Software Engineering Institute

A Framework for Modeling the Software Assurance Ecosystem: Insights from the Software Assurance Landscape Project

Lisa Brownsword
Carol C. Woody, Ph.D.
Christopher J. Alberts
Andrew P. Moore

August 2010

TECHNICAL REPORT
CMU/SEI-2010-TR-028
ESC-TR-2010-028

Research, Technology, and System Solutions (RTSS) Program
Networked Systems Survivability (NSS) Program
Acquisition Support Program (ASP)

http://www.sei.cmu.edu

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

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

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.

Table of Contents

Acknowledgments vii

Abstract ix

1 Introduction 1
- 1.1 Background 1
- 1.2 SEI Software Assurance Landscape Project 1
- 1.3 Purpose and Structure of this Report 2

2 Framework Overview 5
- 2.1 Context for the Assurance Modeling Framework 5
- 2.2 Information the Framework Should Address 6
- 2.3 Structure of the Modeling Framework 7
  - 2.3.1 Activity Categories 8
  - 2.3.2 Views 9
  - 2.3.3 Methods 10

3 Pilot of Assurance Modeling Framework 13
- 3.1 Scope of the Pilot 13
  - 3.1.1 Overview of the Selected Assurance Capability 14
  - 3.1.2 Selecting Assurance Solutions for the Pilot 15
- 3.2 Structure of the Pilot 16

4 Principal Perspectives and Influences View 17
- 4.1 Method Summary 17
- 4.2 Applying the Method 18
- 4.3 Observations 20

5 Value Exchanged View 21
- 5.1 Method Summary 21
- 5.2 Applying the Method 23
- 5.3 Observations 26

6 Potential Assurance Results View 29
- 6.1 Method Summary 29
- 6.2 Applying the Method 30
- 6.3 Observations 32

7 Motivations View 33
- 7.1 Method Summary 33
- 7.2 Applying the Method 34
- 7.3 Observations 35

8 Critical Behaviors View 37
- 8.1 Method Summary 37
- 8.2 Applying the Method 39
- 8.3 Observations 43

9 Adoption of Products View 45
- 9.1 Method Summary 45
- 9.2 Applying the Method 47

9.3 Observations 50

10 Future Drivers View 51
10.1 Method Summary 51
10.2 Applying the Method 52
10.3 Observations 55

11 Conclusions and Next Steps 57
11.1 Example Insights from the Assurance Capability Area Profile 58
11.2 Lessons Learned in Applying the Framework 59
11.3 Next Steps 60

Appendix A – Value Mapping 63

Appendix B – Driver Identification and Analysis 73

Appendix C – System Dynamics 77

Appendix D – Technology Transition Analysis 81

Appendix E – Strategic Alternatives Analysis 85

Glossary 89

References 93

List of Figures

Figure 1:	Conceptual Context of the Assurance Modeling Framework	5
Figure 2:	Assurance Modeling Framework	8
Figure 3:	Applying the Assurance Modeling Framework to a Specific Capability Area	13
Figure 4:	Conceptual Model for Identifying the Major Stakeholders	18
Figure 5:	Critical Context Analysis for CVE Support of Software Vulnerability Management	19
Figure 6:	Sample Value Map	22
Figure 7:	Value Map Notation	23
Figure 8:	Value Map for CVE (as of 31 March 2009)	25
Figure 9:	Value Map for CWE (as of 29 June 2009)	26
Figure 10:	Generic Six-Layer Model to Align Supplied Capabilities with Demand	30
Figure 11:	SoS Focus Analysis Alignment Model for CVE	31
Figure 12:	System Dynamics Notation Used in Abstract Models	38
Figure 13:	Reactive Product Vulnerability Patching	40
Figure 14:	Reactivity Degrading Long-Term Security Improvements	41
Figure 15:	Operational Community's Response to Product Vulnerabilities	42
Figure 16:	Reinforcing Security Orientation Due to CVE-Facilitated Vulnerability Comparison	43
Figure 17:	Jolly Model for Commercializing Technology [Jolly 1997]	46
Figure 18:	Axes of Uncertainty Identified in Pilot	52
Figure 19:	Scenario Characteristics Matrix for Team 1	54
Figure 20:	Implications of Paradigm Shift and Digital Natives	55

List of Tables

Table 1:	Key Questions the Framework is Designed to Answer	6
Table 2:	Map of Activity Category to Framework Questions	9
Table 3:	Summary of Views and Methods	16
Table 4:	Primary Stakeholders for CVE	19
Table 5:	Early Value-Exchange Criteria	24
Table 6:	Summary of Roles and Responsibilities for CVE Alignment Model	31
Table 7:	Driver Attributes	34
Table 8:	Candidate Drivers of Vulnerability Management	35
Table 9:	Summary of Subprocesses and Bridges	46
Table 10:	CVE Maturation and Adoption Timeline Using Jolly Model	47
Table 11:	Success Indicators for CVE	49
Table 12:	Success Factors for CVE	49
Table 13:	Coverage of Vulnerability Management for Pilot	58

Acknowledgments

The authors would like to acknowledge the time and contributions of Robert Martin (MITRE), Thomas Rhodes (National Institute of Standards & Technology [NIST]), Michael Kass (NIST), Elizabeth Fong (NIST), Paul E. Black (NIST), and Robert Seacord (Carnegie Mellon® Software Engineering Institute [SEI]) as we developed the value mapping models. We would also like to acknowledge Suzanne Garcia-Miller (SEI) for her time and contribution as we explored future trends and implications using the Strategic Alternatives Analysis method.

We also wish to thank our reviewers Philip Boxer, Linda Levine, and Nancy Mead for their frank comments and insights. Lastly, we would like to acknowledge Jeannine Siviy and John Goodenough for their vision and support to start this discovery path.

Abstract

This report describes the Carnegie Mellon® Software Engineering Institute (SEI) Assurance Modeling Framework. It also discusses an initial piloting of the framework to prove its value and insights gained from that piloting for the adoption of selected assurance solutions. The SEI is developing a way to model key aspects of assurance to accelerate the adoption of assurance solutions within operational settings for the U. S. Department of Defense (DoD) and other government organizations. As part of that undertaking, SEI researchers have developed an Assurance Modeling Framework to build a profile for an assurance capability area such as vulnerability management within an assurance quality such as security. The profile consists of many views developed using selected methods and models. From the analysis of these views, inefficiencies and candidate improvements for assurance adoption can be identified.

1 Introduction

1.1 Background

Today's operational environments are complex and dynamic. User needs and environmental factors are constantly changing, which leads to unanticipated usage, reconfiguration, and continuous evolution of practices and technologies. Operational requirements for software-reliant systems are often ambiguous, incomplete, or incorrect. New defects and vulnerabilities are continually discovered. In environments characterized by these conditions, the effects of complex interrelationships and dependencies among organizations are not well understood, and the incentives that drive people's behavior often form barriers to the adoption of assurance solutions for the software-reliant systems those organizations depend on.

As a result, available assurance solutions are not developed or transitioned quickly and efficiently to operational settings. The U. S. Department of Defense (DoD) defines system assurance as "the justified confidence that a system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle."[1] Building on this definition, software assurance deals with software's contribution to both system and system-of-systems (SoS) assurance. An *assurance solution* is a policy, practice, or technology that contributes to system assurance (i.e., to providing justified confidence that a system will function as intended and is free of exploitable vulnerabilities).

1.2 SEI Software Assurance Landscape Project

Improved operational assurance requires the development of assurance solutions, as well as their adoption and application in operational settings. In recent years, the range of available assurance solutions has increased as more organizations have begun developing them to address software assurance challenges. While a great deal of work has been undertaken to identify and catalog available assurance solutions, little information is available about what is needed to speed their adoption in operational settings. Adoption gaps, barriers, and incentives related to assurance solutions are not well-defined.

In addition, assurance solutions have traditionally been developed for highly structured, tightly controlled operational environments that have limited external connectivity. The degree of structure and control needed to effectively apply traditional assurance solutions will limit their broader use as the general need for net-centricity and interoperability increases. Improving software assurance for highly interconnected systems of systems will require broad adoption of new types of assurance solutions as well as the formation of new ways to evaluate the effectiveness of a broad range of available assurance solutions.

[1] Engineering for System Assurance, NDIA System Assurance Committee, 2008, www.acq.osd.mil/sse/pg/guidance.html

Evaluating and selecting effective options from the range of available assurance solutions can be a challenging endeavor. Assurance solutions can no longer be evaluated in isolation; instead, they must be examined in their contexts of use. Interrelationships and dependencies among organizations and assurance solutions need to be considered when making decisions about the formation, adoption, and usage of assurance solutions. In this report, we use the term *assurance ecosystem* to describe the broad range of interrelated elements that influence operational assurance, including organizations, decision makers, policies, practices, technologies, and people.

The goal of the Software Assurance Landscape Project is to create an Assurance Modeling Framework that can be used to accelerate the adoption of assurance solutions in operational settings. The framework is a way to characterize

- the current portfolio of organizations working in assurance
- assurance solutions (including those being planned, funded, developed, and used)
- the interrelationships among organizations and assurance solutions
- the relative contributions of organizations and solutions to operational assurance
- future trends and their potential impacts on operational assurance

In essence, the framework provides a way to look across the assurance ecosystem and examine the gaps, barriers, and incentives that affect the formation, adoption, and usage of assurance solutions.

There are numerous properties of software assurance, such as reliability and security. For our initial development and use of the framework, we selected the security property of assurance. Within the security property, we focused on vulnerability management, which is an important aspect of an organization's operational security strategy. Deployed software routinely contains defects, and these defects are considered vulnerabilities when they enable an attacker to gain unauthorized access to systems, software, or networks. Vulnerability management defines a security practice that is focused on the prevention, discovery, and correction of vulnerabilities in systems, software, and networks. To demonstrate the viability and usefulness of the framework, we focused its application on two assurance solutions related to vulnerability management: Common Vulnerabilities and Exposures (CVE®) and Common Weakness Enumeration (CWE™).

1.3 Purpose and Structure of this Report

This report describes the current version of the Assurance Modeling Framework and the results of its application to vulnerability management. A previous paper, "Value Mapping and Modeling SoS Assurance Technologies and Supply Chain" [Siviy 2009], described our initial work regarding the Assurance Modeling Framework. In a second paper, "The Landscape of Software Assurance—Participating Organizations and Technologies" [Woody 2009], we described our approach for developing the current version of the framework, the key elements of the framework's structure, and the reasoning underlying our choices when developing the framework's structure.

® CVE is a registered trademark of The MITRE Corporation.
™ CWE is a trademark of The MITRE Corporation.

Multiple audiences can use the information provided in this technical report. Senior decision makers in government and industry can use the framework to support policy and acquisition decisions. People from organizations that fund, develop, mature, and transition assurance solutions can use the framework to better understand how their work relates to work being performed by other organizations. People from the operational community can use the framework to better understand available assurance solutions. Finally, researchers within the software assurance domain can use insights provided in this report to better understand the gaps, barriers, and incentives affecting the formation, adoption, and usage of assurance solutions.

This technical report comprises eleven sections and five appendices. The first three sections provide background and overview for the Assurance Modeling Framework. The focus shifts from the structure of the framework to applying the framework in Section 4. The sections of this report are as follows:

- Section 1 explains the motivation for creating the Assurance Modeling Framework and the Software Assurance Landscape Project.
- Section 2 explains the current version of the framework.
- Section 3 presents an overview of the current set of activities for applying the framework.
- Sections 4 through 10 describe key aspects of the framework in greater detail.
- Section 11 discusses our conclusions and potential next steps.
- Five appendices provide details about our application of the framework.
- A glossary defines terms used in this report.

We recommend sections 1, 2, 3, and 11 for all readers. Those who would like to understand the framework in more detail will find sections 4 through 10 and the appendices of interest.

2 Framework Overview

This section describes the current version of the Assurance Modeling Framework. We explain the relationship of the modeling framework to the assurance ecosystem, the products produced through the use of the modeling framework, and the structure of the framework itself.

2.1 Context for the Assurance Modeling Framework

Software assurance involves many properties, such as security, reliability, and performance, that an operational system or SoS may need to provide. The assurance ecosystem—with decision makers, practices, practitioners, and technologies for all software assurance properties—is sizeable and complex. In addition, software assurance properties are not independent, and they often interact in unanticipated ways. These interactions among assurance properties are currently an area of active research. Analyzing the entire assurance ecosystem would be a huge task. Therefore, we devised an incremental approach to develop and apply the Assurance Modeling Framework for a particular assurance capability area, as shown in Figure 1.

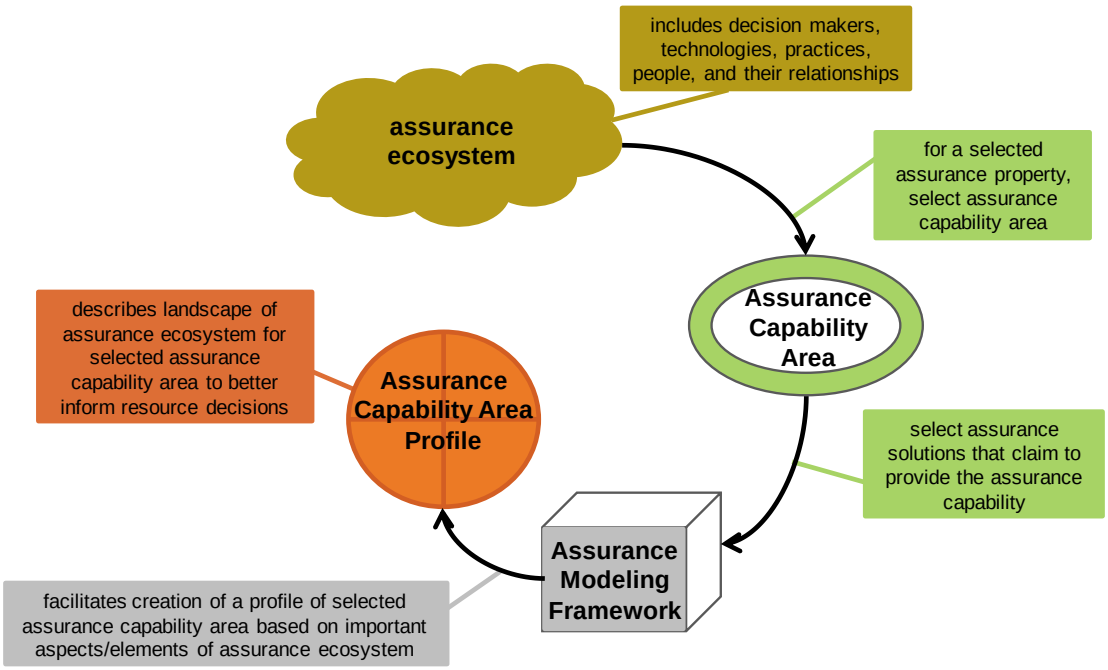


Figure 1: Conceptual Context of the Assurance Modeling Framework

Beginning with the entire assurance ecosystem at the top of Figure 1, we selected a single assurance property, security. We then narrowed the chosen assurance property to a single assurance capability area. Capability areas describe sets of related activities used to achieve an assurance property. For instance, the security property comprises many capability areas, including vulnerability management, incident management, and threat analysis. We selected vulnerability management as the Assurance Capability Area to be analyzed. We then used the Assurance Modeling Framework to create a profile of relevant elements of the assurance ecosystem. The framework provided a structure for applying several analysis methods to create a multi-dimensional Assur-

ance Capability Area Profile (denoted by a segmented circle). The profile describes the landscape of the assurance ecosystem for vulnerability management and provides information for relevant decision makers.

2.2 Information the Framework Should Address

The Assurance Modeling Framework is shaped by information that we have found necessary to characterize gaps, barriers, and incentives affecting the formation, adoption, and usage of collections of assurance solutions. What information should the framework capture in the resulting Assurance Capability Area Profile? To begin answering this question, three aspects of the problem are noteworthy:

1. **A single assurance solution in isolation does not address today's challenges of software assurance.** Rather, multiple assurance solutions are needed. This implies the importance of understanding the interrelationships of collections of assurance solutions within the assurance demands of an operational environment.
2. **Organizations are tightly bound to particular assurance solutions.** This implies that understanding a selected part of the assurance ecosystem requires insight into the interactions among organizations and assurance solutions.
3. **Technology and assurance demands are not static.** Changes in demands may be unpredictable both with respect to timing and direction and may dramatically impact current assurance solutions. This implies gaining awareness of potential future trends and events and determining their potential impact.

Building on the above observations, we formed a list of the key questions that the modeling framework is designed to answer. (See Table 1.)

Table 1: Key Questions the Framework is Designed to Answer

1	How is software assurance value defined for a selected context?
2	Who/what are the participating organizations[2] and assurance solutions?
3	What are the elements of value exchanged among participating organizations and assurance solutions?
4	How do participating organizations and assurance solutions work together to achieve operational assurance?
5	What are the drivers and motivations of participating organizations?
6	What are the critical usage scenarios and behaviors among the participating organizations and assurance solutions?
7	What are the adoption and operational usage mechanisms available for the assurance solutions? How are they aligned with organizational contexts and needs?
8	What is the impact of future trends and events on participating organizations and assurance solutions?
9	What patterns of possible inefficiencies affecting the formation, adoption, and usage of assurance solutions can be identified?
10	What are candidates for improvements? What could be the impact, if implemented?

[2] We use the term *participating organizations* to refer to the breadth of stakeholders that are associated with an assurance solution, such as vendors, suppliers, integrators, researchers, those who fund, customers, and users.

Questions 2, 3, and 4 readily fall out of the three observations listed above. They, along with question 1, provide a basic understanding of who and what assurance solutions are involved in an assurance capability area, their interrelationships, the elements of value exchanged (as it relates to assurance), and how operational assurance is achieved. Questions 5 and 6 expand the basic picture to include the influence of motivations, expectations, and behaviors among the collection of participating organizations and assurance solutions. Question 7 focuses on the adoption and usage characteristics of assurance solutions and their appropriateness for the context. Question 8 has a direct corollary to the observations made about the formation, adoption, and usage of collections of assurance solutions. Questions 9 and 10 represent the ultimate goals for applying the modeling framework.

