attention than low-profile projects.

The risk associated with the profile is another matter. A project that is high profile and high risk to the core business function of the organization will potentially be assigned more resources. An example of this is a website that conducts financial transactions for a banking institution. On the other hand, a high-profile and a low-risk project, such as an intranet phone book application, will potentially be allocated fewer resources.

As always, funding is an issue with all projects. If the funding is not available, regardless of the size of the organization, then resources will simply not be assigned to the project. If funding is available, at a smaller amount than initially requested, then corners are cut to reduce expenditures. Easy targets for reducing expenditures include security testing, ongoing end-user input and feedback, and developer security education and training, but this is potentially a dangerous strategy.

General stakeholders who would be expected to be involved in the development process would include the project sponsor, project manager, business analysts, architect, programmer, tester, risk and security personnel, release personnel, and the end-user.

After methodology customization satisfies the needs of a specific business, it can then be documented for future project replication. Depending on the needs of the organization, this can serve also as an audit trail. The amount of documentation implemented will depend on the needs of the particular organization. For example, a financial institution, due to regulations, will be required to provide detailed documentation of their processes. In contrast, a small local business will probably document only the bare necessities in order to conduct business.

The deliverables that are required, during each stage of the Web Engineering Security development process, depends on the culture of the organization and the development methodology being implemented. A highly regulated industry, such as banking or insurance, will require a greater emphasis on the individual deliverables at each stage of the process. However, the converse is also true; an organization not in a highly regulated business will have fewer deliverables that are required during the various stages of the process.

The application development methodology, that the organization is utilizing for Web development, is usually linked, as well, to the culture and the industry to which the business belongs. An organization that uses a waterfall approach will be more inclined to generate documentation and specific deliverables between the various stages. However, an organization that is implementing an agile approach to application development will, by nature, produce fewer deliverables between the various stages. Understanding the previously mentioned issues, the decision as to whether to create deliverables and to what extent the deliverables are created is left to the organization to determine.

VIII. WES GOALS

The WES goals include upfront integration of security, security comprehensiveness, structured security implementation, and industrial practicality.

Upfront Integration of Security

The WES methodology strives to integrate security from the beginning of the application development process. This is why security discussions are initiated during the business analysis stage of the development process. This up-front integration should help the organization reap benefits ranging from faster application development to positive effects on budgets and time frames by proposing realistic security solutions at the onset of the project. The idea is to move security from the typical viewpoint of an inhibitor to that of an enabler in the eyes of the end-user.

Granted, this move is, to some extent, dependent on the security team that is involved in assisting in the implementation of the WES methodology. They need to not only be defining what is possible in the current organization but to also provide an architectural strategy for the future while providing realistic alternatives to business needs rather than stating that something is not possible, full stop.

Security Comprehensiveness

The WES methodology hopes to address the questions of “How do I build application security into the fabric of my company?” [deJong, 2006]. The solution is to approach the problem from the idea of presenting a proactive comprehensive approach to the security development process. The security methodology should be compatible with the existing application deployment process capitalizing on current core competencies while providing a roadmap for improving security during the application development process.

Structured Security Implementation

The WES methodology provides an overall structure that allows organizations to customize the level of security to its individual needs and implement security into their application development process. This structure can then be hardened to provide an organization with the desired level of continuity, reusability, and audit-ability for future development projects.

Industrial Practicality

The general categories in the WES methodology are not set in stone but are strongly recommended. The items within the categories will need to be tailored and, where necessary, expanded to meet the specific needs of the individual organization and their current policies and procedures. The methodology is designed to complement an organization’s current methodology, while providing guidance to the development process from a security perspective.

The WES methodology provides a roadmap for Web application development that will help guide organizations to a more secure system. The goal is to proactively help developers create applications that are secure by design. Following the WES methodology means that the development process takes into account risk analysis, application security requirements, various organizational policies, organization architecture, code design and coding practices, proper testing procedures, and end-user feedback.

WES provides a practical method by which to address security. There are several solutions in existence that tell you “what” to do to improve general security within an organization and some within the organization’s development process. There are currently a multitude of technical solutions that offer possible solutions to very specific questions which basically answer “how” to solve specific security problems. The technical contribution is growing rapidly daily.

Prior to the WES methodology, nobody designed a security process based on criteria that are specifically tailored to address the needs of a Web Engineering development process. The general solutions that have been proposed in the past tend to lack accurate details that address the practical issue of “where” actions should be performed in the software development process. WES provides the Web Engineering community with a practical methodology to solve the inherent security deficiencies present within generic Web development lifecycles.

IX. WES PRACTICAL APPLICATION

WES was implemented in a Fortune 500 financial organization as a Security Improvement Initiative (SII) project which concentrated on the initiation and design phases of the software development project lifecycle. This organization uses a customized plan driven document centric waterfall approach when conducting Web application development and all other forms of software-based initiatives. Within this process approach the business comes up with an idea and develops a business case to support the project. Once the business case is accepted, then, a project manager is assigned to the project.

The project manager contacts the necessary personnel to have resources assigned to the project. In general, these individuals include the architect and, possibly, a project risk analyst. The architect is responsible for completing a Design Architecture Document (DAD) and presenting it to the Design Architecture Committee (DAC). There are eight voting members on this committee; all have veto authority. If any of the members on the committee vetoes the project, the design is rejected and has to be resubmitted with identified committee objections addressed. It should be noted that all the members had established their seats on the board months before the SII was initiated.

Based on member voting, there are three possible outcomes when a DAD is submitted to the DAC. First, the DAD could be accepted by the committee. Second, the DAD could be accepted by the committee with conditions. Third, the DAD could be rejected. Once the design is approved, the coding teams produce a Detail Design Document (DDD) based on the DAD. This design is then built, tested, and implemented into the production environment under the governance of the architect. All voting members have the right to assign conditions within their area of expertise. If a DAD is accepted with conditions, these conditions must be satisfied prior to progression into the next stage, which in this case would be the build stage. An interesting gage to examine the effects of security on the overall development process is the quarterly analysis of the assigned security conditions.

The organization has customized the individual phases within this approach by subdividing them into stages. The application of the WES developmental methodology up to DAC approval is shown in Table 2: WES Implementation. This table reveals how the process should operate in the first two project phases of the application development lifecycle, the associated generic project stages, and the phases of the WES methodology. The section of the table titled ‘WES Applied Project Stages’ specifically details the integration of the WES methodology with the company’s generic project stages. The application of the WES methodology is conducted in conjunction with the knowledge derived from the survey as summarized in Appendix 1.

The application of WES reveals the opportunity to implement multiple changes to the development process. The group most receptive to the idea of changes to the development process was the architecture group. The Design Architecture Document (DAD) is the primary instrument utilized by the architecture team in the organization. Hence, the logical place to implement changes is the DAD.

It should be noted that the WES methodology does not mandate deliverables from the individual areas within the methodology. The methodology lets the organization determine what is appropriate, based on the size of the organization, the application development methodology that is being utilized, and the corporate culture. In this particular case study, the organization in question is already documentation heavy. Hence, the realistic approach is to expand the current documentation so that it incorporates the new security functionality. The organization already produces a Business Case Document (BCD), Design Architecture Document (DAD), a Detail Design Document (DDD), and Testing Documentation (TD).

