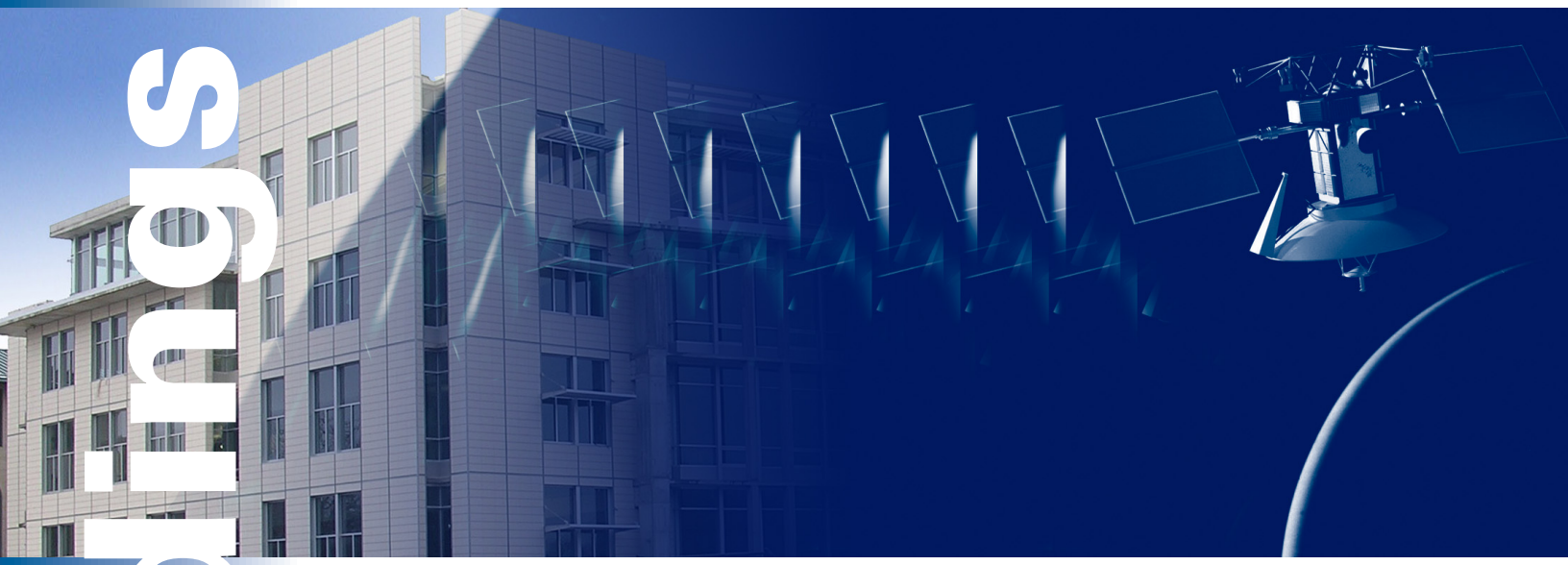
Proceedings



Making the Business Case for Software Assurance Workshop

September 26, 2008
Pittsburgh, Pennsylvania

CERT

Software Engineering Institute
Carnegie Mellon

Carnegie Mellon CyLab
CONFIDENCE FOR A NETWORKED WORLD

Sponsored by the Software Engineering Institute's CERT® Program and Carnegie Mellon CyLab
in support of the Department of Homeland Security Software Assurance Program

Proceedings of the Making the Business Case for Software Assurance Workshop

September 26, 2008

Pittsburgh, Pennsylvania

Sponsored by the Software Engineering Institute's CERT® Program and Carnegie Mellon CyLab in support of the Department of Homeland Security Software Assurance Program

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

The papers in this book compose the proceedings of the meeting mentioned on the cover and title page. They reflect the authors’ opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors or Carnegie Mellon University.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

Additional copies may be ordered from
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-3890
USA

Table of Contents

Organizational Development Issues

Welcome from the Workshop Chairs

The goal of the Making the Business Case for Software Assurance Workshop is to bring together researchers and practitioners from the fields of software engineering, system engineering, software security, and software assurance to exchange ideas and their experiences in support of a business case for software assurance.

This one-day workshop will explore methods for making the business case for software assurance, and associated issues. The workshop will include invited speakers, presentations of refereed papers, and facilitated discussion sessions. Much research and development remains to be done in this area, and together researchers and practitioners need to identify and explore the important issues and the challenges we face. Together we can propose, formulate, and evaluate promising solutions.

A set of topics has been identified for discussion at the workshop. Discussing these topics should clarify common assumptions and important issues. The accepted papers are each associated with at least one of the topics. The topics include the following:

- Measurement
- Process and Decision Making Issues
- Legal Issues
- Globalization
- Risk Issues
- Organizational Development Issues

We thank the speakers and authors for their submissions, the members of the workshop program committee for their constructive reviews, and the sponsors of this workshop. We also appreciate the support received from the Software Engineering Institute and CyLab for the workshop organization and publication process. The support of Pamela Curtis, the SEI editor who prepared these proceedings, and Rita Briston, who was responsible for workshop administration, is especially appreciated.

Jan Vargas, General Chair
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA
jmvargas@cert.org

Nancy R. Mead, Program Chair
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA
nrm@sei.cmu.edu

Organizing Committee

Jan Vargas, General Chair

Nancy Mead, Program Chair

David White, Facilitator

Pamela Curtis, Publications Chair

Bob Fantazier, Design

Rita Briston and Linda Whipkey, Logistics

Program Committee

Julia Allen – Software Engineering Institute

John Bailey – Institute for Defense Analysis

Antonio Drommi – University of Detroit Mercy

Jeff Ingalsbe – Ford Motor Company

Jim McCurley – Software Engineering Institute

Mary Polydys – National Defense University

Sivaram Rajagopalan – Ernst & Young

David Root – Carnegie Mellon University

Mel Rosso-Llopart – Carnegie Mellon University

Dan Shoemaker – University of Detroit Mercy

Rahul Telang – Carnegie Mellon University

Chuck Weinstock – Software Engineering Institute

Bob West – Echelon One

Dan Wolf – Cyber Pack Ventures and Software Assurance Consortium

Agenda

8:30a – 8:45a	Welcome, Introduction, and Workshop Rules	Jan Vargas and Nancy Mead

Keynote (Session Chair: Nancy Mead)

8:45a – 9:30a	Software Assurance: Mitigating Risks to the Enterprise	Joe Jarzombek

Measurement Issues (Session Chair: Dan Wolf)

9:30a – 10:00a	No More Adjectives	Dan Geer
10:00a – 10:15a	What Measures Do Vendors Use for Software Assurance?	Jeremy Epstein
10:15a – 10:30a	Discussion	
10:30a – 10:45a	BREAK	

Process and Decision Making Issues (Session Chair: Carol Woody)

10:45a – 11:15a	Benchmarking Assurance Practices: Contributions to a Business Case for Assurance	Michele Moss and Nadya Bartol
11:15a – 11:30a	Making Business-Based Security Investment Decisions—A Dashboard Approach	Julia Allen
11:30a – 11:45a	Process Investment Value Returns (PIVR) Framework for Measuring Returns on Process Improvement	Balaji Santhanam
11:45a – 12:15p	Discussion	

Legal Issues (Session Chair: Jan Vargas)

12:15p – 1:15p	LUNCH	
12:30p – 1:00p	Prospects for Preserving Software Investment via Patenting	Dennis Carleton
1:00p – 1:15p	Discussion	

Globalization Issues (Session Chair: Julia Allen)

1:15p – 1:45p	Business Impact of and on Software Assurance of the Global Outsourcing of Software Development, Testing, and Use	Warren Axelrod
1:45p – 2:00p	Globalization and the Rise of Mediocrity or Unsafe at Any Speed or Altitude	George Gibbs

2:00p – 2:15p	Inside Track to Offshore Outsourcing Using the Trusted Pipe	Don O'Neill
2:15p – 2:45p	Discussion	
2:45p – 3:00p	BREAK	

Risk Issues (Session Chair: Dan Shoemaker)

3:00p – 3:30p	Where Risk Fails	Brian Chess
3:30p – 3:45p	Three Case Studies in Quantitative Information Risk Analysis	Nicolas Christin
3:45p – 4:00p	Discussion	

Organizational Development Issues (Session Chair: Sivaram Rajagopalan)

4:00p – 4:30p	Promoting Software Assurance on the Front Lines: Industry Proven Practices for Measuring Effective Product Assurance and Employee Training	Paul Kurtz and Dan Reddy
4:30p – 4:45p	It's a Nice Idea but How Do We Get Anyone to Practice It? A Staged Model for Increasing Organizational Capability in Software Assurance	Dan Shoemaker
4:45p – 5:00p	Secure Coding Standards Business Case	Robert Seacord
5:00p – 5:30p	Discussion	
5:30p	Workshop Ends	

Keynote

Software Assurance: Mitigating Risks to the Enterprise

Joe Jarzombek, Director for Software Assurance, National Cyber Security Division, U.S. Department of Homeland Security

The National Cyber Security Division (NCSD) of the U.S. Department of Homeland Security works collaboratively with public, private, and international entities to secure cyberspace and America's cyber assets. In his role as Director for Software Assurance, Joe leads government interagency efforts with industry, academia, and standards organizations to shift the security paradigm away from patch management by addressing security needs in work force education and training, more comprehensive diagnostic capabilities, and security-enhanced development and acquisition practices.

Software is the core constituent of modern products and services—it enables functionality and business operations. In his presentation, Joe will speak to the relevance of software assurance in reducing organizational risk exposure. With today's global IT software supply chain, project management, quality assurance, and software engineering processes must explicitly address security risks posed by exploitable software. In his presentation Joe will highlight how building security in adds value, and discuss how processes should be security-enhanced.

Traditionally, these disciplines have not clearly and directly focused on software security risks that can be passed from projects to the organization. Understanding these risks and the methods to monitor or correct them will guide organizations to improve system predictability and reduce uncertainty.

Software assurance processes and practices span development and acquisition and can be used to enhance processes associated with delivering products, systems, and services. Joe will explain the critical need for adherence to the practices, guidelines, and principles used to build security into every phase of software development and deployment. This includes leveraging existing related models, standards, and schemes. He will discuss free resources that are now available to assist project personnel in managing contracted, outsourcing, and development activities. Joe will also provide an overview of several of the current industry efforts to capture best practices and discuss how they are helping to answer the industry's most pressing questions.

About the Speaker

Joe served in the U.S. Air Force as a Lieutenant Colonel in program management. After retiring from the Air Force, he worked in the cyber security industry as vice president for product and process engineering. Joe also served in two software-related positions within the Office of the Secretary of Defense prior to accepting his current DHS position.

Measurement Issues

No More Adjectives

Dan Geer, Chief Information Security Officer, In-Q-Tel

Assurance, like security, is a means rather than an end. The purpose of risk management around any end is to change the future, not to explain the past. Therefore, assurance metrics are the servants of risk management if and only if they support decision making about risk for the purpose of managing that risk through the adroit choice and steering of means. Adjectives like “faster, cheaper, better” denote ends, but how much faster, cheaper, better, and assured is about choosing amongst means, and it can only be calibrated with numbers. We’ll review the state of the art.

About the Speaker

Dan Geer is Chief Information Security Officer at In-Q-Tel. He’s a security researcher with a quantitative bent, an electrical engineer, a statistician, and someone who thinks truth is best achieved by adversarial procedures.

Milestones: The X Window System and Kerberos (1988), the first information security consulting firm on Wall Street (1992), convenor of the first academic conference on electronic commerce (1995), the “Risk Management Is Where the Money Is” speech that changed the focus of security (1998), the Presidency of USENIX Association (2000), the first call for the eclipse of authentication by accountability (2002), principal author of and spokesman for “Cyberinsecurity: The Cost of Monopoly” (2003), co-founder of SecurityMetrics.Org (2004), and convener of MetriCon (2006-8).

What Measures Do Vendors Use for Software Assurance?

Jeremy Epstein
Cigital, Inc.
jeremy.j.epstein@gmail.com

Books and articles frequently exhort developers to build secure software by designing security in. A few large companies (most notably Microsoft) have completely reengineered their development process to include a focus on security. However, for all except the largest vendors, software security (or software assurance) is a relatively recent phenomenon, and one with an uncertain payoff. In this paper, we examine what real vendors do to ensure that their products are reasonably secure. Our conclusion is that software vendors put significant energy into software security, but there is significant variation in where they invest their money.

1. Introduction

Concern that software products are secure has been around for more than three decades, but until relatively recently was given little attention by the vendor community. The never-ending series of vulnerabilities in Microsoft software galvanized Microsoft, and resulted in their developing a security-focused lifecycle [Howard]. Numerous other texts have described the risks of insecure software, including [Viega] and [McGraw]. More recently, an industry consortium has been formed by some of the larger software companies to define best practices for building secure software [Safecode].

Building on the demand, start-up companies[1] have developed tools to help identify security flaws using techniques such as source code analysis (e.g., Fortify Software, Coverity), binary code analysis (e.g., Veracode), dynamic testing (e.g., SPI Dynamics, NT Objectives, Cenzic), as well as service-focused companies that perform scheduled scans (e.g., Qualys, White Hat Security), education and engineering analysis (e.g., Aspect Security, Cigital), or penetration testing (e.g., Matasano Security).

[1] Inclusion in this non-comprehensive list here should not be interpreted as an endorsement by the author or his employer. Some of the vendors listed here offer products and/or services in addition to those in the list.

Given the choices, vendors, especially those whose primary focus is not security, have difficulty determining where to spend their resources. Additionally, for vendors whose primary products are not security technology, there may be relatively little explicit interest from customers, thus reducing the perceived demand [Epstein].

In order to determine what the "best practices" are that we should follow, we did an informal survey of software vendors to determine how they achieve software security, what motivated them to put energy into software security, and related topics. This paper presents the results of this study, along with its limitations. The paper does *not* make recommendations of what any particular vendor should do, but rather establishes the norms as practiced at this writing.

2. Study Topics & Limitations

The goals of our study were to address four basic questions:

- Who in the organization is involved in software assurance? In particular, we wanted to know:
  - Whether there is a centralized assurance person or team, or whether responsibility is distributed to each engineering team
  - Who has overall responsibility for software assurance, and where that person reports in the organization
  - Whether that person is part of the release decision process, and if so whether they have a veto (i.e., to prevent a product from being released if there are significant security flaws)
- What does the organization do to gain software assurance? In particular, we wanted to know whether the organization:
  - Performs threat modeling to determine the risk factors
  - Performs security design reviews to try to avoid security problems
  - Performs source code reviews (manual or automated) to find implementation flaws
  - Performs automated scans (including, but not limited to, input fuzzing) to find implementation flaws
  - Uses penetration testing (either in-house or third-party) to search for more subtle design or implementation vulnerabilities
  - Provides developer training (and if so, how much and how frequently) so developers can avoid introducing implementation flaws
  - Has an indication (whether by gut feel or metrics) as to which technique(s) are most effective in reducing or eliminating software flaws

- Why does the organization have a software assurance program? For example:
  - Is the interest in software assurance due to direct customer demand, avoiding notoriety, government regulation, etc?
  - How often do customers ask about assurance? Or do they just expect it's there?
  - What words do customers use when asking about assurance?
  - Is the organization seeing procurement language that asks about assurance?
  - Do customers or 3rd parties (e.g., self-styled "security researchers") test the vendor's products for security?
- When did the organization start to focus on software assurance, and how long did it take to see results?

Our study focused exclusively on vendors of shrink-wrapped software. We deliberately eliminated several other types of software developers that might be interesting:

- Custom software developers. Custom software is driven by specific customer requirements, and not by the need to find the common set of capabilities that meet the common needs of a large set of customers. As such, assurance may be given more or less emphasis, depending on the particular customer. This category includes companies that primarily develop software for the government marketplace, including GOTS[2] (Government Off The Shelf).
- Systems integrators. Similar to the custom software developers, these vendors are driven by specific customer requirements, and not by the goal of offering shrink-wrapped software.
- Software as a service. While companies like Salesforce.com and WebEx.com have significant security concerns, they are not (generally) selling their software, but rather use of that software. This would be a logical area to extend the survey, as these vendors are most similar to the shrink-wrapped software market, and are most at-risk due to their products being publicly exposed.
- E-commerce. E-commerce vendors such as Amazon.com have significant software investments, and are at significant risk. However, software is not their primary business, but rather a tool to accomplish their mission.

[2] As distinguished from COTS, or Commercial Off The Shelf software, which is what the commercial software industry calls "shrink wrapped" software.

- Very small vendors. Unless they are specifically focused on security, there is little real motivation or ability for them to put energy into software assurance, although their products may be at risk.
- Embedded systems vendors (e.g., for medical instruments, cash registers). Because these are more likely to run in a constrained environment, and for some categories are more subject to regulation, we did not consider them a useful comparison to our environment.
- Direct competitors to the author's employer. We wouldn't expect cooperation from our competitors, as they might believe that we are gathering information to use against them.

Of course, some companies fit in more than one category. For those, we made an arbitrary decision whether to include them in our survey.

Our emphasis was on medium to large software vendors. We specifically did not seek vendors who are primarily focused on selling security products such as firewalls, IDS, PKI, etc., although some of those vendors are in our sample.

The list of target vendors was selected by reviewing a list of the top 500 software vendors [SWMag][3] and removing those who met one or more of the exclusions listed above. From the remaining list, the author focused initially on those vendors where he knew one or more employees. These employees were usually, but not always, security specialists. In each case, the author asked his contacts for the name of the person or people responsible for software security. In most cases, the author was able to identify an appropriate person, and in most cases, the vendors supplied the information requested in the form of a telephone interview.

Because the author started with those vendors where he had contacts, the list of targeted vendors is somewhat skewed. Most of the author's professional peers are in the security business, and he knows many people in the industry. Thus, if the author does not have any contacts in a vendor, it may be an indication that the vendor does not have a focus on security. To reduce this bias, the author reviewed lists of attendees at security conferences to identify security specialists, and attempted to contact vendors through those security specialists. In some cases, targets were identified through social networks such as LinkedIn. These methods were less successful, as the personal contacts were more willing to be forthcoming than people who did not know the author and therefore, had no reason to trust him.

We specifically excluded Microsoft from this survey, because their security processes are well known and have been described in numerous presentations and books, especially [Howard]. Had we included them, their results would have

[3] This list is admittedly dated, but for purposes of this study was adequate.

shown that they use all of the techniques addressed in this paper, and have numerous motivations for practicing software assurance, most notably the impact on their reputation.

3. Study Results

Our study included eight vendors, which ranged from small (less than $100M in annual sales) to very large (more than $10B in annual sales). Sales volumes were estimated from [SWMag].

Vendors were classified as "security" or "non-security" depending on the predominance of their sales. This distinction was useful because companies perceived as being security vendors have a higher expectation from the marketplace – customers assume that security vendors will be less likely to have security flaws than non-security vendors.

Motivations for security assurance varied significantly, including:

- It's the right thing to do for customers.
- Avoiding being seen as "another Microsoft"[4].
- Fear of the "CNN moment" that affects stock price.
- Loss of sales due to customer concerns.

Additional details of the vendor responses will be in a forthcoming extended paper.

We found significant variation in the processes and motivations of the vendors studied. Not surprisingly, large vendors invest more in software assurance than small vendors, and security vendors put more emphasis on assurance than non-security vendors.

Every vendor asked to remain anonymous, and are therefore represented by letters in the following tables, which summarize our key findings:

Table 1. Techniques used for assurance

Vendor	*Training?*	*Design reviews?*	*Pentesting?*	*Source analysis?*	*Dynamic testing?*
M	Informal	Informal	Internal & external	Manual	Yes
W	Formal & refresher	Not a focus	Internal, external, & customers	Proprietary tools	Yes

[4] This fear of being compared to Microsoft is perhaps misplaced, since Microsoft is arguably in the forefront of securing their products.

Vendor	*Training?*	*Design reviews?*	*Pentesting?*	*Source analysis?*	*Dynamic testing?*
F	Informal & seminars	Performed by developers	Extensive internal, some external	Manual & proprietary tools	Yes
H	Formal	Informal	Internal, external & customers	Company-wide automated	Yes
B	Formal, extensive	Workshop with experts	Internal but discouraged	Company-wide automated	Yes
S	Seminars	Workshop with experts	Field only	Manual, simple tools	Minimal
K	Formal, mandatory	Performed by security expert	Varies by product	Varies by product; some automated	Yes
R	Minimal	Minimal	Not internally, but regular target by hackers	Primary focus	Minimal

Table 2. Motivations for investments

Vendor	*Customer expectations*	*Fear of publicity*	*Explicit requests*
M	Primary	Yes	Minor
W	Primary	Minor	Govt customers only
F	Primary	Yes	Occasional
H	Yes	Primary	Govt customers only
B	Secondary	Minor	Primary
S	Yes	Primary	Minor
K	Primary	Second	Minor
R	Primary	Minor	Govt customers only

From this limited survey, we conclude that:

- Software vendors are aware of the risks of insecure software, and are generally motivated by fear of bad publicity to minimize the security vulnerabilities in their products.
- Few non-government customers explicitly ask for software assurance, but vendors believe that it's an unspoken expectation.
- Most organizations have centralized security organizations that hold the expertise, with outreach into the product development teams to provide software assurance. The head of software security typically reports directly to the head of product development, and has a reasonable degree of influence that allows him/her to prevent product release in case of serious security flaws.
- The techniques used to gain assurance vary among vendors, but nearly all agree that developer training is one of the most valuable uses of limited resources. While everyone agrees that penetration testing has its limitations, it is still helpful as a way to know how good or bad a product is.
- Source code analysis is still early in the acceptance phase, both because tools are expensive and difficult to use effectively. Dynamic testing, including fuzzing, seems to be more cost-effective.
- Common Criteria was mentioned by nearly all vendors, and all but one felt it was a paperwork exercise that had almost no impact on the assurance of their products.
- Most organizations started focusing on software assurance several years ago (perhaps influenced by the famous "Trustworthy Computing" memo [Gates]), and took several years to see results.

Security engineers frequently ask why vendors sell software that has significant security problems. This survey is a step towards answering that question—customers rarely ask about assurance, but despite that, vendors are making significant strides in improving the assurance of their software.

4. Conclusions

Vendors are motivated by customer demand and profit. Thus far, vendors do not see profit in improved software assurance, and explicit customer demand has been minimal. Therefore, they invest primarily because of fear of bad publicity and the notion that assurance is the right thing to do.

Having noted that limitation, vendors are investing in the areas where they perceive the greatest effectiveness: developer training, penetration testing, and

dynamic (black-box) testing, with a smaller level of investment in source code analysis.

Changing the level of investment and types of investment will require a substantial change in customer behavior, by explicitly demanding assurance rather than assuming it's already done.

5. Acknowledgements

The author thanks his contacts in each of the vendors. As each of the vendors provided information about their processes on a non-attribution basis, he regrets that he is unable to thank them by name.

6. References

[CC] *Common Criteria for Information Technology Security Evaluation*, ISO/IEC 15408.

[Epstein] *Software Security and SOA, Danger Will Robinson!,* J. Epstein, S. Matsumoto, and G. McGraw, IEEE Security & Privacy magazine, February 2006.

[Gates] *Trustworthy Computing* (memo), Bill Gates, Microsoft, January 15 2002, http://www.microsoft.com/mscorp/execmail/2002/07-18twc.mspx

[Howard] *The Security Development Lifecycle*, Michael Howard and Steve Lipner, Microsoft Press, 2006.

[McGraw] *Software Security: Building Security In*, Gary McGraw, Addison-Wesley, 2006.

[Safecode] *Software Assurance: An Overview of Current Industry Best Practices*, February 2008, www.safecode.org

[SWMag] *The Complete Searchable 2007 Software 500 Database*, Software Magazine, http://www.softwaremag.com/S_FocusAreas.cfm?Doc=The500

[Viega] *Building Secure Software: How to Avoid Security Problems the Right Way*, John Viega and Gary McGraw, Addison-Wesley, 2001.

Jeremy Epstein has been involved in information security for over 20 years, and is an internationally recognized researcher in the area. He has recently joined Cigital, Inc. after working as Senior Director of Product Security at Software AG, where he was responsible for analyzing and improving the security of all products, designing security for new products, and complying with security standards. Prior to joining Software AG, he led a security research group at Network Associates, and was responsible for the C2 security evaluation of Novell NetWare. He has published over 20 papers in refereed research conferences including USENIX, IEEE, and ACSAC, as well as several articles in trade magazines. In his spare time, Epstein works to improve the security of electronic voting systems as a consultant to several state governments, and serves on technical advisory boards for security startups.

Process and Decision Making Issues

Benchmarking Assurance Practices: Contributions to a Business Case for Assurance

Michele Moss and Nadya Bartol

An increasing number of organizations are committed to addressing software and systems assurance in their products and services. These organizations are leveraging a combination of process improvement approaches to manage the implementation of assurance practices, define key organizational elements required for assurance, and propose ways to leverage measurement to demonstrate the value of initiating and maintaining them.

The first part of this presentation will provide an update on industry efforts to benchmark assurance activities in an organization that is implementing frameworks such as CMMI. The update will include a summary of industry concerns related to benchmarking assurance practices, existing standards, efforts, accomplishments, and organizational practices critical to the integration of assurance into the development of quality products and services. Quantitative and qualitative data resulting from early efforts is critical to maintain momentum and continued funding. Measurement can motivate stakeholders (i.e., the executives, developers, and vendors) to make dramatic changes in the way they perform their jobs. This presentation will provide an update on the Software Assurance Forum efforts to establish a comprehensive framework for software assurance (SwA) and security measurement. The framework addresses measuring achievement of SwA goals and objectives within the context of individual projects, programs, or enterprises and identifies commonalities among the methodologies to help organizations integrate SwA measurement in their overall measurement efforts in a cost-effective and seamless manner. Finally, this presentation will also present emerging ways of quantifying the presence or absence of SwA in software and systems.

About the Speakers

Michele Moss, CISSP, is a security engineer with more than 12 years of experience in process improvement. She has assisted numerous organizations with maturing their information technology, information assurance, project management, and support practices through the use of the capability maturity models including the CMMI, and the SSE-CMM. She specializes in integrating security processes and practices into project lifecycles. Michele is the Co-Chair of the DHS Software Assurance Working Group on Processes & Practices and Practices. Ms.

Moss has also taught classes on the subject of information security process improvement.

Ms. Bartol has over 15 years of information technology (IT) and information assurance (IA) experience, including IT security/IA performance measurement; security process improvement; security policy development; security architecture design; IT security requirements analysis and traceability; IT security configuration documentation development; risk assessments; certification and accreditation (C&A); project management; process analysis; strategic planning; database management; configuration control; and system analysis, design, development, implementation and maintenance.