2.3 Structure of the Modeling Framework

This modeling framework provides an approach for systematically gaining and analyzing the required information within an assurance capability area. The general structure of the framework is shown in Figure 2. The modeling framework is composed of multiple *activity categories* (indicated by rounded rectangles). For example, *Determine Context and Scope* is the first activity category at the top of Figure 2. Each activity category provides insights on one or more of the framework information questions and produces one or more *views* (indicated by rounded capsules). Continuing with the example, the *Determine Context and Scope* activity category produces the Principal Perspectives & Influences view. Each view is a collection of models and data formed using one or more *methods* (indicated by rectangles). The method that forms the Principal Perspectives & Influences view is the **Critical Context Analysis** method. A *profile* is a set of views that collectively describe the relevant elements of the assurance ecosystem landscape for the selected assurance capability area. (As the preceding discussion shows, we follow some typographic conventions when referring to the element: *activity categories* are shown in *italics*, views are shown in regular typestyle; and **methods** are shown in **boldface**.**)**

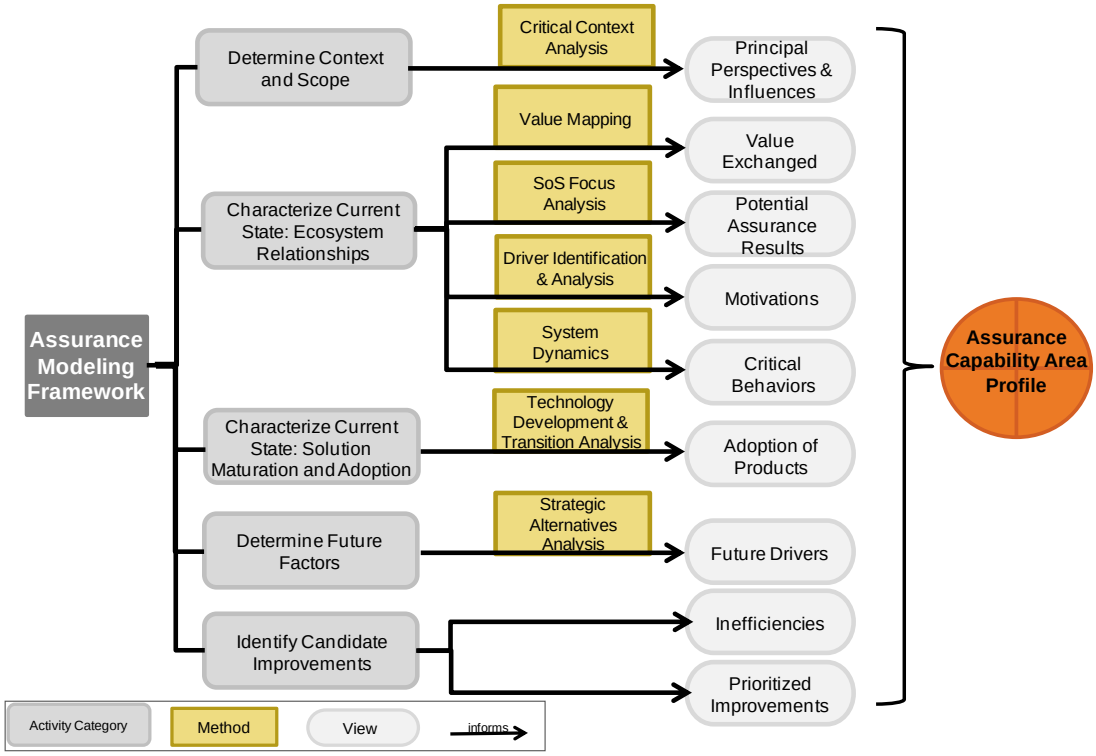


Figure 2: Assurance Modeling Framework

We next describe each of the three elements of the framework: activity categories, views, and methods.

2.3.1 Activity Categories

The modeling framework identifies five required activity categories:

- *Determine Context and Scope* provides the big picture and general scope. What is the assurance capability area to be analyzed and to what granularity? Who are the major groups of organizations forming, adopting, and using particular assurance solutions? Why is the selected assurance capability area important to each group of participating organizations?
- *Characterize Current State: Ecosystem Relationships* provides a more detailed understanding of the current participating organizations, assurance solutions, and the relationships within the assurance ecosystem. How does it all work today?
- *Characterize Current State: Solution Maturation and Adoption* provides an understanding of the current state of the formation, maturation, adoption, and use of assurance solutions. How do they work today?
- *Determine Future Factors* gains an understanding of potential future factors such as operational business and mission needs, technologies, economic, political, or environmental. What might change? What might be the impact?
- *Identify Candidate Improvements* generates the ultimate objectives. What are the inefficiencies in forming, adopting, and using assurance solutions? Where are major candidates for improvements? What could be the impact, if implemented?

Each activity category is focused on addressing particular framework questions. Table 2 shows the *primary* focus for each activity category. Note that multiple activity categories are needed to address some of the questions. Additional activity categories may be needed as further work with the framework uncovers other ways to analyze content relevant to an assurance capability area.

Table 2: Map of Activity Category to Framework Questions

Activity Category	Framework Questions
Determine Context and Scope	1. How is software assurance value defined for a selected context?
	2. Who/what are the participating organizations and assurance solutions? [high-level]
Characterize Current State: Ecosystem Relationships	2. Who/what are the participating organizations and assurance solutions? [provides further detail]
	3. What are the elements of value exchanged among participants?
	4. How do collections of participating organizations and assurance solutions work together to achieve operational assurance?
	5. What are the drivers and motivations of participating organizations?
	6. What are the critical usage scenarios and behaviors among the participating organizations and assurance solutions?
Characterize Current State: Solution Maturation and Adoption	7. What are the adoption and operational usage mechanisms available for assurance solutions? How are they aligned with organizational context and need?
Determine Future Factors	8. What is the impact of future trends and events on participating organizations and assurance solutions?
Identify Candidate Improvements	9. What patterns of possible inefficiencies affecting the formation, adoption, and usage of assurance solutions can be identified?
	10. What are candidates for improvements? What could be the impact, if implemented?

2.3.2 Views

As seen in Figure 2, each activity category generates one or more views. The framework currently consists of nine views, each representing a particular aspect of relevant information, models, and associated analyses. The views in the framework include

- Principal Perspectives and Influences—captures the broad context for the selected assurance capability area and characterizes the critical stakeholders and primary relationships
- Value Exchanged—captures the interrelationships and high-level value exchanged among pairs of participating organizations and assurance solutions
- Potential Assurance Results—captures and characterizes the ways in which assurance solutions align with what operational users do to achieve operational assurance results and identify high-level gaps and inefficiencies
- Motivations—captures and evaluates drivers[3] that are critical to achieving operational assurance objectives

[3] Drivers refer to critical circumstances or situations that strongly influence an outcome or result.

- Critical Behaviors—captures the causal relationships among collections of participating organizations and assurance solutions to identify primary variables of interest and their influences that drive critical behaviors
- Adoption of Products—captures the maturation and adoption mechanisms used and their effectiveness for collections of related assurance solutions
- Future Drivers—captures a range of future trends, influences, and uncertainties that may shape new operational demands and assurance solutions
- Inefficiencies—captures patterns of possible inefficiencies or gaps in assurance solutions and in their adoption and usage
- Prioritized Improvements—captures candidate improvements and their relative priorities

Although not depicted on the diagram, views are interrelated. Information, models, and analysis elements associated with a particular view can be used in the formation of other views.

2.3.3 Methods

The methods used by the framework are intimately tied to a specific view. A method examines an assurance capability area from a distinctive point of view. The methods currently included in the framework were selected according to their applicability for large-scale environments with social and technical elements, such as those found with systems of systems, and ready access by the SEI team to knowledgeable practitioners. Other comparable methods could be substituted.

The current framework uses seven methods:

- **Critical Context Analysis**[4] rapidly reveals and characterizes key stakeholders. The method also elicits the major operational scenarios associated with a domain, along with the primary stakeholder responsibilities and relationships.
- **Value Mapping**[5] provides a visual representation of the interactions between organizations. For a specified assurance solution, each participating organization interacts with a subset of other participating organizations based on some perceived software assurance needs, and each interaction involves an exchange of something of value. Value exchanges can take many forms, such as funding, product information, or governance.
- **SoS Focus Analysis**[6] examines in what ways suppliers of assurance solutions provide capabilities or services and how these are composed and synchronized such that operational users achieve operational assurance results. This method begins to identify potential areas of inefficiencies that are analyzed further.

---

[4] Critical Context Analysis is based on the work of Philip Boxer and draws on the projective analysis methods of Boxer Research Ltd (BRL). Permission to use Projective Analysis (PAN) technology is under license from BRL. For more information on Critical Context Analysis, go to http://www.sei.cmu.edu/interoperability/consulting/sos/criticalcontext.cfm.

[5] Value Mapping as described in an article by Green and Jack [Green 2004].

[6] SoS Focus Analysis is based on the stratification work of Philip Boxer and draws on the projective analysis methods of Boxer Research Ltd (BRL). Permission to use Projective Analysis (PAN) technology is under license from BRL. For more information about SoS Focus Analysis, go to http://www.sei.cmu.edu/interoperability/consulting/sos/sosfocusanalysis.cfm.

- **Driver Identification and Analysis**[7] establishes the key objectives for the assurance capability area being analyzed, the critical factors, or drivers, which influence achievement of those objectives, and in what ways these influence an outcome. This method determines the likelihood that a set of activities will produce a desired outcome.
- **System Dynamics**[8] creates models to understand the critical behavior within a socio-technical domain—that is, how technologies and organizational structures in a domain work together to achieve or inhibit a goal. This method encourages the additional inclusion of soft factors in the model, such as policy, procedural, administrative, or cultural factors that are often key to understanding a particular situation.
- **Technology Development and Transition Analysis**[9] captures maturation, adoption, and usage information over time for assurance solutions. Most technology maturation and adoption methods are oriented toward a single technology. This is an experimental approach to understand maturation and adoption within a context of collections of related technologies.
- **Strategic Alternatives Analysis**[10] provides a picture of key trends, implications, and "watch-points" about current and future elements that may impact an enterprise or community of interest. The method characterizes and explores contrasting trends and uncertainties using scenarios as a means of understanding the potential impact of sociological, technological, political, economic, cultural, and environmental changes.

We describe the methods and their use in our pilot of the modeling framework in sections 4 through 10. The following section explains our initial application of the framework to vulnerability management.

---

[7] Driver Identification and Analysis is based on the work of Alberts and Dorofee on characterizing the success for specific programs [Alberts 2009].

[8] System Dynamics as described in *Business Dynamics* [Sterman 2000].

[9] Technology Development and Transition Analysis draws from Jolly's model and process for commercializing technology [Jolly 1997] and from the SEI's life-cycle model for guiding the maturation and transition of its products [Forrester 2003].

[10] Strategic Alternatives Analysis is based on the work of Schwarz and Van Der Heijden's scenario-based planning techniques that rely on intuitive logics [Schwarz 1996, Van der Heijden 2005].

3 Pilot of Assurance Modeling Framework

We considered our initial application of the Assurance Modeling Framework to be a pilot showing how it could be used. Our goal for the pilot was to establish the viability of the framework and confirm its utility in understanding the relationships of assurance solutions to implemented assurance results for a selected assurance capability area. To meet this goal, the pilot would show that we can identify potential levers for change with a better understanding of how constituent elements (e.g., organizations and assurance solutions) interoperate to achieve (or at times, hinder) software assurance. In this section, we summarize the scope of the pilot and its structure.

3.1 Scope of the Pilot

To start validating the framework, we used the context schematic for the Assurance Modeling Framework, as shown in Figure 3, to select *security* as the assurance property of interest from the assurance ecosystem. The SEI has extensive background and experience in security, which accelerated the piloting effort. Within the security property, we selected *vulnerability management* as the assurance capability area because of its importance as a practice area for security. It is identified as a key practice in both the Building-Security-In Maturity Model[11] (BSIMM) and the OWASP Software Assurance Maturity Model[12] (OpenSAMM), recently released models describing how organizations are addressing security and assurance for software.

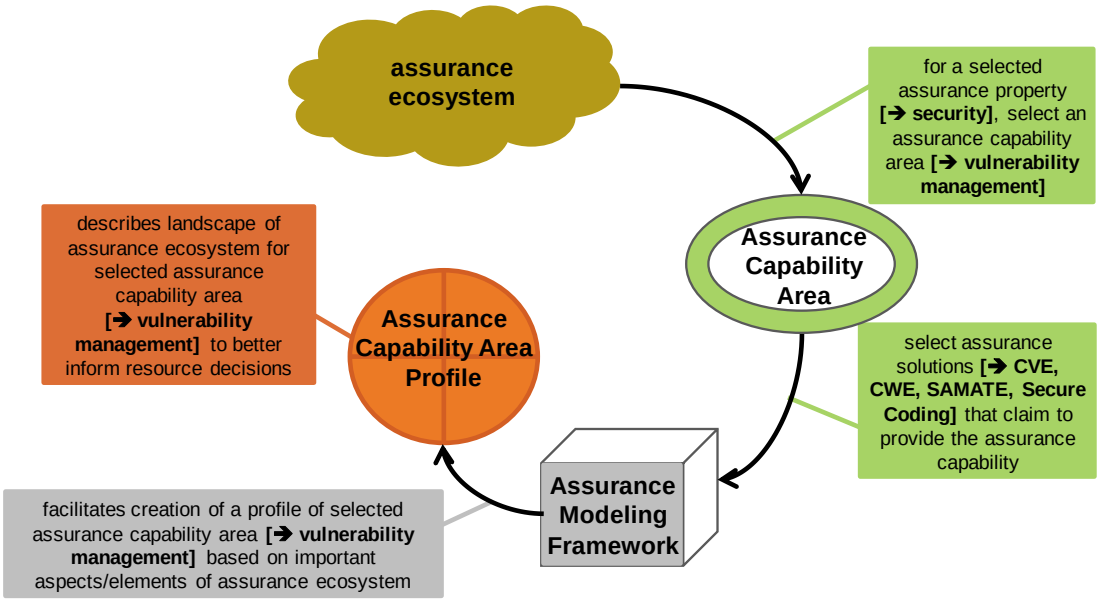


Figure 3: Applying the Assurance Modeling Framework to a Specific Capability Area

[11] More information on the BSIMM is provided at http://bsi-mm.com/ssf/.

[12] More information on the OpenSAMM is provided at http://www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model.

Within this segment of the report, we provide an overview of vulnerability management for readers who are not familiar with this capability followed by a description of the elements selected from the assurance ecosystem to use in the framework to represent this capability area.

3.1.1 Overview of the Selected Assurance Capability

Software is rarely defect free, and these defects are considered to be *vulnerabilities* if they allow someone to gain unauthorized access to software, a system, or a network. The intentional exploitation of such defects is referred to as an *attack.* Attackers may also gain unauthorized access through an exposed weakness resulting from an incorrect configuration or software error that discloses information about the system or network and that can be used as a stepping stone to a further attack. As a result of an attack, the security policies for the system or network may be compromised, allowing the attacker to access information that should be protected (confidentiality failure), change or destroy protected data (integrity failure), or block authorized users from accessing and using the system or network (availability failure such as denial of service).

Vulnerability management is built around a process of prevention, discovery, and correction. Currently, discovery and correction are heavily emphasized. The response to an attack requires identification of the way in which an attacker successfully violated the security policy and correction of the software defect (via a *patch*) or configuration error that allowed the attack to succeed. If the defect cannot be eliminated completely, deterrents to make it harder for the attacker to succeed are needed. These efforts to correct and deter an attacker are referred to as *mitigations.* New vulnerabilities are discovered daily, and attacks are growing in sophistication as attackers develop better tools and gain experience with the constantly expanding layers of technology that are becoming ubiquitous in all operational environments. As organizations become more dependent on technology, the potential impact of such attacks grows. A wide range of operational tools are in use to monitor the operational environment to identify attacks, scan for vulnerabilities, and support patch management.

Many libraries of information about vulnerabilities and appropriate mitigations have been collected and shared since 1988 when a software defect, referred to as the "Morris Worm," was triggered by William Morris and used to successfully attack a majority of the systems on the Internet. None of these libraries achieved wide acceptance as a standard way of characterizing vulnerabilities and mitigations. The Common Vulnerabilities and Exposures (CVE) list was conceived and established to provide a consistent structure for naming vulnerabilities and exposures, so that information from numerous existing libraries could be assembled, cross-referenced, and used more effectively.[13] Its CVE Identifiers uniquely tag publicly known information security vulnerabilities and exposures. The unique identifiers (1) enable data to be exchanged between security products and (2) provide a baseline set of vulnerabilities that can be used to evaluate the security posture of a technology component and the coverage of vulnerability tools and services in finding vulnerabilities. It is an essential part of supplying information about vulnerabilities to the operational community. The CVE list primarily supports discovery and correction. Vendors can specify what

[13] More information on CVE is provided at http://cve.mitre.org/.

CVEs are addressed with each patch, and information repositories about vulnerabilities can be cross-referenced using the unique CVE Identifiers.

Many types of vulnerabilities such as "improper input validation"[14] result from coding mistakes that a software developer did not correct. Since software has many errors and not all of them can be corrected within the resources available, it is important to flag, for special consideration, the types of errors that could become vulnerabilities. The Common Weakness Enumeration (CWE) is a dictionary that provides a description of each coding and design error and the ways in which it should be corrected.[15] CWE is built from the knowledge gained in applying CVE in the detection and correction of vulnerabilities. CWE is structured to address vulnerability prevention. Coding techniques to avoid weaknesses that lead to vulnerabilities have been developed for some coding languages (C++ and Java) [Seacord 2005].

Software companies are developing tools (called static analysis tools) to locate coding errors, and the CWE dictionary provides a means of explaining the types of problems each tool can address. The Software Assurance Metrics and Tool Evaluation (SAMATE)[16] project is structuring testing approaches that will allow validation and comparison of the static analysis tools used by developers to discover security coding errors for CWE coverage.

3.1.2 Selecting Assurance Solutions for the Pilot

To select a representative and manageable set of vulnerability-related organizations and assurance solutions for the framework pilot, we characterized vulnerability management using two general responses to vulnerabilities: reactive and proactive. *Reactive vulnerability management* waits until vulnerability is detected and then responds. Typical actions include vulnerability identification, determination of potential impact, and taking action to correct or mitigate these impacts. The selected technology associated with reactive vulnerability management is CVE, managed by MITRE.

In contrast, *proactive vulnerability management* seeks to prevent or significantly reduce the impact of vulnerabilities by addressing weaknesses in the software that contribute to vulnerabilities prior to their detection in operational settings. The chosen technologies[17] associated with proactive vulnerability management are CWE, also managed by MITRE; SAMATE, managed by the U.S. National Institute of Standards and Technology (NIST); and the SEI's Secure Coding Initiative [Seacord 2005].

---

[14] "Improper input validation" enables an attacker "to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution" [MITRE 2009a]. Improper input validation is the highest ranked software error on the Common Weakness Enumeration's Top 25 list of the most dangerous programming errors [MITRE 2009b].

[15] More information on CWE is provided at http://cwe.mitre.org/.

[16] More information on SAMATE is provided at http://samate.nist.gov/index.php/Main_Page.html.

[17] We apply the definitions from Merriam-Webster: (1) practical application of knowledge especially in a particular area; (2) a manner of accomplishing a task using technical processes, methods, or knowledge.

3.2 Structure of the Pilot

Each element of the framework was exercised but with varying degrees of coverage. Our goal was breadth rather than depth. We initially placed greater emphasis on the activity categories *Determine Context and Scope* and *Characterize Current State: Ecosystem Relationships*. We worked with each view and its associated method sufficiently to determine the kind of information that could be obtained and the relevance and utility of the information to the activity category.

Table 3 provides a quick-reference list of the views and the methods used with each view. Each of the following views and the method used in its execution are described in more detail in the indicated section of the report.

Table 3: Summary of Views and Methods

Framework View	Method Used	Report Section
Principal Perspectives and Influences	Critical Context Analysis	4
Value Exchanged	Value Mapping	5
Potential Assurance Results	SoS Focus Analysis	6
Motivations	Driver Identification and Analysis	7
Critical Behaviors	System Dynamics	8
Adoption of Products	Technology Development and Transition Analysis	9
Future Drivers	Strategic Alternatives Analysis	10

Each section provides

- the objective of the framework view
- a summary of the method associated with the view
- a summary of how the method was used in the pilot and the resulting view
- a set of observations about the method and how it contributes to the view

4 Principal Perspectives and Influences View

The Principal Perspectives and Influences view is associated with the activity category *Determine Context and Scope*, which provides the big picture and general scope. For complex environments, gaining a high-level picture of the domain, general boundaries, the essential stakeholders, primary responsibilities, and critical relationships is crucial. Within a particular domain, many stakeholders are known only to a portion of other stakeholders because not every stakeholder interacts directly with all others. Often, the roles, responsibilities, and critical relationships with some stakeholders are largely hidden from others or only implicitly understood.

The objective for this view is to (1) identify and characterize, at a high level, the major groups of participating organizations and assurance solutions and (2) explain why the selected assurance capability area is important to each group of participating organizations. This view helps to answer the following framework questions:

- How is software assurance value defined for a selected context?
- Who are the key participants?