Table 2: WES Implementation

Project Phases	Generic Project Stages	WES Stages
Initiate & Assess	Idea	**Project Development Risk Assessment** *(Cost/Risk/Effort/Probability of Success)* Data Protection Legislation, Attack Trees Risk Analysis Techniques
	Concept Development	
	Business Case (BCD)	
Design	Business Requirements	**Application Security Requirements** *(What needs to be secured and for how long for this specific project?)* *Organizational Compatibility* Security Policy Compatibility Corporate Culture Compatibility Technological Compatibility
	Initial Design (PDAD)	**Security Design** *(Effectively Secure Individual Security Requirements)* Satisfactorily address risk identified in risk assessment and application security requirements Verify security requirement compliancy with organizational compatibility Establish intended use of W3C Standards, Coding Practices Describe the Establishment of Secure Data, Establishment of Accountability and Trust State the use of Standards (Encryption, Architecture, Infrastructure) Security verification of project viability
	Initial Technical Evaluation (DAC)	
	Design Architecture Document (DAD)	
	Technical Evaluation (DAC)	
Build	Construction (DDD)	**Security Coding** *(Effectively Secure Individual Security Requirements)* Implement W3C Standards, Coding Practices, Code Reviews Secure Data, Establishment of Accountability and Trust, Utilization of Reusable Components
	Testing	**Controlled Environment Implementation** Application Environmental Compatibility Regression Testing Load Testing
		Testing *(Prompt, Rigorous, Security Testing and Evaluation)* Application Testing Verification of risk and requirements satisfaction Incident Management Disaster Recovery Management
Implement	Implementation	**Deployment in Production** Personnel Availability Production Deployment Verification
Feedback	Feedback	**End-user Feedback** Usability Feedback Appropriateness Feedback Patching

Project Development and Risk Analysis

During concept development, the project risk analyst should be conducting a risk analysis. The Project Risk Analyst should also be speaking simultaneously with the architect and the appropriate coding teams in order to determine the project risk and help the business unit develop the project's business case. The results of the survey support the need for early interaction between the business unit and the technical side of the organization. An issue that should be addressed during the risk analysis is the risk compatibility. The application of the WES methodology indicates that the risk analysis should be determining critical functionality within the application, determining appropriate service levels, identifying all possible threats, the probability of attack, the probability of success, and the cost associated with the desired level of protection [Phaltankar, 2000].

Application Security Requirements

When the business requirements are being gathered, members of the business unit should be interacting with the project manager, the architect, a project risk analyst, and members from appropriate coding teams. The idea behind this interaction is to gather a fairly comprehensive listing of the security requirements. Specific security requirements explicitly recognize all the security requirements from the business unit so that they can be addressed successfully. The security requirements should identify specific environmental requirements along with addressing identity, authentication, and authorization. Once the security requirements are gathered, they should be examined from a critical perspective in order to determine how they will comply with the organization's security policy, corporate culture and technology compatibility.

Security Design

Once the security requirements have been ascertained and they have been examined in reference to the security policy, corporate culture, and technology compatibility, the design should take place with this information in mind. The proposed design improvements concentrated on the architecture team's main instrument for creating solutions, which is the Design Architecture Document (DAD). The following changes were proposed to the design process, and some of them are reflected in the DAD.

1. Owner/Creator Contact Information
2. Conversation Checklist for Security Requirements Gathering
3. Signature Section
4. Risk Compatibility Section
5. Identity Management
6. Threat Management
7. Trust Model
8. DAD Socialization

Owner/Creator Information

The idea behind capturing the owner-creator information, the conversion checklist for security requirements gathering, and the signature section, is not only to expedite communication but to assign accountability. Regardless of the existence of questions around various topics in the document, it is necessary to assign ownership of the proposed architecture solution. The owner/creator information tells anyone, who picks up the documented solution, who created the solution's architecture.

Conversation Checklist for Security Requirements Gathering

The conversation checklist, for security requirements gathering, helps aid the project manager to ensure that all of the necessary parties are involved in the creation of the DAD and in the overall project. This, realistically, should be the job of the project manager. However, the survey and observation alludes to the fact that the skill level among the project managers in the organization varies widely. The goal of having the conversation checklist for the security requirements is that there is increased communication with the project members encouraging a higher level of security awareness.

Signature Section

The proposed signature section consisted of three names: the project manager, the individual in the business unit who was responsible for signing off on the business requirements, and the individual on the DAC who is responsible for matching the design to the actual product prior to going into production. (The present system lacked this crucial functionality). The project managers should be included in the DAD since the document is being created at their request. The inclusion of the project manager in this process will help educate management on the design process and help foster solution buy-in; including the name and the contact information for the individual who is responsible for providing approved user requirements helps to encourage communication when there are questions and to assign responsibility. The DAC signature should list the individual (and his/her contact information) who has agreed to follow up on the proposed application to verify that the application being delivered is the same as the application that was proposed in the design submitted to the DAC. This signature provides member and application accountability to the DAC.

Risk Compatibility Section

The idea behind the Risk Compatibility section stage is to ensure that the security design proposed by the architect is compatible with the Risk team's policy requirements. A Risk Compatibility section examines tier trust policy compatibility, proposed low level security practices, and data security. The trust model is the foundation for the policies that are put into place in an organization. Hence, applications can be applied to the policies or the trust model in order to determine compliancy. The reality is that both should be checked for each solution so that there is a system of checks and balances. This insures that the solution is compatible with the organization trust model and policies, while verifying that there are no discrepancies between the policies and the trust model.

The trust model compliance can be examined from two perspectives. The first is the network architecture perspective and the second is the application architecture perspective. The architect would need to learn the organization's network architecture trust model and the application trust model. An assessment of an organization's trust models naturally leads to a discussion about new application integration into both architecture models, making sure to identify any violations to the model and the acquisition of appropriate exception authorizations.

Examples of the issues that the application should address would include:

- Assurance that the application does not violate the Internet network trust model
- That the application uses the organization's Identity Management (IM) system, and, if not, explain why not, and acquires the necessary exception
- How does the application establish direct trust?
- How does the application maintain trust?
- Does the application implement proper encryption policies? Do these policies comply with the Internet network trust model?

Another point that surfaced with the application of the WES methodology is an overview of the application and its impact on the organization's low-level security practices. The idea behind identifying the proposed low-level security practices is to ensure that the application's low-level practices are compatible with established security policies. If they are not compatible, they have to be acknowledged appropriately. Example: An application that has to have access to the kernel level of a UNIX box would need to be acknowledged via an appropriate risk analysis. If the risk is deemed a necessary risk, the appropriate exceptions would have to be sought and granted within the organization. The thought behind data security is that the organization needs to identify any sensitive data held within the proposed system design. The organization also needs to document that this data is being protected by successfully addressing appropriate risk encryption policies, transaction policies, and storage policies.