Making Business-Based Security Investment Decisions—A Dashboard Approach

Julia H. Allen
Carnegie Mellon University, Software Engineering Institute, CERT Program
jha@sei.cmu.edu

This paper presents one approach for selecting security investments using business-based criteria. The approach and supporting tool define seven decision criteria categories, each supported by three or more indicators. Categories and indicators are ranked and applied to a series of investments. Individual investment scores are presented for discussion and evaluation by decision makers. Our intent is that this approach can be use to rationalize and prioritize any class of security investments including software assurance.

Keywords: security investment decisions, business decision criteria, ranking security investments, evaluating security investments, comparing security investments

Introduction

In today's business climate, we are constantly dealing with the demand to do more with less. The resources required to run the business, let alone to invest in new initiatives, are always at a premium—time, money, staff expertise, information, and facilities, not to mention energy and attention span. All investment decisions are about doing what is best for the organization (and its stakeholders). However, what is best is sometimes hard to define, hard to quantify, and even harder to defend when the demand for investment dollars exceeds the supply.

Business leaders are becoming more aware of the need to invest in information and software assurance—to meet compliance requirements and optimize their total cost of ownership for software-intensive applications and systems. So how do we ensure that security investments are subject to the same decision criteria as other business investments? And by so doing, how are we able to justify investments that increase our confidence in our ability to protect digital information using software that is more able to resist, tolerate, and recover from attack?

One approach may begin to shed some light on this topic. It is based on recent CERT research on how to make well-informed security investment decisions using business-based criteria. Over the past four years, CERT has developed a body

of knowledge in enterprise and information security governance, including a detailed framework and implementation guide that describe a robust security governance program.[5] When faced with this framework of tasks, actions, roles and responsibilities, and outcomes, senior leaders say "This is all well and good, but I have many more pressing issues to deal with than security governance. Can you provide me with an aid to select and prioritize these and other security-related actions that I can use as an input to normal planning and capital investment processes?"

This article describes one such approach that is in early demonstration and pilot testing. Organizations that have participated in reviews and initial pilot projects represent the commercial, defense contracting, U.S. federal agency, non-profit, and security vendor sectors. Our intent in presenting it here is to obtain additional feedback about whether it serves as a promising structure and tool for making business-based investment decisions in information and software assurance.

Foundation and Structure

The Security Investment Decision Dashboard (SIDD) provides a means for evaluating and comparing several candidate security investments. A foundational principle of the dashboard is that the priorities for candidate investments are driven by the organization's *desired outcome for any given investment*, not just security investments. This ensures that security investments are subject to the same decision criteria as other business investments. They can then be presented, reviewed, analyzed, debated, and compared using the same scales, factors, and investment-selection criteria and processes.

SIDD describes seven decision criteria *categories*, each supported by three or more decision *indicators*, totaling 33 in all. Two CERT reports [1], [2] served as the starting point for selecting business-based criteria that could be used to evaluate candidate investments. In addition, a number of relevant business and security sources [3]-[7] were analyzed for business-based questions and factors that could help inform security investment decisions. The collected set of questions and factors are reflected in the current set of 33 indicators. The seven categories were derived through affinity grouping of the 33 indicators.

Each category is defined in the form of one or two questions to ask. Categories are presented in shaded text in Table 1 and include Cost, Criticality & Risk, Feasibility, Positive Interdependencies, Involvement, Measurability, and Time & Effort Required. The importance of each category is determined by considering the question "What should *any* candidate investment do for the organization and its stakeholders?" or alternatively, "What is the basis or criteria for selecting *any* candidate investment?"

[5] http://www.cert.org/governance

For example, is it most important that an investment (1) be low cost, (2) be critical to meet business objectives or mitigate a high degree of risk, or (3) be feasible in terms of likelihood of success? Priorities or rankings are then assigned to the category based on the importance of the category to the organization's investment selection process. Each category is further elaborated by three or more indicators that are listed following each category in Table 1. This is a "starter set" that can be tailored to reflect a specific organization's decision factors.

Table 1. SIDD categories and indicators

Category	Description
Cost	What is the estimated total cost to accomplish this investment, taking into account the potential cost savings and/or risk reduction to the organization?
	Overt cost in dollars at outset to accomplish this investment?
	Estimated life cycle cost in dollars over time to sustain this investment?
	Cost of NOT doing this investment, in terms of potential exposure and residual risk (high = investment is more necessary)?
	Potential cost savings to organization beyond breakeven point, if quantifiable (ROI), over time (high = better)?
Criticality & Risk	What is the degree to which this investment contributes to meeting the organization's business objectives and risk management goals?
	Degree to which this investment is key or mainstream in helping the organization meet its primary objectives and critical success factors?
	Degree of risk (as assessed in terms of likelihood and potential impact—high/medium/low priority) mitigated by this investment?
	Degree to which this investment helps the organization protect stakeholders' (shareholders') interests?
Feasibility	How likely is this investment to succeed?
	Likelihood of success on first try?
	Likelihood of success on subsequent tries (if first try fails)?
	Likelihood that turnover among management and/or board of directors will negate work expended on this investment (low likelihood = better)?
	Likelihood that this investment will need to be rolled back (low = better)?
Positive Interdependencies	(1) To what degree does this investment integrate with or represent reasonable changes to existing organizational processes and practices, rather than requiring new ones? (2) To what degree does this investment pave the way for future investments (compliance, policy, risk management, etc.)?
	Degree to which other investments/tasks are dependent on this one (i.e., degree to which this investment makes it easier to accomplish additional tasks)?
	Degree to which the accomplishment of this investment makes it easier to comply with **current** laws and regulations?
	Degree to which the accomplishment of this investment makes it easier to comply with potential **new** laws and regulations in the future?
	Degree to which existing knowledge and/or skills can be used to accomplish this investment, rather than requiring new skills/knowledge?
	Degree to which this investment produces positive side effects (e.g., enhancing brand/reputation, building customer trust, benefiting supply chain partners)?
Involvement	What level of involvement and buy-in are required from various parties for investment success—both within and outside of the organization?
	Level of buy-in required throughout the organization? (Must all employees be on

	board 100% for this to work? Or only a subset, such as management and certain key employees?)
	To what extent does this investment require the active involvement of many departments across the organization?
	Number of people who need to be actively involved?
	Level of involvement by third parties required (partners, consultants, vendors, etc.)?
	Degree of external, independent assessment/auditing (vs. in-house assessment/auditing) required?
Measurability	How measurable is the outcome of this investment?
	Degree to which this investment can be evaluated using existing approaches and reporting mechanisms?
	What is the measurability of the outcome? Can it be quantified in tangible terms (revenue, market share, stock price, etc.)?
	If the outcome is intangible (e.g., goodwill, increased customer trust, enhanced brand), can the benefits be demonstrated against meaningful business success factors?
Time & Effort Required	(1) What level of staff-hours will be required to accomplish this investment? (2) How long will it take to reach break-even cost for this investment?
	Board of directors time required?
	Senior management time required?
	Cross-organizational team/steering committee time required?
	Middle and lower management time required?
	Other key staff time required?
	Time likely needed to achieve the required level of buy-in?
	Time required to achieve first demonstrated results?
	Time required to achieve full adoption and use of the investment results across all affected business units?
	Time to achieve breakeven, if quantifiable?

A business leader may determine that there are other factors or different factors that they use in their investment decision making processes. The SIDD is designed so that categories and indicators can be changed, added, and deleted, and the dashboard will continue to present meaningful comparisons.

Dashboard results are presented in a comparative bar graph form. Score totals are presented for the 7 categories and the 33 indicators for each investment. An additional result is calculated based on the scores for the 6 indicators ranked highest (1-6). This result has been included to accommodate the situation where a subset of indicators is important for investment selection as a companion to the total scores for all categories and for all indicators.

Using the Dashboard

Investment priorities and comparative scores are determined using a two-phased approach. In Phase 1, a decision maker prioritizes categories (Step 1) and indicators (Step 2). The idea here is to determine the importance of each category and each indicator when making *any* organizational investment decision.

These priorities (or rankings) are preserved and applied to all candidate investments during Phase 2.

Phase 2 defines the candidate investments that are to be evaluated (Step 3). There is no upper bound but typically 3-5 investments are evaluated in one use of the dashboard. The decision maker then answers the category and indicator questions (Step 4) for each investment. Scores are calculated by applying the priorities specified in Phase 1 to these answers. Each step is further described below.

1. Phase 1: Establish priorities for all types of organizational investments

   Step 1: Rank categories 1-7 (shaded entries in Table 1) based on their relative importance for any organizational investment decision, 1 being most important and 7 being least important. Do not consider any specific security investment when performing this ranking.

   Step 2: Rank indicators 1-33 (more detailed entries in Table 1), again, based on their relative importance for any organizational investment decision with a ranking of "1" as the most important. Given that 33 indicators is a long list to prioritize, some reviewers grouped these into three sets of ten and then ranked the group of ten. Others created larger scale granularity by assigning a value of, say, 1, 5, or 10 to all 33, which then produced a larger numeric difference between investment scores.

   The current version of the dashboard does not enforce a correlation between category and indicator rankings. This means that one category could be ranked as having the highest priority, while indicators in other categories could be ranked as being more important.

   Steps 1 and 2 are intended to be done once and then applied during all subsequent investment analyses. This helps ensure that results are based on the same ranking and thus can be meaningfully compared. Rankings are periodically reviewed during normal planning cycles or following key events (such as a merger or acquisition) to ensure that they continue to reflect current business priorities. The intent is that these rankings have a fairly long shelf life.

   Some reviewers have suggested that one or more senior C-level leaders perform the category ranking and another group, such as a cross-organizational steering committee, performs the indicator rankings. When category and indicator rankings are done independently, these can then be compared to see if they are consistent or reveal misunderstandings or differences of opinion. In several cases, the shared understanding that resulted from doing these rankings was of equal or greater value than the dashboard results.

2. Phase 2: Evaluate each investment

   For each candidate security investment:

Step 3: Define the investment so that those evaluating it have a common understanding of its scope and intent. While SIDD has been used to evaluate security governance and IT investments (such as policy development, specifying segregation of duties, developing an asset inventory, deploying wireless, creating a new operations center), four example software assurance investments are selected here and further illustrated in Appendix A for the purposes of this workshop.

- A – Integrate architectural risk analysis into the standard SDLC
- B – Integrate secure coding practices into the standard SDLC
- C – Integrate static code analysis into the standard SDLC
- D – Integrate security requirements engineering using SQUARE into the SDLC

Deciding to use SIDD assumes that resources are insufficient to start up all of these now, so we use this approach to help inform which ones to fund.

Step 4: Answer the category and indicator questions (Table 1) for each investment by using a dashboard screen, one per investment. Determining an answer for each question is accomplished by selecting a value from 1 to 5. Based on the question, answers range from very high to very low or very low to very high.

Step 5: Review and discuss the results.

Dashboard outcomes identify the highest priority (highest scoring) investments based on the category rank, the indicator rank, and the answers to the questions for each investment. Given that the category and indicator ranks are fixed (and weighted to normalize the scores[6]), the dashboard results can be meaningfully compared and used to help select which investments to fund, as well as providing a defensible rationale for those that were not selected.

If, based on other factors, these highest scoring investment choices are not justified, this is a valuable opportunity to re-examine the category and indicators rankings and answers to determine if they do indeed reflect how the organization makes investment decisions.

This tool is not intended as a substitute for human judgment. It can be used to make judgments more explicit, to apply a consistent set of decision criteria to all investments which can then be communicated, and to capture trends over time.

Current Status

The SIDD review process started in September 2007 and is ongoing as of the date of this article. The current version of the tool executes as a series of Excel

[6] Category and indicator ranks are converted into weights that are used as multipliers to normalize dashboard scores. This is necessary due to a priority of "1" being highest, yet the highest total score reflects the highest priority investment.

spreadsheets. Comments have been received from eight organizations representing large commercial, large defense contracting, not-for-profit, U.S. federal civilian agency, and security consulting/products and services sectors.

Development is in progress to present the tool as a standalone application. We expect to have this improved version of the tool available as a demonstration by the September workshop.

References

[1] Allen, Julia. *Governing for Enterprise Security* (CMU/SEI-2005-TN-023). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, June 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tn023.html.

[2] Westby, Jody R. & Allen, Julia H. *Governing for Enterprise Security (GES) Implementation Guide* (CMU/SEI-2007-TN-020). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, August 2007. http://www.sei.cmu.edu/publications/documents/07.reports/07tn020.html.

[3] Campbell, George K. "Measures and Metrics in Corporate Security: Communicating Business Value." CSO Executive Council, 2006. https://www.csoexecutivecouncil.com/content/Metrics_Mini_Update_060706.pdf.

[4] Corporate Information Security Working Group. Adam H. Putnam, Chairman; Subcommittee on Technology, Information Policy, Intergovernmental Relations & the Census Government Reform Committee, U.S. House of Representatives. "Report of the Best Practices and Metrics Teams." November 17, 2004; updated January 10, 2005. http://www.educause.edu/ir/library/pdf/CSD3661.pdf

[5] Drugescu, Cezar. "Maximizing the Return on Investment of Information Security Programs: Program Governance and Metrics." Information Systems Security Journal, Taylor & Francis, December 2006.

[6] "ISO/IEC 27001 & 27002 implementation guidance and metrics, Version 0.7" ISO 27001 Security Implementers' Forum, 5 June 2007.

[7] Kleinfeld, Abe. "Measuring Security." Information Systems Security Journal, Taylor & Francis, November 2006.

Appendix A

This appendix contains the following three sections:

- A.1 Category and Indicator Rankings
- A.2 Scores for One Investment in One Category
- A.3 Summary Results for Four Investments

Please note that larger versions of the tables and charts in this appendix are available in a version of this paper in the Governance & Management content area on Build Security In (https://buildsecurityin.us-cert.gov/).

A.1 Category and Indicator Rankings

In this example, the "Criticality and Risk" category is ranked as "1" and is the most important category for any organizational investment decision. "Measurability" is ranked as "7" and is thus the least important category-level criteria.

The indicator that has the highest priority here is the "Cost of NOT doing this investment, in terms of potential exposure and residual risk." It is ranked as "1" and is the most important indicator for any organizational investment decision. As you might expect, the three indicators under "Measurability" are the least important indicators.

	Category / Indicator	Cat Rank (1-7)	Ind Rank (1-33)
Cat	**Cost**	2	
Consider	What is the estimated total cost in dollars of accomplishing this investment, taking into account the potential cost savings and/or risk reduction to the organization?		
Indicators	Overt cost in dollars at outset to accomplish this investment?		6
	Estimated life cycle cost in dollars over time to sustain this investment?		7
	Cost of NOT doing this investment, in terms of potential exposure and residual risk (high = investment is more necessary)?		1
	Potential cost savings to organization beyond breakeven point, if quantifiable (ROI), over time? (high = better)		9
Cat	**Criticality and Risk**	1	
Consider	What is the degree to which this investment contributes to meeting the organization's business objectives and risk management goals?		
Indicators	Degree to which this investment is key or mainstream in helping the organization meet its primary objectives and critical success factors?		4
	Degree of risk (as assessed in terms of likelihood and potential impact -- high/medium/low priority) mitigated by this investment?		3
	Degree to which this investment helps the organization protect stakeholders' (shareholders) interests?		5
Cat	**Feasibility**	3	
Consider	How likely is this investment to succeed?		
Indicators	Likelihood of success on first try?		10
	Likelihood of success on subsequent tries (if first try fails)?		11
	Likelihood that turnover among management and/or board of directors will negate work expended on this investment (low likelihood = better)?		20
	Likelihood that this investment will need to be rolled back (low = better)?		16

Cat	**Positive Interdependencies**	6	
Consider	(1) To what degree does this investment integrate with or represent reasonable changes to existing organizational processes and practices, rather than requiring new ones?		
	(2) To what degree does this investment pave the way for future investments (compliance, policy, risk management, etc.)?		
Indicators	Degree to which other investments/tasks are dependent on this one (i.e., degree to which this investment makes it easier to accomplish additional tasks)?		28
	Degree to which the accomplishment of this investment makes it easier for the organization to comply with **current** laws and regulations?		2
	Degree to which the accomplishment of this investment makes it easier for the organization to comply with potential **new** laws and regulations in the future?		29
	Degree to which existing knowledge and/or skills can be used to accomplish this investment, rather than requiring new skills/knowledge?		25
	Degree to which this investment produces positive side effects (e.g., enhancing brand/reputation, building customer trust, benefiting supply chain partners)?		27
Cat	**Involvement**	5	
Consider	What level of involvement and buy-in are required from various parties for investment success -- both within and outside of the organization?		
Indicators	Level of buy-in required throughout the organization? (Must all employees be on board 100% for this to work? Or only a subset, such as management and certain key employees?)		12
	To what extent does this investment require the active involvement of many departments across the organization?		21
	Number of people who need to be actively involved?		24
	Level of involvement by third parties required (partners, consultants, vendors, etc.)?		13
	Degree of external, independent assessment/auditing (vs. in-house assessment/auditing) required?		30
Cat	**Measurability**	7	
Consider	How measurable is the outcome of this investment?		
Indicators	Degree to which this investment can be evaluated using existing approaches and reporting mechanisms?		32
	What is the measurability of the outcome? Can it be quantified in tangible terms (revenue, market share, stock price, etc.)?		31
	If the outcome is intangible (e.g., goodwill, increased customer trust, enhanced brand), can the benefits be demonstrated against meaningful business success factors?		33
Cat	**Time and effort required**	4	
Consider	(1) What level of staff-hours will be required to accomplish this investment?		

	(2) How long will it take to reach break-even cost for this investment?		
Indicators	Board of directors time required?		17
	Senior management time required?		18
	Cross-organizational team/steering committee time required?		19
	Middle and lower management time required?		23
	Other key staff time required?		22
	Time likely needed to achieve the required level of buy-in?		14
	Time required to achieve first demonstrated results of task?		26
	Time required to achieve full adoption and use of investment results across all affected business units?		15
	Time to achieve breakeven, if quantifiable?		8

A.2 Scores for One Investment in One Category

In this example and for the investment being considered, the answer to the Cost category question "What is the estimated total cost of accomplishing this investment . . ." is low. So the word "low" is replaced by the number "4" (indicated at the top of the column) to allow for a numeric calculation. Given the weighting factors that are applied based on the category rank, the score for the Cost category is calculated to be "4." This score is added to the other six category scores to arrive at the CAT TOTAL score that appears in Appendix A.3.

The indicators are assigned the following values and corresponding scores:

- Overt cost at outset is medium, so the word "med" is replaced by the number "3" and a resulting score of "3" is calculated based on the indicator rank.
- Estimated life cycle cost is very low, so the word "v low" is replaced by the number "5" and a resulting score of "3.75" is calculated.
- Cost of NOT doing this investment is high, so the word "high" is replaced by the number "4" and a resulting score of "4" is calculated.
- Potential cost savings is high, so the word "high" is replaced by the number "4" and resulting score of "3" is calculated.

These indicator scores are added to the other 29 indicator scores for this investment to produce the IND TOTAL score that appears in Appendix A.3.

The red, orange, yellow, light green, and green colors and text are intended to serve as visual cues. Questions that have category and indicator answers that tend to the red end of the spectrum will likely result in a "don't do" this investment decision. Questions that have category and indicator answers that tend to the green end of the spectrum will likely result in a "do" this investment decision.

	Category / Indicator	Cat Rank (1-7)	Ind Rank (1-33)	1	2	3	4	5	MULT	SCORE
				don't do	unlikely	later?	soon	do		
Cat	Cost	2		v high	high	med	4	v low	4	4
Consider	What is the estimated total cost in dollars of accomplishing this investment, taking into account the potential cost savings and/or risk reduction to the organization?									
Indicators	Overt cost in dollars at outset to accomplish this investment?		6	v high	high	3	low	v low	3	3
	Estimated life cycle cost in dollars over time to sustain this investment?		7	v high	high	med	low	5	5	3.75
	Cost of NOT doing this investment, in terms of potential exposure and residual risk (high = investment is more necessary)?		1	v low	low	med	4	v high	4	4
	Potential cost savings to organization beyond breakeven point, if quantifiable (ROI), over time? (high = better)		9	v low	low	med	4	v high	4	3

A.3 Summary Results for Four Investments

Summary results are calculated as follows:

- CAT TOTAL: the numeric sum of the scores for all 7 categories
- IND TOTAL: the numeric sum of the scores for all 33 indicators
- TOP 6 TOTAL: the numeric sum of the scores for the 6 highest priority indicators. This provides an alternative view in the event that 6 specific indicators are of equal or greater relevance to the investment decision.

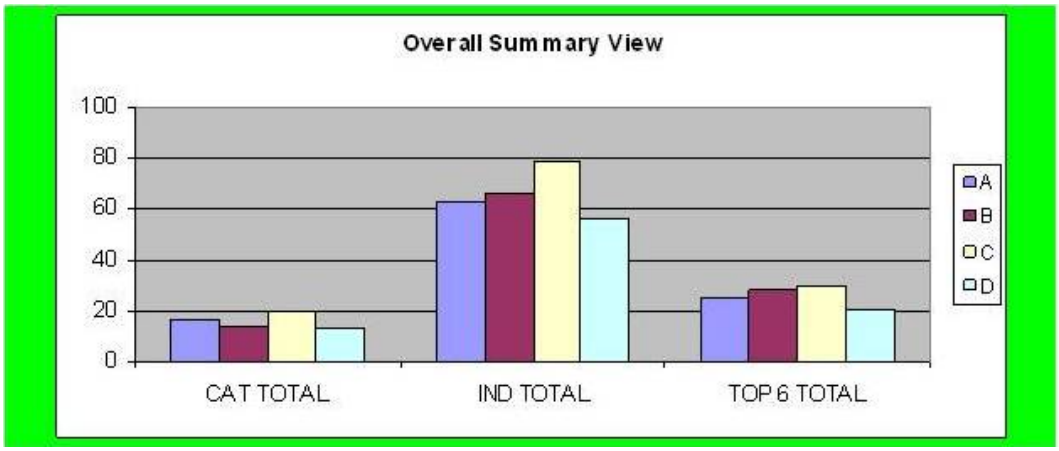
The "Overall Summary View" provides a bar chart comparison of CAT TOTAL, IND TOTAL, and TOP 6 TOTAL. The elements of the Summary View are then displayed individually in the following Summary displays.

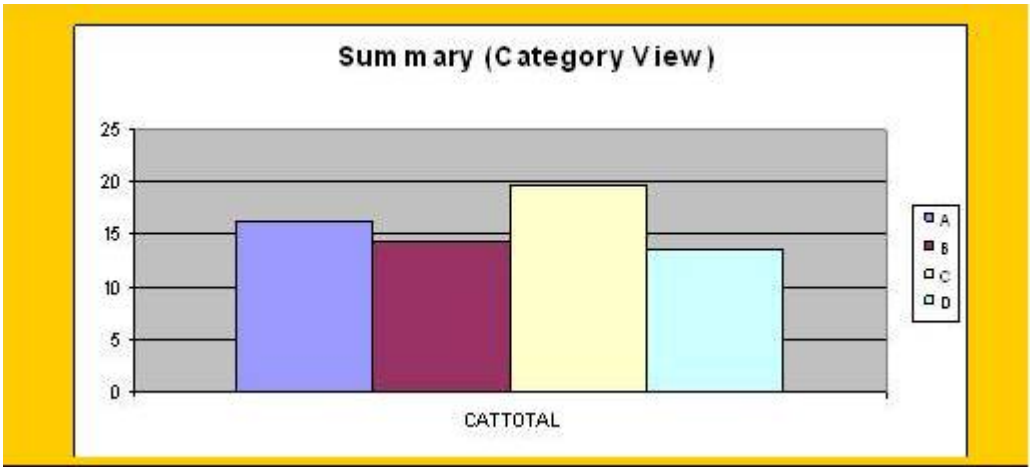
In this particular example, Investment C: *Integrate static code analysis into the standard SDLC* has the highest score (sum of CAT TOTAL and IND TOTAL; confirmed by TOP 6 TOTAL) so should be considered as the first software assurance investment to fund. It is closely followed by Investment B: *Integrate secure coding practices into the standard SDLC*, which should be funded next assuming funds are available. Investment A: *Integrate architectural risk analysis into the standard SDLC* and Investment D: *Integrate security requirements engineering using SQUARE into the SDLC* are next in line respectively, subject to available resources.

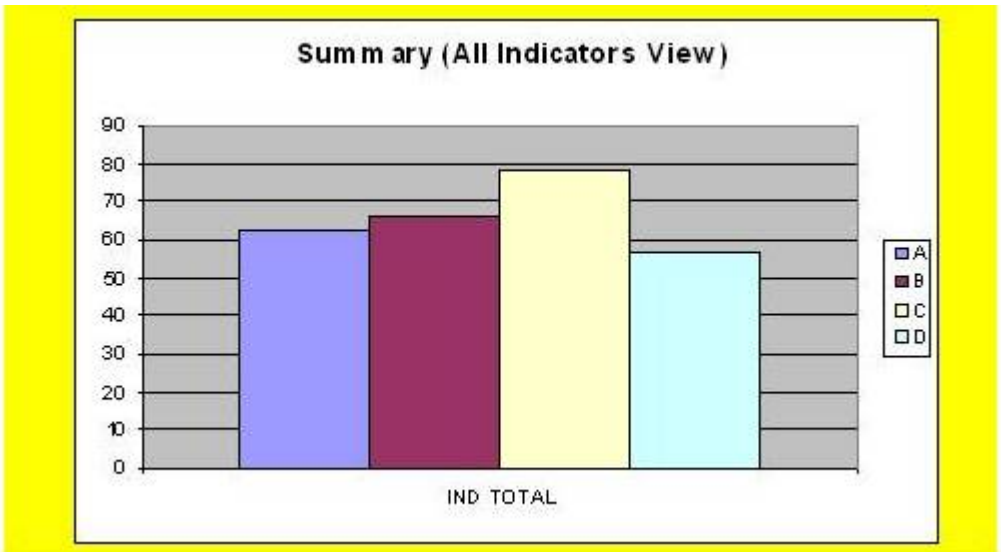
DASHBOARD SAMPLE

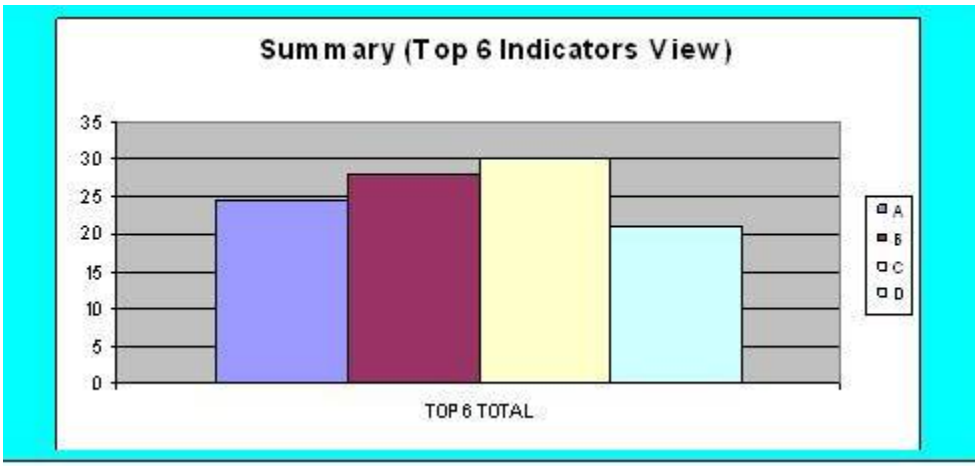
	Candidate Investments					
	A	B	C	D	etc.	
CAT TOTAL	16.25	14.25	19.75	13.5		
IND TOTAL	62.5	66.25	78.75	56.5		
TOP 6 TOTAL	24.75	28	30	21		

Project	Description
A	Integrate architectural risk analysis into the standard SDLC
B	Integrate secure coding practices into the standard SDLC
C	Integrate static code analysis into the standard SDLC
D	Integrate security requirements engineering using SQUARE into the SDLC


Overall Summary View
100
80
60
40
20
0
CAT TOTAL
IND TOTAL
TOP 6 TOTAL
A
B
C
D


Summary (Category View)
25
20
15
10
5
0
CATTOTAL
A
B
C
D


Summary (All Indicators View)
90
80
70
60
50
40
30
20
10
0
IND TOTAL
A
B
C
D



Julia Allen is a senior researcher with the CERT® Program, Software Engineering Institute, Carnegie Mellon University. Allen conducts research in enterprise security governance and software assurance. Prior to this assignment, Allen served as acting Director of the SEI for six months as well as Deputy Director/Chief Operating Officer for three years.