The modeling framework uses the **Critical Context Analysis**[18] method to implement this view.

4.1 Method Summary

Critical Context Analysis rapidly reveals and characterizes key stakeholders. The method also elicits the major operational scenarios associated with a domain, along with the primary responsibilities and relationships.

The domain of interest is the starting point for the **Critical Context Analysis** and sets the boundary for the analysis. Within this domain context, the analysts next identify and characterize the major stakeholders within the domain of interest through structured interviews with representatives from the domain of interest. The objective is to systematically uncover the kinds of major stakeholders associated with the following four perspectives:

- *what*: What do suppliers do? Who is involved?
- *how:* How do suppliers organize and constrain their essential capabilities? Who shapes how it is done?
- *for whom:* Whom are suppliers serving? What is the nature of their clients' operational needs?
- *why:* Who is affected? What is going on in the larger operational context that makes what suppliers do of value?

[18] Critical Context Analysis is based on the work of Philip Boxer and draws on the projective analysis methods of Boxer Research Ltd (BRL). Permission to use Projective Analysis (PAN) technology in this analysis is under license from BRL. More information on Critical Context Analysis is provided at http://www.sei.cmu.edu/interoperability/consulting/sos/criticalcontext.cfm.

Figure 4 shows a conceptual model of the stakeholder perspectives. To gain a balanced picture for the selected domain, the analysts ensure that stakeholders from both the *supply side* (associated with stakeholders who provide capabilities or services) and the *demand side* (associated with operational stakeholders who are in the operational space and experience the business or mission demands) are represented. "What" and "how" reflect supply side perspectives and "for whom" and "why" reflect demand side perspectives.

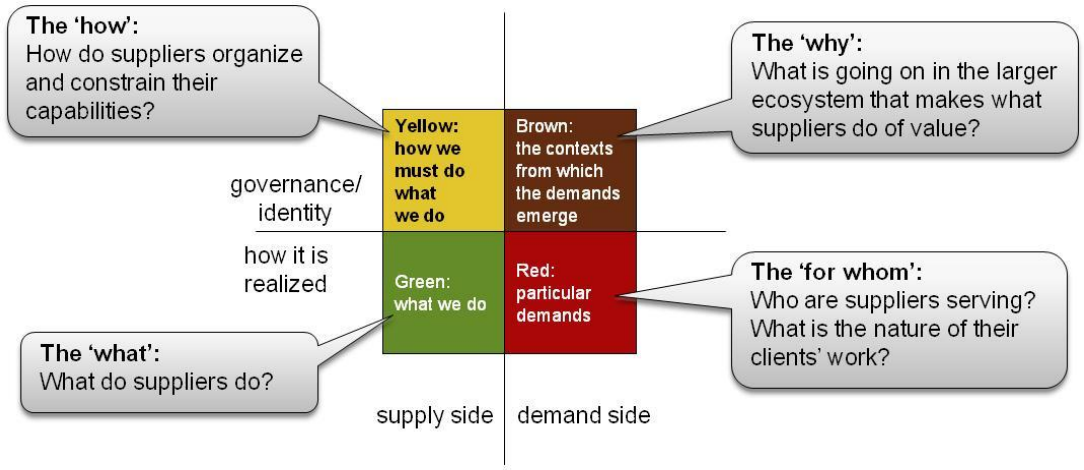


Figure 4: Conceptual Model for Identifying the Major Stakeholders

Throughout the interviews, the analysts populate the quadrants with actual stakeholders or groups of stakeholders and capture the one or two primary roles and responsibilities. Initial relationships are also identified and characterized.

4.2 Applying the Method

For the framework pilot, we used the management of vulnerabilities as supported by CVEs as the domain of interest for our analysis. Using questions similar to those shown in the conceptual model in Figure 4, we identified the primary stakeholders for CVEs. Table 4 lists the groups of stakeholders associated for the quadrants.

Table 4: Primary Stakeholders for CVE

Quadrant in Conceptual Model	Stakeholder Group	Specific Stakeholders
What	Suppliers	• Software application vendors • Software security product vendors
For Whom	Consumers	• IT operations; organizations that run and support computer installations • Site security analysts
How	Governance	• NIST National Vulnerability Database[19] (NVD) • MITRE CVE board
Why	Operational context	U.S. commercial, DoD, and other government operational organizations that rely on computers, networks, software applications, and data storage media to perform their missions

We then formed a critical context matrix with the four quadrants to capture the identified roles and responsibilities, as shown in Figure 5. Using several typical usage scenarios for vulnerability management and the matrix, we iteratively added and refined roles and responsibilities. For example, we revised the "for whom" to add the site analyst's role and responsibilities. This change also required modifications to the IT organization's responsibilities.

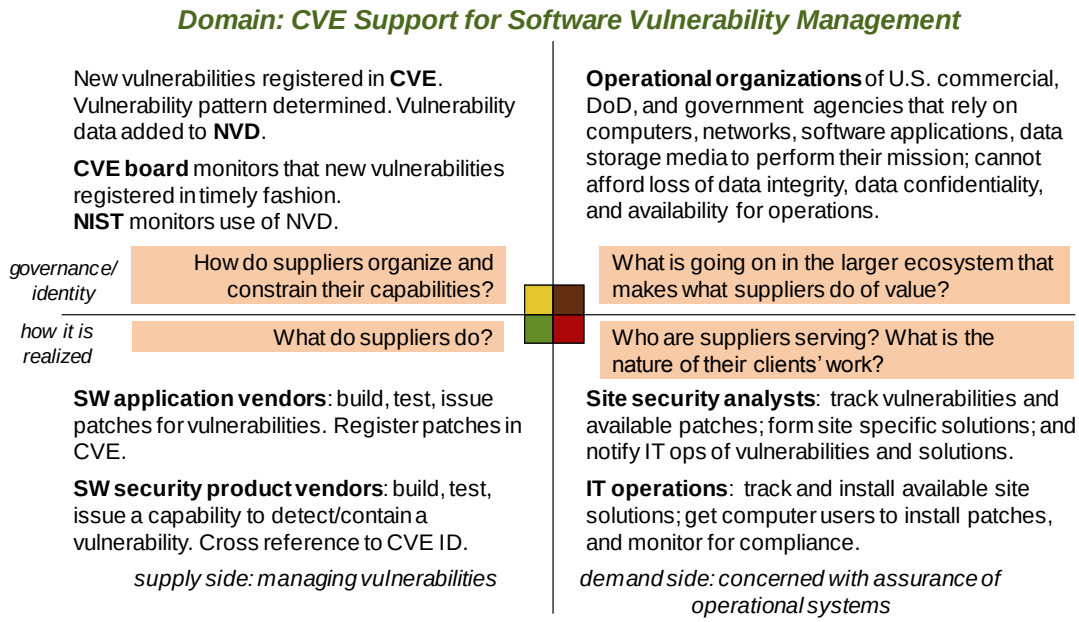


Figure 5: Critical Context Analysis for CVE Support of Software Vulnerability Management

[19] See glossary for description.

4.3 Observations

From the **Critical Context Analysis**, we first identified the tensions between reactive and proactive responses[20] by vendors and operational organizations to the management of vulnerabilities and, at a high-level, their influence on assurance solutions. These types of responses determine many of the major influences and relationships among the identified stakeholders, which were further characterized and analyzed in the other views, such as the Critical Behaviors view.

The results from the **Critical Context Analysis** identified groups of demand-side and supply-side organizations. This established who should be interviewed and revealed the elements (e.g., organizational entities, assurance solutions, and value exchanged) to include in the Value Mapping method. Thus, we gained a more balanced Value Exchanged view.

Critical Context Analysis also provided the initial characterization of how supplier and operational organizations think about and respond to vulnerabilities. This information gave insight into the tradeoffs and relationships based on the particular perspective (i.e., proactive or reactive). How organizations respond to vulnerabilities becomes an important consideration, since organizations provide funding based on their perspective. We then delved into these areas with the **Driver Identification and Analysis** in the Motivations view.

[20] Reactive and proactive vulnerability management are defined in Section 3.1.2.

5 Value Exchanged View

The Value Exchanged view is associated with the activity category *Characterize Current State: Ecosystem Relationships* which provides a detailed understanding of the current state of participating organizations, assurance solutions, and the relationships within the assurance ecosystem.

Determining software assurance solutions relevant to a selected assurance capability and the ways in which organizations participate in the software assurance ecosystem to address a selected capability is an extremely complex problem. The software assurance ecosystem includes organizations from government, academia, and industry. Participants are responsible for research and development, project management, system and software design, system and software engineering, software development, system integration, IT operations, policy and oversight, and compliance. Who interacts with whom and why? By selecting a specific assurance solution known to be relevant to a particular capability and identifying the organizations that own, contribute to, and use the selected assurance solution and the types of value exchanges involved in these interactions, we can identify patterns of interaction for analysis.

The objective for this view is to capture the interrelationships and high-level value exchanged among participating organizations and assurance solutions. This view helps to answer the following framework questions:

- Who/what are the participating organizations and assurance solutions?
- What are the elements of value exchanged among participants?

The modeling framework currently uses the **Value Mapping** method to implement the Value Exchanged view.

5.1 Method Summary

Value Mapping provides a visual representation of the *interactions* between organizations and assurance solutions. For a specified assurance solution, each organization interacts with a subset of other organizations based on their role(s) relative to the assurance solution, and each interaction involves an exchange of something of value. *Value exchanges* can take many forms, such as funding; products that include shared information and tools; services, such as consulting and training; approvals and controls for governance and compliance; and endorsements.

The interrelationships among organizations, the assurance-related value linked to the relationship, and the direction of the exchange can all be identified in a value map. Information from the **Critical Context Analysis** method provides indicators of which organizations will be participating in value exchanges.

Value Mapping depicts some quantification of an organizational value driver [Green 2004]. This technique has been used to measure various areas of organizational performance that do not readily translate into simple metrics [Jack 2002]. Use of this method requires determining who participates in the value exchange and what is being exchanged. As shown in Figure 6, *participants* are denoted by circles—are actors in the ecosystem that include assurance solutions, individuals, standards bodies, or organizations (e.g., commercial, government, or academic). If a generic term

is used to reference a participant, such as "Organization 2," an annotation is included, such as "Organization 2 is a consortium with representatives from government, academia, and industry." The annotation is not intended to be exhaustive. Elements of *exchanged value* are denoted by lines with arrows. From our analysis so far, we have identified six categories of exchanges that provide value to software assurance: funding, product, service, governance, compliance, and endorsement. Each exchange category, or type, is depicted by different line colors. A description of the content of the type of exchange is provided as an annotation to the line on the diagram. For example, data or reports may constitute a product exchange; and knowledge and effort could be elements of a service exchange.

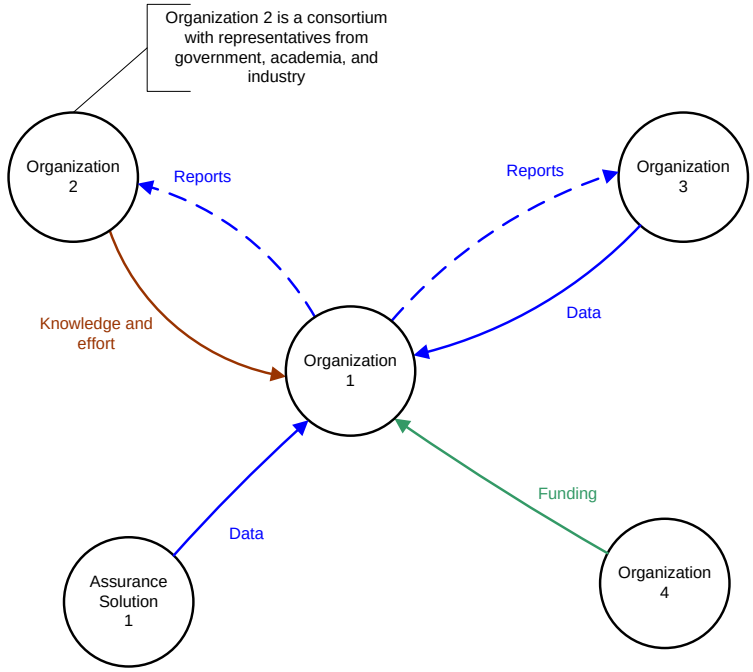


Figure 6: Sample Value Map

A short explanation of the element that describes the value exchanged appears on the line, such as "Data" exchanged between "Assurance Solution 1" and "Organization 1." Each value exchange includes a source from which the value of the exchange originates and a destination to which the value is provided. An arrow is used to denote the direction of a value exchange, pointing from the source to the destination. In addition, the diagram presents the initiator of each value exchanged using a specific convention. A solid line indicates that an exchange is initiated by the source, while a dotted line indicates it is initiated by the destination. Figure 7 summarizes the notation used in the value maps including the color coding.

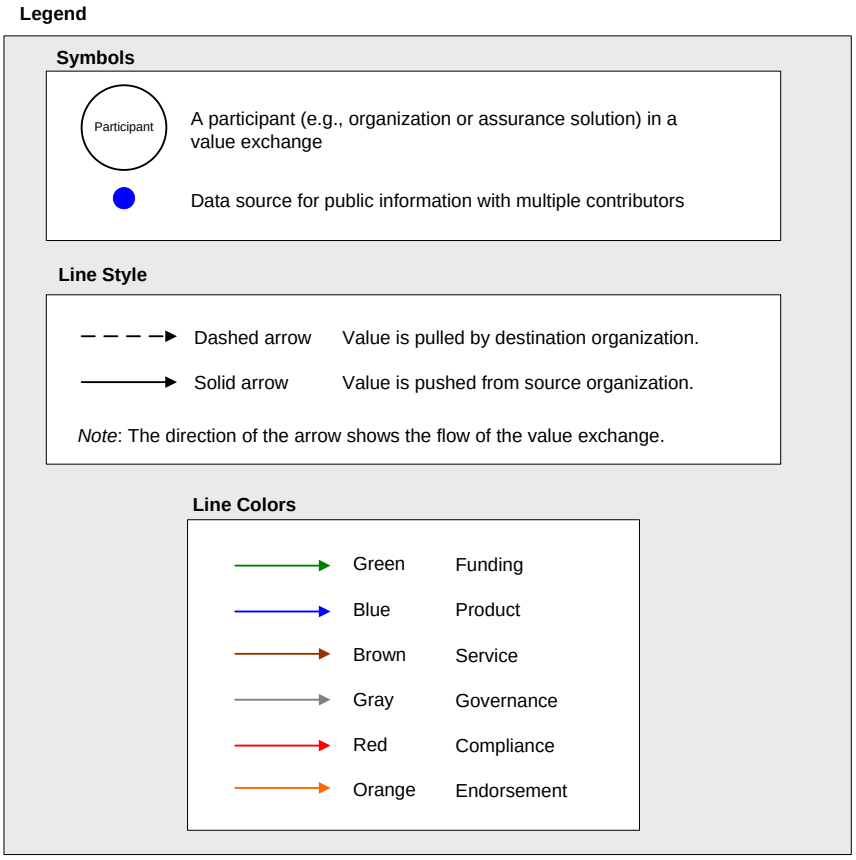


Figure 7: Value Map Notation

5.2 Applying the Method

Publically available information did not provide sufficient insight into the organizational relationships to determine their roles in the assurance ecosystem. Structured interviews with individuals leading the organizational assurance work were needed to provide useful detail to construct the view of the values exchanged.

The interactions shown in a value map are valid as of a specific point in time and reflect the perceptions of the assurance solution representatives providing the input. Templates to guide the interview process were developed and piloted with 20 different assurance solutions (see Appendix A). The selection of exchange types and diagram notation evolved from data gathered during these interviews.

The first diagrams were extremely cluttered, with all of the organizations connected to a capability and values identified, and thus provided little insight. We made an early attempt to determine the importance of each exchange (high, medium, low). In addition, we postulated criteria for considering each exchange type (see Table 5), but this level of measurement was primarily subjective and based on limited available data, which only increased the visual clutter and was soon discarded.

Table 5: Early Value-Exchange Criteria

Output Type	Criteria	Definition
Funding	Size	The amount of assurance funding per year
	Duration	The number of years for which assurance funding is provided
Services	Breadth	The number of different assurance services provided by an organization (e.g., audit, certification of products, evaluations, consulting, training)
	Volume	The quantity of assurance services provided by an organization (e.g., market share, annual sales, sales, installed base of underlying technology)
	Authority	The credibility of the organization with respect to the assurance services it provides (e.g., name recognition, experience and expertise, maturity of offerings, certification of skill by a third party)
Products	Breadth	The number of different assurance products provided by an organization
	Significance	The degree to which an organization's assurance products solve an important assurance problem
	Authority	The credibility of the organization with respect to the assurance products it builds (e.g., perceived penetration of market, perceived influence of products on consumers' practices and system performance)
Publications	Breadth	The range of publications provided by an organization (e.g., product information, best practices, standards, guidance)
	Significance	The degree to which an organization's assurance publications influences the behaviors of consumers or impacts consumers
	Authority	The credibility of the organization that is providing assurance publications
Governance	Breadth	The breadth of an organization's governance and oversight responsibilities with respect to assurance (e.g., policy, regulation, law, contract, decision authority, technical/progress review, budget and funding oversight, direction or mandate)
	Authority	The degree to which a constituency's actions regarding assurance are influenced or affected by governance and oversight (e.g., enforceability, penalty for noncompliance, degree of respect)

As a refinement, we selected a cluster of organizations addressing a single assurance solution (for example, the CVE list) for a selected capability. This provided a manageable level of content and proved useful for analysis.

We started with a typical usage scenario for the assurance solution (a reactive vulnerability response for vulnerability management), and then progressed to the next usage scenario (a proactive response for vulnerability management). As we examined a usage scenario, we captured the par-

ticipants and each type of value exchanged between each participant pair. Since each pair of organizations can have many types of exchanges, color coding was used to visually differentiate each line between them.

Several iterations were needed to refine the types of value that were relevant to software assurance and to clarify and refine the participants. We conducted reviews within the project team and with external parties while developing this view for the capability area. Figure 8 shows approximately half of the CVE value map. CVE is widely used and considered to be a standard for software assurance. For our external reviewers, we used a full-page, easy-to-read version of this map.