Identity Management

Identity is a key factor in establishing and maintaining the security of a system. The physical world places multiple meanings on the term *identity* depending on the context to which it is applied [Mont, Bramhall, Gittler, Pato, et al., 2002]. These meanings include everything from names to addresses, to financial information, to citizenship [Mont et al., 2002]. "'Digital identity' is, at the core, an effort to recreate, organise, automate and integrate all those aspects in the online electronic world and (increasingly) link them to existing 'offline' identities" [Mont et al., 2002]. Hence, Identity Management (IM) has the potential to impact business processes, policies, and the organization's technology in order to attempt to provide access and user control to Web applications. "In this context, identity management is also a key e-business enabler: being able to recognize the digital identity of people and Web services, to understand, manage and validate their profiles and rights is fundamental in order to underpin accountability in business relationships and enable commercial transactions" [Mont et al., 2002]. As far as the organization is concerned, IM should be viewed as a re-useable component within the organization. Under the IM heading, architects should be addressing issues such as role-based access and controls, authentication and authorization, user provisioning, access and control to environments, as well as audit and archive design.

Threat Management

Threat Management attempts to identify all the known threats to the proposed solution and how these threats are being mitigated. This solution should also take into consideration interaction with existing software like host-based intrusion detection systems, network-based intrusion-detection systems, firewalls, and antivirus software. Another area that needs attention is the use of any compliance tools that are being utilized by the organization and the solution's compatibility with the current configuration of the organization.

Trust Model

Trust is critical when establishing security. The architect should describe how trust will be established and maintained between the various application tiers. Another issue that needs to be addressed is how deep a user's identification (id) can be traced within the application. This helps the organization identify a level of risk that it is willing to live with when it comes to identifying the actions of a user.

DAD Socialization

The organization has an interesting environment where the architect is supposed to formalize a solution with all of the members of the group. To formalize a solution with the members of the Design Architecture Committee (DAC) means that the architect meets with each member individually to discuss the proposed solution. This gives the architect and the group member the opportunity to work out any issues prior to the formal DAC meeting. However, observation indicated that this was not taking place effectively. The proposed solution to the problem is built on the idea that improved communication improves overall security. Hence, a mandatory socialization table that included the names and titles of all of the voting members of the DAC was implemented in the Design Architecture Document (DAD).

Implementation Results

As a general indicator of the effectiveness of the Security Improvement Initiative (SII), all of the conditions assigned to new and existing projects during the SII were captured for analysis. The numbers used in the observations are simply very general indicators as to the impact of the SII and nothing more. When the observations first started the security conditions appeared to be on the rise and then they appeared to decrease for the remainder of the study. The security conditions data are available in Table 3: Security Conditions assigned by DAC to Projects.

Table 3: Security Conditions Assigned by DAC to Projects

Quarter	Security Conditions	Total Conditions	Projects	Ratio Security Conditions to Total Conditions	Average Security Condition per Project
1st	12	24	12	.5	1
2nd	35	78	17	.45	2.05
3rd	29	77	14	.38	2.07
4th	24	69	17	.35	1.41
5th	18	48	12	.38	1.5

The SII appears to have had a positive effect on the organization by providing a decreasing trend in quarterly security conditions assigned to projects. The decreasing security conditions helps to improve overall development times by decreasing issues that have to be resolved and reducing tensions between deployment staff and security personnel. However, it cannot be claimed that applying WES is the sole cause for the change in the trend. This is due to the fact that it is impossible to totally lock down a business environment where multiple groups interact, quantify the impact of having a researcher investigating security in a corporate environment, and link results to a single action.

Further Implementation

Due to the focus of the SII, the WES methodology was not implemented to the balance of the application development lifecycle. However, it was applied to highlight areas where the organization could focus future security improvement initiative projects. Hence, the application of the WES project indicated that the build phase should verify that they are addressing the following issues in the area of secure product development: Authentication Development, Authorization Development, Coding Standards, Proper Data Encryption, and the Assignment of architects to consult with individual engineering rooms. In the area of Controlled environment testing, it indicated the verification of the following: Compliance Testing (CT)—Environment Testing, Assurance Testing (AT)—Certificate Testing, Regression Testing, and Load Testing. In the Testing part of the build phase it indicated that the following should be verified: Test Scalability/Fail over, Third Party Penetration Testing, Testing Criteria from Solutions Design, Testing Scripts from Project Risk, Incident Management, and a Disaster Recovery Plan. The application of the WES methodology also suggested that there should be a security sign-off prior to application implementation into production. The methodology also suggested that end-user feedback of the security functionality should be sought in order to determine security effectiveness and to identify security bugs.

X. SUMMARY

A real world understanding of application security indicates that it is a multifaceted issue in an increasingly complex environment. This becomes especially apparent when examining Web-facing applications. The need to address security in application development increased over the past several years. However, one of the major challenges facing organizations in today's Web-enabled environment is balancing technological needs with the business needs of the organization. Another potential challenge for organizations is structuring the overall development process so that there is not a general frustration within the organization in terms of overall process efficiency. A lack of process efficiency potentially hinders aggressive Web development from a business perspective. A lack of security integration and understanding of the application development process creates an environment that is conducive to fostering security deficiencies.

WES is a proactive approach that is designed to operate at a high level of abstraction. There are advantages and disadvantages to a high level abstract solution. The advantage that a high-level of abstraction provides is inclusiveness to the overall process. A high-level process is naturally conducive to security issues, business issues, software development issues, and organizational issues being more inclusive. If these issues are narrowed through too much detail, there is the possibility that the details will be biased in some way or that they will simply miss an important issue. The disadvantage of an abstract approach to a security methodology is that the implementation of the process is demanding from an individual knowledge perspective. WES is constructed from empirical research that consisted of two surveys and relevant literature. Empirical research bases the WES methodology in reality, in that, the goal of the WES methodology is to strengthen security in Web development applications. The principles of security education, good communication, and cultural support provide the foundation for the WES methodology, which aims to create an environment that is conducive to initially fostering and continually encouraging security in an organization's application development environment. Due to resource constraints, WES was only partially implemented in a large organization. However, it is desirable to complete the WES implementation with the existing partner. It would also be desirable to implement WES in other organizations for comparison.

Future research in the area of secure application development is currently examining the development of cloud computing applications. The security process in a Fortune 500 organization is currently being examined to determine if it is appropriate for cloud computing applications. It is also being scrutinized from the perspective of cloud computing forensics. This research seeks to investigate the security controls in cloud computing development lifecycles and determine if they are appropriate for a cloud environment. It also investigates the effectiveness of existing tools to acquire forensically sound data in cloud environments. Future work is also underway to examine security from a large organizational pattern perspective. The idea is to develop security patterns for large businesses that can be applied at the organizational level.

ACKNOWLEDGMENTS

The authors thank the Fortune 500 Organization for its support and collaboration throughout the project.

REFERENCES

Editor's Note: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the article on the Web, can gain direct access to these linked references. Readers are warned, however, that:

1. These links existed as of the date of publication but are not guaranteed to be working thereafter.
2. The contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. The author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.
4. The author(s) of this article, not AIS, is (are) responsible for the accuracy of the URL and version information.

Afuah, A., and C.L. Tucci (2003) *Internet Business Models and Strategies, second edition,* International Edition edition, Boston, MA: MacGraw-Hill.

Agile Alliance Organization (2007) "Agile Alliance", Agile Alliance, http://www.agilealliance.org/ (current May 21, 2007).

Ahmad Al-Omari, O.E.-G., and A, Deokar (2012) "Information Security Policy Compliance: The Role of Information Security Awareness", *Americas Conference on Information Systems,* Seattle, Washington: AMCIS 2012 Proceedings.