In addition to her work in security governance,[7] Allen is the author of *The CERT Guide to System and Network Security Practices* (Addison-Wesley, June 2001) and the CERT Podcast Series: Security for Business Leaders (2006-2008).[8] She is co-author of *Software Security Engineering: A Guide for Project Managers* (Addison-Wesley, May 2008).[9]

[7] http://www.cert.org/governance
[8] http://www.cert.org/podcast
[9] http://www.sei.cmu.edu/publications/books/cert/software-security-engineering.html

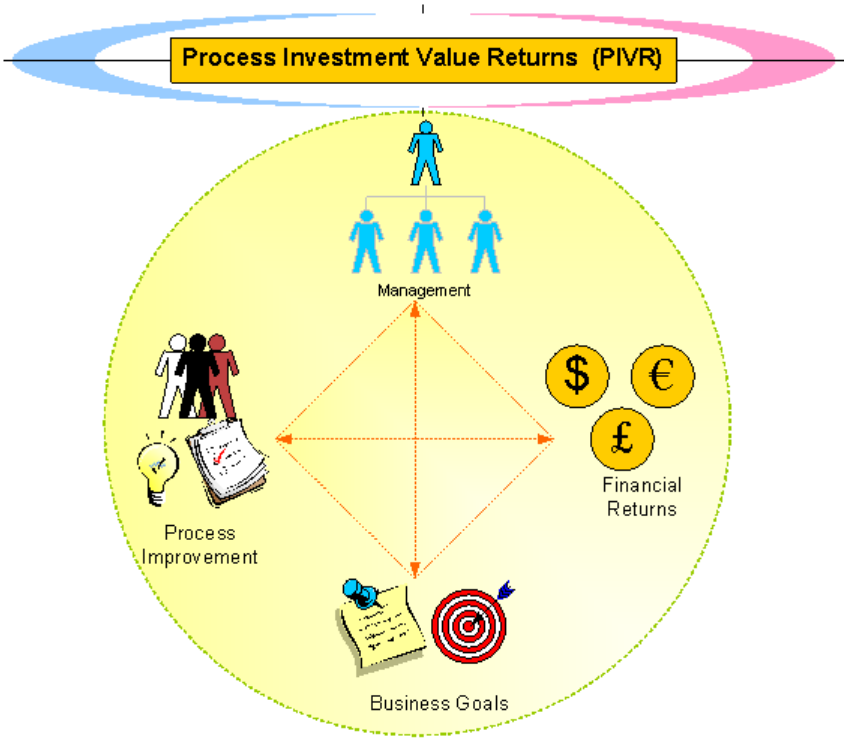
Process Investment Value Returns (PIVR) Framework for Measuring Returns on Process Improvement

Balaji Santhanam
WIPRO Consulting Services
balaji.santhanam@wipro.com

Software assurance activities yield the required results when they are looked upon as set of processes. A process oriented approach puts in place a clear mechanism to plan, monitor, and improve the entire gamut of software assurance activities.

In this context, an ROI framework titled Process Investment Value Returns (PIVR) could be used as a strategic measurement tool. 'Value Point' is proposed as a simple approach for measuring indirect returns, thus addressing the challenges involved in measurement of indirect returns.

In PIVR, returns are measured directly through monetary value and indirectly in terms of Value Point.



1. Introduction

As information technology revolutionized the global economy, sharing of knowledge and best practices across organizations gained prominence .The role of internationally accepted models and frameworks in promoting such knowledge exchanges has been significant. They have paved the way for organizations in setting up a common platform to share and learn from the rest. On the flip side, this enablement has led to proliferation of models and frameworks leaving the senior management in a spot of bother.

For any process improvement initiative, selection of right process methodologies and tools is a key decision that needs to be taken upfront .Several factors like: organization objectives, business model, culture, financial support, workforce competencies highly influence such decisions. Thus it is evident that, when huge investments are made in such initiatives, the returns need to justify the investment.

Return on Investment (ROI) is an often heard buzzword every CEO/CIO would like to talk about day in day out. It highlights the enormous responsibility and accountability entrusted with executives in all investment decisions. It's also quite common to observe that the level of reasoning and analysis, assumes significant proportions when the investment decisions are related to process improvements.

2. Complicating Simplicity

Process improvement initiatives can never be brushed aside as pet project of process assurance group, as it cuts through the length and breadth of an organization. Such initiatives when approached in right sense and direction has yielded compelling business benefits to organizations. Managing such initiatives in certain phases is more an art than science.

"Not everything that can be counted counts, and not everything that counts can be counted," stated Albert Einstein. It stands as an undisputed fact as in today's business where the executive's mantra is to see substantial benefits in every dollar spent.

When measuring these benefits / return on investment, it becomes essential to quantify them in financial terms. However, with certain metrics it is difficult not only to quantify financially but also to measure objectively. This acts as impediment for the advocates of process improvements, from building a strong business case to convince the organization's top brass. In the following paras, a framework to address such challenges is described.

3. Software Assurance: A Necessity

The primary objective of any software oriented organization is to build products or applications that meet the customer requirements. Traditionally, this has been achieved by a well defined process approach that is best described as Software Development Lifecycle.

Requirements, Design, Coding, and Testing are the key processes that comprise the software development lifecycle .Lifecycle models like Waterfall, Modified waterfall, Spiral, Iterative etc prescribe approaches to build software.

In the entire SDLC , Quality Assurance is an critical aspect that cuts across all phases from project initiation to release aimed in ensuring that the product that is been built has adhered to the defined processes and meets the intended requirements.

However over the years, the complexities and need for secure software has grown the discipline of Software Assurance.

4. Process Approach to Software Assurance

Software Assurance is set of activities primarily aimed at delivering products that are free from vulnerabilities and also meets the intended requirements.

When customer expects a final product it is implied that all requirements are met. Addressing, security related goes beyond from just having well defined business cases and user scenarios.

This is where, software assurance positions itself to become a key process as part of an organization's development methodology.

As in a typical process, software assurance is a collection of inputs, tasks and outputs, when well integrated to the core development process fulfills specific objectives towards developing a quality product.

Like any other initiative deployment of software assurance process is taken up at two levels:

- Organization level – focused towards policies, process, awareness and training
- Department /Division/project Level – focused towards implementation of processes

As it could be seen, measurements are ingrained at all critical stages of process to enable analysis of investment and returns at both organizational level and project level.

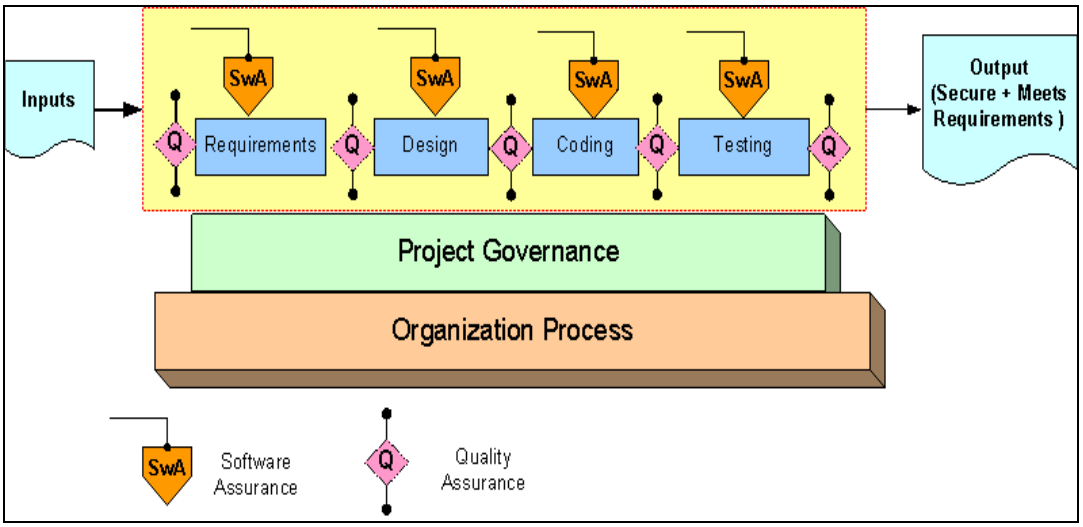


Figure 1. SwA in core delivery

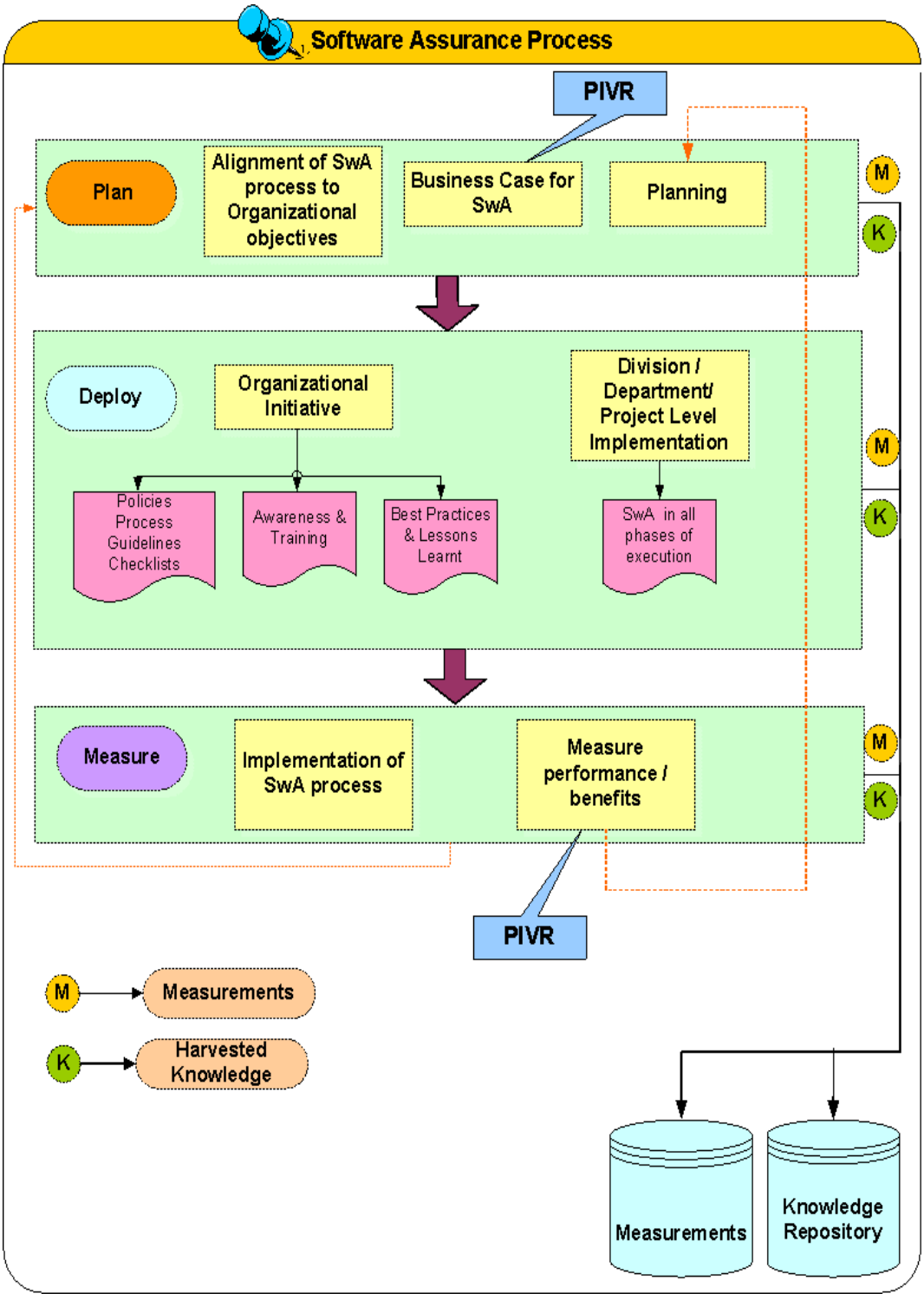


Figure 2. Software Assurance process flow

5. Measurement Paradox in Software Assurance

Building a business case for software assurance before organization wide process roll out is one of the challenging task for the managers.

Every business case defines a set of direct/tangible metrics and indirect /intangible metrics .In case of software assurance challenges in measurement is more pronounced. The returns on software assurance initiatives are measured through set of metrics that are identified and aligned with organization's business objectives. Some of the measures are direct and can be easily measured and financially quantified, while many do not fulfill these criteria.

Measure - Quantify matrix (M-Q matrix), describes the measurement paradox while setting up a business case.

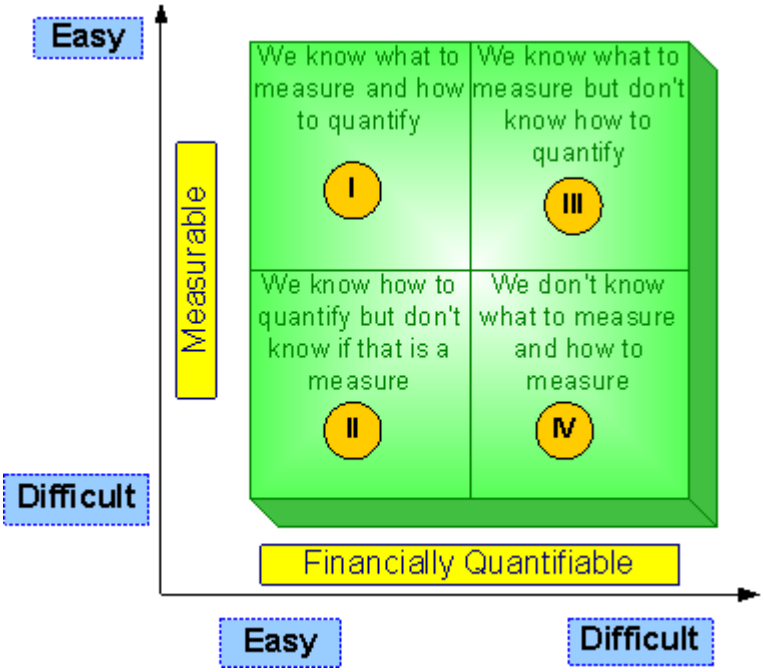


Figure 3. M-Q Matrix

It could be inferred that in majority of cases, the measures that falls in Quadrant –II, III and IV are generally overlooked. In other words 60-75 % of measures that is related to software assurance are not measured.

The root cause of this paradox is, the high level of importance attached to financially quantifiable measures. But conventional wisdom would state that, not all measures can be financially quantified.

Looking at sample metrics for an organization, M-Q matrix could be depicted as:

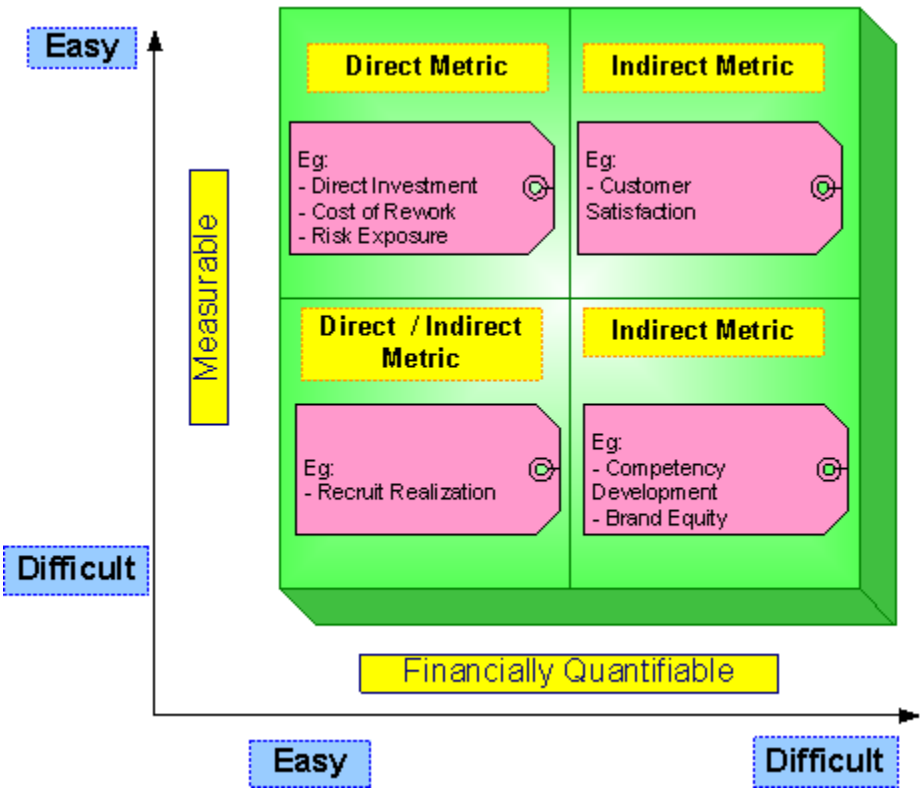


Figure 4. M-Q Matrix example

While measuring the returns on software assurance, metrics could be classified as 'Direct 'metrics and 'Indirect 'metrics. Indirect metrics are those which cannot be either easily quantified financially or measured objectively. They fill in the III and IV quadrants in M-Q matrix and could also figure in Quadrant II. Direct Metrics as name indicates facilitate direct measurement and quantification in financial terms.

Given the scenario, it builds up certain constraints in measuring returns on software assurance activities

Table 1. Constraint – Impact

#	Constraint	Impact
1	Not all measures can be financially quantified	Cannot compare 'Apples to Apples' Comparing 'Apples to Oranges' would dilute the actual measurement
2	Not all returns can be easily measured	Genuine improvements /returns gained may get missed out

6. Tip of Iceberg Effect

Just with looking at the direct benefits as means to measure ROI leaves the organization with a situation akin to ' Tip of Iceberg Effect '.

The pitfalls in such scenario:

- Indirect benefits due to the fallacy of ROI measurement are overlooked
- Most of the benefits are visible only through indirect measures

7. Process – Investment-Value-Returns framework

To address the constraints and challenges, Process Investment Value Returns (PIVR) is a proposed framework for measuring ROI on process improvements.

PIVR framework is built on six basic rules:

1. All metrics cannot be easily measured. All metrics cannot be financially qualified.
2. While the value of direct benefits is undisputed, the value of indirect benefits cannot and must not be negated
3. Difficulty in measuring or financial quantification should not be criterion for disqualifying a metric when the business value it creates is evident
4. When identifying metrics, look for a logical cause and effect relationship!
5. Value Point is used as logical substitute to measure the indirect metrics
6. ROI is analyzed by comparing the direct returns in terms of cost and indirect returns in terms of Value Point

Benefits of PIVR:

- Simple way for measuring direct and indirect metrics
- Reduces complexity involved in conversion of non financial measures to financial measures
- Involvement of key stakeholders in agreeing upon the measures
- Comprehensive coverage of all benefits gained

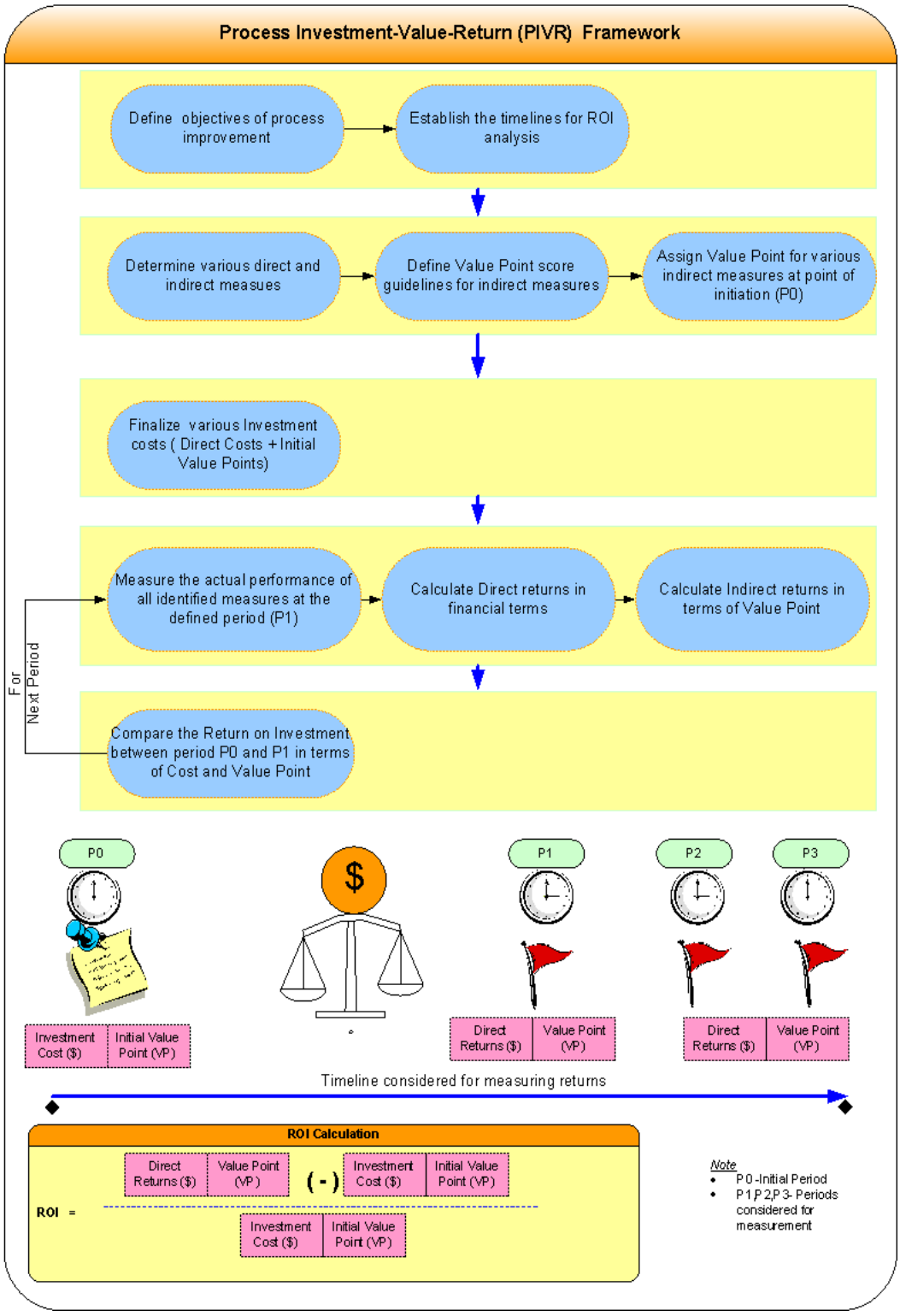


Figure 5. PIVR Framework

8. Value Point as measure for indirect metrics

One approach to break such impasse revolving measurement of indirect metrics is through usage of 'Value Point'.

What is a Value Point?

1. Value Point is a unitless notation of measurement system to denote:
    - business value of metrics that cannot be financially quantified
    - metrics that cannot be measured through conventional units of measurement system
2. Value Point could also be used as conversion factor for converting measurement value of indirect measures to common unit for analysis.
3. Value Point scores are additive in nature i.e. Value Point scores of two metrics can be added

Where Value Point is used?

Let's consider certain indirect measures and see the usage of Value Point.

(a) where the business value of metrics that cannot be financially quantified

For example, Customer Satisfaction Index is a critical measurement to gauge the acceptance of a quality product. Generally; organizations administer a survey through a form and measure customer satisfaction in a rating scale of 1-5 or 1-10.

In such cases, the business value such metrics indicate can be indicated through Value Point, by a conversion scale.

For example:

Table 2. CSI rating

Customer Satisfaction Index (CSI)	Lower Rating	Higher Rating
Customer Satisfaction Measurement Rating	1	5
Value Point	1000	3000

Here when customer rates organization performance, the ratings are given in scale of 1 to 5, 1 being the lowest and 5 being the highest.

Based on the importance the organizations assign for Customer Satisfaction Index (CSI), the Value Point score for lower and upper limits are decided. In this case, CSI of 1 equals 1000 value points and CSI of 5 are awarded 3000 points.

Scoring ranges for converting measurement value to Value Point score is defined at the beginning for all indirect metric. The range values can be revisited at end of every analysis period.

9. How to assign Value Point to different indirect measures?

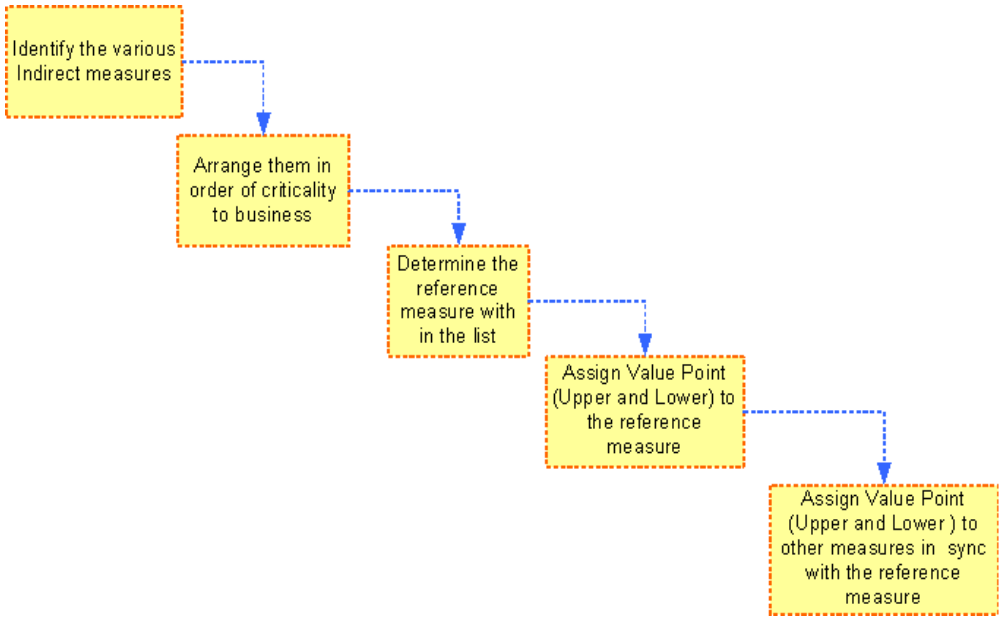


Figure 6. Value point for indirect measures

Identify the various indirect measures:

The indirect measures are decided based on the value that each of the measure signifies to business.