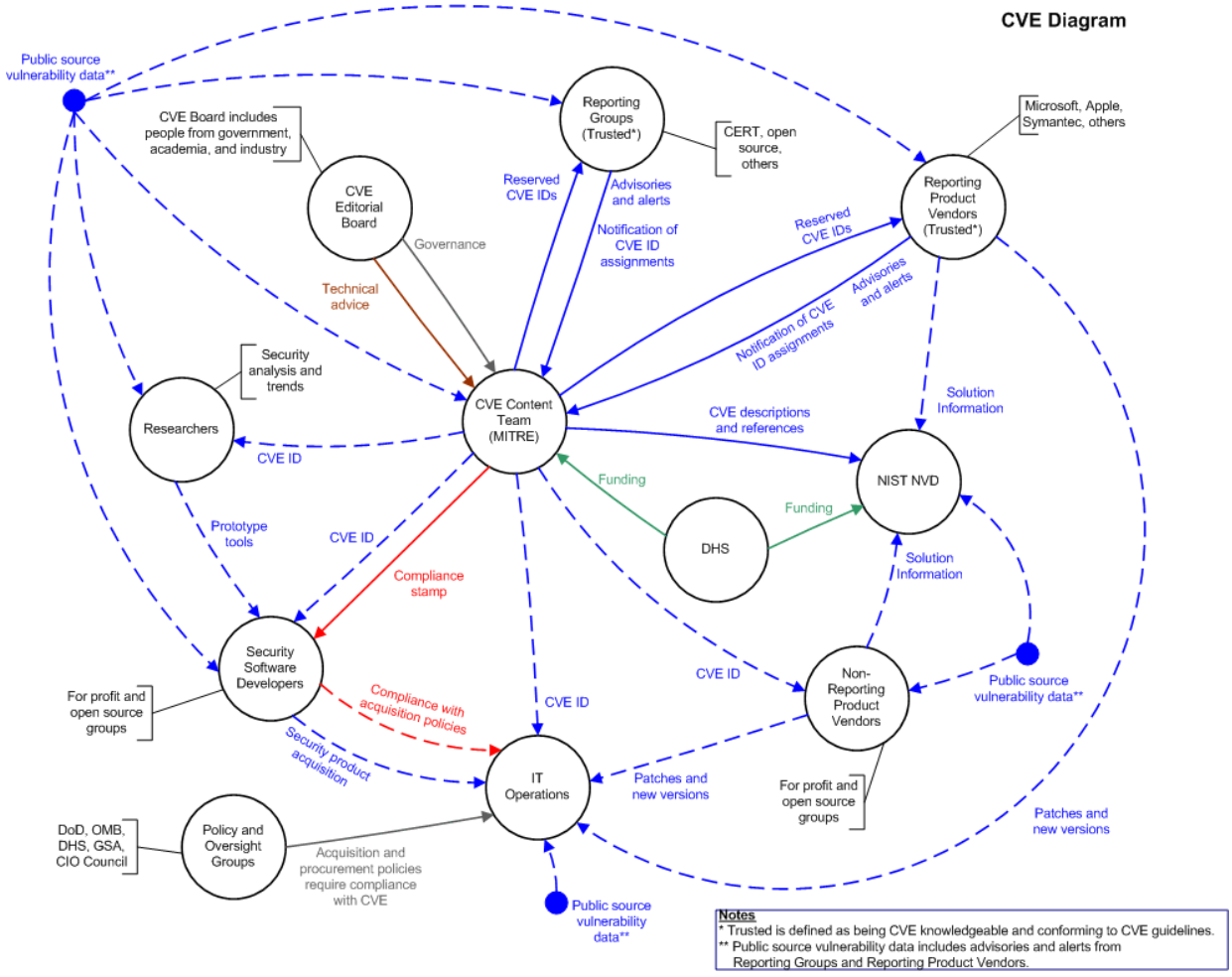


Figure 8: Value Map for CVE (as of 31 March 2009)

In contrast, Figure 9 shows the value map for CWE, which has not been in existence as long as CVE and is used by a more limited set of participants. Value maps for Secure Code and SAMATE are shown in Appendix A.

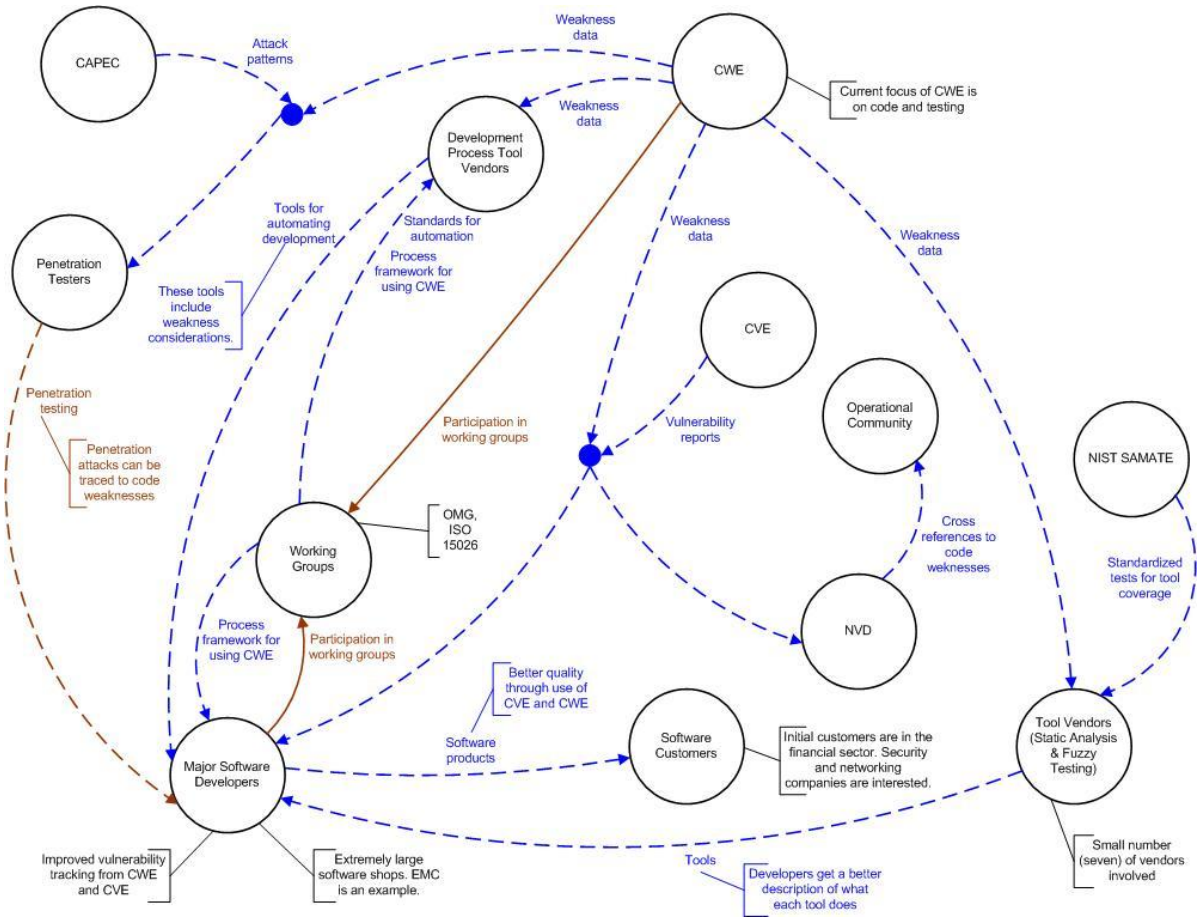


Figure 9: Value Map for CWE (as of 29 June 2009)

5.3 Observations

From the value maps, it was clear that the importance of an assurance solution resulted in a wider range of participants and value interactions. None of the assurance solutions functioned in isolation. The relative importance (and possibly the maturity) of an assurance solution were shown by the broader range of exchange types and the higher number of organization types participating in the realm of influence. For example, governance is part of the CVE value map, but does not yet appear in the CWE value map. We speculated that the ecosystem changed over time but did not spend the time creating value maps for different points in time. The **Technology Development and Transition** method (see Section 9) was included in the framework to build a view of changes over time.

The values identified in these organizational exchanges only have an indirect effect on operational software assurance. The value maps show a great deal of interaction among organizations but offer little insight into how the content from these interactions is used by each organization.

For CVE, organizations will most likely use the reported response information about each vulnerability to protect themselves, but the motivation for doing this and the value to the reporting organizations are unclear, and other views are needed to provide this understanding.

External participants were able to review and augment the value maps with minimal explanation. The visual structure is intuitive to assurance solution designers and builders. While each value map captures only a snapshot in time, external reviewers found that the diagrams provided a succinct way to describe the key organizational relationships for their assurance solution. Previously, this information was only tacitly understood. The maps supported in-depth conversations about software assurance with the solution owners, thus allowing the interviewees to better communicate their solution's contributions. Many interviewees were surprised by the number of layers of indirect connections that could be identified and captured. Also, it is possible to "zoom in" further on a specific relationship or subgroup of relationships to capture greater detail about the organizations and value exchanges. The value mapping models provide a useful way to describe an organization's roles in as assurance capability area.

Our use of value maps was limited by the tools available for construction. Visio performed well to support the visual examination of the information, but each review and update required extensive rebuilding of the diagram. Multiple diagrams were needed to show different points in time and different levels of granularity. A tool that would allow focusing in on subgroups of relationships to be explored in greater detail within the higher level view would be extremely useful for analysis and comparison.

6 Potential Assurance Results View

The Potential Assurance Results view is associated with the activity category *Characterize Current State: Ecosystem Relationships* which provides a detailed understanding of the current state of participating organizations, assurance solutions, and the relationships within the assurance ecosystem.

The objective for this view is to capture and characterize the ways in which assurance solutions align with what operational users do to achieve operational assurance results and identify high-level gaps and inefficiencies. This view helps to answer the following framework question: How do collections of participating organizations and assurance solutions work together to achieve operational assurance? The modeling framework currently uses the **SoS Focus Analysis** method to implement the Potential Assurance Results view.

6.1 Method Summary

SoS Focus Analysis examines technical and organizational elements of a complex environment and models various connections among them. Various elements form capabilities or services that are supplied, composed, and synchronized such that operational users achieve the operational effects that their mission or business demands. Understanding in what ways supplied capabilities align with operational demands reveals imbalances or inefficiencies and permits the identification of potential improvements.

Information from the following methods provides input to **SoS Focus Analysis**:

- **Critical Context Analysis** identifies the key stakeholders and their responsibilities that are associated with the four quadrant perspectives (why, for whom, how, and what)
- **Value Mapping** establishes interrelationships and high-level value exchanged among pairs of participants

Each of these methods helps define sufficient context for using **SoS Focus Analysis**. This initial information is then augmented by the following:

- a high-level description of *how* each stakeholder performs his or her responsibilities. Interviews with representatives of the various stakeholder groups provide this information.
- an understanding of the operational mission or business goals and the critical constraints that impact these goals

The analysts then construct a layered model of the way in which the alignment of supplied capabilities with the operational business or mission goals is managed. Previous work with complex systems indicates that six layers usefully capture the alignment model for sets of capabilities or services between technology elements and the operational demand or context of use. Figure 10 shows this layering for a generic set of capabilities.[21] The *layers* are aligned from left to right and

[21] This research by Phil Boxer will be published in an upcoming SEI technical note.

indicated by the numerals at the bottom of the diagram. Layers 1 through 3 focus on the supply side; whereas, layers 4 through 6 focus on the demand side. The *capabilities* or services at a given layer are used by the layers to its right—thus providing composite capabilities. For example, the capability "technical elements" at layer 1 is used by the capability "technical integration of elements" at layer 2.

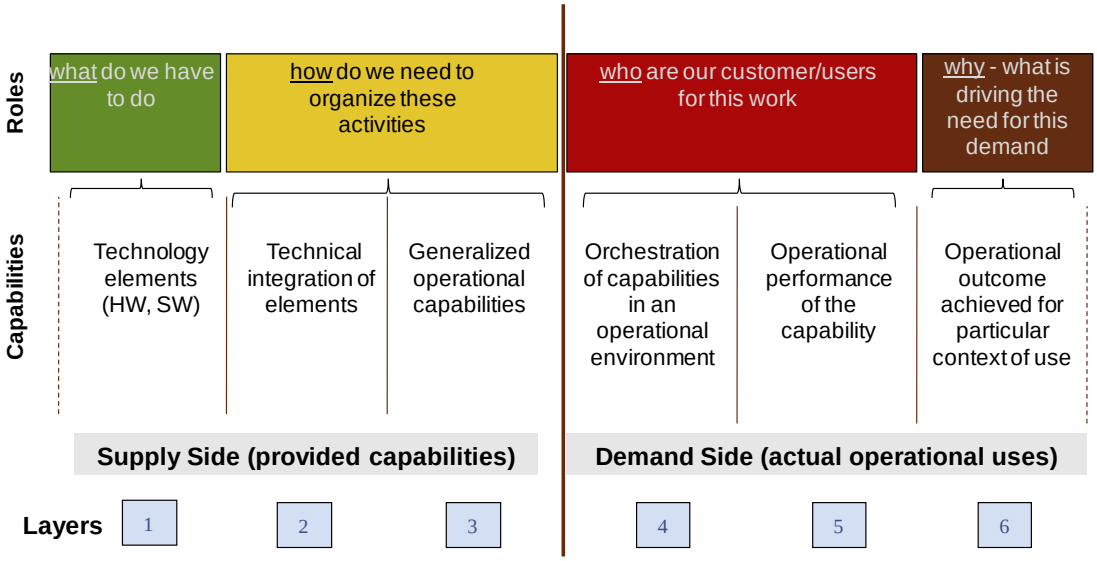


Figure 10: Generic Six-Layer Model to Align Supplied Capabilities with Demand

The *roles* associated with the four perspectives (what, how, for whom, and why) identified with the **Critical Context Analysis** map to the six-layer model in specific ways, as shown at the top of Figure 10. The roles for the *what* perspective support layer 1. The roles for the *how* perspective support layers 2 and 3, and so on.

6.2 Applying the Method

For the pilot project, we started with the roles and responsibilities as characterized in the **Critical Context Analysis**. We used the value maps for CVE in lieu of interviews with stakeholder representatives. From this information and complex systems expertise, we constructed the alignment model for CVE, which is shown in Figure 11.

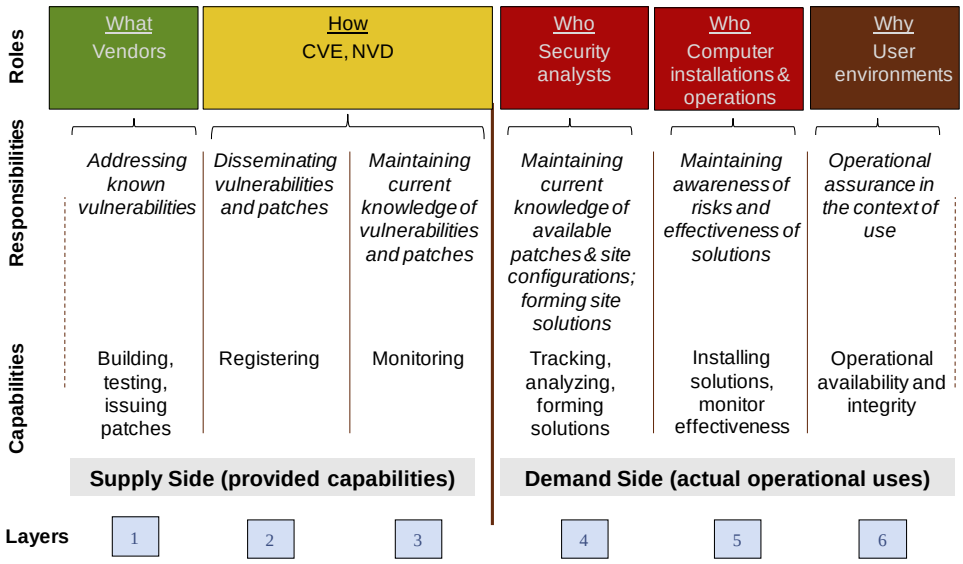


Figure 11: SoS Focus Analysis Alignment Model for CVE

The generic capabilities were tailored for the domain of interest, CVE support for vulnerability management. The roles are denoted in the shaded boxes at the top of Figure 11. A short description of the primary responsibilities for each role, at a given layer, is indicated in italics below the shaded boxes. Table 6 provides further details of each role and its responsibilities.

Table 6: Summary of Roles and Responsibilities for CVE Alignment Model

Role	Description of Responsibilities	Associated Layer
Vendor	Builds and issues patches to address known vulnerabilities	1
CVE and NVD	Registers, disseminates, and monitors vulnerability information and patches	2 and 3
Site security analyst	Manually tracks information on known vulnerabilities. As relevant patches for the site become available, the analyst pulls the relevant patches and forms solutions based on their tacit knowledge of the site network, application configurations, and operational objectives and priorities.	4
IT operations	Manually installs the site-specific solution and alerts the operational users and monitors the operational environment for effectiveness of the solutions. Some IT operations use a patch-management system to assist with the patch installations, but many do not.	5
End users in operational environments	Comply with patch-solution instructions	6

While the alignment model provides a concise way to represent the roles, responsibilities, capabilities and their interrelationships, it is through understanding *how* these capabilities are composed and synchronized to support the demands of operational units that potential areas of inefficiencies are identified. This final step of the analysis is summarized in the following observations section.

6.3 Observations

The effect a technology or other mechanism has on achieving software assurance is often not direct; rather, it comes about through a network of relationships among participating organizations and assurance solutions that must be understood within their operational context. By analyzing the alignment of supplied capabilities to the operational demand, potential over- and under-investments can be identified, areas where tacit knowledge is held can be revealed, and roles that manually synthesize significant information from multiple sources can be identified. Each of these situations represents potential inefficiencies and candidate improvements. From the analysis of the current snapshot of the CVE ecosystem, each of these situations was identified.

- The majority of the participating organizations and assurance solutions support layers 1–3. There are relatively few assurance solutions to support layers 4 and 5.
- Performance of the site security analyst role is largely manual in pulling and integrating information on the available patches with site priorities and configurations to form site-specific solutions (layer 4). Tacit knowledge is typically needed by the security analysts of where to put patches (often multiple locations), what the operational priorities are, and what potential interactions can occur when multiple products are patched.
- Some IT operations use a patch-management system to augment the installation of patches. The IT operations role is largely manual for monitoring the effectiveness of a solution, ensuring compliance by operational users, or ensuring achievement of the assurance objectives of the operational organization (layer 5).

In addition, the analysis revealed that the roles associated with layers 4–6 keep the operational system available and responsive. That is in sharp contrast to the roles associated with layers 1–3, which get patches out as fast as possible. This difference is significant because stakeholders associated with these roles determine value differently. In turn, this affects how the value for particular assurance solutions should be optimally expressed, and impacts the transition and adoption of an assurance solution.

7 Motivations View

The Motivations view is associated with the activity category *Characterize Current State: Ecosystem Relationships*, which provides a detailed understanding of the current state of participating organizations, assurance solutions, and the relationships within the assurance ecosystem. An essential step when evaluating operational assurance is to establish the key factors, or motivations, that strongly influence whether or not the objectives of an assurance capability area will be achieved. Stakeholders can gain an appreciation of what is currently working well, identify gaps and inefficiencies related to performance, and chart a course for improving that capability area.

The objective of this view is to capture and evaluate drivers that are critical to achieving the objectives of a given assurance capability area. This view helps to answer the following Assurance Modeling Framework question: What are the drivers and motivations of participating organizations? The framework currently uses **Driver Identification and Analysis** to implement the Motivations view.

7.1 Method Summary

Driver Identification and Analysis is a method for determining the likelihood that a set of activities will produce a desired outcome. It was originally developed by SEI to characterize a specific program's potential for success. During the course of this project, we adapted it for use within the Assurance Modeling Framework by abstracting up from the perspective of a particular program to the more generic view. More detailed information about this method can be found in *A Framework for Categorizing Key Drivers of Risk* [Alberts 09].

The starting point for **Driver Identification and Analysis** is establishing the key objectives for the assurance capability area being analyzed. A *key objective* is defined as a vital outcome intended to be achieved in the future; it provides a benchmark for measuring success. Once the key objectives have been explicitly articulated, the critical factors that influence achievement of those objectives are identified. These critical factors are referred to as *drivers*—circumstances or situations that strongly influence the eventual outcome or result. Drivers are important because they define a small set of items, usually about ten to twenty, which can be used to gauge the potential for achieving a successful outcome. Once a set of drivers is identified, each driver in the set is then analyzed to determine exactly how it is influencing the outcome.

Information from the following methods provides input to **Driver Identification and Analysis**:

- **Critical Context Analysis** frames the broad context of a domain, including critical participants and basic influences.
- **Value Mapping** establishes interrelationships and high-level value exchanged among pairs of participants.
- **SoS Focus Analysis** identifies how assurance operational demands are aligned with participants, and then identifies capabilities of value and gaps.