Alberts, C. (2003) *Introduction to the OCTAVE® Approach,* Dorofee, J.S.A., and C. Woody (eds.), Pittsburgh, PA: Carnegie Mellon Software Engineering Institute, pp. 1–37.

Ayoub, R. (2011) *The 2011 (ISC)2 Global Information Security Workforce Study,* New York: Frost & Sullivan.

Balfanz, D., G. Durfee, D.K. Smetters, and R.E. Grinter (2004) "In Search of Usable Security: Five Lessons from the Field", *Security & Privacy Magazine,* IEEE, (2)5, pp. 19–24.

Baskerville, R.L. (1989) “Logical Controls Specification: An Approach to Information Systems Security”, *Systems Development for Human Progress*, pp. 241–255.

Baskerville, R. (1992) “The Developmental Duality of Information Systems Security”, *Journal of Management Systems,* (4)1, pp. 1–12.

Baskerville, R. (1993) “Information Systems Security Design Methods: Implications for Information Systems Development”, *ACM Computing Surveys,* (25)4, pp. 375–414.

Baskerville, R., B. Ramesh, L. Levine, J. Pries-Heje, et al. (2003) “Is Internet-Speed Software Development Different?”, *IEEE Software,* (20)6, pp. 70–77.

BBC (2011) “Japan Earthquake: Tsunami Hits North-east”, BBC, http://www.bbc.co.uk/news/world-asia-pacific-12709598 (current October 28, 2011).

Beck, K. (2001) “Manifesto for Agile Software Development, The Agile Alliance”, http://www.agilealliance.org/ (January 25, 2014).

Beck, K., and C. Andres (2004) Extreme *Programming Explained: Embrace Change, second edition,* Indianapolis, IN: Addison-Wesley Professional.

Berinato, S. (2003) “The Bugs Stop Here”, http://www.cio.com/archive/051503/bugs.html.

Boman, M. (1997) *Conceptual Modelling,* London: Prentice Hall.

Booysen, H.A.S., and J.H.P. Eloff. (1995) “A Methodology for the Development of Secure Application Systems”, *Proceedings of the 11th IFIP TC11 International Conference on Information Security,* 1995.

Bostrom, R.B., and J.S. Heinen (1977) “MIS Problems and Failures. A Socio-technical Perspective: Part I: The Causes," *MIS Quarterly,* (1)3, pp. 17–32.

Bradley, T. (2011) “Microsoft Calls for Safer and Healthier Internet”, PCWorld, http://www.pcworld.com/businesscenter/article/219788/microsoft_calls_for_safer_and_healthier_internet.html (current October 29, 2011).

Bulgurcu, B., H. Cavusoglu, and I. Benbasat (2010) “Information Security Policy Compliance: An Empirical Study of Rationality-based Beliefs and Information Security Awareness”, *MIS Quarterly,* (34)3, pp. 523–A7.

C&A Systems Security Limited (2012) “COBRA”, *C&A Systems Security Limited,* http://www.riskworld.net/ (current 2012).

Common Criteria (2012) “Common Criteria”, http://www.commoncriteriaportal.org/ (current 2012).

Compagna, L., P. El Khoury, A. Krausová, F. Massacci, et al. (2009) “How to Integrate Legal Requirements into a Requirements Engineering Methodology for the Development of Security and Privacy Patterns”, *Artificial*

27000.org Directory (2014) "The ISO 27000 Directory," 27000.org Directory, http://www.27000.org/iso-27002.htm (current January 25, 2014).

Computer Security Institute (2009) “CSI Computer Crime and Security Survey 2009”, *Computer Security Institute,* http://gocsi.com/survey (current April 25, 2010).

Computer Security Institute (2010/2011) *CSI Computer Crime and Security Survey,* Computer Security Institute,

comScore (2011) “The 2010 U.S. Digital Year in Review”, comScore, http://www.comscore.com/Press_Events/Press_Releases/2011/2/comScore_Releases_The_2010_U.S._Digital_Year_in_Review (current October, 28, 2011).

D’Arcy, J., and A. Hovav (2009) “Does One Size Fit All? Examining the Differential Effects of IS Security Countermeasures”, *Journal of Business Ethics,* (89)1, pp. 59–71.

D’Arcy, J., A. Hovav, and D. Galletta (2009) “User Awareness of Security Countermeasures and Its Impact on Information Systems Misuse: A Deterrence Approach”, *Information Systems Research,* (20)1, pp. 79–98.

Dacey, R.F. (2003) *INFORMATION SECURITY Effective Patch Management Is Critical to Mitigating Software Vulnerabilities,* United States General Accounting Office.

deJong, J. (2006) “Slipping in the Side Door with App Security Message”, BZ Media LLC, http://www.sdtimes.com/article/special-20060815-01.html (current August 28, 2006).

Deloitte (2010) *2010 Financial Services Global Security Study.*

Deshpande, Y. (2004) Web Engineering Curriculum: A Case Study of an Evolving Framework, *Web Engineering 4th International Conference, ICE 2004,* Munich, Germany, p. 526.

Deshpande, Y., S. Murugesan, A. Ginige, S. Hansen, et al. (2002) "Web Engineering", *Journal of Web Engineering*, (1)1, pp. 3–17.

Dictionary.com (2005) "Trust", Lexico Publishing Group, LLC, http://dictionary.reference.com/search?q=Trust (current January 25, 2014).

Ellis, J., and T. Speed (2001) *The Internet Security Guidebook: From Planning to Deployment,* San Diego: Academic Press.

Fernandez, E.B. (1999) "Coordination of Security Levels for Internet Architectures", *Proceedings of the 10th International Workshop on Database and Expert Systems Applications,* pp. 837–841.

Fildes, J. (2010) "Stuxnet Worm 'Targeted High-value Iranian Assets'", BBC, http://www.bbc.co.uk/news/technology-11388018 (current October 28, 2011).

Foster, J.C. (2004) "Five Hidden Tactics for Secure Programming", *Ounce Labs,* March 25, 2004.

Fowler, M., and J. Highsmith (2001) "The Agile Manifesto", Dr. Dobb's Portal, http://www.ddj.com/dept/architect/184414755#sidebar (current June 1, 2006).

Glass, R.L. (2003) *Facts and Fallacies of Software Engineering,* Boston, MA: Addison-Wesley.

Glisson, W.B. (2008) *The Web Engineering Security (WES) Methodology,* Glasgow, Scotland: University of Glasgow.

Glisson, W.B., L.M. Glisson, and R. Welland (2006a) "Web Development Evolution: The Business Perspective on Security", Thirty-Fifth Annual Western Decision Sciences Institute, Hawaii, 2006a.

Glisson, W.B., A. McDonald, and R. Welland (2006b) "Web Engineering Security: A Practitioner's Perspective", International Conference on Web Engineering, Palo Alto, California, 2006b.

Glisson, W.B., and R. Welland (2005) "Web Development Evolution: The Assimilation of Web Engineering Security", Third Latin American Web Congress, Buenos Aires, Argentina, 2005.

Glisson, W.B., and R. Welland (2006) "Web Engineering Security (WES) Application Survey Technical Report", Glasgow, Scotland: University of Glasgow.

Glisson, W.B., and R. Welland (2007a) "Web Engineering Security: Essential Elements", The Second International Conference on Availability, Reliability and Security (ARES) Vienna, Austria, 2007a.