Though there might be several indirect measures care needs to be taken:

- to ensure that the measures considered , provide returns on investment in significant form
- any ‘ double count ‘ is avoided during calculation (e.g., measuring rework efforts directly and again converting defect rate to rework efforts and counting them separately attributes to ‘double count’ of rework)

Arrange them in order of criticality to business

The management group decides the weightage that needs to be given to each measure. This is based on the value each measure could signify to business. For example, for an organization brand equity and obtaining compliance certifications might be of more importance than internal process compliance scores.

Determine the reference measure in the list

From the various indirect measures listed, one measure is identified as reference point. This could be one in the mid zone of measures ranked based on business criticality.

Assign Value Point (Upper and Lower) to the reference measure

Value Point is assigned to the reference measure. Upper limit and Lower limit scores are set.

Upper limit Value Point score is the maximum score that can be awarded to a metric if it performs as expected. Lower limit Value Point score is the minimum score that can be awarded to a metric if performs far below expectations.

Assign Value Point (Upper and Lower) to other measures in sync with the reference measure

Based on the Value Point score given to the reference metric, guideline scores are set up for all other metrics.

Essential aspects in Value Point scoring

- Involvement of senior management with key stakeholders in defining and awarding the Value Point score
- Consensus among stakeholders on the Upper and Lower Value Point scores
- Clear guidelines to award the actual scores with reference to the guideline values
- Signing off the Value Point guideline scores by the Senior Management during the initial phases of process improvement initiative

10. ROI Measurement through PIVR

Case Study

For clarity of understanding, let us consider a hypothetical business case of Resonance Info systems.

Background:

IT Service provider involved in Application, Development and Maintenance of applications (ADM) in Banking, Financial Services and Insurance domain

Clients:

Mid size and large organizations across the globe

Current challenges:

1. Customer complaints on quality of products delivered primarily attributed to application security features (Reported by three customers)
2. Security breaches reported by one customer due to certain vulnerabilities present in final application
3. Non compliance to security framework/compliance leading to disqualification from participating in three big tenders
4. Discomfort amongst employees due to customer complaint
5. Loss in brand equity

Analysis:

Analysis of the issues, had spelt out the following shortcomings:

- Absence of clear processes for software assurance in software development
- Lack of awareness amongst staff on security related aspects
- No mechanism to quantify the risks related to security matters
- Losing competitive advantage due to lack of compliance certifications

Business Case:

Delivery Manager of Resonance, is entrusted with the job to present a business case to management emphasizing the need to implement software assurance processes as part of the core delivery processes.

Delivery Manager draws up an alignment of the business objectives with the need to deploy software assurance process. He would like to project a ROI over a period of three years.

11. PIVR: ROI Measurement approach

I. Define objectives of process improvement

The first and foremost step in any process improvement initiative is to clearly identify the objectives and alignment to business goals. This gives the direction for organization to channelize their efforts in right places.

Tools like Quality Function Deployment, Goal Question Metric could be used to define the objectives and establish the alignment

II. Establish the timelines for ROI analysis

The time frame for analyzing the returns is determined based on the business environment in which organizations are operating on.

P0- Initial period where the Investment is made

P1, P2, P3 – Periods in which analysis is performed e.g.: could be one year periodicity

III. Determine various direct and indirect measures

Various direct and indirect measures to be considered are identified and listed

IV. Define Value Point score guidelines for indirect measures

Value Point guidelines are decided by senior management along with other stakeholders for all indirect measures

V. Assign Value Point for various indirect measures at point of initiation (P0)

Based on the guidelines set, initial Value Point score is assigned to indirect measures

As the next logical step, the direct and indirect metrics are identified.

Adopting the Value Point approach, the guideline values for the indirect metrics are assigned.

Indirect Metrics					
#	Metric	Business Impact Rating	Lower Value Point	Upper Value Point	P0 (Value Point)
1	Customer Satisfaction Index	1	1000	3000	2000
2	Time to Market Improvement	2	1000	2000	1500
3	Lead qualifications enablement	3	0	1500	175
4	Process Compliance Score	4	400	750	0
5	Compliance Certification	5	250	400	250
6	Technical Competency	6	150	250	450
7	Recruitment Realization Score	7	0	150	0
	Total				4375

Figure 7. Indirect metrics

The values assigned under P0 are the Value Point scores awarded at the beginning of the process roll out.

The Value Point scores are given based on the measurement value of every metric. It is important to note that, there is an underlying measurement before the Value Point scores are given. Value Point only converts those diverse measurement units to a common scale for facilitating comparison.

For example:

#	Metric	Description	UOM	Organization Goal	LVP	UVP
1	Customer Satisfaction Index	Measured through Customer Satisfaction Survey	Index	4	1000	3000

Range	VP Score	Range	VP Score	Range	VP Score	Range	Range	VP Score
CSI < 3	1000	CSI between 3.1 -4	1500	CSI between 3.6 and 4	2000	CSI between 4 and 4.5	CSI > 4.5	3000

Figure 8. Value point guidelines

VI. Finalize various Investment costs (Direct Costs + Initial Value Points)

Investment costs are attributed in financial numbers. This along with the initial score of Value Point becomes the total investment for the process improvement initiative.

Investment Costs		
#	Activity	Cost ($)
1	Staff Salary involved in deploying software assurance activities	
	Full time resources	54000
	Part time resources (Efforts(in hrs) * Cost / hr	9000
2	Consultant Cost	50000
3	Training	8000
4	Cost towards certification / compliance checks	15000
5	Hardware cost	4000
6	Software cost	4000
7	Rewards to people for process improvement	1800
8	Process Maintenance & Overheads Cost	2500
	Total	148300

Figure 9. Investment costs

VII. Measure the actual performance of all identified measures at the define period (P1)

At the assigned period (P1) actual measures are collected and analyzed for both direct and indirect measures.

VIII. Calculate direct returns in financial terms

Direct measures are converted to financial terms as defined by organization guidelines.

IX. Calculate indirect returns in terms of Value Point

Indirect returns are measured in terms of Value Point in accordance with the scoring guideline established by the organization.

X. Compare the Return on Investment between period P0 and P1 in terms of Cost and Value Point

The process is repeated at the defined periods.

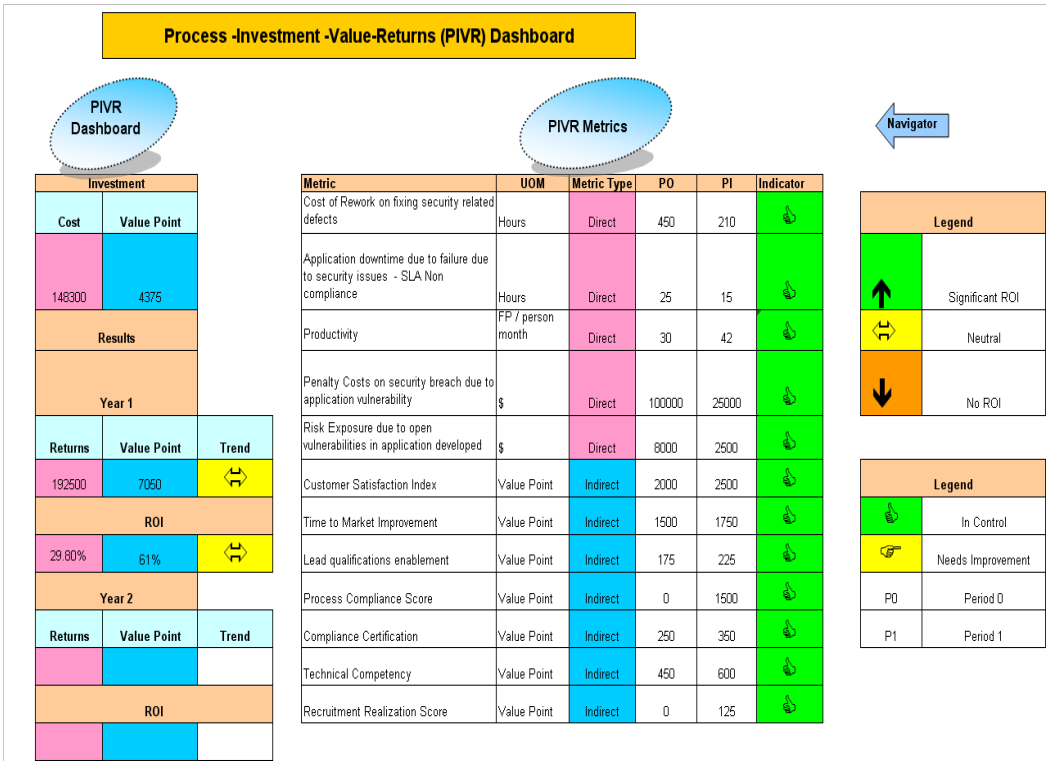


Figure 10. PIVR dashboard (PIVR tool)

12. Conclusion

Many organizations today are successful in their initial implementation of processes but over a period of time, sustaining the momentum and maintaining process compliance on software assurance activities becomes a herculean task.

Like any other product launched in market, processes too end up following a typical product life cycle, with initial success, maturity and end of life. This primarily happens to organization's failure to clearly demonstrate the true benefits and showcase them in the right way.

PIVR framework tries to address these finer aspects by helping organizations to periodically measure returns on process improvement initiatives .This enables senior management to focus on these initiatives with renewed energy and steward them for better business results.

Notes:

1. In all the discussions, the word metrics and measures are used interchangeably and primarily intended to convey the concept of measuring performance of an activity.
2. Metrics considered as Direct and Indirect may differ from organization to organization and discussed in the paper are more from generic point of view
3. Organizations may have specific measurement system for all type of metrics in which cases ,usage of Value Point need to be defined in Value Point guideline accordingly
4. PIVR framework conceptualized by the author has been piloted in few organizations. PIVR tool to compliment PIVR framework, the screen shots of the tool has been used as illustration to better explain the concepts involved.

References

Management Accounting, ICFAI Publications

Planning Poker Estimation, http://en.wikipedia.org/wiki/Planning_poker

The ROI from Software Quality, Khaled El Emam

Calculating CMMI based ROI-Why, When, What and How, Rolf W.Reitzig, Dennis R. Goldenson, Diane Gibson, Mark .R. Cavanaugh

Balaji OS is a management graduate with an engineering background having rich experience in areas of process improvements in IT and non-IT business domains. He has been involved in definition and deployment of processes, implementing process management systems in sync with standards and improvement frameworks (ISO, BS7799, CMMI, Balanced Score Card) in areas of software process improvement, security governance, and people practices. He has consulted organizations in their process improvement initiatives. He also contributes actively to the community through his blog and publications. He works as Senior Consultant in WIPRO Consulting Services, India.

Legal Issues

Prospects for Preserving Software Investment via Patenting

Dennis Carleton, Partner, Intellectual Property Group, Fox Rothschild LLP

Investments made in quality assurance in software development can be recaptured through patenting and the commercialization and licensing of the software. However, recent legal developments bring into question the ability to protect software with patents in the U.S. This talk will discuss recent developments and cases that will have an impact in the future on the ability to protect software and the extent of protection that may or may not be available to developers of software. In particular, will the US move toward a European model where software in and of itself is incapable of receiving patent protection, or will the US patent system become more liberal, allowing protection for software and business methods not currently eligible for patent protection? In particular, the case of In re Bilski, a case currently pending before the Federal Circuit Court, has the potential to re-define the parameters of patentable subject matter with respect to business methods and software. The Bilski case, a patent application claiming a method of managing the consumption risks of a commodity, was twice rejected by the patent office as being based on unpatentable subject matter, and was appealed to the Federal Circuit Court. The case is so controversial that the Federal Circuit Court required not only the normal initial hearing but a rare en banc hearing, where all 12 judges of the Federal Circuit Court listen to arguments. The details of the case, and its potential effect of the patentability of software systems, will be explored during the discussion.

About the Speaker

Dennis Carleton is a partner in the law firm of Fox Rothschild LLP, concentrating his practice in intellectual property law, and, in particular, software and business method patents. Dennis is a graduate of Carnegie Mellon University, having received a BS degree in Electrical Engineering in 1983 and a Masters of Software Engineering degree in 1990 and was a software engineer prior to graduating from the University of Pittsburgh School of Law in 1996.

Globalization Issues

Business Impact of and on Software Assurance of the Global Outsourcing of Software Development, Testing, and Use

Warren Axelrod, Research Director for Financial Services, United States Cyber Consequences Unit

There are well-recognized risks inherent in the mere outsourcing of the design, development, assurance, implementation, and post-implementation support of software. However, the increased use of offshore third-party development and production capabilities for custom-built, open source and commercial off-the-shelf (COTS) software has introduced additional risk factors beyond those regularly encountered with in-house development shops and domestic outsourcing.

The combined impact of outsourcing and offshoring on one's ability to ensure that software operates as specified and in a predictable fashion, and that it is trustworthy, will vary based on a multitude of factors. Many of these factors are frequently omitted from outsourcing/offshoring analysis and decision-making. The presentation will endeavor to provide an extensive list of the more relevant factors.

The cost-effectiveness of outsourcing and offshoring software-related decisions is very much influenced by one's ability to mitigate the inherent risks of software developed and run offshore through the use of internal or outsourced software assurance. In addition, when the software assurance function itself is sent off-shore, an additional set of risks is encountered. These risks have even greater impact if the software was also developed offshore, since the local nature of the software assurance process may bias the results.

Both customer organizations and service providers are subject to risks when they engage in software development, assurance, support, and operational relation-ships. In this presentation, we discuss the nature of those risks, from a business perspective, as they pertain to software assurance. We examine what needs to be done to avoid, deter, or prevent any adverse impact on system security, data integrity, privacy, intellectual property and resiliency of inadequately reviewed software, which is increasingly being developed and operated in a global envi-ronment.

About the Speaker

C. Warren Axelrod is the Research Director for Financial Services for the United States Cyber Consequences Unit. He is also an Executive Adviser to the Financial Services Technology Consortium. Previously, he was the Chief Privacy Officer and Business Information Security Officer for a bank. There he interfaced with the firm's business units to identify and assess privacy and security risks and mitigate them, to have employees become familiar with security policies, standards, and procedures, and to ensure that they were followed.

Warren was honored with the prestigious Information Security Executive Luminary Leadership Award 2007. Warren has published three books, two of which are on computer management, and numerous articles on a variety of information technology and information security topics, including computer and network security, contingency planning, and computer-related risks. His third book, *Outsourcing Information Security*, was published by Artech House in September 2004.

Warren holds a PhD in managerial economics from the Johnson Graduate School of Management at Cornell. He is certified as a CISSP and CISM.

Globalization and the Rise of Mediocrity or Unsafe at Any Speed or Altitude

George Gibbs
Northrop Grumman
gibbsgr.ctr@efv.usmc.mil

With globalization the ability to track and monitor product quality is made exceedingly difficult. This paper will examine a fictional global organization that builds complex transit systems for the global market. This organization must compete on cost, quality and brand recognition in an increasingly competitive business climate. Economic factors take precedence over safety considerations and publicity, unless fatalities and publicity force reconsideration.

This paper hypothesizes that product quality is sacrificed as the organization cuts cost and finances projects in new and innovative ways. The organization working within the bounds of industry regulation understands jurisdiction issues and seeks to avoid oversight. Globalization enables organizations to distribute accountability and mitigate negative press coverage.

What We Believe

We believe the products we use on a daily basis are safe and have undergone rigorous health and safety inspections. We believe our aircraft are well maintained, our food is safe and the cars we drive have been engineered to the standards set forth by the US government. We believe in our system of checks and balances, a partnership between government and industry that places the safety of the public above the profits of any one industry. We believe that government is big and always has an oversight of all major projects.

We believe in a free market place, and when a product's safety is in question the free press will break the story and inform the public. We believe organizations don't buy shoddy products.

We believe that organizations outsource to "concentrate on core services and products" and by outsourcing the organization can contract for specialized services for a limited time and achieve greater operational effectiveness.

The Facts

Regulatory agencies have limitations set upon their authority. The NTSB (National Transportation Safety Board) for example will get involved in most public transit accidents provided the project was funded with Federal dollars. If a transit project is built upon federal land, but financed by the sale of publicly traded bonds, the NTSB lacks jurisdiction unless a fatality occurs. When a transit system is privately funded, the set of federal safety standards don't apply and the contractor is left to his own safety standards. Amusement parks may have more stringent safety and accountability standards than some privately funded transit systems.

Major aircraft manufacturers rigorously test airframes and must undergo a certification process as mandated by the FAA but an airframe is more than the sum of its parts. A passenger aircraft must have certified pilots, pilots certified for that model aircraft, certified maintenance and ground crew, and highly trained aircraft controllers and documentation detailing maintenance and operating procedures.

Airlines, in recent years, have outsourced maintenance operations to foreign organizations that do not have certified FAA maintenance personnel. Some aircraft are fractionally owned. Fractional ownership is a business model that allows an owner to buy a share of an aircraft. The cost of operation and maintenance is distributed among as many as eight owners. Pilots may fly more than a single aircraft model and may not be certified for each model. Advisories specific to an aircraft model may not be effectively communicated to the aircraft pilot and simple corrective actions may not be taken.

The NTSB tracks aircraft accidents when they occur in the US. Accidents that occur in foreign countries are not within the jurisdiction of the NTSB, thus a series of accidents that occur in Asia and a series of accidents that occur in Europe are not correlated by the NTSB. Each country must argue the merits of each accident on an individual basis in accordance with local law.

Press coverage is not uniform; stories of local interest receive more air time. Correlation of failures that occur over several years is simply not made and not reported.

Going Global

Transit Manufacturing is a small company that specializes in the design and construction of small transit systems. These systems, in use at amusement parks and some other public locations, are highly reliable and have been in service for 30 years. Transit manufacturing recently has expanded its product line. Increasingly complex systems are up for bid requiring complex software. Transit Manufacturing projects typically require expertise in civil engineering, mechanical engineering and safety engineering but this software design effort stretches the organiza-

tion's engineering capability. For business reasons Transit Manufacturing is sold to a larger (still US) corporation to better compete in the world market. AB Global Transit offers creative solutions and funds projects with the use of bonds which they issue. AB Global Transit merges its standard software solutions with the automated solutions offered by Transit Manufacturing. Safety standards differ and the pedigree of the software becomes uncertain as product lines are merged. AB Global Transit is not big enough for the competitive global market and seeks to be acquired. AB Global Transit aggressively bid contracts. The successes of Transit Manufacturing projects is used as selling points for the new complex transit systems even though the past systems are not nearly as complex as the proposed systems. AB Global Transit, a US Company, bids with high risk schedules and optimistic cost estimates based upon software reuse.

AB Global Transit is sold and now becomes ABC Global Transit. ABC Global Transit, a German Company, is a foreign owned company and is known for its quality and dependability. It has facilities in the US, Canada, Sweden, Germany, Mexico, India and China and as a result of acquisitions of as many as 50 companies, it has redundant facilities. The only real value ABC Global Transit has to the transit industry is increased efficiency through consolidation of facilities. Some of the former Transit Manufacturing Company projects are behind schedule.

ABC Global Transit is sold once again. Bounder Global Transit produces trains, planes and automobiles and has manufacturing/design operations around the world. In three years a single enterprise has changed names four times. Software used in amusement park rides, now is bid for use in driverless transit applications and projects are funded by the use of bonds, eliminating federal oversight and safety regulations.

The Predictable the Unpredictable and the Expected Outcome

(1) West Coast, a Software Glitch

The first failure was on the **West Coast. A software glitch** was responsible for the crash of three rail cars and the damage of a fifty foot section of guide way.

Analysis: When Transit Manufacturing bid the system, they underbid simply to get the contract knowing the business would be sold. Backorders were everything; the goal was to increase the paper value of the organization. The project was financed by bonds; no federal oversight.

Safety certification would cost $2 million (high estimate); the cost to repair the guide way, and three rail cars was $5 million (insurance records). This incident and was described as a fender bender by the host facility. No one was hurt so just fix and go on. Few records were maintained, i.e. the detailed test process, and this lack of traceability was considered an asset not a liability. Another incident occurred at the same site several months later, the fire department re-

sponded after smoke was detected. Project was bid by Transit Manufacturing and delivered by Bounder Global Transit. After 9/11, Bounder Global minimized software assurance verification and outsourced some of the work. (August 2002, February 2003)

(2) East Coast, the Detection of Cracking

The second failure occurred about two weeks later on the **East Coast. Cracking of brackets** was detected. To the best of anyone's knowledge not a single bracket failed in the field, but if the bracket had failed a derailment could occur. This type of incident occurs in many new systems; an anomaly is detected and fixed before a critical failure occurs. Service was interrupted for months and lost revenue was in the millions.

Analysis: The defective bracket was corrected and no one was injured. Three months later the trains were in service again. (August 2002)

A year later another flaw was detected when one of the brake rotors "disintegrated in an inspector's hands." The supplier of the rotor claimed the organization was informed of the flaw two years earlier. Bounder Global Transit denied any knowledge of the defect. When a second major transportation disruption occurs on a system which a US senator uses, it becomes news. Senate hearings were held. Project bid by AB Global Transit and delivered by Bounder Global.

(3) East Coast, Loss of Life

The third failure occurred again on the **East Coast. This time there is loss of life.** A customer service representative was made a test engineer for a day. He did not survive. The NTSB investigation found that test procedures were not on site prior to the start of the test.

Not a Mickey Mouse Operation

The technology for the monorail vehicles came directly from the well-tested and safe monorail train systems running in an amusement park. Having an independent non-profit corporation in charge of financing, maintaining and running the entire system was a relatively new idea with very few precedents. From the start, extremely high standards and great financial demands were set for what was a new, unproven management structure in the transit domain. The pressure to perform without losing money was great.

Figure 1. Rail project analysis

Analysis: When a 23 year old single father raising a 5 year old daughter is killed on the job the most negative publicity is generated. Bounder Global Transit was the bidder of the project and had successfully delivered a similar project. What went wrong? (September 2002)

Bounder Global was in financial discord. The aerospace sector was down, way down due to 9/11. Cost and schedules dominated while safety came last. Engineering discipline was out and adaptive "seat of your pants"

management practices took over. These practices included lack of training for the customer service agent/test engineer, removal of speed regulation equipment, and adding 18,000 lbs of unsecured concrete as ballast. The ballast slid and crushed the operator after the derailment. A nine month schedule slip on a billion dollar transit system project in a city with one of the world's largest transit system makes national news. Any wonder why they didn't get a 1 billion dollar contract for a rail system offered in the same city two years later?

(4) Bolts on a Small Business Jet

The next year a rather unusual near failure. **The bolts on a small business jet** were shown to crack. The fix was simple but critical. If these bolts would fail in flight the jet could, in the words of the FAA, "go into an uncontrollable dive." The fix took about 10 minutes but when the FAA checked for compliance, the Bounder Global Transit could not verify that the repair had been made.

The FAA subsequently ordered an Emergency AD (Aircraft Directive). This level of severity basically states if you're in the air consider landing at the earliest opportunity.

(5-6) A Rail System Failed, Tire

The fifth failure came a year later. **A rail system failed when a tire fell 25 feet** to the parking lot below. This generated rather extensive negative publicity. Executives claimed the failure was a result of a worker failing to install the tire correctly. The system was tested, and a week later returned to passenger service.

(6) The system was reopened to the public and the next day a **two pound, six inch diameter washer** detached from one of the cars, shorted out the power rail and fell to the street below.

Analysis: If the situation was bad before the falling washer failure, it rapidly became worse. The credibility of senior management was directly challenged. Independent failure analysis experts were called in. Test procedures were rewritten. Test results had to be demonstrated and the system could not be opened to the public until an adequate test profile was completed.

The day before the failure, 140 error indications were logged against the defective train. **No one read the error log.** The local press reported that trains shuttered on certain sections of the guide-way months before the tire fell off the train. Furthermore the tire was not the first part that fell to the street below. A twenty pound part of the drive train fell as the system was first tested. Bounder Global Transit, still in financial distress, found a new and creative way to finance the system. They calculated the predicted ridership, they issued bonds to finance the project and in short they offered a cradle to grave solution that did not depend on federal or city funding. The basic problem is a guide-way with turns that

were too tight. With no oversight, no detailed design, and no traceability it is easy to shift blame. Negotiate rather than litigate or engineer.

The next week the system was put back into service, and failed one day later. Needless to say the credibility of the organization was challenged. A failure analysis firm was called in for an independent analysis. This firm looked at the total picture, management methods, design and finance. This firm had a history of analyzing catastrophic failures. One incident in particular involved the collapse of walkways at the Kansas hotel. This hotel failure was traced to the failure to install a dollar part in the overhead walkway. This small omission caused the deaths of over 100 people. Bounder Global Transit now infamous for their lack of detailed specifications and test procedures didn't sweat the details. (September 2004)

Replacement of GE Jet Engine Seals

The regional jet crashed near a city after the engines flamed out. Two pilots were killed after a fatal attempt to glide the plane to an airport.

When the engines lose power mid-flight, parts of the seal can quickly cool in the cold high altitude air. The cooling makes the seals shrink …causing them to lock up if they aren't spinning fast enough, the Washington based regulator said.

Figure 2. Engine seals

Four months later, with lost revenue of $70,000 per day, the rail system reopened. The rail system continues to operate to this day at 50% of the predicted ridership. The four hundred million dollar contract to extend the system was cancelled.

(7) Cargo Jet Crash

The Bounder Global aerospace sector began having similar problems. **A cargo jet crashed**. Two pilots decided to join the *40,000 Foot Club,* an exclusive club of Mountain Top Cargo, where the crew would fly to an altitude of 40,000 feet, the maximum certified altitude of the aircraft. Obtaining clearance to change altitude, they climbed to 40,000 feet and joined the club. Suddenly, the port engine stalled and within five minutes the starboard engine failed. The pilots declared an emergency and attempted to restart the engines. They vectored to an emergency landing site but fell short. This happened about a month after the falling tire/washer problem.

Analysis: This is not a simple accident. It is true the pilots took an unneeded risk by flying at the 40,000 ft altitude but it was within the aircraft flight envelope. Contributing factors to the accident include a problem with the jet engines. The GE engines need a minimum flow of air to prevent a turbine lock problem. This fact was known before the crash but the flight manuals were not updated and the pilots were not properly trained.