7.2 Applying the Method

For the pilot, the assurance capability area selected for analysis was vulnerability management. When applying **Driver Identification and Analysis**, we further narrowed the focus to the *reactive component of vulnerability management*, where the emphasis is on identifying and correcting vulnerabilities during operations. We then identified three objectives for the reactive component of vulnerability management:

1. maintain a low-risk operational environment
2. respond to reports of vulnerabilities in a timely manner (e.g., advisories and alerts, product vendor patches)
3. ensure that any adverse effects from vulnerability solutions have minimal impact on users and operations

Next, we used our expertise in the area of vulnerability management to identify a set of critical factors, which would strongly influence the three objectives noted above. We brainstormed answers to the following questions:

- What circumstances, conditions, and events will drive an organization toward a *successful* outcome (i.e., achieving key objectives)?
- What circumstances, conditions, and events will drive an organization toward a *failed* outcome (i.e., not achieving key objectives)?

After generating a list of items, we organized the items into 18 groups that share a central idea or theme; a candidate driver is the central idea or theme of each group. Each candidate driver comprises four key attributes—*name, success state, failure state, and considerations*—which are described in more detail in Table 7. The names of the 18 candidate drivers for vulnerability management are shown in Table 8. A complete list and their attributes can be found in Appendix B.

Table 7: Driver Attributes

Attribute	Description	Example
Name	A concise label that describes the basic nature of the driver	Distribution Mechanisms
Success State	A driver exerts a positive influence on the outcome	Mechanisms for distributing vulnerability information and solutions are sufficient.
Failure State	A driver exerts a negative influence on the outcome	Mechanisms for distributing vulnerability information and solutions are insufficient.
Considerations	Circumstances that must be considered when evaluating a driver	Distribution of advisories and alerts Application of patches Changes to system configurations

Table 8: Candidate Drivers of Vulnerability Management

Driver		Driver	
1.	Vulnerability Management Objectives	10.	Technology
2.	Plan	11.	Facilities and Equipment
3.	Process	12.	Organizational Conditions
4.	Distribution Mechanisms	13.	Compliance
5.	Situational Awareness	14.	Event Management
6.	Task Execution	15.	Requirements
7.	Coordination	16.	Solution Tracking
8.	External Interfaces	17.	Risk Tolerance
9.	Information Management	18.	Unintended Consequences

7.3 Observations

We analyzed which candidate drivers affecting vulnerability management are strongly influenced by the operational use of CVE. Based on the analysis, we identified the following four drivers of successful outcomes:

- Distribution Mechanisms (Driver 4)—This driver focuses on mechanisms for distributing vulnerability information and solutions. Mechanisms include advisories and alerts that provide information about vulnerabilities, solutions that correct or mitigate vulnerabilities, and patches.
- Situational Awareness (Driver 5)—Situational awareness is determining how information, events, and actions will affect vulnerability management objectives. This driver focuses on the system and network environments, which include an up-to-date documented baseline of all systems and networks, awareness of new vulnerabilities, documentation of patches applied, and network topology diagrams.
- Coordination (Driver 7)—This driver addresses how well vulnerability management tasks and activities are coordinated within each team and across teams.
- External Interfaces (Driver 8)—The interfaces with external parties are addressed by this driver. In particular, it looks at whether advisories and alerts, solutions, patches, and vulnerability information are correct, complete, and received in a timely manner.

We applied the conceptual approach of this method during the pilot. A next step would be to use the selected drivers with a specific organization to assess performance of vulnerability management within an actual operational environment. Within parts of a particular ecosystem, we could use **Driver Identification and Analysis** to better characterize inefficiencies identified with some of the other methods.

8 Critical Behaviors View

The Critical Behaviors view is associated with the activity category *Characterize Current State: Ecosystem Relationships*, which provides a more detailed understanding of how the current state is working and the interactions among organizations and assurance solutions addressing a selected assurance capability area within the assurance ecosystem. By understanding the critical behaviors exhibited by organizations within the selected capability, it is possible to project the ways in which assurance solutions are addressing the desired results and identify opportunities for improvement in meeting this target. The behaviors of various organizations that build, maintain, and apply assurance solutions work together to achieve or inhibit the achievement of an assurance capability. Once the critical behaviors are understood, patterns of inefficiency can be identified and alternative options can be analyzed.

The objective for this view is to capture the causal relationships among collections of participating organizations and assurance solutions to identify primary variables of interest and their influences that drive critical behavior. This view helps to answer the following framework question: What are the critical usage scenarios and behaviors among participating organizations and assurance solutions? The Assurance Modeling Framework currently uses the **System Dynamics** method to implement the critical behaviors view.

Information from the following methods provides input to **System Dynamics**:

- **Critical Context Analysis** provides the objectives of the assurance capability area.
- **Value Mapping** establishes the participating organizations involved in the selected assurance solutions within an assurance capability area and the values exchanged between these organizations.
- **SoS Focus Analysis** provides the chain of values exchanged from suppliers to operational users. It also provides a structure for organizing the **System Dynamics** model and constraints.
- **Driver Identification and Analysis** provides the factors that would help or hinder achievement of objectives for an assurance capability.

8.1 Method Summary

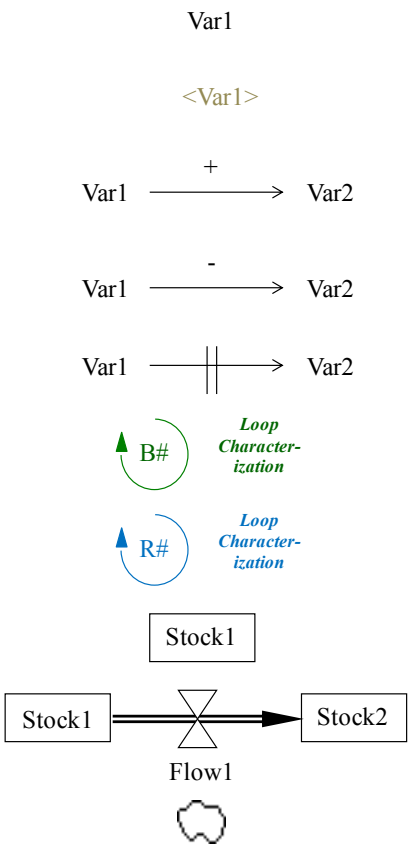
The **System Dynamics** method helps analysts model and analyze critical behavior as it evolves over time within complex socio-technical domains. A powerful tenet of this method is that the dynamic complexity of critical behavior can be captured by the underlying feedback structure of that behavior. The boundaries of a system dynamics model are drawn such that all the enterprise elements necessary to generate and understand problematic behavior are contained within them. The method has a long history and is described in the following primary sources:

- Sterman's comprehensive treatment of the System Dynamics method [Sterman 2000]
- Meadows' description of thinking in systems rather than components in isolation as a means to analyze and solve problems [Meadows 2008]

System Dynamics and the related area of systems thinking encourage the inclusion of soft factors in the model, such as policy, procedural, administrative, or cultural factors. The exclusion of soft

factors in other modeling techniques essentially treats their influence as negligible, which is often an inappropriate assumption. This holistic modeling perspective helps identify mitigations to problematic behaviors that are often overlooked by other approaches.

Figure 12summarizes the notation used by **System Dynamics** modeling. The primary elements are variables of interest, stocks (which represent collection points of resources), and flows (which represent the transition of resources between stocks). Signed arrows represent causal relationships, where the sign indicates how the variable at the arrow's source influences the variable at the arrow's target. Basically, a positive (+) influence indicates that the values of the variables move in the same direction, whereas a negative (-) influence indicates that they move in opposite directions. A connected group of variables, stocks, and flows can create a path that is referred to as a feedback loop. The type of feedback loop is determined by counting the number of negative influences along the path of the loop. An odd number of negative influences indicates a balancing loop; an even (or zero) number of negative influences indicates a reinforcing loop.



Variable – anything of interest in the problem being modeled

Ghost Variable – variable acting as a placeholder for a variable occurring somewhere else

Positive Influence – values of variables move in the same direction (e.g., source increases, target increases)

Negative Influence – values of variables move in the opposite direction (e.g., source increases, the target decreases)

Delay – significant delay from when Var1 changes to when Var2 changes

Balancing Loop – a feedback loop that moves variable values to a goal state; loop color identifies circular influence path

Reinforcing Loop – a feedback loop that moves variable values consistently upward or downward; loop color identifies circular influence path

Stock – special variable representing a pool of materials, money, people, or other resources

Flow – special variable representing a process that directly adds to or subtracts from a stock

Cloud – source or sink (represents a stock outside the model boundary)

Figure 12: System Dynamics Notation Used in Abstract Models

System Dynamics models identify two types of feedback loops: balancing and reinforcing. Significant feedback loops identified within a model are indicated a loop symbol and a loop name in italics. Balancing loops—indicated with a label "B" followed by a number in the loop symbol—describe aspects of the system that oppose change, seeking to drive variables to some goal state. Balancing loops often represent actions that an organization takes to mitigate a problem. The feedback loop in Figure 13 provides an example of a balancing loop that depicts the feedback

control. In this example the loop describes the behavior of vendors whose products are discovered to have vulnerabilities. As the number of product vulnerabilities increases, so does the amount of vendor resources needed to patch application code. And as products are patched, the number of vulnerabilities decreases, so fewer product patching resources are needed.

Reinforcing loops—indicated with a label "R" and followed by a number in the loop symbol—describe system aspects that tend to drive variable values consistently upward or downward. Reinforcing loops often represent the escalation of problems, but may also include problem mitigation behaviors.

8.2 Applying the Method

The **System Dynamics** model developed thus far has helped to document and describe how organizations use CVE and CWE to promote improved vulnerability management assurance solutions. Based on the analysis of behaviors described by subject matter experts, it is possible to hypothesize measures for improving the efficiency of the vulnerability management supply-side processes.

Two versions of a System Dynamics model were created during the pilot: a detailed version (shown in Appendix C on page 78) and a simplified version (shown on page 79). The simplified version helps to convey the essence of the detailed model using a simpler notation. The detailed model documents additional aspects such as the mapping to the stratification layers identified by SoS Focus Analysis and more details about the influences on critical behavior. Diagram 2 in Appendix C shows the connection of the six-layer stratification of the SoS Focus Analysis to the detailed System Dynamics model. The primary feedback loops influencing the critical behavior are represented in the simplified as well as the detailed model. The labels, loop colors, and general layout of the detailed model are preserved in the simplified version to make it easier to cross-reference between the two models. For an abstract view of what the model describes and its essential insights, the simplified model is sufficient. However, the detailed model provides a vehicle for future validation of critical behaviors and evaluation of improvement strategies.

To understand how to interpret a system dynamics model, the following description uses a series of figures drawn from our application of the framework to vulnerability management. Start with Figure 13, which depicts a single balancing feedback loop relevant to vulnerability management. The diagram shows that vendor product vulnerabilities are connected positively ("+") with the level of vendor community resources available to develop patches. This positive influence implies that as product vulnerabilities increase, vendor community resources for developing patches also increase. Likewise, as resources to patch increase, a greater number of patches is produced and the patching of product vulnerabilities increases. (Released patches are tagged with the appropriate CVE Identifiers to notify customers which vulnerabilities are addressed, cross-referencing to the CVE registry of product vulnerabilities available at the CVE website.[22]) As patching occurs, the number of known vulnerabilities in the vendor product is expected to decrease, which leads to a decrease in the level of community resources needed to patch vulnerabilities. Because the connec-

[22] The CVE information publicizes vulnerabilities in a vendor's product in a way that encourages that vendor to patch those vulnerabilities.

tion between "patching product vulnerabilities" and "product vulnerabilities" is marked with a minus sign, *reduced* product patching implies an *increase* in product vulnerabilities, which then requires an increase in resources, and so forth. This balancing feedback, as indicated by the odd number of negative influences along its path, characterizes efforts to keep the number of product vulnerabilities in check. This type of organizational behavior is reactive.

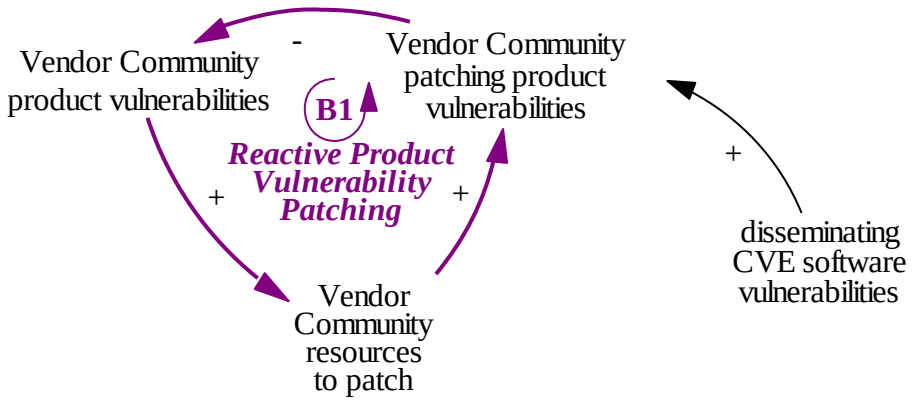


Figure 13*: Reactive Product Vulnerability Patching*[23]

Figure 14 adds a second feedback loop to describe a proactive behavior response which is also available to an organization to handle the problem of product vulnerability.[24] The organization's management must decide how to split organizational resources between reactive and proactive activities. Allocating resources to work on patching existing product vulnerabilities is expected, especially when these vulnerabilities represent a high risk that the customers for this organization will suffer from exposure and compromise.

The feedback loop labeled B2 in Figure 14 depicts the proactive solution to product vulnerability. Added steps for training and experimentation with alternate approaches for preventing vulnerabilities from being introduced into vendor product software in the first place are required, and the use of a different assurance solution (CWE, which identifies specific software weaknesses and approaches to eliminating vulnerabilities from the software development process) is needed.

While proactive solutions are important to long-term security improvement, it is clear that some immediate relief must go to addressing the problem of current product exposure and compromise. However, as shown in the R1 loop of Figure 14, too much focus on reactive activities that reassign personnel from vulnerability prevention efforts to vulnerability patching can overly constrain resources available to address practices for avoiding vulnerabilities in new software development

[23] System dynamics diagrams are generated using the VenSim tool. For more information, go to http://www.vensim.com.

[24] In this and subsequent figures, callout boxes are used to describe the "flow" of the model, telling the story of how problems associated with vulnerability management unfold. Callout boxes are labeled numerically and should be read in that order.

and lead to increased vulnerabilities in released products. This reinforcing loop can result in a downward spiral of increasingly vulnerable products and increased assurance problems.

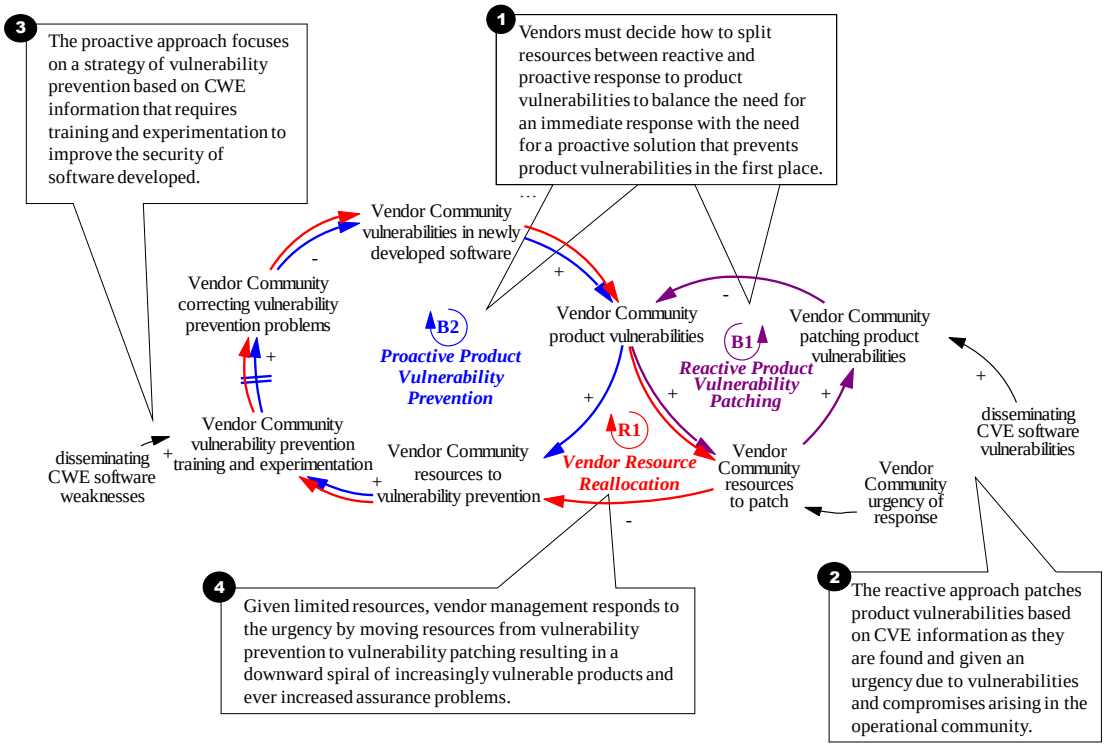


Figure 14: Reactivity Degrading Long-Term Security Improvements[25]

Figure 15 augments the two feedback loops to include the behaviors in the operational community that create the sense of urgency in the vendor community and reinforce the vendors' shifting of resources from proactive to reactive vulnerability management. The balancing feedback loop labeled B3 represents the operational community's attempt to manage vulnerabilities through technical scanning. The analyst community packages patch-based solutions to these vulnerabilities based on CVE reporting and dissemination of vulnerability information. For vulnerabilities not found by technical scanning, security incidents may occur and emergency solutions may need to be identified and implemented, as shown in the balancing loop labeled B4. It is the customer response to high-impact security incidents and the general exposure to attack that pressures vendors to have immediate fixes.

[25] Redundant arrows of different colors are provided between variables to clarify the path of the feedback loops enumerated in the model.