Glisson, W.B., and R. Welland (2007b) *Web Survey Technical Report,* Glasgow, Scotland: University of Glasgow.

Graff, M.G., and K.R. van Wyk (2003) *Secure Coding Principles & Practices,* Sebastopol, CA: O'Reilly & Associates Inc.

Hansche, S., J. Berti, and C. Hare (2004) *Official (ISC)2 Guide to the CISSP Exam,* Boca Raton, FL: Auerbach.

Hare, C. (2004) *Policy Development, fifth edition,* Tipton, H.F., and M. Krause (eds.), Information Security Managment Handbook, Boca Raton, FL: Auerbach Publications, pp. 925–943.

Hertzum, M., N. Jørgensen, and M. Nørgaard (2004) "Usable Security and e-Banking: Ease of Use vis-a-vis Security", *Australasian Journal of Information Systems,* (11)2.

Hirschheim, R.A. (1985) "Information Systems Epistemology: An Historical Perspective", in Mumford, E., R. Hirschheim, G. Fitzgerald, and T. Wood-Harper (eds.), *Research Methods in Information Systems,* North-Holland, Amsterdam, pp. pp.3–9.

Howard, P.D. (2004) "The Security Policy Life Cycle: Functions and Responsibilities", in Tipton, H.F., and M. Krause (eds.), *Information Security Management Handbook, fifth edition,* Boca Raton, FL: Auerbach Publications.

IBM (2011) *IBM X-Force 2011 Mid-year Trend and Risk Report*, Armonk, NY: IBM.

IBM Global Services (1999) "Business Continuity: New Risks, New Imperatives and a New Approach", IBM Global Services, http://www-935.ibm.com/services/us/index.wss/summary/bcrs/a1000475?cntxt=a1000388 (current June 6, 2006).

International Organization for Standardization, "ISO/IEC 12207:2008", International Organization for Standardization, http://www.iso.org/iso/catalogue_detail?csnumber=43447 (current January 25, 2014).

ISO (2012) "International Organization for Standards", http://www.iso.org/iso/en/ISOOnline.frontpage (current January 25, 2014).

Joshi, J.B.D., W.G. Aref, A. Ghafoor, and E.H. Spafford (2001) "Security Models for Web-based Applications", *Communications of the ACM,* (44)2, pp. 38–44.

Kaplan, R. (2004) "A Matter of Trust", in Tipton, H.F., and M. Krause (eds.), *Information Security Management Handbook, fifth edition,* Boca Raton, FL: Auerbach Publications.

Karjalainen, M., and M. Siponen (2011) "Toward a New Meta-theory for Designing Information Systems (IS) Security Training Approaches", *Journal of the Association for Information Systems* (12)8.

Klein, H.K., and M.D. Myers (1999) "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems", *MIS Quarterly,* (23)1, pp. 67–93.

Kroeger, A. (2011) "Flood Recovery in Australia, Sri Lanka, the Philippines", BBC, http://www.bbc.co.uk/news/world-asia-pacific-12179996 (current October 28, 2011).

Lipner, S. (2004) "The Trustworthy Computing Security Development Lifecycle", 2004 Annual Computer Security Applications Conference, Tucson, Arizona, 2004.

McDermott, J., and C. Fox. (1999) "Using Abuse Case Models for Security Requirements Analysis", Computer Security Applications Conference 1999 (ACSAC '99), Phoenix, AZ, 1999, pp. 55–64.

McDonald, A. (2004) *The Agile Web Engineering (AWE) Process,* Ph.D. Thesis, Glasgow, Scotland: University of Glasgow.

McDonald, A., and R. Welland (2005) "Agile Web Engineering (AWE) Process: Perceptions Within a Fortune 500 Financial Services Company", *Journal of Web Engineering,* (4)4, pp. 283–312.

Mimoso, M.S. (2003) "Top Web Application Security Problems Identified SearchSecurity.com", SearchSecurity.com, http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci873823,00.html?NewsEL=9.25 (current April 12, 2005).

Mitnick, K. (2002) *The Art of Deception: Controlling the Human Element of Security,* Mitnick, K.D., and W.L. Simon (eds.), Indianapolis, IN: Wiley.

Mont, M.C., P. Bramhall, M. Gittler, J. Pato, et al. (2002) *Identity Management: A Key e-Business Enabler,* Trusted E-Services Laboratory, HP Laboratories, Bristol, CT.

Mouratidis, H., J. Jürjens, and J. Fox (2006) "Towards a Comprehensive Framework for Secure Systems Development", in Dubois, E., and K. Pohl (eds.), *Advanced Information Systems Engineering, vol. 4001,* Berlin/Heidelberg, Germany: Springer, pp. 48–62.

Niekerk, J.V., and R.V. Solms (2004) "Corporate Informaiton Security Education", in Deswarte, Y., F. Cuppens, S. Jajodia, and L. Wang (eds), *Information Security Management, Education and Privacy,* IFIP 18th World Computer Congress, TC11 19th International Information Security Workshops, Toulouse, France.

Oates, B. (2006) *Researching Information Systems and Computing,* London: Sage Publications.

Organization for Internet Safety (2004) "Guidelines for Security Vulnerability Reporting and Response", http://www.symantec.com/index.jsp.

Ounce Labs (2004) *Weapons for the Hunt: Methods for Software Risk Assessment,* Waltham, MA: Ounce Labs, Inc.

Ozier, W. (2004) "Risk Analysis and Assessment", in Tipton, H.F., and M. Krause (eds.), *Information Security Managment Handbook, fifth edition,* Boca Raton, FL: Auerbach Publications, pp. 795–820.

Peltier Associates (2005) "Peltier Associates Facilitated Risk Analysis Process (FRAP)", http://www.peltierassociates.com/frap.htm (current October 16, 2007).

Peltier, T. (2000) "Effective Risk Analysis", 23rd National Information Systems Security Conference, Baltimore, MD, 2000.

Pfleeger, C.P., and S.L. Pfleeger (2003) *Security in Computing, third edition,* Upper Saddle River, NJ: Prentice Hall.

Phaltankar, K.M. (2000) *Practical Guide for Implementing Secure Intranets and Extranets* Boston, MA: Artech House, Inc.

Premkumar, T., S.S. Devanbu (2000) "Software Engineering for Security: A Roadmap", *Proceedings of the Conference on The Future of Software Engineering,* International Conference on Software Engineering Limerick, Ireland, 2000, pp. 227–239.

PricewaterhouseCoopers (2011) *Global State of Information Security Survey*, New York.

Puhakainen, P., and M. Siponen (2010) "Improving Employees' Compliance Through Information Systems Security Training: An Action Research Study", *MIS Quarterly,* (34)4, p. 21.

Rakitin, S.R. (1997) *Software Verification and Validation: A Practitioner's Guide,* Boston, MA: Artech House.

Ramsin, R., and R.F. Paige (2008) "Process-centered Review of Object Oriented Software Development Methodologies", *ACM Computer Survey,* (40)1, pp. 1–89.

Rothke, B. (2004) "A Look at the Common Criteria", in Tiption, H.F., and M. Krause (eds.), *Information Security Managment Handbook, fifth edition,* Boca Raton, FL: Auerbach, pp. 969–977.