Bounder Global Urged to Revise Jet Guide

The US National Transportation Safety Board recommended Monday that Bounder Global revise its reference book for the Jet Aircraft to include more detailed instruction for takeoff.

The board recommended that the takeoff stabilizer settings…should be included in the handbook that is carried into the cockpit and that the operators be informed of the changes.

"Our aircraft flight manual already has the information… It's already done!" said a spokeswoman for Bounder Global.

Any changes would be done by the aviation training company which is responsible for the handbooks.

Figure 3. Guide revision

(8) Passenger Jet Crash

A passenger aircraft with 50 passengers crashed. The crash made the newspapers but since the crash occurred in the Far East little attention was paid to the incident. After an investigation, the cause of the accident was ice on the aircraft wings. Inexperienced staff did not deice the aircraft even though the aircraft was parked outside in bitter cold temperatures overnight. This occurred about a month after the cargo jet crash.

Analysis: Maintenance and Pilot training problem. The aircraft was left outside overnight on a very cold night with no deicing. A contributing factor may be the wing design that has too much play on a control surface.

(9) Business Jet Crash

The next month another crash occurred. **Ice on the wings** and pilot error were the causes. This incident had far more press coverage than the previous two since one of the fatalities was an eight year old boy.

Analysis: Pilot training problem. The aircraft was not deiced. Contributing factor may be the wing design that has too much play on a control surface.

(10) Business Jet Crash

The fourth incident occurred on the **East Coast. A business jet failed to take off**. Again, pilot error was the cause because he did not properly check the weight distribution within the aircraft.

Analysis: Pilot error but lack of training also must be considered. This was the second such accident resulting in an accident. Documentation and training is important and can avert disaster.

(11) Nose Landing Gear

In 2007 the organization continued to have problems. There were a series of failures in Asia, the most notable was the **nose landing gear failed to lock**. No one was killed but an investigation revealed that a bolt was never inserted in the aircraft by Bounder Global maintenance staff.

(12) Series of Landing Gear Failures

A series of **landing gear failures** continued in Europe, two failures in three days and a third failure within a month. This was settled out of court for $165 million.

Figure 4. Asian landing gear

(13) Door Falls Off

In 2008 a **door fell off** a business jet on takeoff, just an isolated incident proclaimed Bounder Global.

Analysis 11, 12, 13: Lack of any proactive investigation and action by Bounder Global. The first accident that occurred in Asia and this incident was preceded by several that forced unscheduled landings. The problem was so bad the Ministry of Transportation traveled to the home country to discuss the failure history. Bounder Global had ample opportunity to request inspection of the landing gear for the entire fleet before the series of European landing gear failures occurred. Items 11 and 12 occurred with a turbo prop aircraft and item 13 occurred with a business jet it seems that the entire company lacks a staff of well trained maintenance mechanics.

Consolidated Analysis

Bounder Global Transit's failure rate is due to several factors not unique to the aviation or rail industries. The lack of design and analysis, including requirements analysis, is one of the leading causes. Aggressive management, cost estimates, and lack of a failure reporting mechanism also contribute. The management style that placed a premium on salesmanship, image and the ability to negotiate through a difficult situation over engineering discipline is the major contributing factor.

When the trains crashed on the West Cost the automated control system design was far behind schedule. This new design subcontracted out much of the engineering effort on a partnership basis. The subcontractor/partner used proprietary equipment that due to the limited production run far exceeded the cost of standard rail equipment that performed the equivalent function. The lack of engineering analysis at the conception of the project, failed to identify that similar commercial grade equipment could have been used. The result was a design with excessive costs that was never safety certified (fire certified). The fire certification for the West Coast project was an acceptable risk but other projects required the equipment to be installed in tunnels where fire poses a much higher risk.

Bounder Global Transit's aggressive management style advertized the maximum ceiling of the aircraft as 40,000 ft. The statement implies greater fuel economy because of a decreased aerodynamic drag but the company failed to disclose that a minimum airflow must be maintained through the jet turbines. This turbine lock problem was discovered in the test phase of the aircraft. Both the engine manufacturer and the aircraft manufactured were aware of the problem but the aircraft flight manuals failed to disclose the condition. The pilots were heard thumbing through the flight manuals looking for restart instruction as the aircraft plummeted to earth. The flight manuals were subcontracted out. Blame is easily distributed to the subcontractor.

Still Interested In Streetcar Despite Rejection

..says it is still interested in pursuing the 1.25 Billion contract to supply a cities transit system with new streetcars despite the cities rejection …

The Transit Commission has told the company that its proposed vehicle would literally derail if used on the cities streetcar tracks.

Figure 5. Still bidding

When the Business Jet failed to take off because of a weight distribution problem, this was the second occurrence of the problem. The first occurred in 2001, and after years of investigation, the NTSB identified the weight distribution problem. To understand the cause of the second crash you must examine not only the aircraft crash but the ownership and management of the aircraft itself. The Business Jet that crashed was owned by one company leased to a second company and flown with pilots from a third company. Flight documentation was supplied by a subcontractor. In this complex management structure were the pilots aware of the issue? Was documentation updated? Who has the responsibility for initiating the documentation updates? How proactive was the process?

When a tire fell from a train 25 feet to the parking lot below the root cause of the failure was the lack of a detailed analysis of the guide-way. The turns were too sharp and some grades too steep. A similar east coast guide-way failed about two years before. The guide-way construction was, in both cases, subcontracted out and the failure was blamed upon shoddy workmanship of the guide-way subcontractor/partner. Bounder failed to have a program of lessons learned and under engineered the guide-way for a second time. When this second project lost the tire, the true cause of the failure was covered up and the trains were placed back into service. Lack of a detailed guide-way design resulted in the failure of the system and a significant contract loss (400 million).

Recently, Bounder Global bid a rail project claiming that their off the shelf design was compliant with all requirements. The customer reviewed the proposal and declared the design non-compliant because the rail cars would derail on certain sections of the track. The company lost another billion dollar contract due to under engineering of the guide-way.

The series of landing gear failures, the design was adequate but proactive maintenance inspection actions were not taken. Bounder Global could have issued inspection requests for the entire aircraft fleet but they failed to do so. This lack of action cost the organization 165 million dollars but the both airlines acquired more of the same model aircraft.

Europe: *Finally the three parties came to a very strange agreement. RBR Airlines will receive a compensation of approximately 165 million dollars. The weird part of the agreement is that RBR Airlines orders 27 new Bounder Global aircraft and 13 of the new planes will be RZ80's. This means that a few months after RBR Airlines refused to continue the operation of the RZ80's, it orders 13 new ones*.

Asia: *the RZ80 order from Asian Air Airways, worth about US$80 million, comes less than a month after Asian transport investigators determined that an error by Bounder Global maintenance workers led to a highly publicized RZ80 landing without a nose wheel. The Asian probe concluded last month that workers had failed to attach a bolt while repairing the front landing-gear doors of the RZ80.*

Figure 6. New aircraft orders

Both RBR Air lines and Asian Air made selections based upon fuel economy rather than safety. Economic stress forces most organizations become very short sighted and risk the loss of life rather than choose safety. In the case of the falling tire, the organization lost a $400 hundred million dollar follow on contract. Several months later, the organization lost a one billion dollar transit contract. This billion dollar loss came a few months after the "two pound washer incident," and after a rash of four Bounder aircraft failures.

Conclusion

Bounder Global Transit's exceedingly poor safety record is distributed between the rail and aerospace industries. No one government or news agency correlated the entire spectrum of failures that included design, maintenance, management and human errors. The rail sectors failures evoked outrage largely because of press coverage. These failures occurred in a single country not distributed around the globe. The aerospace failures were globally distributed.

This is a work of fiction; the names of the organizations were changed.

Worker killed customer service rep made test engineer

A system is more than just parts, it is a collection of interfaces, in this case training, maintenance, and documentation that all must play together. This is a challenge that requires diligence and commitment to produce a product or service that exceeds requirements. The fact that hardware systems predominate herein is of little consequence, if the key decision makers march full speed ahead without a system of checks and balances that marry quality and safety with cost and schedule realities then costs dramatically rise and preventable catastrophic failures occur.

George Gibbs is an engineer with Northrop Grumman with over twenty years of experience. Mr. Gibbs has a BS in Physics and a MS in Electro physics from Polytechnic University of New York. His diverse experience includes the F-14D, Ring Laser Gyro Navigation Systems, satellite communications and projects for the intelligence community.

Presently Mr. Gibbs is assented to the Joint Interoperability Test Command (JITC) and is responsible for the evaluation and validation of the Expeditionary Fighting Vehicle (EFV) and its ability to communicate Global Information Grid (GIG).

Inside Track to Offshore Outsourcing Using the Trusted Pipe: What Global Enterprises Look for in Offshore Outsourcing

Don O'Neill
Independent Consultant
ONeillDon@aol.com

Studies on global software competitiveness reveal that offshore outsourcing is an asymmetric tactic that delivers a competitive advantage. As global enterprises increasingly seek to achieve competitiveness on the cheap, global outsourcing is becoming more widespread. But due diligence is needed if success is to be achieved.
The Trusted Pipe™ architecture is a preferred approach to offshore outsourcing that will enable management and engineering personnel ("intelligent middlemen") located in offshore areas to facilitate the exchange of multi-dimensional messages spanning subjects that are cultural, technical and legal rights and remedies, software engineering and various other business skills, from buyer to seller.[10] Primarily, the Trusted Pipe™ will manage a network of global enterprises (GE) seeking to outsource software development and operations offshore to offshore vendors (OV). There are two major types of control points (CP), at least one GECP that operates in the U.S. and manages the network of global enterprises seeking to outsource software development and operations offshore and at least one OVCP that operates in the target country and manages the network of outsource vendors.

The object is to minimize the risks and maintain the benefits of an economic globally based, enterprise that deals with software producers in off shore nations that would otherwise be barred by adverse risks associated with such global enterprises.
® Trusted Pipe is registered with the U.S. Patent and Trademark Office

Keywords: global enterprise, global enterprise control point, intelligent middlemen, multi-dimensional messages, outsource vendor control point, outsource vendor, Trusted Pipe™ architecture

1. Need

Offshore outsourcing is an asymmetric tactic that delivers a competitive advantage [Florida 05, Hira 05]. As global enterprises increasingly seek to achieve

[10] Title of invention "Business management and procedures involving intelligent middleman", Inventor Donald O'Neill, Publication Number US20060015384 A1, Submission Date July 14, 2004

competitiveness on the cheap, global outsourcing is becoming more widespread [NAE 08, Software 2015]. But due diligence is needed if success is to be achieved. It is especially necessary to exercise due diligence in anticipating and avoiding the risks and threats that could occur through the use of the Global Supply Chain [CrossTalk 08]. What should global enterprises look for in offshore outsourcing?

While global outsourcing can be used to project enterprise competitiveness [Dobbs 04, Friedman 05], access to the world's high skilled, low cost software providers may be barred to the risk adverse global enterprise unless it establishes a multi-dimensional channel capable of rapid exchange of essential management, engineering, process, business, legal, and cultural packets in a predictable, reliable, and safe manner. The intended purpose of these multi-dimensional packets is to facilitate a coordinated interaction between the Global Enterprise and the Outsource Vendor, one that avoids conflict, smoothes out misunderstandings, and dampens down reaction to shortfall and mismatch in expectation and delivery.

2. Managing Scale and Value

The inside track for offshore outsourcing using the Trusted Pipe™ represents innovation in the outsourcing space [USPTO 04, Elders 05]. As offshore outsourcing moves to smaller projects, a dependable mechanism is needed to efficiently and effectively manage to scale the initiation of global enterprise projects and their fulfillment by offshore vendors. The Trusted Pipe™ architecture is literally the inside track to offshore outsourcing. Managing an arrangement of global participants and functional tasks into an innovation-driven, value hierarchy is a maxima-minima problem of pushing the highest skill work to the lowest cost of performance.

The convergence of cheap telecommunications, a defined software development life cycle and roadmap to software process maturity, commoditized programming skills, and low wages enable the offshore outsourcing of computer program software projects. For computer programming software and information technology projects, the highest value may be assigned the legal and business functions within the initiating global enterprise, and the lowest value may be assigned the engineering function of the fulfilling outsource vendor (see Figure 1). The Global Enterprise business need is met by the Outsource Vendor engineering solution. The process, management, and culture functions performed by the Global Enterprise and Outsource Vendor Control Points are necessary to eliminate friction. In the international outsourcing environment, this is what the outsourcing integrator does... selects and organizes the parts and eliminates friction thereby improving the predictability of the outcome.

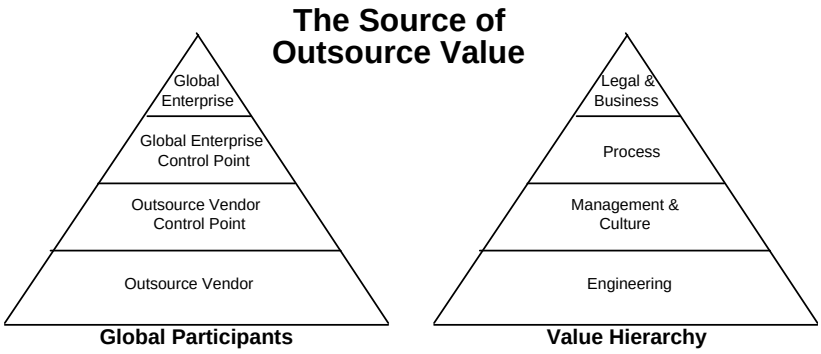


Figure 1. The Source of Outsource Value

3. Preferred Approach

The "Trusted Pipe™" architecture teaches how to conduct offshore outsourcing in the best possible way using a Trusted Pipe™ staffed with intelligent middlemen protecting bits at the water's edge. The Trusted Pipe™ architecture is a preferred approach to offshore outsourcing, one that will manage a network of global enterprises seeking to outsource software development and operations offshore (see Figure 2). There are two major types of control points, at least one Global Enterprise Control Point (GECP) (see Figure 4) that operates in the U.S. and manages the network of global enterprises (see Figure 3) seeking to outsource software development and operations offshore and at least one Outsource Vendor Control Point (OVCP) (see Figure 5) that operates in the target country and manages the network of outsource vendors (see Figure 6). These control points are staffed by intelligent middlemen capable of composing and interpreting the multi-dimensional messages.

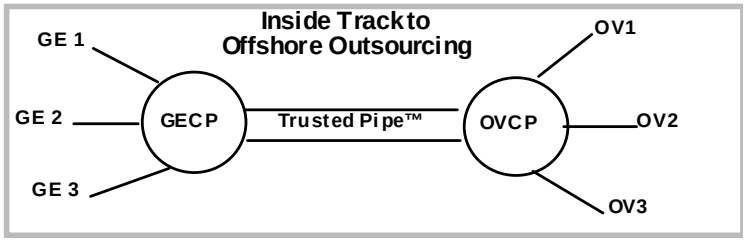


Figure 2. Inside Track to Offshore Outsourcing

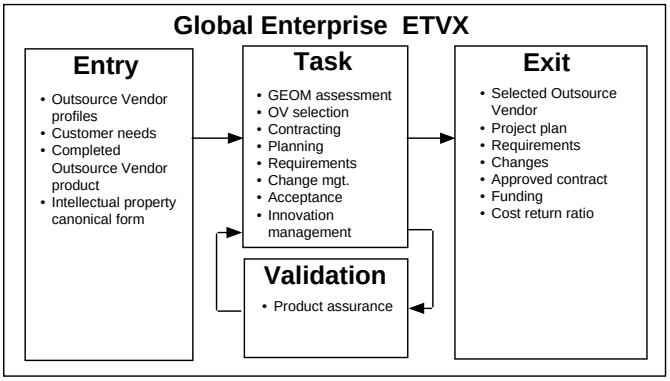


Figure 3. Global Enterprise ETVX

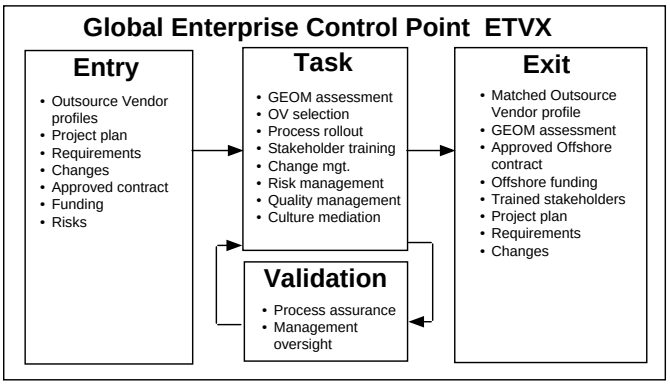


Figure 4. Global Enterprise Control Point ETVX

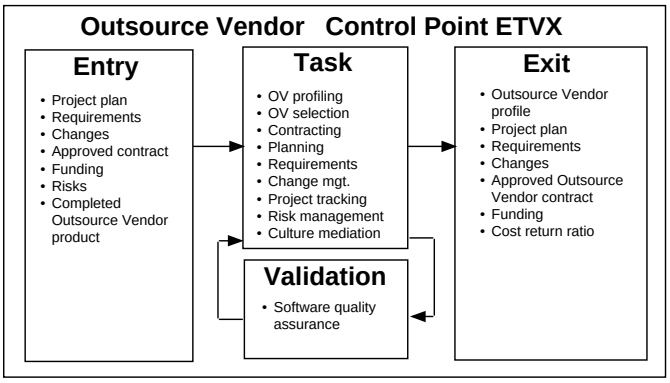


Figure 5. Outsource Vendor Control Point ETVX

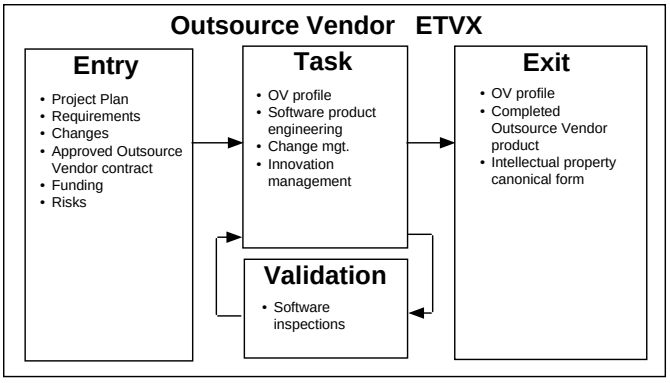


Figure 6. Outsource Vendor ETVX

Trusted Pipe™ features an in-country control point connected by high speed line, secure line to an out-country control point with capabilities and protocols organized into seven layers (see Figure 7). These aren't the usual seven layers. These intelligent layers comprise hard and soft skills spanning ethical dimensions, cultural mediation, intellectual property safeguards, security and privacy safeguards, management and engineering practice, domain knowledge, and technology infrastructure.

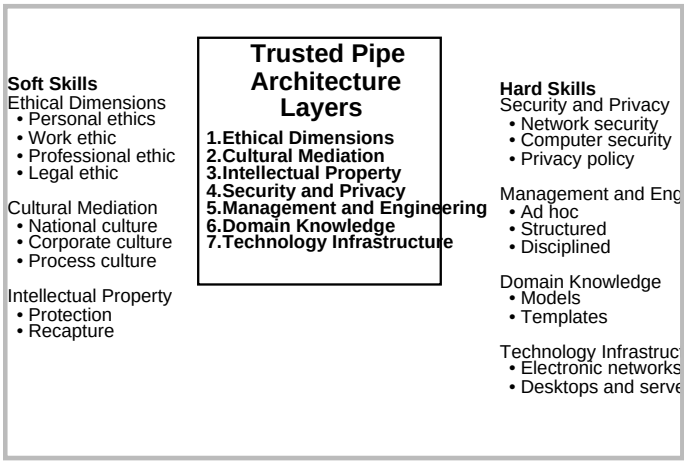


Figure 7. Trusted Pipe Architecture Layers

The knowledge, skills, and behaviors for installing and operating the control points and for using the Trusted Pipe™ and interacting with a control point reside in an on the shelf training program for rolling out Trusted Pipe™ (see Figure 8). The candidate for a roll out may be a company or a country.

Figure 8. Inside Track Training Product Foundation

4. Issues and Benefits

What are the benefits of the Inside Track Using the Trusted Pipe™? The benefits derive from the ability of the mechanism to pro actively address the concerns of global enterprises associated with due diligence. These issues are the source of outsourcing resistance and span business and legal, cultural, technical and legal, and software engineering and management.

Business and Legal

1. The management and control of intellectual property is accomplished with increased confidence by spanning the boundary between legal and technical factors.
2. Furthermore, the global enterprise is assured of recapturing intellectual property derived during the engagement. This is accomplished by employing a standard template to record processes, designs, and algorithms. These standard templates permit the Global Enterprise legal staff to fashion artifacts for use in the United States Patent and Trademark Office (USPTO) process of patents, copyrights, trademarks, and trade secrets.
3. The balance between commodity and strategic outsourcing can be shifted towards strategic content with greater confidence.
4. Additional privacy, increased anonymity, and increased safeguard of proprietary assets are guaranteed.

Cultural

1. Misunderstandings and expectation shortfall can be dampened without damaging network relationships between global enterprises and outsource vendors.

2. Issues can be handled in the best possible way across all dimensions including management, engineering, process, business, legal, and culture.
3. Any impedance mismatch in cultural style can be accommodated. For example, no push back to extreme militancy.

Technical and Legal

1. The computer and network security operations of the offshore vendor can be controlled.
2. Piracy of software packages is controlled.
3. The background of workers can be vetted.

Software Engineering and Management

1. The outsource vendor focus shifts from software process maturity to software product engineering. The OVCP shoulders the burden of software project management and quality assurance. The GECP performs oversight and governance and process management.
2. The process of experimentation inherent in software development and essential for innovation is facilitated through a rapid, predictable, and reliable operation.
3. Predictability in cost, schedule, and quality is managed and controlled.
4. The cost and billing model can be well coordinated with the change management mechanism in such a way as to diminish shortfall in expectation by spanning the boundary between management, engineering, and business factors.
5. The software development activity is accorded the best possible management, oversight, and governance.
6. The software operation is accorded a seat in the boardroom of the Global Enterprise.

5. Matching Global Enterprise and Outsource Vendor

The successful brokering of buyers and sellers in the complex environment of international trade depends heavily on matching the right buyers with the right sellers and buffering misunderstandings and maintaining the calibration of expectation and delivery. Finding the right matches is greatly assisted by assessing the global enterprise outsourcing maturity capability and profiling the leading indicators of the outsource vendor.

The GECP utilizes the Global Enterprise Outsource Maturity (GEOM) Assessment Instrument to identify findings and their consequences and formulate recommendations and plans for improvement. The global enterprise will understand how mature it is in seeking to achieve global software competitiveness, what it seeks to accomplish with offshore outsourcing, and what steps it must take to better position itself for success.

The OVCP utilizes the Outsource Vendor Profile (OVP) to characterize the vendor space. The leading indicators in the profile instrument may suggest avenues and directions for vendor improvement.

With a repository of Global Enterprise assessments and Outsource Vendor profiles it is possible to match buyers and sellers that promise a well aligned operation. Maintaining a collection of these assessments and profiles yields a repository of valuable information that serves to authenticate and professionalize the broker, clearinghouse, and gatekeeper role envisioned. This will assist in transitioning the business from the initial push to one of constant pull for the sustaining operation.

6. Global Enterprise Outsource Maturity (GEOM)

Global outsourcing is used to project the competitiveness of the enterprise. It is a defined process that operates as a disruptive technology in distinguishing an enterprise from its competition. Like any technology, user enablement transitions from novice to expert. Global Enterprise Outsourcing Maturity (GEOM) plots these transitions and pinpoints the capabilities that underlie them (see Figure 9).

GEOM is intended for use by both the buyer and seller of outsourcing services as a means to calibrate buyer expectations and align seller capabilities. It provides criteria for source selection useful to buyers. It provides a benchmark for sellers to strive for. GEOM is composed of a maturity model, an assessment instrument, and a database of practicum. GEOM is composed of five process elements.

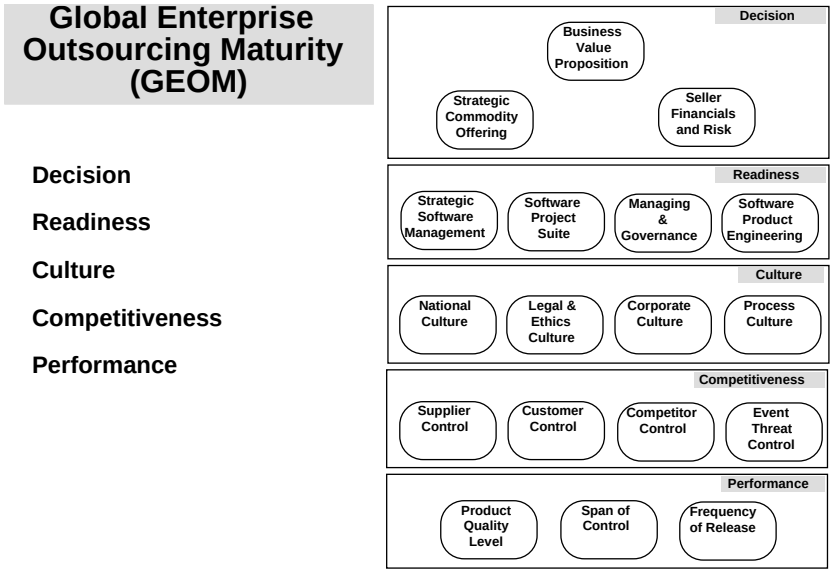


Figure 9. Global Enterprise Outsourcing Maturity (GEOM)

1. *Decision* focuses on the Business Value Proposition and relies on Strategic/Commodity Offering and Seller Financials and Risk. Strategic/Commodity Offering distinguishes between software assets that are strategically essential and those that are simply commodities in selecting outsourcing candidates and determines the optimum life cycle scope for the outsourcing engagement. Business Value Proposition features a return on investment calculation that draws upon the Seller Financials and Risk process for in-house cost estimates and outsource cost estimates, risk identification, and termination cost estimate and factors in wage scale elasticity for both in-house and outsource operations.
2. *Readiness* ensures there is a focus on strategic software management and the software project suite practices needed both in-house and outsource as well as well developed managing and governance capabilities and software product engineering practices. Strategic Software Management includes shared vision among stakeholders, software engineering process, software project management, software product engineering, domain architecture, and operations support. Software Project Suite practices include Planning, Tracking and Oversight, Requirements Determination, Software Product Engineering, Software Configuration Management, Risk Management, and Metrics. Managing & Governance capabilities include the Source Selection Process, Requirements Process, Configuration Management, and Governance and Oversight Process. It is recognized that Software Product Engineering practice may vary and may include Ad Hoc Programming, Structured Software Engineering, and Disciplined Software Engineering.
3. *Culture* spans national, legal and ethics, corporate, and process [Carmel 99]. National Culture includes considerations, such as, language, work ethic, ethics, and militancy [CIO 00, Moitre 01]. Legal & Ethics Culture spans Intellectual Property, Piracy, Security, Privacy, Trustworthy Software, and worker vetting. Corporate Culture may favor commitment management, product perfection, or personnel resources. Process Culture is based on the Software Engineering Institute's Capability Maturity Model [Paulk 95].
4. *Competitiveness* focuses on Supplier Control, Customer Control, Competitor Control, and Event Threat Control. Supplier Control includes establishing attractive workplace culture, achieving maturity in process and skills, fostering deep industry relationships within supplier community, and retaining personnel. Customer Control includes fostering deep customer relationships, balancing business factors, and achieving total customer satisfaction. Competitor Control includes fostering deep community relationships, fielding superior products, and leading niche direction Event Threat Control includes guarding against government intrusion, applying strategic software management, performing due diligence, and understanding reality.
5. *Performance* focuses on product quality level, span of control and frequency of release. Product Quality Level measured in defects per thousand lines

spans 10-100/1000, 1-10/1000, .1-1/1000, and .01-.1/1000. Span of Control measured in lines of code per individual spans under 12,500, 12,501-25,000, 25,001-50,000, 50,001-100,000, 100,001-200,000, and above 200,000. Frequency of Release spans daily, weekly, monthly, quarterly, semi-annually, and annually.