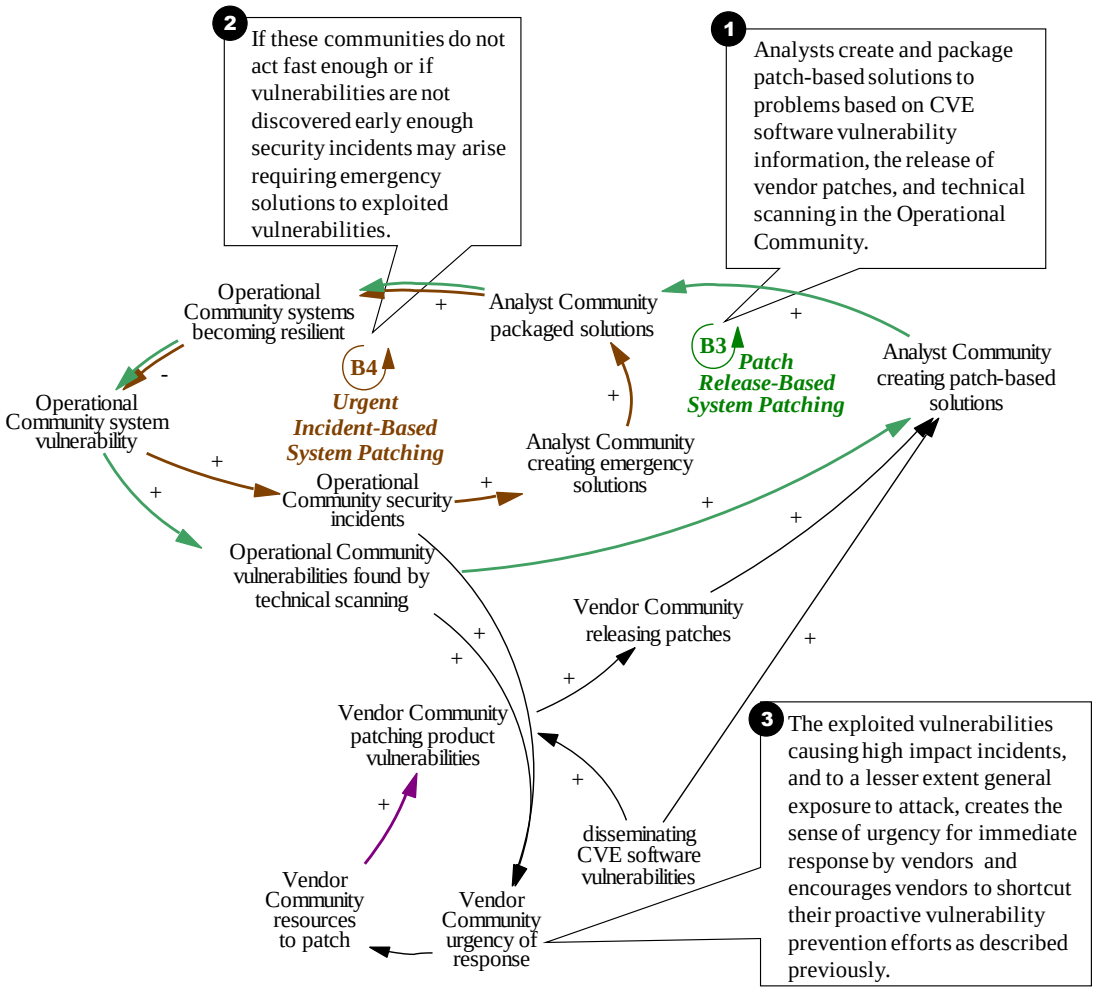


Figure 15: Operational Community's Response to Product Vulnerabilities

Figure 16 reflects what may be an emerging trend in the vendor community. The increased exposure of vendor product vulnerabilities brought by the CVE initiative is making it easier for buyers to compare different products based on their security track record. There is evidence that buyers are considering this issue when deciding on what products to use—a trend that is likely to continue as vendors understand its importance for maintaining and growing their market share. This trend can push vendors toward more proactive approaches to prevent vulnerabilities from getting into their products in the first place. As shown in the reinforcing loop R2 in the figure, this increased focus on preventative techniques is likely to show up in vendors' advertising over time as their products become more secure and the number of incidents that exploit their product vulnerabilities diminishes. This increased emphasis on vulnerability prevention claims in advertising should make it even easier for buyers to compare the vulnerability of different products, thus reinforcing the trend.

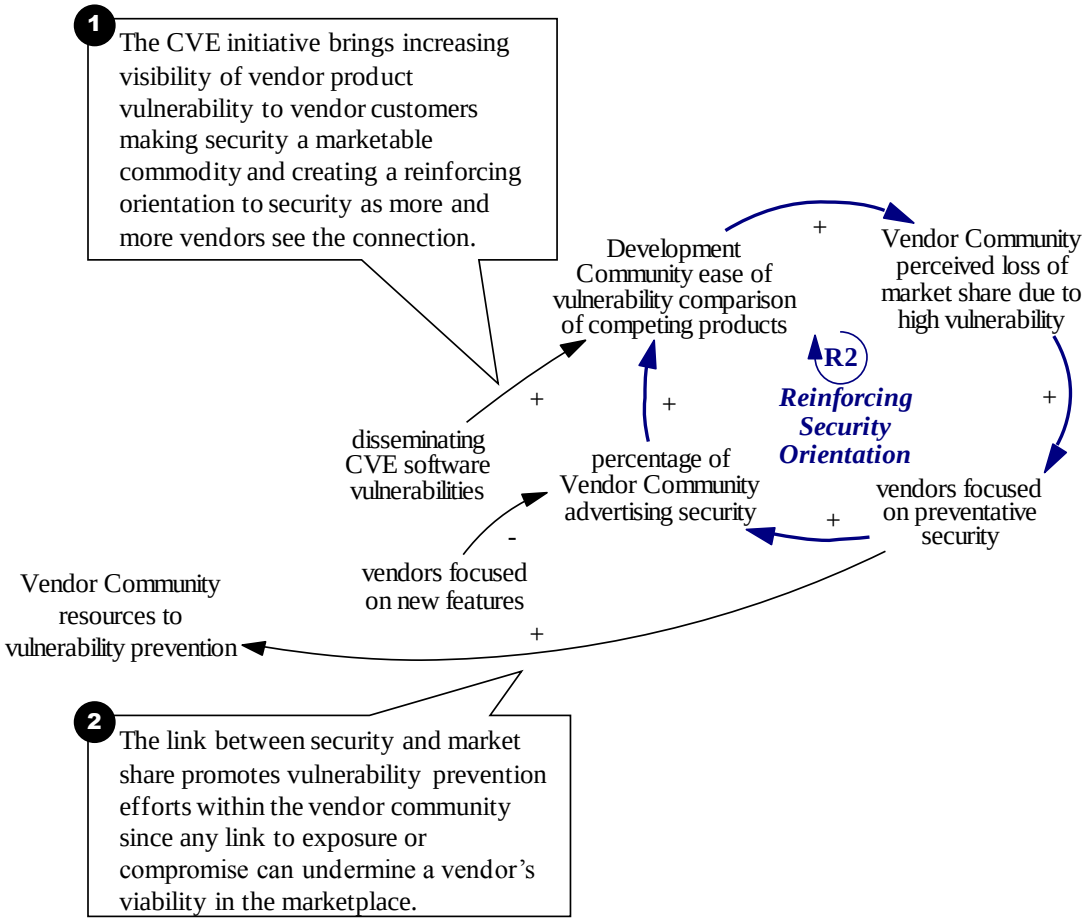


Figure 16: Reinforcing Security Orientation Due to CVE-Facilitated Vulnerability Comparison

8.3 Observations

The **System Dynamics** model developed for the pilot use of the framework suggests a number of critical behaviors in the vulnerability management capability area that are useful in analyzing opportunities for improvement. Developers and operators faced with vulnerability problems in vendor products are expected to demand fixes to those problems. Faced with customer pressure, the vendor community focuses its efforts on creating patches for its products to address vulnerabilities and other defects. By cataloging and cross-referencing available information about vulnerabilities, the CVE brings increasing visibility of vendor product vulnerability to customers. As customers become more aware of their exposure to product vulnerabilities, they are expected to look for vendors with products that exhibit less vulnerability, thereby increasing the pressure for vendors to improve their products so that less vulnerability are identified. As vendors take action to prevent vulnerabilities in their products, their behavior encourages their use of vulnerability prevention techniques such as CWE to ensure that vulnerabilities in their products do not cause operational problems for their customers.

The structure of the behavior feedback loops suggests there might be a tipping point for optimal assurance performance, where a good balance of proactive software vulnerability prevention practices and reactive patch generation and release is needed. Too much focus on vulnerability prevention neglects the necessary response to critical patches needed (sacrificing short-term needs).

Too much focus on patch generation to the exclusion of vulnerability prevention sacrifices long-term advancement in SoS operational security assurance. Refinement of the system dynamics model to permit simulation would be one means of assessing the impact of possible improvements to the vulnerability management capability.

9 Adoption of Products View

The Adoption of Products view is associated with the activity category *Characterize Current State: Solution Maturation and Adoption* which provides an understanding of the current state of the formation, maturation, adoption, and use of assurance solutions. Creating an assurance solution is only part of the assurance ecosystem. Fostering faster and more effective adoption of solutions requires understanding the current maturation of the solution, the mechanisms employed and their impact, and the participants involved.

The objective of this view is to capture the maturation and adoption mechanisms used and their effectiveness as collections of related assurance solutions. This view helps to answer the following framework questions:

- What are the adoption and operational usage mechanisms available for assurance solutions?
- How are the mechanisms aligned with organizational context and need?

Numerous models and processes for managing the activities of technology innovation, maturation, transition, and adoption exist. Most are focused on the adoption of a single technology. Yet for software assurance, many of the technologies are interdependent because they address different aspects of the assurance problem. As a result, understanding their maturation and adoption requires working within a context of *collections* of related assurance solutions at varying states of maturity and adoption. We have not found a suitable existing method and are experimenting with several models and processes to determine if and how they might be adapted for the more complex needs of the assurance ecosystem. **Technology Development and Transition Analysis** is a method that the project team is developing to address this need. We expect this method to change significantly in subsequent phases of our research.

9.1 Method Summary

The Assurance Modeling Framework currently uses **Technology Development and Transition Analysis** to

- identify a timeline of maturation and adoption for two interrelated assurance solutions
- identify and characterize the major transition mechanisms applied by the solution owners
- analyze the applicability of the maturation approaches and transition mechanisms

Information from the following methods provides input to **Technology Development and Technology Analysis**:

- **Value Mapping** establishes the collection of assurance solutions, participating organizations, and the types of value exchanged.
- **SoS Focus Analysis** frames the operational context, major roles and their responsibilities, and the supplied capabilities or services.

As previously noted, **Technology Development and Transition Analysis** is under development. At this point in the project, we used the following single-technology maturation and transition models as the basis of our approach:

- Jolly's model and process for commercializing technology [Jolly 1997]
- Tornatzky and Fleischer's general concepts for technological innovation [Tornatzky 1990]
- Moore's whole-product approach for technology adoption [Moore 1999]
- SEI's life-cycle model for guiding the maturation and transition of its products [Forrester 2003]

We selected the Jolly model because it explicitly includes the processes that build value for a new technology as it matures and the associated bridges that motivate organizations and communities to adopt that technology. In Figure 17, we replicate the Jolly model. It comprises nine overlapping elements that Jolly refers to as *segments*. There are two kinds of segments: (1) *subprocesses* that build the value of a new technology at each stage and (2) *bridges* that satisfy stakeholders of the current stage and mobilize stakeholders for the next stage. We have used the convention for this report to show subprocess names as uppercase.

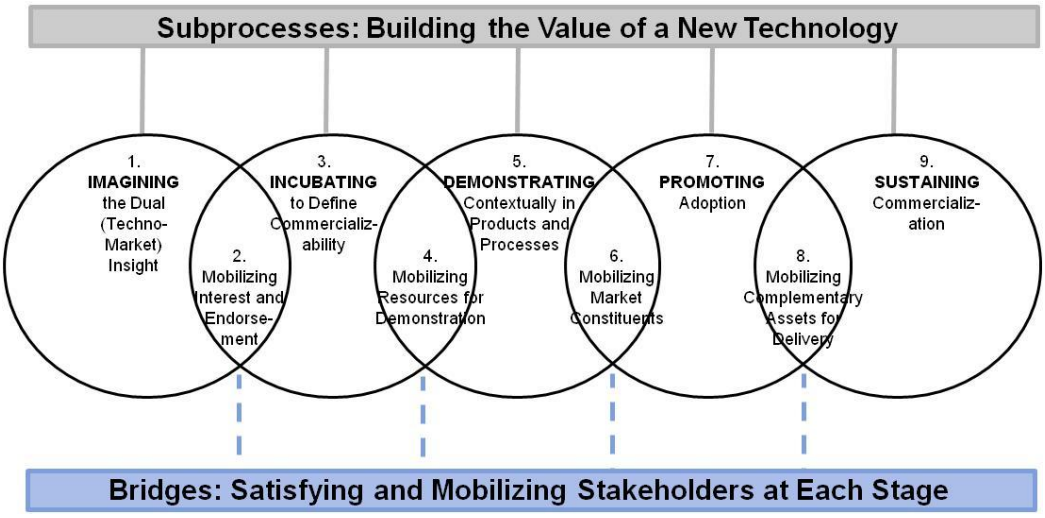


Figure 17: Jolly Model for Commercializing Technology [Jolly 1997]

A short description of Jolly's segments is provided in Table 9. The entries for bridges do not include a focus description because the bridge name provides that information. Similarly, the bridge entries omit completion points, since this is captured by the outcome information.

Table 9: Summary of Subprocesses and Bridges

Subprocess	Bridge	Focus	Outcome	Completion Points
IMAGINING		Establish Technical Credibility	Exciting, preferably unique technology-based idea linked to a market need	Technical proof of principle, filing key patents, preliminary vision for the technology
	Mobilizing Interest & Endorsement		Early endorsement by those who matter	
INCUBATING		Show Technical Value and Transitionability	Definition of idea's technical feasibility, commercial potential, and plan for taking it	Prepared business case and plan for commercialization, crafting the technol-

Subprocess	Bridge	Focus	Outcome	Completion Points
			further	ogy or product platforms, testing with lead customers
	Mobilizing Resources for Demonstration		Resources needed for demonstrations	
DEMONSTRATING		Establish Whole Product	Incorporating the technology in attractive, market-ready products and/or processes	Launch of commercial version of product or process
	Mobilizing Market Constituent		Develop a market	
PROMOTING		Create an Installed Base	Getting product or process rapidly accepted by various market constituents	Capturing a profitable share of market quickly
	Mobilizing Complementary Assets		Establish approach to determine the extent to which technology is shared while keeping control over its exploitation	
SUSTAINING		Keep Products Competitive	Generating long-term value by entrenching and expanding use of the technology and retaining a lead	Adequate ROI made in technology and infrastructure for commercializing it

9.2 Applying the Method

For the pilot, we selected CVE and CWE as the interrelated assurance solutions for analysis. We first built a timeline for each assurance solution, structured by the Jolly subprocesses and bridges. We captured events, actors, actions, and decisions along with the year of the occurrence. Table 10 shows the resulting timeline information for CVE mapped onto the Jolly model.

Table 10: CVE Maturation and Adoption Timeline Using Jolly Model

Segment	Activities
IMAGINING	1998 – need being articulated; trusted third party as a broker for a solution identified by community 1999 – Cerias Purdue workshop: proof of concept - vendors align with MITRE to develop CVE concept; others in the audience question its utility
Mobilizing Interest & Endorsement	1999 – MITRE forms CVE Editorial Board with vendors and research organizations 1999 – Website initiated
INCUBATING	1999 – CVE Initiative unveiled with initial list of vulnerabilities, conference booth staffed by CVE Editorial Board, press release 1999 – Form of CVE entries and adjudication process defined and tested 2000 – CVE list expanded by Editorial Board from broader set of sources
Mobilizing Resources for Demonstration	2000 – NSA & ESC funding secured to continue 2001 – Senior Advisory Council formed: used to elicit funding and support

Segment	Activities
	from those who were feeling the pain 2000/2001 – big presence at targeted conferences to get the word out, build broader interest 2000/2001 – expanded website for information dissemination and distribution
DEMONSTRATING	1999/2000 – Operational demonstration of multiple vendors using CVE conducted during IDSnet at SANS conferences 2000 – Candidate numbering authorities established along with roles, responsibilities and communication protocols 2001 – Microsoft starts using CVE 2001 – MITRE expands support staff, improves processes and utilities to deal with increasing volume of vulnerabilities 2002 – Red Hat & Open Source start using CVE (April) 2002 – Articulation definition of compatibility requirements – output, searchability, documentation 2002 – NIST special publication about use of CVE (recommendation for use) 2002 – DoD rewrite of 8500.2 implements IAVA process with CVE referenced 2003 – questionnaire for self-reporting of vendor compatibility: results submitted to MITRE for initial review and published on website
Mobilizing Market Constituent	2001 – coordination with NIST NVD (originally I-CAT) 2002 – continuing big presence at targeted conferences (e.g., RSA, SANS, Black Hat) to broaden awareness of extent of community involvement and usage as well as a method for networking with vendors, researchers, and thought leaders 2002-2004 – website expands to involve and recognize partners
PROMOTING	2004 – March and Nov compatibility award ceremonies (mechanism for incentivizing quality participation): CVE declared as de facto standard by community 2005 – 2009 – continuing research into usage patterns
Mobilizing Complementary Assets	2008 – increasing self-reporting and self-assessments – moving to more community self-policing
SUSTAINING	2007–present – vendors and researchers continue to engage with CVE becoming candidate numbering authorities for their product/project on their own 2009 – considering move to self-regulation: community and MITRE considering elimination of compatibility verification – self-evaluation to be sufficient

Next, we analyzed the timelines for patterns of potential factors that could contribute to maturation and adoption successes and failures of the assurance solutions. We first identified *indicators* of success. The success indicators for CVE are listed in Table 11.

Table 11: Success Indicators for CVE

Indicators of Maturation and Adoption Success for CVE
CVE is accepted throughout the supplier community.
CVE is considered a de-facto standard by the community.
Vendors advertise that they are CVE compliant.
Content providers/list makers reference vulnerabilities using CVE.
NVD explicitly uses CVE.

We then identified and characterized the potential *factors* that contributed to that success or failure. The success factors for CVE are summarized in Table 12. A detailed list of the factors and their characterization can be found in Appendix D.

Table 12: Success Factors for CVE

Factors Contributing to Success for CVE
MITRE identified a clear market need (from a community perspective).
Vendors were motivated to participate.
MITRE's strategy allowed it to partner with researchers and content providers/list makers.
A growing amount of vulnerability information was distributed across multiple databases (operated by competing groups).
MITRE filled an unmet community need with CVE.
MITRE signed agreements with vendors to get information earlier.
MITRE's proof of concept using public data convinced vendors of the value of the CVE approach.
MITRE identified the right stakeholders and did a good job of getting them involved in building the solution
MITRE explicitly focused on reducing the barriers to adoption
MITRE's solution did not force adopters to change the way they did business.
Government policy – DoD IAVA was rewritten to include CVE.
MITRE continues CVE "marketing" and product evolution.
There is continued investment in infrastructure.
Community articulated "standard" before MITRE used the term.
Focus on building collaborations.

We conducted similar data gathering and analysis with CWE, and provide those results in Appendix D.

The final step was to evaluate the interactions and effectiveness of adoption mechanisms within the context of *collections* of technologies and participants. Our current approach admittedly deals with a very limited collection—CVE and CWE. Using the technologies' connections and the organizations associated with those technologies (as indicated on the value maps), the timelines with transition mechanisms captured via the Jolly model, and the success factors, we identified patterns that might not have surfaced if we had looked at the technologies individually. We discuss these results in the following section.

9.3 Observations

The technology maturation and transition mechanisms for CWE are being patterned after those used with CVE. While they were quite successful for CVE, they may have less success for CWE—and the maturation timeframe may be quite different. CVE required little behavioral change on the part of its primary users (e.g., suppliers of IT products, such as Microsoft or Oracle, and suppliers of vulnerability management products, such as Symantec or McAfee). CWE, however, will require extensive behavioral and process changes on the part of its users such as software development organizations. For CWE different development techniques and processes that proactively address known weaknesses will need to be institutionalized. These types of changes will require additional transition mechanisms and perhaps other participants and incentives to accelerate their adoption.

While there is overlap among the user communities of CVE and CWE, there are also key differences. The primary focus of CVE is characterizing vulnerabilities from an *operational* perspective. Those characterizations are written in the language of operations. In contrast, the primary focus for CWEs is characterizing weaknesses associated with vulnerabilities from a software *development* perspective. CWEs are written in the language of software engineering; a large part of their success will be tied to bridging the "language" gap between operations-centric and software engineering vocabularies. There is an increasing trend to write CVE vulnerability descriptions so the connections to CWE will be easier to identify, which may have an impact on their use by operational organizations.

An ability to capture and express the current state in maturation and transition for a collection of related technologies and organizations—an ecosystem—is a critical need for this project. Further research is recommended to identify and adapt a suitable method.

10 Future Drivers View

The Future Drivers view is associated with the activity category *Determine Future Factors* which provides an understanding of potential future factors, such as operational business and mission needs, technologies, economic, political, or environmental, and their impact on assurance solutions and participating organizations.

The objective of this view is to capture an applicable range of future trends, influences, and uncertainties that may shape new operational demands and assurance solutions. This view helps to answer the following framework question: What is the impact of future trends and events on participating organizations and assurance solutions? The Assurance Modeling Framework currently uses the **Strategic Alternatives Analysis** method to implement the Future Drivers view.