Royce, W.W. (1987/1970) "Managing the Development of Large Software Systems: Concepts and Techniques", *Proceedings of the 9th International Conference on Software Engineering,* 1987/1970, pp. 328–338.

Saltzer, J.H., and M.D. Schroeder (1975) "The Protection of Information in Computer Systems", University of Virginia, Department of Computer Science, http://www.cs.virginia.edu/~evans/cs551/saltzer/ (current June 5, 2006).

Schwaber, K, and J. Sutherland (2011) "The Scrum Guide", Scrum.org, http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf.

Siponen, M. (2000) "A Conceptual Foundation for Organizational Information Security Awareness", *Information Management & Computer Security,* (8)1, p. 10.

Siponen, M., R. Baskerville, and J. Heikka (2006) "A Design Theory for Secure Information Systems Design Methods", *Journal of the Association for Information Systems*, (7)11.

Siponen, M., R. Baskerville, and T. Kuivalainen (2005) "Integrating Security into Agile Development Methods", *Proceedings of the 38th Hawaii International Conference on System Sciences,* Hawaii, 2005.

Siponen, M., and J. Heikka (2008) "Do Secure Information System Design Methods Provide Adequate Modeling Support?", *Information Software Technology,* (50)9–10, pp. 1035–1053.

Siponen, M.T. (2005a) "Analysis of Modern IS Security Development Approaches: Towards the Next Generation of Social and Adaptable ISS Methods", *Information Organization,* (15)4, pp. 339–375.

Siponen, M.T. (2005b) "An Analysis of the Traditional IS Security Approaches: Implications for Research and Practice", *European Journal of Information Systems,* (14)3, pp. 303–315.

Siponen, M.T. (2006) "Secure-system Design Methods: Evolution and Future Directions", *IT Professional,* (8)3, pp. 40–44.

SSE-CMM (2003) Systems *Security Engineering—Capability Maturity Model (SSE-CMM),* Model Description Document, Pittsburgh, PA: Carnegie Mellon University.

Symantec (2005) "Importance of Corporate Security Policy Defining Corporate Security Policies, Basing Them on Industry Standards, Measuring Compliance, and Outsourced Services Are Keys to Successful Policy Management", Symantec, http://securityresponse.symantec.com/avcenter/security/Content/security.articles/corp.security.policy.html (current July 15, 2005).

The Open Web Application Security Project (2004) "The Ten Most Critical Web Application Security Vulnerabilities", *The Open Web Application Security Project,* http://www.owasp.org/index.jsp.

The Open Web Application Security Project (OWASP) (2004) "The Ten Most Critical Web Application Security Vulnerabilities", *The Open Web Application Security Project,* http://www.owasp.org/index.jsp.

Tiller, J.S. (2004) "Outsourcing Security", in Tiption, H.F., and M. Krause (eds.), *Information Security Management Handbook, fifth edition,* Boca Raton, FL: Auerbach Publication, pp. 1061–1072.

Travis, D. (2009) "247 Web Usability Guidelines", *UserFocus,* http://www.userfocus.co.uk/resources/guidelines.html (current September 12, 2012).

U.S. Department of Commerce (2011) *Quarterly Retail e-Commerce Sales,* Washington, DC: U.S. Department of Commerce.

U.S. Department of Health and Human Services (2006) "Research-based Web Design & Usability Guidelines", Washington, DC: U.S. Government Printing Office, http://www.usability.gov/guidelines/index.html (current September 12, 2012).

Viega, J., and B. McGraw (2005) *Building Secure Software,* Boston, MA: Addison-Wesley.

W3C (2005) “W3C World Wide Web Consortium”, W3C World Wide Web Consortium, http://www.w3.org/ (current April 24, 2005).

W3C (2008) “Web Content Accessibility Guidelines (WCAG) 2.0”, W3C, September 12, 2012.

Walsham, G. (2006) “Doing Interpretive Research”, *European Journal of Information Systems,* (15)3, p. 10.

Williams, P., T. Shimeall, and C. Dunlevy (2002) *Intelligence Analysis for Internet Security, volume 23,* pp. 1–38. New York: Routledge, part of the Taylor & Francis Group.

Wylder, J.O. (2004) “Towards Enforcing Security Policy: Encouraging Personal Accountability for Corporate Information Security Policy”, in Tipton, H.F., and M. Krause (eds.), *Information Security Management Handbook, fifth edition,* Boca Raton, FL: Auerbach Publications, pp. 945–952.

APPENDIX 1: ORGANIZATIONAL SURVEY AND INTERVIEWS

An in-depth survey was conducted during July and August of 2005, at a Global Fortune 500 financial organization, which focussed on security. The primary objective of the survey was to determine the areas where security practices were being successfully applied and to gain an accurate understanding of the role that security plays in a large organization’s application development process. A secondary objective was to establish a base line to identify the impact of using WES within the organization.

The Survey Method

Sixteen one-to-one interviews were conducted with selected employees representing a variety of roles with a diversity of work experiences within the technical side of the organization. All of the employees were directly involved in the application development process and some of these were security specialists. Our aim was to ensure that these employees covered all aspects of the software development process.

All interviews were conducted and recorded by one of the authors (Glisson) using an interview template consisting of thirty questions. Most of the questions were structured to elicit an initial response (yes/no/do not know) followed by supplementary questions to drill down into the reasons for these responses. To comprehend the security challenges, the application development process was examined first in order to understand the environment. Then the security implications of the environment were scrutinized.

The interview template was split into four parts to:

1. gather information about the interviewee demographics (four questions)
2. understand the application process within the organization, the interviewees’ role in it and their understanding of the process (nine questions)
3. focus on the security, the existing security architecture and its perceived strengths and weaknesses (fifteen questions)
4. wrap up the interview by getting the interviewees’ feedback on the questions and to give them the opportunity to offer any other information they thought was relevant (two questions)

The Interviews

Interviewee Demographics

The first four questions were used to establish the interviewee’s current role in the organization, his/her number of years of experience and a brief idea of the individual’s history. These questions revealed that the interviewees were experienced IT professionals who had a variety of technology backgrounds and, in general, several years of experience. The average number of years of experience among the sixteen respondents was just under fourteen years.

The Application Development Process

Question 5 firmly established the existence of a documented application development process. There was some discrepancy among individuals regarding the process specifics but the general idea is that the organization uses a customized plan driven version of the waterfall approach. The good points identified included providing structure to the development environment, being well understood in the organization, providing accountability, and flexibility at the more detailed level. The bad points of the process were business time-to-market, heavy documentation, and one-size-fits all (nonflexible) approach.

Question 6 identified the areas in which the interviewees are engaged in the product lifecycle and was used to check that we had a good coverage of all relevant development roles.

Question 7 asked about the effectiveness of the application development process. Only six out of sixteen interviewees indicated that is effective. Four indicated that it was not successful and six indicated that it was successful "Sometimes."

Out of the four who indicated that the development process is not effective, these individuals indicated that the process was not cost effective; too heavy on the documentation, too slow, and applications are chosen based on business need and not how they fit with the IT structure. Out of the respondents who indicated effectiveness "Sometimes," they thought that the application development methodology was good for project structure and repeating projects. They thought it was not effective when considering time-to-market issues and rapid application development needs, introduction of new technology, and the bureaucracy and overheads of the process.