7. Outsource Vendor Profile (OVP)

The Outsource Vendor Profile (OVP) assesses factors associated with initial conditions, infrastructure, and experience (see Figure 10).

1. Initial condition factors include English language fluency, low wage structure, financial literacy, worker compliance, and privacy and anonymity.
2. Infrastructure factors include software education, software process maturity, access to technology cluster support, telecommunications, legal structure, mutuality in trade, and IP protection and recapture.
3. Experience factors include product offering experience, service offering experience, package application skills, open source experience, and application domain skill.

Outsource Vendor Profile (OVP)

Initial Conditions

Infrastructure

Experience

Factors	China	Russ.	India	Ireland	U.S.
Initial Conditions					
English language	no	no	yes	yes	yes
Low wage structure	yes	yes	yes	yes	no
Financial literacy	no	no	yes	yes	yes
Established companies	no	no	yes	no	yes
Vet for security	no	no	no	no	yes
Worker compliance	yes	no	no	yes	yes
Privacy & anonymity	yes	yes	yes	yes	yes
Infrastructure					
Software education	no	no	yes	no	yes
CMM experienced	no	no	yes	no	yes
Software management	no	no	no	yes	yes
Disciplined engineering	no	no	no	no	yes
Technology clusters	no	no	yes	no	yes
Modern telecommunications	no	no	yes	yes	yes
Security protection	no	no	no	no	no
Government support	yes	no	yes	yes	yes
Legal structure supp.	no	no	no	yes	yes
Mutuality in trade	no	no	no	yes	yes
IP protection and recapture	no	no	no	no	yes
Experience					
Product offerings	no	no	no	no	yes
Services offerings	yes	yes	yes	yes	yes
Source for mainten.	yes	yes	yes	yes	yes
Packaged application skills	no	no	yes	yes	yes
Open source exper.	yes	no	no	no	yes
Tradition of innovation	yes	yes	no	no	yes
Application domain skills	yes	yes	no	no	yes
Totals					
yes	9	6	13	13	23
no	16	19	12	12	2

Figure 10. Outsource Vendor Profile (OVP)

8. Findings: Facts Matter

Let's consider a software project. Done in the US a project takes 100% of effort, about 33% high value jobs and 67% low value jobs. When outsourced offshore, this software project takes 130% effort. The 33% of old high value jobs (business, legal, and program management functions) remains in the US. The 67% low value jobs (engineering) move offshore. The added 30% created by offshoring are

medium value jobs evenly split between US (process functions) and offshore (software management and culture mediation). Important to note, US workload is full cost; offshore workload is one-sixth to one-third of US full cost.

As a result, the US performs 48% (33% + 15%) of workload in high and medium value jobs. The offshore operation performs 82% (67% + 15%) of workload in medium and low value jobs. Outsourcing delivers 11% cost savings at higher offshore rates to 22% cost savings at lower offshore rates (see Figure 13a). Outsourcing cost savings stimulate additional project initiation. If software project demand doubles, the number of US jobs will be restored to current levels but these jobs will operate higher in the value chain.

9. Tracing the Offshore Delta

The Offshore Delta is traced in Figure 11.

- Development activities are listed.
- The percent of Onshore Base effort for each activity is listed. These total 100%.
- The percent of Offshore Delta effort for each activity is listed. In this example, it totals 33%.
- The percent of Onshore Base and Offshore Delta are assigned to the In-house Portion and Offshore Portion. These total 55% and 78% respectively.
- The percent of Onshore Base and Offshore Delta are assigned to the In-house Portion and Offshore Portion. These total 55% and 78% respectively.

Development Activities	Onshore Base	Offshore Delta	Inhouse Portion	Offshore Portion
Planning	10	5	10	5
Requirements Determination	20		20	
Offshore Vendor Selection		2	2	
Transition Offshore		8	4	4
Specification/Design	20			20
Code	20			20
Test	20	5	5	20
Transition Inhouse		8	4	4
Oversight	10	5	10	5
	100	33	55	78

Figure 11. Tracing the Offshore Delta

10. Cost Return Ratio

The Cost Return Ratio is the metric that best quantifies the benefits of an offshore outsourcing engagement. The Cost Return Ratio is calculated using the following expression:

Cost Return Ratio:=[Onshore Base-(In-house Portion + Offshore Portion)]/Onshore Base

We begin by calculating a Wage Weighted Resource metric for the Onshore Base (see Figure 11).

1. The In-house Portion of 25% is multiplied by the High Value labor wage of $120/hr. to obtain the Wage Weighted Resource metric of 3000.
2. The Offshore Portion of 75% is multiplied by the Low Value labor wage of $60/hr. to obtain the Wage Weighted Resource metric of 4500.
3. Summing these, the Wage Weighted Resource metric for the Onshore Base is 7500.

The In-house Portion is similarly calculated.

1. The In-house Portion of 25% is multiplied by the High Value labor wage of $120/hr. to obtain the Wage Weighted Resource metric of 3000.
2. The Offshore Delta Portion of 15% is multiplied by the Mid Value labor wage of $90/hr. to obtain the Wage Weighted Resource metric of 1350.
3. Summing these, the Wage Weighted Resource metric for the In-house Portion is 4350.

The Offshore Portion is also similarly calculated using the Offshore Rate to obtain the offshore labor wage.

1. The Offshore Portion of 75% is multiplied by the Low Value labor wage of $60/hr. adjusted by the 1/3 Offshore Rate to obtain the Wage Weighted Resource metric of 1500.
2. The Offshore Delta Portion of 15% is multiplied by the Mid Value labor wage of $90/hr. adjusted by the 1/3 Offshore Rate to obtain the Wage Weighted Resource metric of 450.
3. Summing these, the Wage Weighted Resource metric for the Offshore Portion is 1950.

Finally the Cost Return Ratio of 0.16 is calculated by substituting the values calculated.

1. CRR:=[Onshore Base-(In-house Portion + Offshore Portion)]/Onshore Base
2. CRR:=[7500 - (4350 + 1950)]/7500= [7500-6300]/7500= 1200/7500= 0.16

nshore Base	In-house Portion	Offshore Portion	Onshore Base	In-house Portion	Offshore Portion	Cost Return Ratio	Offshore Rate Versus U.S.
100%	25%	75%	7500	4350	1950	0.16	(1/3)

Calculate Wage Weighted Resources [i.e., Percent Dollars] as Follows:

7500 [Onshore Base]
- **25% * $120/hr. =3000** — **High Value Labor**
- **75% * $60/hr. =4500** — **Low Value Labor**

4350 [In-house Portion]
- **25% * $120/hr. =3000** — **High Value Labor**
- **15% * $90/hr. =1350** — **Mid Value Labor [Offshore Delta]**

1950 [Offshore Portion]
- **75% * (1/3*$60/hr.) =1500** — **Low Value Labor**
- **15% * (1/3*$90/hr.) =450** — **Mid Value Labor[Offshore Delta]**

Cost Return Ratio=[Onshore Base-(In-house Portion+Offshore Portion)]/Onshore Base
Cost Return Ratio=[7500-(4350+1950)]/7500
Cost Return Ratio=[7500-6300]/7500=1200/7500
Cost Return Ratio=0.16

Figure 12. Calculating Wage Weighted Resource

The Onshore Base, In-house Portion, Offshore Portion, and Cost Return Ratio for various onshore/offshore mixes and offshore rates are calculated using the Wage Weighted Resource metric for year 1, 2, and 3 (see Figures 13a-c).

Year 1 (+30% delta) **Cost Return Ratio**

Cost Return Ratio=[Onshore-(Inhouse Portion+Offshore Portion)]/Onshore

Onshore Base	Inhouse Portion	Offshore Portion	Onshore Base	Inhouse Portion	Offshore Portion	Cost Return Ratio	Offshore Rate Versus U.S.
100	25	75	7500	4350	1950	0.16	(1/3)
			7500	4350	975	0.29	(1/6)
			7500	4350	650	0.33	(1/9)
100	33	67	7980	5310	1790	0.11	(1/3)
			7980	5310	895	0.22	(1/6)
			7980	5310	597	0.26	(1/9)
100	50	50	9000	7350	1450	0.02	(1/3)
			9000	7350	725	0.10	(1/6)
			9000	7350	483	0.13	(1/9)
100	67	33	10020	9390	1110	-0.05	(1/3)
			10020	9390	555	0.01	(1/6)
			10020	9390	370	0.03	(1/9)

US rate	Offshore (1/3)	Offshore (1/6)	Offshore (1/9)	Onshore Base	Inhouse Portion	Offshore Portion
120	40	20	13.33	*	*	
90	30	15	10		* (15%)	* (15%)
60	20	10	6.67	*		*

Figure 13a. Cost Return Ratio: Year 1

Year 2 (+15% delta) **Cost Return Ratio**

Cost Return Ratio=[Onshore-(Inhouse Portion+Offshore Portion)]/Onshore

Onshore Base	Inhouse Portion	Offshore Portion	Onshore Base	Inhouse Portion	Offshore Portion	Cost Return Ratio	Offshore Rate Versus U.S.
100	25	75	7500	3675	1725	0.28	(1/3)
			7500	3675	863	0.40	(1/6)
			7500	3675	575	0.43	(1/9)
100	33	67	7980	4635	1565	0.22	(1/3)
			7980	4635	783	0.32	(1/6)
			7980	4635	522	0.35	(1/9)
100	50	50	9000	6675	1225	0.12	(1/3)
			9000	6675	613	0.19	(1/6)
			9000	6675	408	0.21	(1/9)
100	67	33	10020	8715	885	0.04	(1/3)
			10020	8715	443	0.09	(1/6)
			10020	8715	295	0.10	(1/9)

US rate	Offshore (1/3)	Offshore (1/6)	Offshore (1/9)	Onshore Base	Inhouse Portion	Offshore Portion
120	40	20	13.33	*	*	
90	30	15	10		* (7.5%)	* (7.5%)
60	20	10	6.67	*		*

Figure 13b. Cost Return Ratio: Year 2

Year 3 (+10% delta) **Cost Return Ratio**

Cost Return Ratio=[Onshore-(Inhouse Portion+Offshore Portion)]/Onshore

Onshore Base	Inhouse Portion	Offshore Portion	Onshore Base	Inhouse Portion	Offshore Portion	Cost Return Ratio	Offshore Rate Versus U.S.
100	25	75	7500	3450	1650	0.32	(1/3)
			7500	3450	825	0.43	(1/6)
			7500	3450	550	0.47	(1/9)
100	33	67	7980	4410	1490	0.26	(1/3)
			7980	4410	745	0.35	(1/6)
			7980	4410	497	0.39	(1/9)
100	50	50	9000	6450	1150	0.16	(1/3)
			9000	6450	575	0.22	(1/6)
			9000	6450	383	0.24	(1/9)
100	67	33	10020	8490	810	0.07	(1/3)
			10020	8490	405	0.11	(1/6)
			10020	8490	270	0.13	(1/9)

US rate	Offshore (1/3)	Offshore (1/6)	Offshore (1/9)	Onshore Base	Inhouse Portion	Offshore Portion
120	40	20	13.33	*	*	
90	30	15	10		* (5%)	* (5%)
60	20	10	6.67	*		*

Figure 13c. Cost Return Ratio: Year 3

Year 1	**25/75**	**33/67**	**50/50**	**67/33**
1/9	0.33	0.26	0.13	0.03
1/6	0.29	0.22	0.10	0.01
1/3	0.16	0.11	0.02	-0.05
Year 2	**25/75**	**33/67**	**50/50**	**67/33**
1/9	0.43	0.35	0.21	0.10
1/6	0.40	0.32	0.19	0.09
1/3	0.28	0.22	0.12	0.04
Year 3	**25/75**	**33/67**	**50/50**	**67/33**
1/9	0.47	0.39	0.24	0.13
1/6	0.43	0.35	0.22	0.11
1/3	0.32	0.26	0.16	0.07

Figure 14. Cost Return Ratio by Onshore/Offshore Mix

The Cost Return Ratio provides the basis for comparing the cost benefits of various options (see Figure 14). The Cost Return Ratio is the proportion of savings achieved by offshore outsourcing (see Figure 15a-c).

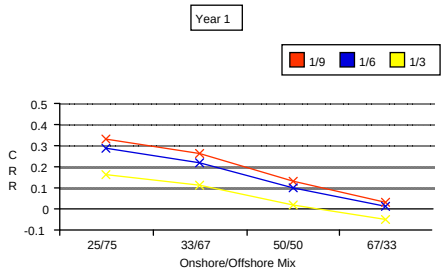


Figure 15a. Cost Return Ratio by Onshore/Offshore Mix- Year 1

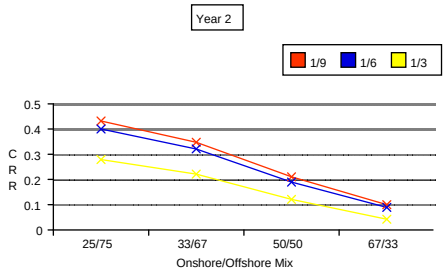


Figure 15b. Cost Return Ratio by Onshore/Offshore Mix- Year 2

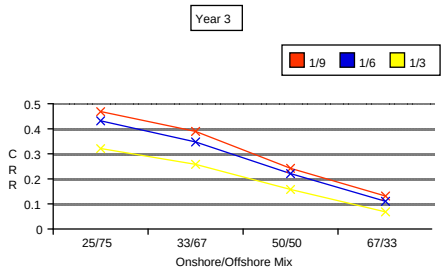


Figure 15c. Cost Return Ratio by Onshore/Offshore Mix- Year 3

11. Further Efficiencies Sought

The intermediate functions of requirements determination, product architecture and specification, project management, process management, and quality assurance are most subject to rearrangement where the criteria for rearrangement are tied to the prospect for innovative contribution. For example, in the innovation-driven arrangement requirements determination is tightly coupled with consumer innovation, and product architecture and specification are tightly coupled with producer innovation; and so these are accorded high value. On the other hand, certain process, management, assurance, and culture functions necessary to eliminate friction and improve the predictability of the outcome are only loosely coupled with innovation-driven activities and are thereby accorded less value. It is here among the job descriptions of these intelligent middlemen that standards-based commoditization can be further advanced to assist predic-

tability and increase software industry efficiency, where additional opportunities for disintermediation may yield further productivity gains, and where the residue of intelligent middlemen with their essential software job descriptions and functional activities can be elevated within the value hierarchy.

12. Questions Posed

Understanding why companies engage in outsourcing, managing the risks inherent in outsourcing engagements, identifying the essential elements of global outsourcing maturity, and reasoning about the cost return ratio for typical outsourcing scenarios are all topics of current research and study.

1. The enterprise contemplating outsourcing must obtain maturity in global outsourcing. What is global outsourcing maturity?
2. The outsource vendor must understand the assessment criteria and beyond that must demonstrate compliance with the criteria. What are the criteria for global outsourcing vendor assessment?
3. Without credible cost savings there is no basis for global sourcing. What is the cost return ratio, and what wage structures work for typical global outsource scenarios?
4. Without a realistic recognition of the risks of global outsourcing, the global outsource engagement cannot succeed. What are the risks for certain country destinations and outsource scenarios?

Bibliography

[Carmel 99]
Carmel, Erran, *Global Software Teams*, Prentice Hall, 1999, 269 pages, ISBN 0-13-924218-X

[CIO 00]
"A Passage to India", *CIO Magazine*, December 2000

[CrossTalk 08]
Defense Science Board, "Mission Impact of Foreign Influence on DOD Software", *CrossTalk: The Journal of Defense Software Engineering*, Vol. 21 No. 5, May 2008

[Dobbs 04]
Dobbs, Lou, *Exporting America: Why Corporate Greed is Shipping American Jobs Overseas*, Warner Books, 196 pages, August 2004, ISBN 0-446-57744-8

[Elders 05]
"High Paying Jobs From Offshore Outsourcing: An Oxymoron?", Bill Elder interviews Don O'Neill for *Technical Support Magazine*, December 2005, http://www.naspa.com/05articlesbymonth.htm#december

[Florida 05]
Florida, Richard L., *The Flight of the Creative Class: The New Global Competition for Talent*, Harper Collins, New York, 2005, 326 pages, ISBN 0-06-075690-X

[Friedman 05]
Friedman, Thomas L., *The World Is Flat: A Brief History of the Twenty-First Century*, Farrar, Straus, and Giroux. New York, 2005, 488 pages, ISBN -13: 978-0-374-29288-1

[Hira 05]
Hira, Ron and Anil Hira, *Outsourcing America: What's Behind Our National Crisis and How We Can Reclaim American Jobs*, AMACOM, 2005, 236 pages, ISBN 0-8144-0868-0

[Moitre 01]
Moitre, Deependra, "Country Report on India's Software Industry", *IEEE Software Magazine*, January 2001

[NAE 08]
"The Offshoring of Engineering: Facts, Unknowns, and Potential Implications", Committee on Offshoring of Engineering, National Academy of Engineering of the National Academies, Washington, D.C., 7 August 2008 http://www.nap.edu/catalog.php?record_id==12067

[Paulk 95]
Paulk, Mark C., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley Publishing Company, 1995, 441 pages, ISBN 0-201-54664-7

[Software 2015]
"Software 2015: A National Software Strategy to Ensure U.S. Security and Competitiveness", Report of the Second National Software Summit, Center for National Software Studies, 29 April 2005, http://www.CNsoftware.org

[USPTO 04]
Title of invention "Business management and procedures involving intelligent middleman", Inventor Donald O'Neill, Publication Number US20060015384 A1, Submission Date July 14, 2004

Don O'Neill is a seasoned software engineering manager and technologist currently serving as an independent consultant. Following his twenty-seven year career with IBM's Federal Systems Division, Mr. O'Neill completed a three-year residency at Carnegie Mellon University's Software Engineering Institute (SEI) under IBM's Technical Academic Career Program and has served as an SEI Visiting Scientist.

Mr. O'Neill served on the Executive Board of the IEEE Software Engineering Technical Committee and as a Distinguished Visitor of the IEEE. He is a founding member of the Washington DC Software Process Improvement Network (SPIN) and the National Software Council (NSC) and served as the President of the Center for National Software Studies (CNSS) from 2005 to 2008. He was a contributing author of "Software 2015: A National Software Strategy to Ensure U.S. Security and Competitiveness", a report on the Second National Software Summit. Mr. O'Neill has served as a reviewer of National Science Foundation (NSF) software engineering research proposals and has served as a member of the NIST Software Assurance Metrics and Tool Evaluation (SAMATE) Advisory Committee (2006-2008). He has authored Business Case articles that have appeared on the CERT Build Security In (BSI) web site. His current research is directed at public policy strategies for deploying resiliency in the nation's critical infrastructure.

Risk Issues

Where Risk Fails

Brian Chess, Founder and Chief Scientist,
Fortify Software

Security accounts for roughly 9% of IT spending, but what we've gotten for our 9% seems to be an ever-increasing number of headlines about IT security failures. It is time to rethink our approach to security beginning with our first principles. For today's security community, our most commonly stated principle is that security is about risk management, so this talk will consider the way we apply the concept of risk.

The notion of risk is attractive to the security community because our disciple offers few absolute guarantees, but a mathematical view of risk often fails to serve security practitioners well. It is a poor foundation for building a case for software assurance. Instead we need to promote security as part of a broader engineering endeavor and address the unknowable in the same way it is addressed when we combat other risks to the public.

This talk will look at the attributes that make risk appealing concept for application to security problems and why some of those same attributes undercut our ability to quantifiably improve security using risk-based analysis. We will contrast computer security problems to the problems found in some common non-digital systems such as the health code and the fire code where quantified risk is not used as a primary tool for limiting failure and show how these systems succeed with a combination of culture, standards and good engineering.

About the Speaker

Brian Chess is a founder of Fortify Software and serves as Fortify's Chief Scientist, where his work focuses on practical methods for creating secure systems. His book, Secure Programming with Static Analysis, shows how static source code analysis is an indispensable tool for getting security right. Brian holds a PhD in computer engineering from the University of California at Santa Cruz, where he studied the application of static analysis to the problem of finding security-relevant defects in source code. Before settling on security, Brian spent a decade in Silicon Valley working at huge companies and small startups. He has done research on a broad set of topics, ranging from integrated circuit design all the way to delivering software as a service.

Three Case Studies in Quantitative Information Risk Analysis

Mohammed A. Bashir and Nicolas Christin
Carnegie Mellon University, INI/CyLab Japan
ashbashir@cmu.edu, nicolasc@cmu.edu

In this paper, we build on existing literature and on a dialog with several decision-making partners (e.g., CISOs) to propose a simple methodology to quantitatively assess the value of security. We use this methodology to provide quantitative data gathered from three case studies of real organizations. The vastly different results we obtain across the three organizations considered emphasize the dependence between the security investments and the nature of the organization implementing them.

1 Introduction

Implementing security is potentially costly, may be partially ineffective, and does not generate any direct revenue for an organization. In addition, organizations are faced with trade-offs when they consider mitigation strategies to prevent attacks. A countermeasure may mitigate an attack, but is also likely to make tasks for the organization's end users more difficult. Take the example of spam: Not only it is extremely difficult for a spam filter to block all undesirable emails, but the spam filter may also block legitimate traffic, further impeding productivity.

As such, convincing non-technical decision-makers to invest in security is a daunting task for the technical managers or security officers who have to justify security expenditures. It is nevertheless a mandatory undertaking to avoid monetary losses due to security breaches [10].

Peltier [9] argues that, in information security, qualitative risk analysis is far easier to conduct than quantitative risk analysis, notably due to the complexity of the computations involved in quantitative models, and the lesser amount of security expertise needed. A further criticism against quantitative models is that, while offering seemingly precise estimates of the damage and recovery costs, they more often than not have considerable margins of error, due to the core assumptions on which they rely.

However, the key advantage of quantitative models is that they provide an actual dollar amount or "bottomline," which makes them appealing to non-technical decision makers. Among quantitative models designed for information security risk analysis, we can cite (in chronological order) the models proposed by Meritt [8], Tan [12], Blakley [4], Greer et al. [6], or Arora et al. [3]; but it is worth noting that most organizations doing quantitative risk analysis use their own model, tailored to their own specifics [9]. Publicly available decision-aid tools, for their part, have been either focusing on qualitative aspects [2], or, on the other hand, on very specific aspects, e.g., network topology [11], consistency between risk analysis and investments [1].

This paper argues in favor of quantitative models for information security. We rely on a simple methodology, described in Section 2, and present three case studies based on actual organizations in Section 3. Our case studies outline the dependencies between the size of an organization, the threats it faces, and the security measures it has in place. We discuss our results and conclude in Section 4.

2 Methodology

To describe our case studies, we rely on a simple quantitative model. A key feature of the model is its simplicity, motivated by usability constraints: we want it to be available as a tool usable by technical as well as management personnel. As such, our model relies on a few numbers to input, and provides a relatively small set of output metrics. Intermediate calculations may be relatively complicated, but can be automated. An implementation of our model is available as an Excel spreadsheet from `http://arima.okoze.net/isra`.

As simple as it may be, our model tries to address a wide variety of threats, not just IT risks. For instance, it also tries to capture risks posed to information held in different media (e.g., paper), and takes into account a comprehensive list of threats ranging from corrupt backup to dumpster diving.

This model is the product of an iterative approach: we created an original version of the model, using, as a basis, the information security risk analysis framework developed at LBNL [3]. We then presented our model to ten external partners, whose backgrounds and affiliations cover a fairly large range, both from the technical and management aspects; partners include researchers, security managers, and well as senior management, and are located in Asia, Europe, United States, and the Middle East. After gathering and integrating feedback from our partners, we revised our model, and arrived at the version we describe next.

The model revolves around attacks, outcomes of these attacks, countermeasures, countermeasure efficacy, and uses two primary types of output: (a) a Value at Risk (VaR) analysis, based on differing countermeasures, and (b) a Risk-based Return on Investment (RROI), that is, the ratio between the net benefit in implementing countermeasures and the cost for such countermeasures) of security controls [3].

The rationale for the Value at Risk analysis is that it is seemingly the best understood language from the financial community. At the same time, Risk-based ROI measures how effectively an organization uses its resources to avoid or reduce risk, and appears a necessary input for budgeting considerations.

Attacks and countermeasures To arrive at the VaR and RROI, we first take the attack types as input and the frequency of attacks over the past year. Using this and the percent coverage of recorded data, we calculate the estimated number of attacks per year.

Here our model uses a key assumption, that the recent past is a good indicator of what will happen in the near future. In other words, the threats are not expected to change drastically from one year to the next. While this assumption may be considered relatively stringent, it is relatively hard to avoid it without resorting to pure speculation about future events.

For attack i, consider the attack frequency F_i, and the coverage $G_i \leq 1$. The coverage corresponds to an assessment of how much data has been recorded, compared to the number of incidents that actually happened; ideally G_i should be equal to 1, but we need to take into account possible weaknesses in the audit trail maintenance. The estimated number of attacks per year is $A_i = F_i/G_i$.

The model also takes the countermeasures in place as input along with the effectiveness of each of the countermeasures against each of the attack types defined above (denoted by C_{ij} for countermeasure j against attack i) . Using this data and the attack frequency, we can calculate the estimated number of attacks that the organization would have suffered had there not been any countermeasures in place.

First, the probability all countermeasures fail against an attack i is $CF_i = \prod_j (1 - C_{ij})$. This formula makes the assumption that countermeasures are independent of each other. This assumption is reasonable for countermeasures that are largely orthogonal, for instance, physical security on the one hand, and data encryption on the other hand.

From there we get the number of attacks we would have in absence of countermeasures, $B_i = A_i/CF_i$, and finally the number of attacks prevented by countermeasures $R_i = B_i - A_i$.