10.1 Method Summary

The **Strategic Alternatives Analysis** method projects different plausible but contrasting sets of future conditions over a long-range horizon. The method is designed to reveal the impacts of candidate policies and practices as a response to potential changes in the external environment. The method characterizes and explores contrasting trends and *uncertainties* that may impact an enterprise or community of interest. Key uncertainties are transformed using s*cenarios* as a means of understanding the potential impact of sociological, technological, political, economic, cultural, and environmental changes. The method facilitates the generation of key trends, implications, and "watchpoints" to inform policy and practice decisions.

Strategic Alignment Analysis provides a view of the contextual environment in which certain technical or management activities are likely to occur over the long term, typically 10-20 years. The method is adapted from the following sources:

- Schwarz and Van Der Heijden's scenario-based planning techniques that rely on intuitive logics [Schwarz 1996, Van der Heijden 2005]
- Kelly's general categories of future uncertainties that can be mined to start the set of uncertainties for specific contexts [Kelly 2005]

Strategic Alignment Analysis uses an outside-in approach, wherein participants are asked to look at different uncertainties they foresee. Those uncertainties are then grouped and characterized as dimensions of uncertainty. They have endpoints that are high contrast (e.g., a dimension called "Character of Technology Change" could have endpoints of "Incremental" and "Disruptive"). Interesting pairs of dimensions are then looked at, to define characteristics of various futures. From the futures characteristics, the participants work backwards through time to verify the plausibility of the envisioned futures, effectively creating scenarios of the future. Then, participants work through implications of each chosen future scenario (usually three or four scenarios are selected). At this point in the analysis, it is often quite easy to determine robust strategies—those that would create a benefit to the organization in multiple contrasting futures—as well as specialty strategies that may optimize only in one or two futures. The final step in most instances is to establish watchpoints related to different scenarios to help the community of interest understand how unfolding events may affect their strategies.

10.2 Applying the Method

For the pilot, the first step was to establish the focus question for the analysis. The focus question provides a basis for discovery and exploration and an anchor for subsequent investigation. The focus question for the pilot was as follows: How will the SoS software assurance environment evolve between now and 2020?

Next, we brainstormed and prioritized external drivers of change. Our objective was to produce a prioritized list of uncertainties and a small set of axes of uncertainty. To stimulate our thinking, we used a generic set of factors as a starting point:

- market size, growth, volatility
- customers
- competitors
- suppliers
- owners
- communities
- partners
- demographics (aging, immigration patterns)
- values (lifestyles, political or spiritual movements)
- technological breakthroughs
- industry competitive structures
- legislation and regulation
- emergent "rules" (standards, trade practices)

After a period of facilitated brainstorming, we identified six high-priority uncertainties that we thought were appropriate for the focus questions. Figure 18 shows the resulting uncertainties, along with a characterization for each endpoint.

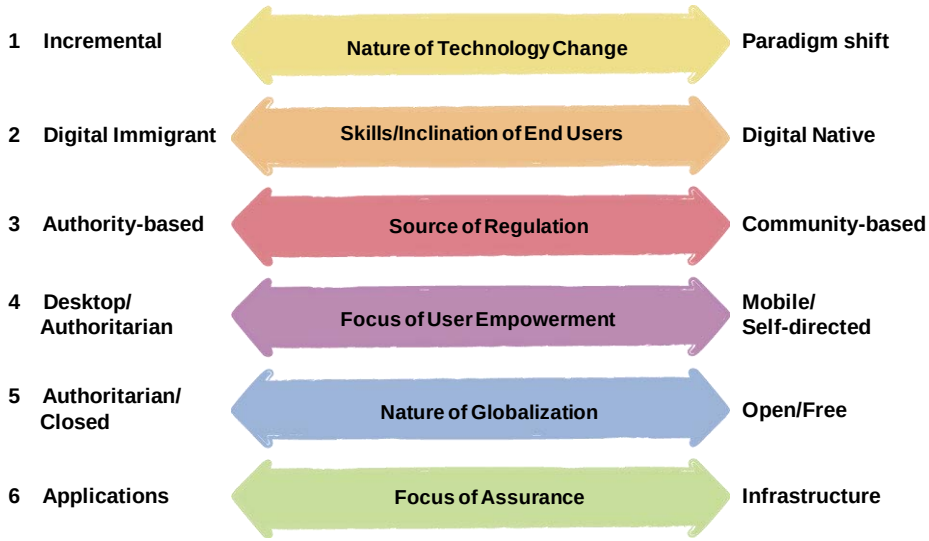


Figure 18: Axes of Uncertainty Identified in Pilot

From the group above, two were selected for further processing:

- **Nature of Technology Change:** This dimension focuses less on the speed of technology change and more on what a technology is likely to demand from its adopters. Incremental technology changes tend to have easier adoption cycles than technologies calling for paradigm shifts. For example, in scenarios where paradigm shifts are likely to occur, one of the common implications is a technology backlash that must be accounted for by those promulgating new technologies.
- **Skills/Inclinations of End Users:** In his book *Digital Game-Based Learning*, Prensky divides our current workforce into two categories: Digital Immigrants, who grew up and were educated into the workforce without constant access to computers and the internet, and Digital Natives, who had constant access computers and the internet and associated digital technologies as part of their formative years [Prensky 2000]. One might argue that, as we move forward, the shift from a digital immigrant to a digital native population is a certainty. However, access to technology is not uniform across socioeconomic sectors, so within the project's time horizon, there is still uncertainty as to which part of the population will dominate. This uncertainty has implications for everything from how a new technology is positioned to the types of training and other transition mechanisms that are required to ensure its adoption.

Although both of these axes of uncertainty relate to technology, they actually explore the sociological aspects of technology adoption rather than technology itself. The focus on sociological aspects provides more useful insights to participants than axes of uncertainty that focus explicitly on characteristics of a particular technology.

For the next step, we reframed the selected axes of uncertainty into scenarios. To do that, we formed two teams to experiment with juxtapositions of different axes of uncertainty, create scenario characteristics matrices, and analyze the candidates for the most promising one or two scenarios. Figure 19 illustrates the scenario characteristics matrix generated by Team 1, which explores the intersection of the Nature of Technology Change (horizontal axis) with the Skills/Inclinations of End Users (vertical axis). Each quadrant records three to five characteristics that best represent the particular intersection. For ease of reference, we labeled each quadrant with a descriptive phrase that summarized the characteristics for that quadrant.

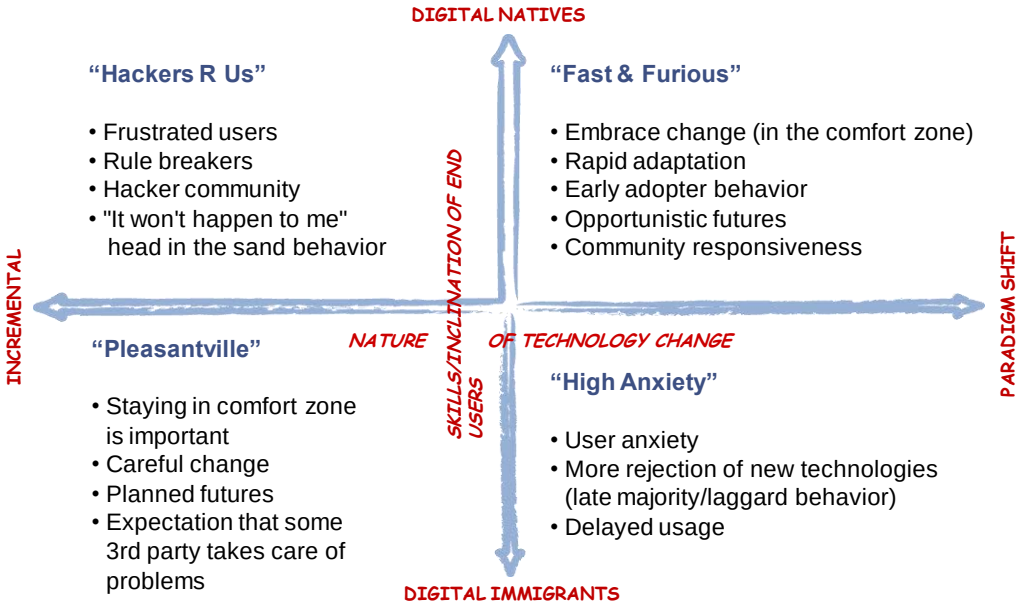


Figure 19: Scenario Characteristics Matrix for Team 1

To stimulate discussion, we used the phrase "This is a world in which…" to start identifying characteristics of each quadrant. To encourage breadth of the characteristics, we used the STEEP elements (Sociological, Technological, Environmental, Economic, and Political) as guides. Once a reasonable set of characteristics was captured, we labeled each quadrant in a way that summarized the overall characteristics of the quadrant. For example, the upper right hand quadrant, "Fast & Furious" captures the characteristics of crossing the Nature of Technology Change–Paradigm Shift with the Skills/Inclinations of End Users–Digital Natives. The label "Fast & Furious" gives a sense of the frenetic nature of an environment where users readily accept new technologies with radical innovations.

Next, we used the upper right-hand quadrant as the basis for developing a scenario using the "Headlines from the Future" scenario generation mechanism. This approach lays out a timeline derived from the focus questions and through brainstorming creates news headlines that would characterize the future. We explored such diverse topics as the changing nature of the workplace, new geographic locations serving as centers of technology innovation, and the effects of different types of digital communities. Appendix E shows the scenario headlines for the "Fast & Furious" quadrant generated by Team 1. These scenario snippets then fed into the next step that focused on deriving strategy implications and watchpoints.

Figure 20 illustrates some of the implications that Team 1 generated when processing the futures characteristics and the Headlines from the Future. At this point, the focus of the method moved inward, to look at implications for the organization or community asking the question, more than looking outward at the world within which the community or organization exists. Note that the implications deal more with assurance-specific items than the axes of uncertainty or the futures characteristics did.

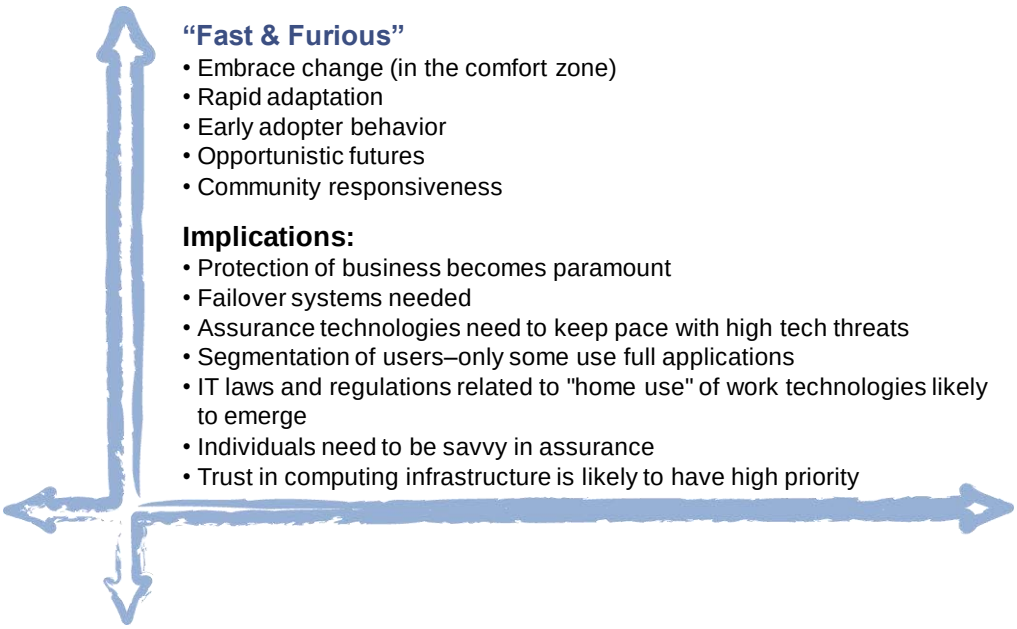


Figure 20: Implications of Paradigm Shift and Digital Natives

10.3 Observations

With the fast pace of technology advances and changes within operational environments, understanding potential future trends and external events sufficiently for them to be factored into today's decisions and plans is critical. While more could be done with the data derived from our **Strategic Alternatives Analysis**, at the very least it has revealed this critical need and demonstrated a viable approach for making potential trends and events explicit and for exploring their potential impact.

11 Conclusions and Next Steps

Assurance solution developers, as well as those who fund the formation, maturation, and transition of assurance solutions, need to understand the landscape of the assurance ecosystem to describe how their solutions address assurance. They need to determine where resources should be invested to gain the most assurance benefit, to identify the critical gaps in available solutions, and to accelerate the formation, adoption, and usage of solutions. The goal of the Software Assurance Landscape Project is to create an Assurance Modeling Framework that can be used to accelerate the adoption of software assurance solutions in operational settings.

To create an effective analysis approach, the landscape project team identified software assurance capabilities within desired assurance properties, such as security. To analyze a selected software assurance capability, we created a modeling framework that facilitates the systematic capture and analysis of relevant information to address a necessary set of research questions for a selected capability.

Use of the framework allows us to understand the relationships of assurance solutions to implemented assurance results. The modeling framework produces a profile for a selected assurance capability area through five activity categories. Each activity category focuses on developing insights on one or more of the framework information questions and produces one or more views. The set of views describes the profile. Each view is formed using a method. While the framework currently uses seven specific methods suitable for large complex socio-technical environments, other comparable methods could be substituted.

In a pilot, we chose to analyze vulnerability management to demonstrate the potential of the current framework for understanding and analyzing a selected software assurance capability. Each element of the framework was exercised but with varying degrees of coverage. Our goal was breadth rather than depth. Table 13 summarizes for each view in the framework, the method used and highlights which aspects of the vulnerability management capability area we explored. We placed greater emphasis on the activity categories *Determine Context and Scope* and *Characterize Current State: Ecosystem Relationships*. We worked with each view and its associated method long enough to determine the kind of information that could be obtained and the relevance of the information to the problem. To that end, we worked with each method until we had sufficient results to determine that we could use it well for other aspects of assurance. When we identified areas where a method was providing interesting insights we pursued it further. We stopped when the task became repetitive.

Table 13: Coverage of Vulnerability Management for Pilot

Framework View	Primary Focus of View	Method Used	Focus of Modeling for Pilot
Principal Perspectives and Influences	Captures the broad context for the selected assurance capability area and characterizes the critical stakeholders and primary relationships	Critical Context Analysis	CVE
Value Exchanged	Captures the interrelationships and high-level value exchanged among pairs of participating organizations and assurance solutions	Value Mapping	CVE, CWE, SAMATE, Secure Code
Potential Assurance Results	Captures and characterizes the ways in which assurance solutions align with what operational users do to achieve operational assurance results and identify high-level gaps and inefficiencies	SoS Focus Analysis	CVE
Motivations	Captures and evaluates drivers that are critical to achieving operational assurance objectives	Driver Identification and Analysis	Operational environments
Critical Behaviors	Captures the causal relationships among collections of participating organizations and assurance solutions to identify primary variables of interest and their influences that drive critical behaviors	System Dynamics	Software application providers: tension of reactive versus proactive responses to vulnerabilities
Adoption of Products	Captures the maturation and adoption mechanisms used and their effectiveness for collections of related assurance solutions	Technology Development and Transition Analysis	CVE, CWE
Future Drivers	Captures a range of future trends, influences, and uncertainties that may shape new operational demands and assurance solutions	Strategic Alternatives Analysis	Vulnerability management projected out to 2020

We have shown that we can identify potential levers for change with a better understanding of how constituent elements (e.g., organizations and assurance solutions) interoperate to achieve (or at times, hinder) software assurance. Several examples of the insights about vulnerability management are provided in the following section.

11.1 Example Insights from the Assurance Capability Area Profile

Part of evaluating the modeling framework includes determining the kinds of insights that are possible. The following are examples of the insights we gained through the pilot use of the modeling framework for vulnerability management:

- **The majority of the assurance solutions and participating organizations support supplier-related capabilities.** The focus is on the technology products themselves; the connections to operational assurance are assumed. Thus, the broader view on operational assurance is lost. We saw limited support for the operational side where several roles rely primarily on manual or home-grown approaches. These are areas of potential gaps and inefficiencies that offer opportunities for improvement. Without building the Potential Assurance Result view, the gaps and inefficiencies are difficult to perceive.

- **What motivates assurance solution suppliers is distinctly different from what motivates operational organizations.** Assurance solution suppliers within the vulnerability management capability area are motivated to identify and produce vulnerability patches quickly. In contrast, organizations associated with operational environments are motivated to maintain system availability and responsiveness. Security practices are seen as extra work by managers of operational environment. Thus, the return on investment for assurance solutions for operational organizations is not direct or compelling. Assurance solution suppliers prioritize their work and position their solutions from their perspective on quick release of patches. Yet operational organizations relate to the value of assurance solutions based on system availability and not timely patches. Bridging these conflicts in perspective could offer potential improvements for overall operational assurance. The Potential Assurance Result view highlights potential conflicts, which the Motivations view further expands and characterizes.
- **Understanding the similarities and differences in user communities for seemingly similar assurance solutions can be critical to the successful adoption and usage of assurance solutions.** While both CVE and CWE are dictionaries (or indexes) for particular aspects of vulnerability management information, their intended user communities are quite distinct. CVE is used primarily by operational roles such as security analysis and IT staff; CWE is oriented toward software developers. While some adoption and transition approaches, at least at a general level, can work for either community, certain key aspects will not. For example, each of these two communities has different terminology, communication sources, and priorities. The Adoption of Products view surfaces these issues.
- **There are important dynamics between the reactive and proactive responses to vulnerability management that affect the formation and adoption of assurance solutions.** The structure of the behavior feedback loops suggests that a balancing point between proactive software vulnerability prevention practices and reactive patch generation and release is needed. Too much focus on vulnerability prevention neglects the necessary response to critical patches needed (sacrificing short-term needs). Too much focus on patch generation sacrifices long-term advancement in SoS operational assurance. The Principal Perspectives and Influences view initially revealed the dichotomy within vulnerability management. Details of interactions were refined through the Value Exchanged and Potential Assurance Results views. The Critical Behaviors view then crystallized the dynamic behavior between reactive and proactive management.

11.2 Lessons Learned in Applying the Framework

Through piloting the framework, we saw that both the process of building each view and the resulting assurance capability area profile are beneficial. While we have captured key observations and lessons learned through applying the various methods described in this report, it is important to step back and look across the work as a whole. Several lessons are particularly noteworthy:

1. **The views, through their associated models, should be built in a particular order.** The optimal order is reflected in the five activity categories, starting with the first activity category shown in Figure 2 on page 8. The framework assists in growing an understanding of the selected assurance capability area as one applies the set of methods to produce the views.

2. **Understanding the assurance ecosystem of a selected capability requires multiple, interrelated perspectives.** Each view that makes up an assurance capability area profile provides useful and typically unique insights. Early prototypes showed that independently modeling collections of participating organizations and collections of assurance solutions is insufficient. Organizations are tightly bound to particular solutions. To better understand the assurance ecosystem, models must capture the interactions among organizations *in concert with* the assurance solutions and the values exchanged. We also saw that simply improving an assurance solution in the abstract does not necessarily improve the software assurance results within an end-user environment.
3. **The views provide a communication vehicle for affected stakeholders.** We have found useful ways in which to model portions of the software assurance ecosystem—at least in the small—that provide a means of communication among those participating in a selected capability area. For example, as we built these models, we reviewed them with vulnerability management assurance solution owners. That review has expanded the solution owners' understanding of how they fit within the landscape of the selected capability area. This information has had the side benefit of providing them with a clearer understanding of (1) other participating organizations and assurance solutions and (2) what they might consider doing in the future. Some of the models generated are easier for stakeholders to review and understand, such as value maps, while others require greater explanation, such as system dynamics models. Analysts will need to factor this into their use of the modeling framework and their targeted audiences.
4. **Future trend and technology maturation and adoption information should be reflected into other views.** While we found that we could readily gather this type of information, it will be more useful to apply additional effort—and research—to connect future focused perspectives back into the other categories of activities to build future views, particularly for values exchanged, motivations, and critical behaviors, to assemble a more comprehensive understanding of future impacts on assurance solutions.
5. **Building a profile of a selected assurance capability area creates a snapshot in time.** To remain useful, it must be kept current, since participating organizations and their roles and assurance solutions are continuously changing and expanding. The assurance ecosystem is a highly dynamic set of relationships. Decisions made at a particular point in time will need to be revisited as more information becomes available.