Question 8 asked about the amount of time it takes to get a project from inception to delivery. There was a range of answers to the inquiry, but the average was about a year, give or take a couple of months.

Question 9 was about project time lines. Thirteen out of sixteen respondents indicated that they felt that the project timelines should be shorter. The reasons behind the desire for a shorter process included the loss of potential business opportunities, market competitiveness, and the need to take advantage of new technologies.

Question 10 asked if projects exceeded their estimated time frames. There was unanimous agreement that projects exceed estimated time frames within the organization. The reason for exceeding time frames suggested were changing business requirements, complex technical environments, lack of technical expertise, inadequate estimation techniques, and inexperienced project managers. Fifteen respondents indicated that security issues contributed to elongated time frames in some way.

Question 11 explored budget overruns. Fifteen out of sixteen respondents indicated that projects exceed the estimated budgets. The main reasons for budget overruns included poor managerial planning, resource issues, and changing business requirements. Seven of the respondents indicated that security issues had contributed to budget over runs and two indicated that it was possible.

Question 12 asked about the existence of any corporate recommendations in terms of optimal overall time frames for development. The effective answer to this question, in the organization, is that one does not explicitly exist. However, there may be expectations from various business units and timeframes exist within specific pieces of the overall development cycle processes.

Question 13 explored how the in-house development process is used. Eight individuals indicated that projects always follow the in-house development project. However, in exploring these answers further, there was some doubt about the extent of interviewees' knowledge of whether the whole process was used. Five of the respondents indicated that all projects did not follow the development process. Two of the respondents indicated that it happened "Sometimes."

The "No" responders indicated that reasons included people attempting to circumvent the process, critical time scales, and poor project planning. One point of interest that did surface during this line of questioning is the fact that after the design approval by Design Architecture Committee the development process has the potential to break down and be discarded in the name of project completion. The individuals who answered "Sometimes" indicate that it is up to the project manager to follow the development process and that exceptions have been made in the past in order to get around following the methodology.

The Role of Security

Question 14 attempted to ascertain what individuals in the industry feel a security development process should contain. There was a wide range of answers for this question with several answers indicating that the security development process should contain specific stages of the development lifecycle. Additional answers also indicated best practices, guidelines, communication, training, and accountability. All of these are valid responses, however, the lack of a clear, straightforward answer indicates a potential discrepancy in the definition of the term *security* and the interpretation of the phrase *a security development process* within the organization.

Question 15 was designed to determine areas where security is not engaged in the existing development lifecycle. The results indicated that there are clear deficiencies in security visibility within the overall development process.

Security is severely lacking in the business analysis stage. Clearly, there were issues with security in the evaluation, maintenance, and evolution stages. Slightly less emphasis was given to security issues in the testing and deployment stages, but it could be argued that there is a potential problem or perception of a problem, in these stages as well. In fact, the only two stages where security is clearly perceived to be involved in the development process are the requirements and the design stages.

Question 16 ascertained the number of people who think there is a documented security development process and explored the strengths and weaknesses of this process. The company does actually have a documented security process maintained by the Project Security team. The responses revealed that the knowledge of the document is restricted to specific groups, and only five interviewees were aware of it.

The good points of the security development process include high structure that helps to provide documentation. The highly structured process creates an environment that is conducive to audits and future reference needs. The documentation was also listed as a drawback to the process along with explicitly making one group responsible for security verses making everyone responsible for security. Security awareness was one point that was mentioned that still needs to be developed within the organization.

The problems that were discussed with the current security process included a lack of emphasis on the employee, a lack of utilization of the current process, a lack of security involvement after the design has been signed off, a lack of security awareness, and a lack of stakeholder buy-in to security. The point of breakdown appears to be around the entire development process. The process takes too long. The business has the power to circumvent the process to keep projects on track from a timeline and budget perspective, while a shortage of personnel and problems around post-implementation and change management need to be addressed.

The general thought behind the lack of a security process within the organization seemed to be that the individuals involved in security do not record the process; they just do what needs to be done. These people are viewed as a resource and are accessed as needed during the development process. However, there is some confusion over when and where the Security Team actually gets involved in the process. This is taken to the point that it is viewed as the solutions designers' problem. There is also the view that security is a bolt-on issue that is addressed after the coding is complete. Hence, the organization is giving security only lip service and not truly pursuing a security architecture infrastructure.

Question 17 attempted to determine how applications are deemed secure within the organization. There were a variety of answers to this query. The answers identified a number of different parts of the application process: requirements, security policies and standards, process, testing, audits, and reviews.

Expanding on these points: Requirements referred to the business and technical application requirements. The Security team sets the security policies and standards and industry standards are used to help ensure security within the organization. Process referred to the creation of the Design Architecture Document and submitting it to the Design Architecture Committee. Testing referred to internal penetration testing and third party testing.

Question 18 asked how an application is deemed secure from a development perspective. The result is that testing is subjective and tailored around the needs of the application based on the functional and non-functional requirements. The general rule is that high-risk applications require more testing and third-party testing. Outwardly facing applications are more rigorously tested than inwardly facing applications.

Question 19 was used to determine the stakeholders who are responsible for security at the various stages of the development lifecycle. The results from this question indicate that there is a lot of confusion about who is responsible for what and at what stages of the lifecycle. The responses from the participants indicate that the Security Team is perceived to have the most responsibility through the various stages of the development lifecycle. However, many other answers were given in response to this question. This confusion over which stakeholders are responsible for security supports the results obtained from question fifteen where there were clearly areas in the development lifecycle where security is not involved in the process. If you do not know which stakeholders are responsible for the security, it stands to reason that it would be difficult to know where security is involved in the process.

Question 20 was designed to try to pinpoint a specific individual role that is responsible for security within the organization. Eleven of the respondents indicated that there was an individual responsible for security within the organization and of these some form of the Security Team was identified by name six times.

Question 21 attempted to determine if conflicts arise between the stakeholders and the individuals responsible for security. Fourteen of the respondents indicated that conflicts arise between the two groups. The types of conflicts were financial and time constraints, or conflicts over security solutions. The disagreement over the security solution appears to have its roots in the perception of the level of risk that is perceived with an application. Hence, a higher level of risk would necessitate a stronger security solution. This disagreement over risk assessment occurs between the business unit and the application developers.

Question 22 was designed to determine the extent contractors are used in the organization and to determine if they present a major risk to the organization. The initial result is that the company uses contractors very heavily. There was only one group that did not use contractors. The majority of the respondents indicated that contractors are held to the same application development methodology as employees. If they do use a different process, the process is examined and approved by the proper individuals within the organization. The majority of the respondents indicated that contractors are also held to the same security requirements as employees. However, reading between the lines in conversation, the organization does not do the testing for them on the applications that they are building. Hence, there is the underlying possibility that there could be discrepancies in application testing. How effectively this is monitored appears to be up to the project manager.

Question 23 sought the interviewee's opinion on the emphasis security is given within the organization. The answers to this question were widely varied. Some individuals thought that the emphasis on security was strong, due to outside factors such as legislation, while others felt that the emphasis was weak. Two individuals felt that the emphasis had improved over the past several months while others felt that the security focus is misaligned with the application development process. Some individuals felt that security played a large role in the organization while others felt that the emphasis was small and that security is effectively seen as an inhibitor rather than an enabler in the development process.