Attacks vs. attack outcomes There is a crucial difference between an instance of an attack (e.g., malicious code infection), and its outcome (e.g., unavailability of a user PC). Countermeasures can thwart attacks; but, only attack outcomes affect the organization's bottomline. The relationship between attacks and attack outcomes is given by a matrix (α_{ij}). For instance, for attack i denoting a malicious code infection, there may be two possible outcomes: destruction of all information with a probability of 100%, and unavailability of the user PC with a probability of 70%. Then, we have $\alpha_{i,1} = 1$, $\alpha_{i,2} = 0.7$, and $\alpha_{i,j} = 0$ for any different outcome j (e.g., unavailability of a print server).

Losses We now shift to the expected losses. Consider the annual salary of an IT employee, M, the employee cost per day is estimated to be $P = 1.5M/365$, where the 1.5 factor has been chosen after discussion with partners to take into account tax and administrative overhead. This cost P will lead us to the expect loss per attack, once we have estimated the number of workdays an attack costs.

We use, as an input, the number of one-person days it would take, for an IT professional, to perform the following type of efforts: 1) The effort needed to diagnose a typical attack, 2) The effort needed to report a typical attack, 3) The effort needed to repair the damage caused by a typical attack, and 4) The effort needed to address any public relations/reputation issues arising from a typical attack.

Here, a typical attack refers to an attack that is most common, that is, one that is closest to the median with respect to severity. With this information, for each attack outcome j, we get the nominal damage N_j as

$$N_j = \sum_k D_{jk} P + C \ ,$$

where C is a parametrized cost noted for other, collateral attack damage not taken into account in other calculations, and D_{jk} represents the attack damage (in days), with k representing the four types of efforts (diagnosis, report, repair, follow-up).

We go from the nominal damage to the expected loss per attack outcome, EL_j by considering the extra severity $S_{j,*}$ of the attack. This is done by specifying the probability that any given attack outcome will be ten ($S_{j,10}$), one hundred ($S_{j,100}$) or one thousand ($S_{j,1000}$) times more severe than the typical outcome(s) for an attack of that type. This accounts for attacks that sometimes result in a high degree of damage. We get

$$EL_j = N_j[10S_{j,10} + 100S_{j,100} + 1000S_{j,1000} + (1 - S_{j,10} - S_{j,100} - S_{j,1000})] \ ,$$

as the expected loss for attack outcome j. This metric is equivalent to the Annual Loss Expectancy for attack outcome j.

Combining the previous outputs, we can calculate the expected loss without countermeasures and the loss avoided due to the use of countermeasures: With these two pieces of information it is now possible to calculate the residual loss per attack outcome (ELC_j). This is the same as the expected loss with countermeasures in place. We have

$$ELC_j = EL_j \sum_i \alpha_{ij} A_i \ \ ,$$

for a total residual risk $RR = \sum_j ELC_j$.

The sum of the expected loss for each attack type yields the total estimated expected loss that the organization incurred in the previous year. This can also be considered the

total residual risk the company is exposed to. All factors being the same, the organization is likely to incur this cost or loss from the attacks specified over the next year.

The expected loss per attack outcome (without countermeasures) is, likewise, $ELwoC_j = EL_j \sum_j \alpha_{i,j} B_i$, and the loss avoided thanks to the countermeasures is simply $LA_j = EL_j - ELwoC_j$.

Benefit of countermeasures We have so far looked at residual risks, but have not assessed the benefit associated with a given type of countermeasure. Consider the cost of capital r, and a time period in years given as t. Consider the total expected loss without any countermeasure, $ELwoC = \sum_j ELwoC_j$, then the benefit associated with only countermeasure k being in place is $BC_k = ELwoC - LC_k$, where LC_k is the total loss when only countermeasure k is in place. From BC_k we can get the current NPV for countermeasure k, NPV_k, as

$$NPV_k = BC_k - CM_k(1-r) \ ,$$

and the NPV over r and t as

$$NPV_{k,r,t} = \sum_l \frac{BC_k - CM_{k,l}}{(1+r)^t} - CM_k(1-r) \ ,$$

where $CM_{k,l}$ is the ongoing cost of countermeasure k, over interval l.

We can also compute the residual risk for each countermeasure acting alone. This is combined with the cost of the countermeasure to produce the net benefit of the countermeasure, and then the ROI for the countermeasure. The net benefit for countermeasure k is $NBCM_k = BC_k - CM_k$, which gives use a ROI for countermeasure k of $ROIC_k = NBC_k / CM_k$.

Simulating the value at risk Some of the inputs, in particular those pertaining to attack severity and number of occurrences, may be subjective. As such, the residual risk computed may be inexact. To solidify the predicted values, we complement the residual risk calculations with Monte-Carlo simulations of the value at risk.

We take the estimated number of attacks and the probability that this figure is 50% higher and 50% lower than the estimate. This data is used as input into a binomial distribution function with a random value to calculate a new attack frequency. This new attack frequency is then input into the model and a new total residual risk value is calculated. This procedure is repeated for a large number of n instances. This then allows for a Value at risk calculation for a specified confidence level.

3 Case studies

We next turn to a description of three case studies on which we use our model to gain a better understanding of the intricacies between each situation (security threats, particularities of the organization), and the effectiveness of selected countermeasures. The first case study is of a small network solutions company, the second case study is of a non-profit organization in the UK, and the third case study is of a major project within a Japanese insurance company. The full input and output stages of the model are available as an online appendix at `http://www.andrew.cmu.edu/user/nicolasc/publications/isra-appendix.pdf`.

3.1 Small network solutions company

This case study is for a small network solutions company. The company has an annual turnover of $4.8m and 22 employees. A typical IT employees salary is $31,000, which

Table 1: **Case study 1: Countermeasure effectiveness**

Countermeasure	Cost	ROI	Curr. NPV	NPV w/ r, t
Anti-virus	$1,000	1057%	$10,728	$8,349
Firewall	$2,000	-22%	-$135	-$339
IDS	$600	219%	$1,403	$1,154
Training and education	$1,500	1437%	$21,777	$18,770
UPS	$2,500	-78%	-$1,571	-$1,643
Active directory	$1,000	956%	$9,709	$8,332
Backup server	$1,200	-51%	-$435	-$511
Spam filtering	$500	1179%	$5,969	$5,135
Network access ctrl.	$2,200	398%	$9,083	$7,654
Email policy enforct.	$2,000	223%	$4,758	$3,916

Table 2: **Case study 1: Expected loss per attack outcome**

Attack Outcomes	EL per Attack Outcome
Information Theft/Disclosure	$322.93
Information Modification	$1,145.85
Information Destruction	$1,178.58
Service (User PC) Unavailable	$211.19
Legal/compliance problems	$6.46

equates to an IT employee cost per day of $129. The company faces the following attacks/threats: (1) Malicious code infections, (2) Administrator account compromise, (3) Regular account compromise, (4) Improper use, (5) Theft, (6) Spam, and (7) Natural disaster.

We calculate the residual risk to be $33,819. This is the amount that the company can expect to lose through the attacks we have considered in our model over the next year, assuming that the attack frequencies, attack outcomes, attack-attack outcome relationships, and countermeasure effectiveness remain the same.

With 95% confidence, we can infer that over the next year the residual risk (the amount the company is likely to lose) will be no more than $41,968. And with 99% confidence we can determine that over the next year the residual risk will be no more than $46,107.

Armed with the Value at Risk, senior management can now decide whether they wish to accept the risk or attempt to reduce it. If they attempt to reduce it, we can again use the model to estimate the ROI and NPV for additional countermeasures.

Table 1 shows the ROI, current NPV and NPV, where $r = 0.15$ and $t = 3$ years for each of the countermeasures in place acting alone. The ROI and NPV for most of the countermeasures is positive, showing these countermeasures are cost-effective. However, the firewall, backup server, and UPS seem not to provide good value for the services they provide.

The ROI for the countermeasures calculated is the ROI for each countermeasure acting alone. Our simple model does not take into account interactions between countermeasures, which may be particularly complex. They are dependent upon the combinations of countermeasures and the network architecture and configurations. Our model assumes that the combined effectiveness of the countermeasures is multiplicative, which may be overly optimistic.

Table 2 identifies the attack outcomes that result in the highest expected loss per attack outcome. The attack outcome with the highest expected loss is information destruction, and is therefore what the manager should try to prevent as much as possible. We can now identify the attacks that lead to the attack outcome, and consider countermeasures that will

Table 3: **Case study 1: Attacks that lead to information destruction and their respective losses**

Attack	Freq.	% result. in info. destruction	Expect. loss
Malicious code infection	20	35%	$8,250
Improper use	30	10%	$4,420
Natural disaster	2	10%	$236

help mitigate these attacks, thus reducing expected loss and residual risk.

Table 3 shows that the attack that results in the highest expected loss for an information destruction attack outcome is the malicious code infection. Using this information, we can explore possible countermeasures to mitigate against malicious code infections. The company currently has an anti-virus in place that is 90% effective in mitigating malicious code infection attacks. The company could invest in a more advanced secondary screening process for files that enter the system; essentially a second anti-virus.[1]

Assuming that we have a new secondary anti-virus (AV2, costing $3,000), which the vendor claims will be 80% effective against the malicious code infections that the company currently faces, the number of malicious code infection type attacks will be reduced from 20 to 4. This leads the residual risk to become $23,527 given the addition of the new anti-virus. Therefore, the benefit from the new anti-virus is $10,292 ($33,819 - $23,527). The net benefit is $7,292 ($10,292 - $3,000). We can then also calculate the ROI for AV2 as follows:

$$\text{ROI for AV2} = \frac{\text{prev. RR - new RR - cost of AV2}}{\text{Cost of AV2}} \approx 242\% \ .$$

The NPV is

$$\text{NPV} = \frac{\text{prev. RR - new RR - ong. cost of AV2}}{(1+r)^t} - \text{cost of AV2} \ .$$

With a capital cost of 15%, a time period of 2 years and an annual cost of $500, the NPV is roughly $4,404.

Using the same method the company can calculate the ROI and NPV of another anti-virus solution, and see which of the two is better.

Another point to note is that the new profit expected from ventures that have been profitably undertaken, thanks to the countermeasure, are not taken into account in the ROI and NPV calculations. These are projects that would not have been possible due to an excessively high risk exposure had the countermeasure not been in place. This is the opinion espoused by Soo Hoo [7], who calculates ROI simply as the annual benefit over the cost of the countermeasure. Blakley [4] however, includes new profit expected from otherwise impossible ventures into the benefit part of the equation. We have chosen not to make this addition to the ROI formula, because of the difficulty of defining the new profit. This difference should be considered when looking at countermeasures effectiveness.

If the company observes that adding additional countermeasures to the information security infrastructure does not reduce the residual risk to an acceptable level in a cost efficient way, it can choose to invest in cyber-security insurance. However, because of the lack of good actuarial data on which insurance companies can base premiums, they tend to include additional risk factors into their calculations, thus increasing premiums [5].

The company can also change the percentage values of the inputs and see the affects on the outputs and thereby identify areas where they need to spend more money. A 10%

[1]The company must also consider the implications of such a countermeasure on productivity. The second anti-virus may slow down the speed of end-users computers, as more operations have to be conducted due to the secondary anti-virus, and thus negatively affect productivity. Users may also become impatient and attempt to bypass the secondary anti-virus. This would have to be supplemented with additional education and training, thereby increasing costs.

Table 4: **Case study 2: Countermeasure effectiveness**
(All amounts in thousands of dollars.) Note the disproportionate SPVs obtained which indicate that the loss numbers reported by the organization are overly pessimistic. ROIs (not shown) are also disproportionately high.

Countermeasure	Cost	Curr. NPV	NPV w/ r, t
Anti-virus	$1K	$26,154K	$59,702K
Firewall	$0.8K	$18,908K	$43,171K
IDS	$1.5K	$10,808K	$24,676K
Training and education	$3K	$8,112K	$18,520K
Backup server	$2K	-$1,700	-$5,100
Spam filtering	$0.4K	$21,609K	$49,335K

increase in estimated attack frequency, results in an approximately 10% increase in the residual risk. Therefore we can conclude that investing resources into more accurate data collections with respect to attack frequencies would not be cost effective.

3.2 Non-profit organization

This case study is for a charity organization based in the United Kingdom. The currency values have been converted to dollars.

The organization has an annual turnover of $12m and 56 employees. A typical IT employees salary is $60,000, which equates to an IT employee cost per day of $250. The company faces malicious code infections and administrative account compromises.

We compute the residual risk to be $145,578. With 95% confidence, the residual risk should be no more than $232,336, and with 99% confidence our model tells us that over the next year the residual risk will be no more than $261,695.

Our model informs us of the potential effectiveness of additional countermeasures. Using, as in the first case study $r = 0.15$ and $t = 3$ years, and considering countermeasures in isolation, we obtain Table 4. The ROI and NPV for all of the countermeasures is positive, except for the backup server.

We note all numbers in the table are astoundingly high, which indicates the organization is highly sensitive to any change in the security policy. Also, these numbers are due to the high losses as reported by the managers from the organization, compared to the relatively modest turnover. Indeed, according to the values in this table, the mere threat of spam could bring this organization down. This leads us to believe that the self-reported values are overly pessimistic, and illustrates the value of a quantitative analysis of the kind as a "sounding board" when planning a budget. Although the values given are too pessimistic, the respective order of importance of each threat appears to be properly assessed.

The ROI/NPV for the backup server is negative here, because in itself, it does not prevent any attacks; however, it is worth noting that it could be very useful to mitigate (or even completely avoid) information destruction. This again, illustrates the point that the purpose of the tool is to act as a decision support tool, and the decisions are ultimately down to management or the user, who base their decisions on numerous other factors, other than ROI and NPV. These include things such as the profit gained from projects that are made possible because of the countermeasure.

Here again, the attack outcome with the highest expected loss is information destruction. In the case of this organization, administrative account compromise is the sole attack that results in information destruction. Using this information the user can identify potential countermeasures to prevent root compromises. This can include improved IDS and firewall capabilities, activity logging and improved policies with user training.

Table 5: **Case study 2: Expected loss per attack outcome**

Attack Outcomes	EL per Attack Outcome
Information Destruction	$3,437.50
Service (User PC) Unavailable	$962.50
Service (Email) Unavailable	$1,375.00

The user can explore the ROI and NPV for each of these countermeasures by using the same procedure highlighted in the previous case study. We will take a look at improved policies with user training as a possible countermeasure against root attack. Assuming the new countermeasure can reduce the current number of attacks by 50% the number of root compromise attacks would reduce from 10 to 5. This will reduce residual risk to $140,852, that is, a reduction of $4,726. Assuming that instigating the new policy and training users will cost in the region of $4,000, we can see that the countermeasure is cost-effective as the net benefit will be greater than zero. However, it would be better to have a countermeasure that would produce a greater net benefit, and a greater reduction in the residual risk.

Also, in this organization we found that the number of malicious code infection type attacks is quite high (225). This is another area where improvement in preventing losses may be possible. One possibility is to improve the effectiveness of the firewall if a large proportion of malicious code infection attacks that result in root compromises originate from outside the organization is high. Alternatively, the organization could add a secondary firewall. If we explore the idea of making the firewall rules stronger, so that the firewall prevents more of the malicious code infections, we would expect there to be fewer malicious code infections, resulting in fewer root compromises and ultimately a lower residual risk. However, stronger firewall rules would also prevent more legitimate traffic from passing through the firewall, and may impact users negatively. If the user is able to estimate the cost of the strengthened firewall rules, it would be possible to calculate the net benefit and ROI for the countermeasure.

Assuming that the annual negative effects of the stronger firewall are estimated at $20,000, the change in permissions costs $100, and the stronger firewall prevents a further 30% of malicious code infection type attacks, the residual risk will reduce to $77,516. This equates to a benefit of $68,062 ($145,578 - $77,516), a net benefit of $47,962 ($68,062 - $20,100), an ROI of 238% ($47,962 / $20,100), and with cost of capital at 15%, over 2 years, the NPV will be as follows:

$$\text{NPV} = \frac{145578 - 77516 - 20100}{(1+0.15)^2} - 20000 \approx \$16,266 \; .$$

The Value at Risk will now be such that, with 95% confidence the organization will not lose more than $122,365 over the next year from the attacks defined earlier.

Hence, our model tells us that, in the current situation, strengthening the rules to the existing firewall will be a cost-effective loss mitigation strategy. This is however, dependent on the reliability of the inputs to the model.

3.3 Project in multinational insurance company

This case study is for a project within a large multinational insurance company located in Japan. The currency values have been converted to dollars.

The project involves a turnover of $10m and 100 employees. A typical IT employees salary is $60,000, which equates to an IT employee cost per day of $250. The company faces the following attacks: (1) Malicious Code Infections, (2) Account Compromise, (3) Theft, (4) Spam, and (5) Natural Disaster.

Table 6: **Case study 3: Countermeasure effectiveness**

Countermeasure	Cost	ROI	Curr. NPV	NPV w/ (r, t)
Anti-virus	$8K	450698%	$36,057K	$82,298K
Firewall	$10K	503975%	$50,399K	$115,037K
IDS	$10K	472471%	$47,248K	$107,844K
Training and education	$5K	287970%	$14,399K	$32,866K
UPS	$10K	-100%	-$8.5K	-$32K
Server Room – Phys. Sec.	$8K	121681%	$9,735K	$22,209K
Employee Monitoring	$5K	194777%	$9,739K	$22,227K
Active Directory	$10K	21%	$3.6K	-$5K
Backup Server	$15K	68458%	$10,271K	$23,413K
Spam Filtering	$10K	367881%	$36,789K	$83,964K
BCP/DR	$30K	32375%	$9,717K	$22,125K

These result in the following attack outcomes: (a) Information Theft/Disclosure, (b) Information Modification, (c) Information Destruction, (d) Service Unavailable - User PC, (e) Service Unavailable - Email, (f) Service Unavailable - Website, and (g) Legal/Compliance Damage.

Our model predicts a residual risk of $4,521. With 95% confidence, over the next year the residual will be no more than $6,334, and with 99% confidence, the residual risk will be no more than $6,994.

Table 6 shows the ROI and NPV for countermeasures, with $r = .15$ and $t = 3$ years for each of the countermeasures acting alone. The ROI and NPV for all of the countermeasures is positive, except for the UPS. The ROI for the UPS is negative because it does not prevent any attacks. The UPS mitigates the loss from the attack outcome instances. It does not prevent any attacks, therefore it has a negative ROI and NPV.

Based on the Value at Risk figures, the company may decide that the current countermeasures in place are sufficient and any further countermeasures are not needed. This is especially the case as, with 99.9% confidence, given the inputs given, the residual risk for the project will be less than $7,654.

4 Discussion and conclusions

We provide a simple model for quantitative risk analysis of information security, and use this model on three cases studies: a small IT company setting, a non-profit organization, and a project within a multinational insurance company. The model has shown itself to be a useful input to decision-makers in that it has allowed the small IT company to make a decision to introduce a secondary anti-virus and thereby reduce residual risk by $10,000, and calculate that the ROI for the secondary anti-virus would be over 200%.

For the non-profit organization our research has helped the organization to make the decision to strengthen its firewall rules, thus approximately halving the organizations information security related residual risk. It has allowed the insurance company to determine the Value at Risk, giving them a better understanding of the projects risk exposure.

This shows, that used effectively, quantitative models can be an effective decision support tool. User feedback reported that such analysis will now allow them to justify to management countermeasures that they previously wanted to introduce, but for which they were unable to provide a suitable business case.

This work is clearly a long-term research endeavor, of which we have only completed the first step, that is, a methodology definition, and acquisition of initial data. The most interesting part of this research lies ahead of us, in the interpretation of the data. What makes

for instance our third case study to have so little value at risk, compared to our second case study? At first glance, both organizations seem to have some security controls in place, so why do we see such a huge disparity? We are in the process of analyzing this data, and determining if we can derive over-arching security principles tying an organizational structure with possible effectiveness of countermeasures.

Of course, such future work hinges on obtaining even more data to inform an analytic model of countermeasure efficiency. It is our hope that, by making our methodology and its instantiation (Excel spreadsheet) available to the public, we will be able to acquire supplemental data, and foster further discussion between the academic and management communities.

Acknowledgments

We thank our industrial partners for the extensive feedback and data, they provided throughout this research. This work greatly benefited from discussions with Ashish Arora and Rahul Telang at Carnegie Mellon's Heinz School. Mohammed A. Bashir's research was fully supported by a scholarship from the Hyogo Institute of Information Education Foundation.

References

[1] Secure insight analysis, 2007. `http://www.msbai.com/`.

[2] C. Alberts and A. Dorofee. An introduction to the OCTAVE method, January 2001. `http://www.cert.org/octave/methodintro.html`.

[3] A. Arora, D. Hall, A. Pinto, D. Ramsey, and R. Telang. Measuring the risk-based value of IT security solutions. *IEEE IT PRO*, November/December 2004.

[4] B. Blakley. A measure of information security in dollars. In *Proceedings (online) of the First Annual Workshop on Economics and Information Security (WEIS'02)*, Berkeley, CA, May 2002.

[5] L. Gordon, M. Loeb, and T. Sohail. A framework for using insurance for cyber risk management. *Communications of ACM*, pages 81–85, March 2003.

[6] D. Greer, K. Hoo, and A. Jacquith. Information security: Why the future belongs to the quants. *IEEE Security and Privacy*, pages 24–32, July/August 2003.

[7] K. Soo Hoo. *How Much Security Is Enough? A Risk Management Approach to Security*. PhD thesis, June 2000.

[8] J. Meritt. A method for quantitative risk analysis. In *Proceedings of the 22nd National Information Security Systems Conference*, Arlington, VA, October 1999.

[9] T. Peltier. *Information security risk analysis*. CRC Press, Boca Raton, FL, 2nd edition, 2005.

[10] S. Scalet. Risk: A whole new game. *CSO Magazine*, December 2002.

[11] SkyBox Security. Skybox view, 2007. `http://www.skyboxsecurity.com/products/overview.html`.

[12] D. Tan. Quantitative risk analysis step-by-step, 2002. SANS Institute Reading Room paper #849. `http://www.sans.org/reading_room/whitepapers/auditing/849.php`.

Organizational Development Issues

Paul Kurtz

Promoting Software Assurance on the Front Lines: Industry Proven Practices for Measuring Effective Product Assurance and Employee Training

Paul Kurtz, Executive Director, SAFECode, partner, Good Harbor Consulting LLC
Dan Reddy, Consulting Product Manager, Product Security Office, EMC

As threats to critical information systems grow more dynamic and sophisticated, never has it been more important to reduce software vulnerabilities and improve software's resistance to attack. Corporations building technology products often struggle with how to effectively promote software assurance practices within their organizations—specifically, how to begin the effort for assurance and sustain it over time.

This presentation will demonstrate how one company launched a business-oriented approach for measuring product assurance leveraging techniques like a Product Security Policy and Lean Six Sigma. These techniques fit within a new Security Development Lifecycle framework to justify and drive the ongoing financial investments needed to promote product assurance. Additionally, the presenter will discuss a number of case studies from individual companies that have successfully fostered their own assurance training programs even in the absence of an industry-accepted assurance training and certification program. SAFECode is working to explore how the requirements and desired experience of the employees of its member organizations can be amplified and replicated throughout industry.

SAFECode is developing a white paper entitled "Training Techniques, Certification & Goals" that will discuss current assurance training and certification programs that foment secure software development ultimately producing strong controls and integrity for commercial application. The paper will detail the skill sets and certifications SAFECode member companies look for in their employees and their opinion on the types of educational programs and certifications that would be desirable in the promotion of software assurance, within and among industry organizations. The paper will also discuss industry case studies of successful assurance training programs. Finally, SAFECode will propose recommendations based on gap analysis of current educational training and certification programs, which will lead to an increase in business practice for software assurance.

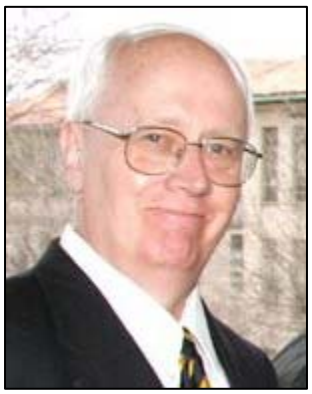
EMC is an industry leader in building software and hardware infrastructure that securely manages its customers' information. EMC has a rich four-year history to share in making the case for a new approach to building security into its products. The presentation will detail how EMC began measuring its internal progress for securing its products against a Product Security Policy.

About the Speakers

Paul Kurtz is the Executive Director of SAFECode and a partner at Good Harbor Consulting LLC. Kurtz is a recognized expert on cyber security and served in senior positions on the White House's National Security and Homeland Security Councils under U.S. Presidents Clinton and Bush. Kurtz served as the founding Executive Director of the Cyber Security Industry Alliance (CSIA), an advocacy group dedicated to ensuring the privacy, reliability, and integrity of information systems. Prior to joining CSIA, Kurtz most recently was special assistant to the President and senior director for critical infrastructure protection on the White House's Homeland Security Council (HSC), where he was responsible for both physical and cyber security. Before joining HSC in 2003, Kurtz served on the White House's National Security Council (NSC) as senior director for national security of the Office of Cyberspace Security and a member of the President's Critical Infrastructure Protection Board.

Kurtz received his bachelor's degree from Holy Cross College and his master's degree in International Public Policy from Johns Hopkins University's School of Advanced International Studies.

Dan Reddy is a Consulting Product Manager in the Product Security Office at EMC, a group that is charged with the continued driving of security improvements into EMC products. In his various roles in his 12 years at EMC, he has been consulting with EMC customers around product security issues and has been involved in numerous IT software development projects. Prior to joining EMC, Dan spent 15 years at New England Electric, a major electric utility with nationally critical infrastructure, where he held a variety of IT and business roles, including Manager of Technical Services in IT and Staff Assistant to the Chief Operating Officer. He also teaches Computer Science courses at Quinsigamond Community College in Massachusetts, where has taught for over 30 years. He holds an M. Ed. in Computer Science from Worcester State College and a B.A. from Tufts University in Education.



It's a Nice Idea but How Do We Get Anyone to Practice It? A Staged Model for Increasing Organizational Capability in Software Assurance

Dan Shoemaker
University of Detroit Mercy
dshoemaker1@twmi.rr.com

This paper presents a standard approach to increasing the security capability of a typical IT function. This five level model involves the development of a common set of security best practices, which are then deployed in a staged fashion to leverage an optimal security capability across the organization. At the lowest level the organization will have minimal assurance of security capability. At the highest level the organization can be trusted to produce products and provide services that are both dependable and secure. The paper will present the practices and the maturity framework. It will also discuss the practical mechanisms for implementing this model in a real world setting.

1. Introduction: Adding a New Challenge to an Existing Problem

Software projects have always been a crapshoot with the odds seriously stacked against the player. For instance, a recent Borland study found that approximately 33% of all projects are canceled prior to deployment, 75% of all projects are completed late and nearly 50% lack originally scheduled features and functions (Borland, 2005, p. 4). In addition, it has been well documented that depending upon project size between 25% and 60% of all projects will fail; where "failure" means that the project is canceled or grossly exceeds its schedule (Jones, 2004).