Question 24 was designed to determine if the elements of the existing in-house security process are always followed. Seven of the sixteen respondents indicated that it was not always followed. The reasons for not following the security process included: time pressures; bureaucracy; lack of awareness; and a lack of security involvement in certain aspects of the process. Other reasons that were mentioned included the complete lack of a security process and where the application sits, i.e., does the application face the Internet or is it internal.

Question 25 revealed that the majority of the individuals who were surveyed (eleven out of sixteen) felt that security should play a larger role in the organization's development environment. The reasons these individuals gave for the security role needing to be larger mainly concerned the nature of the business. They indicated that the organization is relatively small in the financial world, and protection of the reputation is critical. In the current environment, security can be de-scoped for numerous reasons, integrating security into the development process up front would cut development overhead and increase security awareness within the organization.

Question 26 asked whether there was a job-related impact for not following the development security process. The responses were: eight felt there was no impact, six thought there was, and two did not know.

Question 27 attempted to determine the areas that require a greater or reduced emphasis on security within the company process. There were a variety of answers, but there were some recurring themes, all identifying areas requiring greater emphasis: four interviewees talked about business requirements, four interviewees talked about education, and five interviewees talked about testing. These themes indicate that there are problems with these areas in the organization.

Question 28 asked about the major security threats during application development. Common themes included requirements/design/implementation/testing (seven answers), people and behavior (three), policy circumvention and enforcement (two), and viruses (two).

There were a variety of answers to the supplementary question inquiring which of these issues are being met by the existing process, which ranged from "None" to "All." A theme that did surface in a few of the answers is that separation of duty between code reviews and testing is sufficient within the organization. There were several "None" responses to the further question asking which issues the existing process was not satisfying. Other answers included a lack of documentation, internal and external coding issues, and a lack of security in the solution design.

Wrap Up Questions

Question 29 was included to analyze the survey instrument. Eight individuals indicated that there were no questions that were vague or difficult to follow. Three individuals indicated that there was some confusion over the term application development versus the term that the organization uses that is "product lifecycle." One individual thought that question 23 was difficult to follow and prevented him from delivering a clean concise response. Two individuals thought that there were a lot of questions about a security development process that does not exist.

Question 30 allowed interviewees to add any additional comments that they feel are relevant to the survey. Five of the interviewees did not have any additional information to offer. The answers from the balance of the responders were extremely varied. Their answers included discussing interviewee backgrounds, general discussions about the survey, the definition of security and the skill sets and training of employees.

Conclusions

The organization uses a customized plan driven waterfall approach to govern all projects at the higher levels, including Web-based applications. The practical environment operates in the following way. The business gets a project idea approved through a business analysis process, which is followed by a business requirements stage. At that point a project manager from the technical side of the organization is assigned to the project. The project manager acquires an architect from the architecture team, who is charged with the task of creating a high-level design that meets the business requirements, that fits into the existing organization's infrastructure and is acceptable by all relevant parties. The main tool that is used to complete this task is the Design Architecture Document (DAD). Depending on the size of the project the architect can create a preliminary DAD and then a final DAD for really large projects, or the architect can create a one-time DAD for small-to-medium projects.

The DAD is then presented to the Design Architecture Committee (DAC) for approval. At this point the DAC can do one of three things. They can approve the design, approve the design with conditions, or reject the design. If the design is approved, it moves on to the appropriate team to implement the solution using a conventional cycle of code, code reviews, testing, and evaluation. If it is approved with caveats, it is up to the architect who created the DAD to satisfy the condition so that the project can move on to the implementation stage. If any member of the DAC rejects the design, the architect must go back to the drawing board and try again.

It is questionable as to whether the development process was always followed. The survey revealed that the individual teams that actually implemented the projects used their own methodologies. These methodologies have been customized so that they can meet the deliverables mandated by the high-level plan-driven waterfall approach and meet timescales required by the business. Realistically, the organization was operating two different approaches to application development at different levels within the organization. The high-level approach was a customized version of the plan-driven waterfall approach. The low-level approach consisted of a number of ad-hoc processes contrived by the individual coding teams. After going through a formal design approval process (DAD/DAC), there was no verification that the design implemented in production is the design that was originally approved.

The general indication from the interviewee answers was that projects exceed estimated budgets and timeframes on a fairly regular basis. Interviewee answers also indicated that the current application development process is not effective when considering time-to-market issues, rapid application development needs, and the introduction of new technology, resulting in a lack of efficiency.

The results of the survey indicated that there are areas within the organization's development process that are experiencing deficiencies in security and need to be addressed. The existing DAD did address some security issues. An application's security requirements must be listed in a table along with a description of how they are being satisfied. The description of how these requirements are being met should include a discussion on how the proposed solution addresses confidentiality, integrity, and availability. The current document also addresses backup and recovery, purge and archive of information, and disaster recovery plans. However, it was not clear how these issues were addressed during implementation and what steps were taken to verify that appropriate actions had been taken.

Security Education

The organization generally emphasises the importance of security because of the nature of its business. There is a well-defined security policy, but many developers seem to be unaware of this. There is clearly a failure to ensure that security policies are disseminated throughout the organization from senior management. There is also a feeling among developers that security is the responsibility of the security team and, thus, security is somebody else's problem. So there is a clear need for security education for the IT employees, emphasising that security is everybody's problem and that there is an organizational security policy.

Job-related Impact for Not Following Security Processes

The perceived lack of any job-related impact for not following security processes is another example of the feeling that security is not the responsibility of the developers. If there is no accountability for failures to follow correct procedures, there is no incentive to fully adhere to these procedures.

Integration of Security with the Application Development Process

Currently, security is seen as a separate process “bolted on” to the development process. Code reviews and testing are seen as being important aspects of the development process, but these do not seem to be explicitly linked to security requirements. It is particularly disturbing that security issues were not perceived to be a fundamental part of the Business Requirements stage at the beginning of the development process. Conflicts between the business stakeholders and the security specialists also suggest that there is not a clearly defined process for risk analysis. The statement that security is seen as an inhibitor rather than an enabler in the development process suggests that any specified security requirements may not be fully implemented. All these points indicate that security issues need to be integrated into the application development process from the beginning and throughout all stages, and that the security specialists need to be more actively engaged with the developers and not seen as a separate group inhibiting the development process.

A detailed analysis of the survey results and how these contributed to the creation of the Security Criteria for Web Application Development (SCAWD) has been previously published [Glisson et al., 2006b].

ABOUT THE AUTHORS

William Bradley (Brad) Glisson was the Director of the Computer Forensics and E-Discovery MSc Program at the University of Glasgow. He has accepted a post as an Associate Professor in the School of Computing at the University of South Alabama. His current research focuses on examining the challenges and evaluation of practical solutions for cloud computing forensics investigations; practical research into the technical solutions, methodologies, and toolset capabilities used in mobile device forensics; legislative and business implications of application development on the Web, along with forensic investigations of these applications; and accurate representation, preservation, forensic integrity, and recovery of information. He received his Ph.D. in Computing Science from the University of Glasgow.

Ray Welland is emeritus Professor of Software Engineering and former Chair of the School of Computing Science at the University of Glasgow in Scotland. His interests include software design methods and tools, software development processes, agile Web engineering, Web Engineering Security, and usable security.

Communications of the Association for Information Systems

ISSN: 1529-3181