Worse, this is not exactly a new phenomenon. Throughout the 1990s, industry studies reported almost exactly the same outcomes. During that period, the average project exceeded its budget by 90 percent and its schedule by 120 percent and fewer than half of the projects initiated during that time finished on time and on budget (Construx, 1998). Likewise, a similar study done by KPMG Pete Marwick found that 87% of failed projects exceeded their initial schedule estimates by 30% or more. While at the same time, 56% exceeded their budget estimates by 30% or more and 45% failed to produce expected benefits (KPMG, 1996).

The root cause of this less than sterling track record lies in the nature of software itself. Try building something that is invisible or accurately documenting something whose form only exists in the minds-eye of a customer and you will under-

stand the problem. Software development involves translating a customer's abstract ideas about functionality into tangible program behaviors. That makes it hard to ensure anything consistent and repeatable about the process or its outcomes. Given those conditions, it might seem miraculous that anything useful has ever been produced by the industry, but the problem is just getting started. Now the product ALSO HAS TO BE SECURE.

When defects were just quality issues, the problem of buggy code had marketing and customer relations ramifications. Today, the right kind of defect, exploited by the wrong kind of adversary, can lead to a 9/11 style outcome. That is the reason why; no matter what the current list of excuses for defects the "buck" has to "stop" when it comes to producing secure software.

2. Maintaining the Minimum Organizational Capability to Ensure Secure Software

In practical application, it is hard to make the business case for secure software. That is because organizations are composed of people and those people have varying degrees of capability. Variation isn't a problem if a particular level of performance isn't required, because the company can always just keep patching their mistakes. However, where a specific level of proficiency is necessary to ensure a given level of performance, staff capability is a serious issue.

Staff capability is a major concern for business, since it is almost impossible to maintain a specific level of proficiency where constant turnover is a given. In that case, it becomes very important to adopt a well-defined process for developing and then assuring the organization's overall capability. Best practice is essential for secure software work since it defines the proper way to perform a given task. However, all sorts of factors can influence how closely and consistently any particular worker will follow any given practice. As a result, a standard organizational process has to be instituted to ensure that all required best practices are executed as specified in the software assurance plan.

Creating that process is an organizational development issue. It is also a precondition for making the business case for software assurance. It is a given that the organization is only going to be as secure as the capabilities of its people. Therefore, any discussion about the costs and benefits of secure processes is pure speculation until the people who will carry them out can be assured to be capable and willing to follow proper practice.

The term discipline simply denotes that a practice is reliably performed. Software assurance requires disciplined practice because in order to ensure a consistently secure product, all of the right practices have to be executed, by all participants, at all times, in a coordinated fashion. Accordingly, disciplined practice is essential to ensure that all of the products that are produced are secure all the time.

3. Learning to Discipline Cats

In many cases, consistent performance of disciplined practices will ensure the general security of code. But those practices will also impose additional work requirements. Because it is more work, it cannot just be assumed that the people who do that work will naturally accept and follow those new additional requirements. Instead, it should be assumed that people within the software organization have to be consciously motivated to carry out additional security tasks.

Motivation is an important factor in the software assurance process. Motivation initiates, directs, and sustains all forms of human behavior. Motivation is the factor that ensures a person's willingness to consistently execute a given task or achieve a specific goal, even if the performance of the task itself is personally inconvenient. It also dictates the level and persistence of a person's commitment to the overall concept of secure software. Consequently, motivation is the factor that underwrites disciplined performance.

Motivation is typically geared to accountability. This accountability comes from the enforcement of appropriate-practice (not best-practice) policies. Appropriate-practice policies are developed and documented by the organization to guide the entire process by which the software is created. These policies are then monitored for compliance as part of the overall organizational accountability system. The accountability system then rewards appropriate actions and discourages the inappropriate ones. However, it is impossible to enforce accountability if all of the appropriate-practice policies are not known or understood. Therefore, the organization also has to ensure that all of its employees know what they are expected to do, as well as the consequences of non-compliance.

Being able to ensure that everybody in the organization understands his or her exact role, responsibility, and function is the single most critical requirement in ensuring that software is developed correctly. That is because, no matter how potentially correct the security practices might be, if the people responsible for following those practices do not understand what they are supposed to do there is almost no chance that the resulting work products will be secure.

As such, every organization has to undertake a deliberate effort to maintain every worker's up-to-date knowledge of his or her individual security duties and accountabilities. The need to have an organized function in place to ensure a continuous level of security knowledge is particularly essential in light of the fact that the workforce in most businesses is constantly changing. As trained workers leave, or change jobs, and untrained people being added there has to be a consistent effort to maintain a requisite level of knowledge and understanding.

Consequently, besides perfecting the technical end of the software assurance process another aim of the software assurance function has to be to make cer-

tain that the function that ensures that understanding operates as intended. The mechanism that most organizations employ to meet that obligation is called awareness, training, and education (AT&E).

4. Ensuring that Everybody in the Operation Is Knowledgeable

There are three approaches to ensuring knowledge and acceptance of secure practice. Those approaches are awareness, training, and education. In ordinary use, the combination of these three is often called an AT&E program. Each of these delivery models represents a different approach to learning. Each has a distinct application and each is characterized by progressively more rigorous and extensive learning requirements. Because of that progression, these approaches are normally rolled-out in practical application as a hierarchy.

At the basic level, which is awareness, the purpose of the learning is very broad but the learning requirements themselves are limited. The next level up, which is training, builds on the awareness function. However, the application of training is restricted to fewer people and the learning is more in depth. Finally, at the top of the hierarchy, which is education, the application might be limited to a few key people but the learning requirements are very broad and in depth.

4.1 Awareness Programs

Awareness is the lowest rung in the ladder. Effective awareness programs ensure that all employees at every level in the organization appreciate the need for, and are capable of executing, disciplined secure software practice, in a coordinated manner. This meets basic software assurance aims. However, the requirement for awareness varies across the organization. Awareness at the highest levels of the corporation sets the “tone at the top.” So, awareness programs at the executive level are focused on ensuring the strategic policy awareness issues facing the organization, as well as the costs, benefits, and overall implications of security.

At all of the other levels, it is necessary to maintain a relatively high degree of awareness of relevant software assurance practices. Therefore, everybody in the organization must be made aware of the specific security requirements that apply to their position. In addition, they have to be motivated to practice security in a disciplined fashion. Thus, a good awareness program will

- Strengthen motivation—the program must motivate all users to practice security.
- Ensure effective focus—the program must concentrate on relevant and appropriate topics.
- Maintain participant interest—the program must ensure that individual participants will continue to be interested in security.

- Underwrite capable performance—the program must ensure effective security
- Integrate the content—the program must ensure the full integration of the proper set of practices

However, awareness alone does not assure reliable software assurance practice. As such, it is also necessary to ensure that individuals responsible for executing specific assurance functions, such as static tests and inspections are knowledgeable in the precise requirements of their role. That implies the need for a greater degree of knowledge and capability than is typically provided by an awareness function. This is typically underwritten by formal training.

4.2 Training Programs

Training is organized instruction that is intended to produce an explicit outcome. Consequently, it emphasizes job-specific skills. The purpose of training is to make sure that organizational functions, which are required to ensure safe and secure software, are performed correctly. Training ensures that all participants in the process have the specific skills necessary to carry out their assignments and that the level of organizational capability is continuously maintained. Training can be expensive, but it is an effective way to guarantee capable long-term execution of software assurance processes.

Nonetheless, because it is based on skills rather than concepts, training is too narrow to ensure that the software assurance process itself is executed correctly across the entire organization. Instead, training prepares individual workers to execute a series of steps without concern for the context, or the reasons why those might be necessary. Training provides a quick and satisfactory outcome if the known threats to software never change or if adaptation to new threats is not required. However, most assurance situations are dynamic and complex. Therefore, training does not provide the overall strategic understanding that is necessary to establish a lasting security solution. A program of formal education is required to ensure that the organization's code is maintained continuously secure.

4.3 Education Programs

Education is oriented toward knowledge acquisition, rather than the development of short-term skills. It ensures an intelligent, rather than rote, response. It establishes understanding of the principles of secure software development as well as the critical thinking abilities that will be needed to evolve the software development process through a continually changing and uncertain threatscape. For that reason, the few individuals in the organization who are responsible for the long-term guidance of the security function must undergo formal and in-depth education in software assurance principles and practices.

Education can be distinguished from training by its scope, as well as the intent of the learning process. In a training environment, the employee acquires skills as part of a defined set of job criteria. In an educational context, the employee is taught to think more about the implications of what he or she is learning. The learner must be able to analyze, evaluate, and then select the optimum security response from all alternatives. Thus learners are encouraged to critically examine and evaluate the problem and to respond appropriately by tailoring fundamental principles into a solution that precisely fits the situation.

The practical aim of education is to develop the ability to integrate new knowledge and skills into day-to-day security practice. The specific outcome of an institutionalized education process is the ability of executives, managers, and workers to adapt to new situations as they arise. Given what has been said about the constantly changing nature of threats and vulnerabilities, this is an essential survival skill for the leadership of any organization.

5. Increasing Organizational Capability through AT&E

The outcome of a properly administered AT&E program is an increased level of organizational capability. This is a strategic concept. It is based on the achievement of five progressively more capable states of security:

- Recognition—the organization recognizes the need for security.
- Informal Realization—the organization understands informal security practices.
- Security Understanding—the security practices are planned and monitored.
- Deliberate Control—decisions about security practices are based on data.
- Continuous Adaptation— practices adapt to changes and are continuously improving.

The levels of capability are progressively achieved through targeted awareness, training, and education processes.

5.1 Security Recognition

The most fundamental level is simple Recognition. Here, the majority of the participants are able to recognize that secure software is a valid and necessary concern. Until that fundamental state of recognition is achieved, the organization is essentially operating without any concept of secure practice. Once adequate recognition is established, however, individual members begin to understand that exploitation of coding flaws is a concern. This may not necessarily be in any deliberate or actively organized fashion, but it does involve a persistent underlying appreciation that security practice is necessary.

5.2 Informal Realization

At the next level, Informal Realization, members of the organization become more conscious of the need to ensure against software defects. Every worker is aware that those concerns exist. Workers might also follow rudimentary assurance procedures in response to that understanding. Thus, this level is supported by a more involved awareness program.

The awareness program that underlies informal realization presents security issues that have been expressly identified as concerns, such as buffer overflows. It might also present general practices to address these concerns, such as parameter checking. This is done on an ad-hoc or informational basis. The best practices that are designed to avoid common coding errors are not sufficiently specific and their performance is not organized well enough to ensure that security is embedded in the standard operation. That happens in the next step.

5.3 Security Understanding

The third stage, Security Understanding, is the first level where a consciously planned and formal security effort takes place. At this stage, the organization understands and acts on a commonly accepted understanding of the need for some form of formal security practice. The response might not be extensive and it is often dependent on individual willingness, but it is recognizable in that standard software assurance procedures are planned and documented in a systematic fashion.

The fact that security procedures have been formally documented allows the organization to implement a training program. Training is typically done to enforce understanding of the requisite security practices that are associated with each generic role. For instance, there might be targeted programs for executives regarding the business consequences of exploitation, a different one for managers aimed at implementing monitoring and control functions, and another for workers aimed at ensuring that best practices are followed.

The worker training programs might be subdivided by operation, such as development, versus, acquisition versus sustainment. The aim of each program though is to foster understanding of the security procedures that are appropriate to that role or function. These programs are generally not oriented toward ensuring specific skills beyond the understanding of the security practices that are required to carry out basic work. That is done in the next stage.

5.4 Deliberate Control

The fourth stage, Deliberate Control, is typical of a well-organized software assurance operation. Deliberate control is characterized by an institutionalized software assurance response that is built around providing a tailored set of skills

for each relevant position. These skills are defined and managed based on a precise knowledge of the requirements of each individual's role in the organization.

The execution of these security tasks is monitored using quantitative measures of performance, such as defect density. Deliberate control is enforced by defined accountability. Because it is objectively monitored, the security operation is fully managed by the organization's top-level executive team. At this level of functioning, the organization can be considered both safe from common threats and actively practicing the steps that are necessary to maintain that requisite level of security.

This state comprises a targeted mix of training and education. Coordination and administration of the program is designed to achieve specific assurance outcomes. The training and education program communicates the precise knowledge and skills that are needed to correctly perform specific security practices that are required by each function. This is reinforced through periodic retraining.

Training at this level is a carefully planned activity that requires many of the activities performed by the personnel security function, such as job definition, job classification, and privilege setting, to make it successful. The outcome provides a very high level of carefully controlled assurance. However, this is not yet the highest level of education possible.

5.5 Continuous Adaptation

At this final and fifth level, the software assurance function is fully optimizing. It not only carries out all of the practices necessary to ensure secure code within the dictates of the situation, but it continues to evolve those practices as conditions change. Organizations at this level are capable of adapting to new threats as they arise. That allows them to maintain consistently effective software assurance countermeasures as well as an active response to any new threat. They are safe from harm because they are protected from all but the most unforeseen events, and they are capable of a rapid and meaningful reaction to any threat that might occur.

This stage is achieved by ensuring workers master the critical thinking skills necessary to identify and solve problems. That requires a high level of knowledge of the elements and requirements of the field, as well as the thought processes to allow people to adopt these principles to new situations as they arise.

The classic mechanism for reaching this level of competence is a well-designed educational program. Skill training might also be among the factors needed to achieve this level. Nevertheless, the integration of that knowledge into the capability to respond correctly to new or unanticipated events falls within the realm of education.

6. Some General Conclusions

A formal and well-run AT&E program is a critically important advantage for a software organization because, in the end, no matter how well intentioned your staff might be, without sufficient knowledge in secure coding practice, your assurance capability will be limited.

The type of dynamic approach outlined here is an ongoing commitment. Therefore, it cannot be stressed enough that the organizational entity that is given the responsibility for training must constantly monitor and control the development of the program and the personnel resource through formal assessment and review.

The maturation of an AT&E program is a continuous activity that flows from the refinement of security knowledge as well as new knowledge gained through performance of security activities. The training operation requires a total commitment by the organization, particularly the top-level people, to maintaining a dynamic and complete understanding of all necessary requirements and capabilities. This is essential in order to develop the programmatic responses required to meet the demands of an evolving threatscape. Nonetheless, if this dictate is adhered to, AT&E can provide the operational backbone necessary to ensure that the organization will stay secure.

References

Borland Software Corporation, “Software Delivery Optimization Maximizing the Business Value of Software,” Borland Vision and Strategy Solution Whitepaper, 2005

Construx Software Builders, web site @ www.construx.com, 1998, Cited in Shoemaker, D and Vladan Jovanovic, “Engineering a Better Software Organization”, Quest Publishing House, Ann Arbor, 1998

Jones, Capers, *Assessment and Control of Software Risks*, Prentice-Hall: Englewood Cliffs, 1994, NJ

Jones, Capers, “Software Quality in 2005, a Survey of the State of the Art”, Software Productivity Research, Marlborough, Massachusetts, 2005

KPMG Technology and Services Group, web site at www.kpmg.ca 1996, Cited in Shoemaker, D and Vladan Jovanovic, “Engineering a Better Software Organization”, Quest Publishing House, Ann Arbor, 1998

McConnell, Steve, “The Business Case for Software Development”, Construx Software Builders Inc., 2007, http://www.igda.org/qol/IGDA_2005_QoLSummit_Business-Case.pdf, 12/1/2007

McGibbon, Thomas, “A Business Case for Software Process Improvement Revised”, DoD Data Analysis Center for Software (DACS), 1999

Paulk M., B. Curtis, M. Chrissis, C.Weber, "*Capability Maturity Model, Version 1.1*," Technical Report, Software Engineering Institute, Carnegie-Mellon University, 1993

Dan Shoemaker has been teaching software engineering topics since his first experience with the field in 1987. The fact that he has been located in a College of Business has influenced the direction of his research and writing, focusing it on “big picture software engineering processes.” He is also the Director of the NSA sanctioned Centre for Assurance Studies at the University of Detroit Mercy and the Co-Chair of the Workforce Training and Education Task Force at NCSD software assurance initiative. His book *Information Assurance for the Enterprise* is currently Amazon’s number one seller in the field of IA.

Robert Seacord

Secure Coding Standards Business Case

Robert Seacord
CERT Program, Software Engineering Institute,
Carnegie Mellon University
rcs@cert.org

Shaun Hedrick
CERT Program, Software Engineering Institute,
Carnegie Mellon University
hedrick@sei.cmu.edu

The lack of secure software systems is a baffling and disturbing long-term trend. Although consumers of software technology, ranging from compilers to Internet-facing systems, demand security their purchasing decisions frequently do not reflect this—and vendors are aware of this. Consequently, software vendors focus on functionality, performance, and other system attributes that more directly influence purchasing decisions. This consumer behavior results from a lack of shared understanding between vendors and consumers as to what constitutes software security. To sell security, vendors frequently need to sell security features, but consumers are primarily interested in purchasing software that is free from vulnerabilities that would cause their systems and data to be compromised. We propose a solution to this problem in the application of implementable and verifiable secure coding standards to create a shared definition of software security between software consumers and vendors.

1. The Demand for Secure Software

The Morris worm incident, which brought ten percent of Internet systems to a halt in November 1988, resulted in a new and acute awareness of the need for secure software systems. Twenty years later, many security analysts, software developers, software users, and policy makers are asking the question, “Why isn’t software more secure?”

The first problem is that the term software security, as it is used today, is meaningless. Many have attempted to define this term [3, 1] but there is no generally accepted definition. Why does this matter?

There are a variety of reasons given for why software is not more secure. These reasons include inadequate tools, programmers with a lack of sufficient training in security, and schedules that are too short. These however, are all solvable problems, suggesting that the root cause of the issue lies elsewhere.

One explanation for why software is not more secure is because there is no demand for secure software. In simple terms, if one vendor offers a product that

has more features and better performance and is available today and another vendor offers a secure product that has less features and not quite as good performance and will be available in six months, there is really no question as to which product customers will buy, and vendors know this.

So why do customers not buy secure products? Again, this is because the word "secure" is meaningless in this context. Why would a customer pass up tangible benefits to buy a product that has such an ill-defined and intangible property as security?

The only solution to this problem is to change the market dynamic for developing and purchasing software systems. Creating this new market dynamic first requires a shared, detailed understanding between vendors and consumers as to what constitutes software security. This shared understanding can be established by producing common definition of software security for particular software systems. Once established, this shared definition provides a mechanism by which customers can demand secure software systems and vendors can show the value of their investment in secure software development.

2. Secure Coding Standards

The goal of secure coding standards is to change the market dynamic for developing and purchasing software systems, by producing an actionable and measurable definition of software security programs.

An essential element of security is well-documented and enforceable coding standards [4]. Coding standards require programmers to follow a uniform set of rules and guidelines determined by the requirements of the project and organization, rather than by the programmer's familiarity or preference. Once established, these standards can be used as a metric to evaluate source code (using manual or automated processes).

A secure coding standard provides rules and recommendations for writing code in a secure manner. While developing code in compliance with a coding standard does not guarantee the security of a software system, it does provide information about the quality and security of the code. In addition to providing quantitative information as to the degree a particular software system complies with a set of secure coding guidelines, it also informs the consumer that the software developers who produced the code have done so with security in mind.

Secure coding standards benefit software vendors as well as consumers. By implementing their system to conform to industry-standard guidelines, vendors can make a verifiable claim regarding the quality of the code, and this claim can be supported by independent review.

Established secure coding guidelines provide a standard to which customers can assess the security and quality of software systems they evaluate for purchase

and provide a way for vendors to explain their investment in software security in a manner that will drive sales. In other words, the concept of a secure system now has value because the word “secure” has meaning.

3. Application Source Code Review

Source code reviews (also known as code inspections) have been used for many years to reduce errors in program development. Code inspections performed to identify and eliminate security flaws leading to exploitable buffer overflows and other vulnerabilities are referred to as source-code security “audits.” These audits can be effective in finding and eliminating problems that cannot be detected using existing tools. However, they are typically unstructured and rely largely on the experience and tenacity of the programmers performing the review.

Increasingly, application source code reviews are dictated. For example, Section 6.6 of the Payment Card Industry (PCI) Data Security Standard [2] requires that companies with stored credit card or other consumer financial data install application firewalls around all Internet-facing applications or have all the applications' code reviewed for security flaws. This requirement could be met by a manual review of application source code or the proper use of automated application source code analyzer tools.

The use of secure coding standards provides additional structure to application source code reviews, by defining a proscriptive set of rules and recommendations to which the source code can be evaluated for compliance.

4. CERT Secure Coding Standards

The Secure Coding Initiative in the CERT/Coordination Center is developing secure coding standards for C, C++, Java, and other languages using a wiki-based community process at www.securecoding.cert.org. The first of these standards to be completed, The CERT C Secure Coding Standard will be published this fall by Addison-Wesley [5]. Future revisions of this coding standard are being developed on the wiki, along with the initial versions of C++ and Java secure coding standards.

The CERT C Secure Coding standard can be used as a measure of software security by determining the degree to which a software system complies with the rules and recommendations in this standard. Again, compliance does not guarantee the absence of vulnerabilities, but it does guarantee the absence of coding errors that are commonly found to be the root causes of vulnerabilities.

The easiest way to validate code as compliant with the CERT C Secure Coding standard is to use a certified source code analysis tool.

Rules and recommendations in this standard are classified into three levels. Emphasis should be placed on conformance to Level 1 (L1) rules. Software systems

that have been validated as complying with all Level 1 rules are considered to be L1 Conforming. Software systems can likewise be assessed as Level 2 (L2), or fully conforming (L1 and L2) depending on the set of rules to which the system has been validated as conforming.

Conformance to secure coding rules must be demonstrated to claim compliance with the CERT C Secure Coding Standard unless an exceptional condition exists. If an exceptional condition is claimed, the exception must correspond to an exceptional condition defined by the standard and the application of this exception must be documented in the source code.

Compliance with recommendations is not necessary to claim compliance with this standard. It is possible, however, to claim compliance with recommendations in cases in which compliance can be verified.

4.1. Deviation Procedure

Strict adherence to all rules is unlikely. Consequently, deviations associated with individual situations are permissible.

Deviations may occur in response to circumstances which arise during the development process, or for a systematic use of a particular construct in a particular circumstance. Systematic deviations are usually agreed upon at the start of a project.

For these secure coding rules to have authority, it is necessary that a formal procedure be used to authorize these deviations rather than an individual programmer having discretion to deviate at will. The use of a deviation must be justified on the basis of both necessity and security. Rules that have a high severity and/or a high likelihood require a more stringent process for allowing a deviation than rules and recommendations with a low severity that are unlikely to result in a vulnerability.

Software developers must be able to produce documentation as to which systematic and specific deviations have been permitted during development to request a claim of compliance with this standard.

4.2. Evaluation and Certification

The Secure Coding Initiative is developing a process to evaluate and certify conformance to CERT Secure Coding Standards.

The process consists of a comprehensive code review. This review relies heavily on static analysis. Among other tools, the Secure Coding Initiative is using the Compass/ROSE compiler and static analysis suite, which has been extended to check for rules contained within the CERT C Secure Coding Standard. As the

scope of the review is only the standard, output from tools not related to the standard is ignored.

Given the limitations of current static analysis tools, as well as the limitations of static analysis in general, manual inspection is also used sparingly to assess compliance with rules that are not amenable to automated analysis.

For each rule and recommendation, the source code is certified as: provably-nonconforming, deviating, conforming, and provably conforming.

- The code is provably nonconforming if one or more violations of a guideline are discovered for which no deviation has been specified.
- Deviating code is code for which the application developer has a documented deviation. This documentation will be included with the certification.
- The code is conforming if no violations between the code and the rule could be determined.
- Finally, the code is provably conforming if the code has been verified to adhere to the rule in all possible cases.

It is possible for the code to still be considered either conforming or provably conforming with specific deviations, provided the explanation for these deviations is satisfactory.

Once the process is completed, a report detailing the conformance or nonconformance for each CERT C Secure Coding rule is provided to the customer. Along with the conformance classification, this report includes the file name and applicable line numbers, in the event an infraction is detected. For each provably nonconforming guideline, the identification number is provided to assist in repairing the defect.

The entire process is intended to be all-inclusive, with each submission of the code by the client being considered a separate review. However, the process can also be considered iterative as there is an expectation that the client will continue to submit code until compliance is achieved on all possible rules.

5. Summary

Secure coding standards have an important and vital role to play in the development of secure software systems. In addition to providing guidance to developers, they provide a metric for assessing qualities of the code that promote security. While this does not guarantee the security of the system, it does provide a useful measure when evaluating and purchasing software systems that claim to be secure. As such, secure coding standards could become a market enabler in helping suppliers and customers alike assess the value of compliance.

6. References

[1] Robert J. Ellison, Nancy R. Mead, Gary McGraw, Sean Barnum, Julia H. Allen. *Software Security Engineering: A Guide for Project Managers*. Addison-Wesley, May 2008.

[2] PCI Security Standards Council. *Information Supplement: Payment Card Industry Data Security Standard (PCI DSS) Requirement 6.6 Code Reviews and Application Firewalls*. February 2008.

[3] Seacord, Robert C. *Secure Coding in C and C++*. Boston, MA: Addison-Wesley, 2005.

[4] Seacord, Robert C. "Secure Coding Standards." Static Analysis Summit . NIST Special Publication 500-262 [Online]. Gaithersburg, MD: NIST, 2006. 14-16 Available: http://samate.nist.gov/docs/NIST_Special_Publication_500-262.pdf

[5] Seacord, Robert C. *The CERT C Secure Coding Standard*. Boston, MA: Addison-Wesley, 2008.

Robert C. Seacord leads the Secure Coding Initiative at the CERT Coordination Center (CERT/CC) at the Software Engineering Institute (SEI) in Pittsburgh, PA. The CERT/CC, among other security related activities, regularly analyzes software vulnerability reports and assesses the risk to the Internet and other critical infrastructure.

Robert is an adjunct professor in the Carnegie Mellon University School of Computer Science and in the Information Networking Institute and part-time Faculty at the University of Pittsburgh. An eclectic technologist, Robert is author of three previous books, *Secure Coding in C and C++* (Addison-Wesley, 2005), *Building Systems from Commercial Components* (Addison-Wesley, 2002) and *Modernizing Legacy Systems* (Addison-Wesley, 2003), as well as more than 40 papers on software security, component-based software engineering, Web-based system design, legacy-system modernization, component repositories and search engines, and user interface design and development. Robert started programming professionally for IBM in 1982, working in communications and operating system software, processor development, and software engineering. Robert also has worked at the X Consortium, where he developed and maintained code for the Common Desktop Environment and the X Window System. He represents CMU at PL22.11 (ANSI "C") and is a technical expert for the JTC1/SC22/WG14 international standardization working group for the C programming language.