11.3 Next Steps

While we have demonstrated the viability of the framework, additional tasks would expand the applicability of the current work and set the stage for streamlining its use:

- **expanding the analysis of governance mechanisms.** Governance is a central aspect of the formation, maturation, adoption, and usage of assurance solutions. Where governance comes in, what form it takes, its effectiveness, and when the type of governance mechanism should change are key issues. Several of the views capture part of the needed information (e.g., Value Exchanged and Adoption of Products views). Further work is needed to determine how to analyze the effectiveness of a particular governance mechanism within a given context.
- **expanding the use of the framework to another assurance property or assurance capability area.** One option would be to select another assurance capability area within security

that emphasizes the business and operational side of a specific organization or organizational unit, such as threat analysis or incident management. This option would allow the SEI team to expand the applicability of the modeling framework and demonstrate value for operational entities as well as suppliers. In particular, it could show how assurance solutions currently align with the business or mission outcomes expected of operational units—and where there are gaps and thus improvement opportunities.

- **modeling of the future characterizations for vulnerability management.** During the pilot, we focused on gaining a reasonable understanding of how to capture a sufficient level of information about the current state of the vulnerability management assurance landscape. We did several experiments in understanding future trends and unexpected events to determine the importance of characterizing and modeling elements of the current state through the perspective of potential future trends and operational needs. To provide a stronger planning capability within the framework, approaches for characterizing the current profile views within the perspective of the future are needed.
- **modeling of the technology maturation and adoption for vulnerability management.** Available approaches for understanding and modeling technology maturation and adoption within the socio-technical complexities of even one aspect of software assurance appear to be very limited. While we made progress capturing basic information, we need to look for additional methods and continue experiments for modeling technology maturation and adoption of complex collections of assurance solutions and identify elements critical to assurance solution adoption.
- **expanding and vetting the behavioral system dynamics models with industry representatives.** The system dynamics models built during the pilot focus on the suppliers of assurance solutions. Those models should be expanded to incorporate more of the operational perspective and its demands. As part of the pilot, we validated the views built from value mapping and from technology development and transition analysis models with community representatives. This process provided a dual value: the team received feedback along with corrections and additions, but the community representatives found enormous value in the models in helping them to better understand the dynamics surrounding their particular technology. We think this community interaction is also critical for the system dynamics models.
- **expanding the framework to include methods to more formally identify and prioritize gaps and mitigations.** For the pilot, we relied on informal, ad hoc approaches to collect and characterize gaps and inefficiencies and propose recommended mitigations. Further work is needed to identify appropriate methods and adapt them to the needs of the software assurance ecosystem.

While the pilot required significant effort to adapt and, as much as possible, validate the methods and associated models to this new area, future applications of the framework will not need to repeat these activities. In addition, we should note that future applications will benefit from the lessons we have learned. Our work to date has established important groundwork. We have demonstrated the utility and feasibility of modeling important aspects of an assurance ecosystem by applying a range of modeling methods that can effectively be combined in a systematic way. The Assurance Modeling Framework has great potential to help the SEI's SoS software assurance re-

search to (1) identify impediments to creating, maturing, and adopting assurance technologies and (2) provide the DoD and other government organizations with better information to make investment decisions that improve the software assurance results they require. Without a reasonable understanding of how the software assurance ecosystem operates, it is too easy to apply funding, policies, and technology inappropriately.

Appendix A – Value Mapping

Appendix A contains the following items:

- template for technology profiles
- template for organizational profiles
- value map for Secure Code
- value map for SAMATE

Technology Profile

A. Technology Description

A1. Technology Name	A2. Description
Provide name of technology.	*Explain the basic purpose of the technology from an assurance perspective.*

B. Technology Background

B1. Technology Owner	B2. Investment	B3. Development Timeline
Provide name of technology owner.	*Provide number of FTE[26]/year for development.*	*Characterize development timeline (in years).*

B4. Target Audience	B5. Competitors	B6. Related Fields
For whom is this technology being built? *Who will be the user?* *What is the assurance focus of the target audience?*	*Who are the known and perceived competitors for this technology (in general and for assurance)?* *What is the distinguishing feature of this technology?*	*Identify any fields, disciplines, or bodies of knowledge that are related to the technology.*

B7. Notes
Provide any additional notes relevant to technology background.

[26] FTE is full-time equivalent, which is a ratio of the number of hours paid in a work period to the number of working hours in the business days in that period.

B7. Notes
Provide any additional notes relevant to technology background.

C. Technology Maturity

C1. Maturity Level	C2. Additional Information about Technology Maturity
Characterize the maturity level of the technology.	*Provide any information relevant to technology maturity.* *Consider adoption rates, start-up costs for customers, services available to support transition, etc.*
☐ Research ☐ Feasibility ☐ Demonstration ☐ Development ☐ Piloting ☐ Transition ☐ Sustainment	

C3. Risks
Document any risks or potential barriers that could affect development or adoption of this technology.

D. Technology and Whole-Product Characterization

D1. Core Technology

Determine the technology type of the core technology being profiled. Mark an 'X' in the box that describes the main assurance area addressed by that technology.

Assurance Area

Technology Type	Product Functionality	Quality Attributes	Process/Management	Compliance	Knowledge and Skills
Software Tool					
Method					
Data Repository					
Documentation					
Services					
Training					

D2. Transition Support

Mark an 'X' in each box for which a support product exists. Mark a 'P' in each box for which a support product is planned but not currently available.

Technology Type	Support Products
Software Tool	
Method	
Data Repository	
Documentation	
Services	
Training	

D3. Notes

Document any notes that are relevant to technology and the whole-product characterization.

E. Assurance Contribution from the Core Technology

E1. Assurance Area	E2. Description
*Mark an 'X' in the box for each assurance area marked in **Table D: Technology and Whole-Product Characterization**.*	*For each assurance area marked with an 'X,' describe or elaborate how the technology contributes to software assurance.* *Be explicit in your description.* *Also, describe the nature of the improvement to be expected (e.g., step change, reduce uncertainty, improve confidence, capability to measure or visibility into a situation, etc.)*
☐ Product Functionality	
☐ Quality Attributes	
☐ Process/Management	
☐ Compliance	
☐ Knowledge and Skills	

F. Assurance Characterization

F1. Discipline Orientation	F3. Notes: Discipline Orientation	F4. Life Cycle	F8. Notes: Control Categories
Characterize how the technology is oriented.	*Provide any additional notes relevant to discipline orientation.*	*Characterize the life-cycle phase in which the technology is used.*	*Provide any additional notes relevant to life cycle.*
☐ Software Development ☐ System Engineering/Integration ☐ IT Development ☐ IT or Systems Operations ☐ Acquisition		☐ Acquisition ☐ Requirements ☐ Design ☐ Development ☐ Test and Integration ☐ Operations ☐ User Recomposition	

F5. System Orientation	F6. Notes: System Orientation	F7. Control Categories	F8. Notes: Control Categories
Characterize system focus of the technology.	*Provide any additional notes relevant to system orientation.*	*Characterize the control categories addressed by this technology.*	*Provide any additional notes relevant to control categories.*
☐ Component ☐ Single System ☐ System of Systems		☐ Prevent/Avoid ☐ Detect ☐ Correct	

G. Technology Application / Demonstration

G1. Demonstrated	G2. Potential Demonstration	G3. Notes: Technology Application / Demonstration
Characterize where and how technology has been applied. Include results where possible/practical. *Please be specific about the life-cycle phases in which the technology has been piloted.*	*Characterize life-cycle phases where technology could be applied.*	*Provide any additional notes relevant to technology application, such as system types, customer types/niches.*

H. Relationships to other Technologies

H1. Technology Name	H2. Relationship
Provide name of each related technology.	*Describe the relationship to each related technology.*

Organizational Profile Questionnaire

1. What is the basic mission of your organization? In general, what type of work does your organization perform?

   Guidance to Interviewer

   You should get general information that describes the overall mission of the organization.

Notes

2. What role does your organization play with respect to software assurance?

   Guidance to Interviewer

   You need to get enough information to determine which of the following categories applies to the organization: solution seeker, solution identifier, solution standardizer, solution builder, solution evaluator, solution implementation controller, solution implementer, solution approver, regulator, trainer/educator, sustainer/operator.

Notes

3. In the area of software assurance, what does your organization produce or provide?

   Guidance to Interviewer

   You need to get enough information to determine which of the following categories applies: funding, services, products, publications, governance, other.

   Make sure that you provide sufficient context for each output type.

Notes

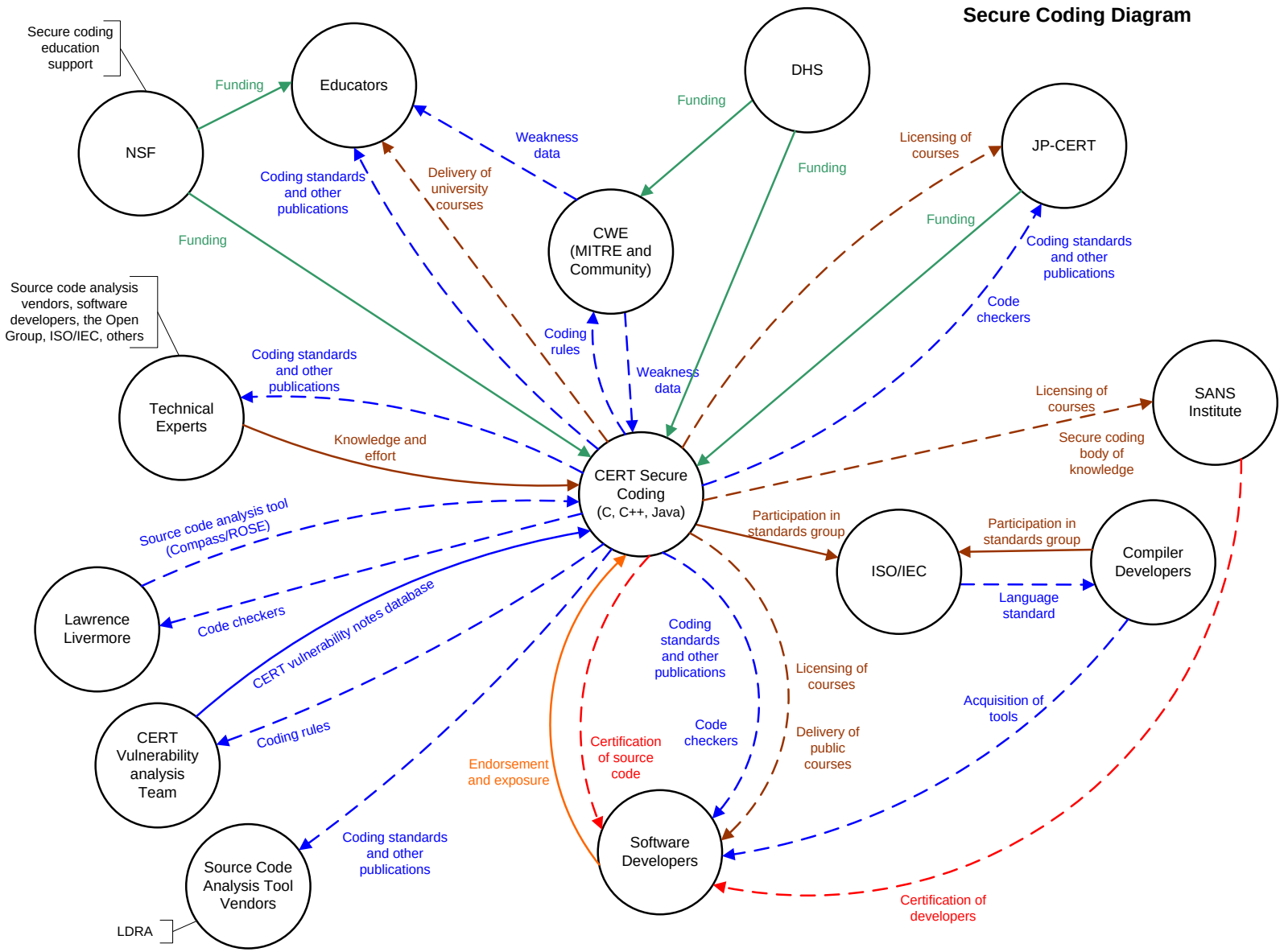
4. Which organizations provide value to your organization? To which organizations does your organization provide value?

   Guidance to Interviewer

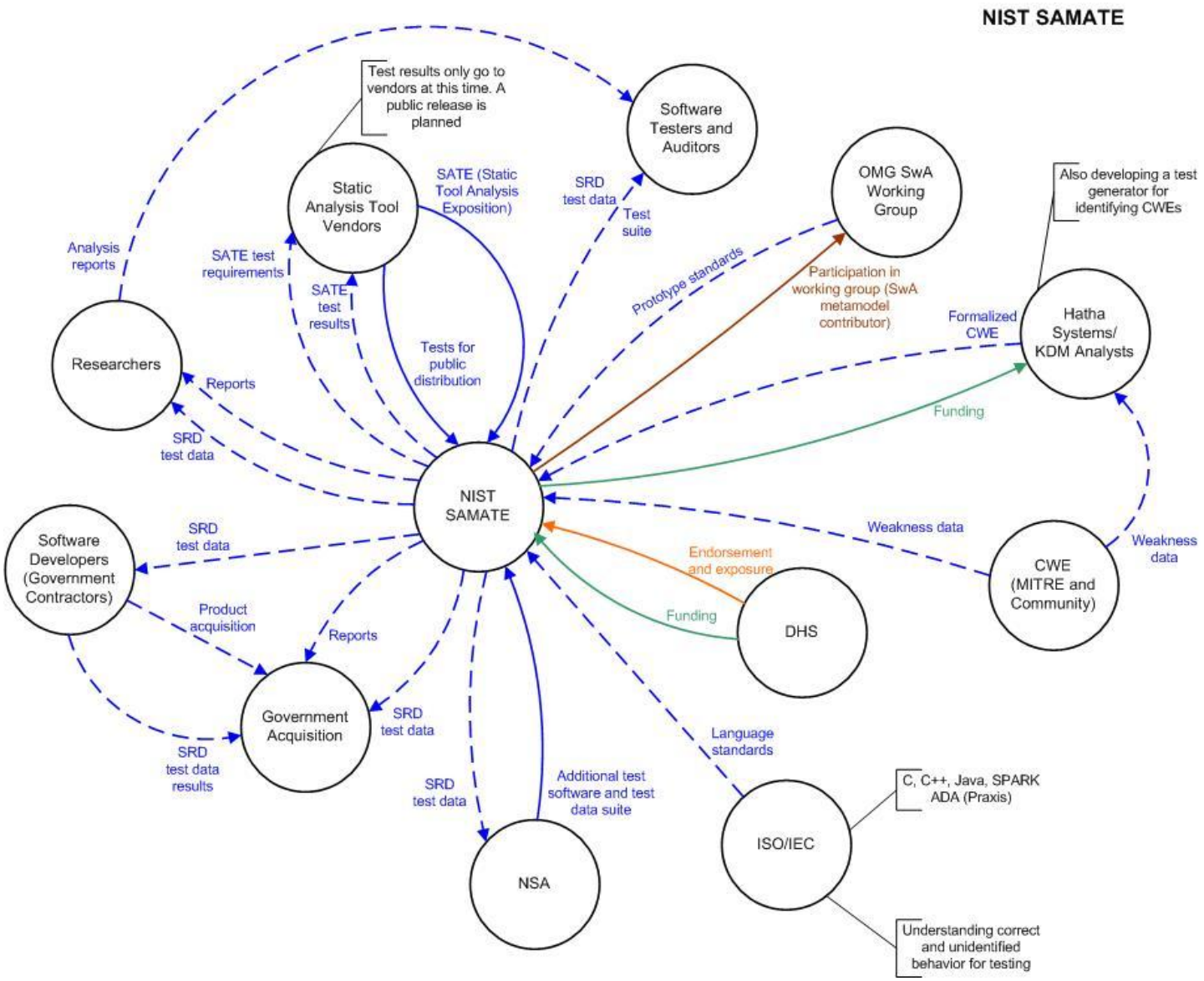
   You are trying to determine which organizations participate in value exchanges with the interviewee's organization.

Notes

Value Map for Secure Code (as of 26 February 2009)



Value Map for NIST SAMATE (as of 3 March 2009)



Appendix B – Driver Identification and Analysis

Appendix B contains the following items:

- Objectives for Vulnerability Management
- Candidate Drivers of Vulnerability Management

Objectives for Vulnerability Management

1.	To maintain a low-risk network environment
2.	To respond to reports of vulnerabilities in a timely manner (e.g., advisories and alerts, product vendor patches)
3.	To ensure that vulnerability solutions minimize adverse effects on users and operations (i.e., strike a site-useful balance between security and performance)

Candidate Drivers of Vulnerability Management

Driver Name		Success State	Failure State	Considerations
1.	Vulnerability Management Objectives	Vulnerability management objectives are realistic and achievable.	Vulnerability management objectives are unrealistic or unachievable.	Risk tolerance Timeliness of response System and network performance Alignment of objectives across all collaborators and partners Resources available
2.	Plan	The plan for managing vulnerabilities is sufficient.	The plan for managing vulnerabilities is insufficient.	Resources Funding Roles and responsibilities

Driver Name		Success State	Failure State	Considerations
3.	Process	The process being used to manage vulnerabilities is sufficient.	The process being used to manage vulnerabilities is insufficient.	Process design Measurements and controls Process efficiency and effectiveness Interoperability of processes among collaborators and partners Training
4.	Distribution Mechanisms	Mechanisms for distributing vulnerability information and solutions are sufficient.	Mechanisms for distributing vulnerability information and solutions are insufficient.	Distribution of advisories and alerts Application of patches Changes to system configurations
5.	Situational Awareness	Situational awareness of the system and network environments is sufficient.	Situational awareness of the system and network environments is insufficient.	Up-to-date documented baseline of all systems and networks Awareness of new vulnerabilities Documentation of patches applied Network topology diagrams IT asset inventory
6.	Task Execution	Vulnerability management tasks and activities are performed effectively and efficiently.	Vulnerability management tasks and activities are not performed effectively and efficiently.	Knowledge, experience, and expertise of management and staff Staffing levels Staff availability
7.	Coordination	Vulnerability management tasks and activities within each team and across teams are coordinated appropriately.	Vulnerability management tasks and activities within each team and across teams are not coordinated appropriately.	IT operations Management Users Vendors Stakeholders Communication Information sharing Dependencies Relationships
8.	External Interfaces	Work products from collaborators and partners meet quality and timeliness requirements.	Work products from collaborators and partners do not meet quality and timeliness requirements.	Vulnerability information Solutions Patches Advisories and alerts

Driver Name		Success State	Failure State	Considerations
9.	Information Management	Vulnerability information is managed appropriately.	Vulnerability information is not managed appropriately.	Usability Confidentiality Integrity Availability
10.	Technology	People have the tools and technologies they need to manage vulnerabilities effectively.	People do not have the tools and technologies they need to manage vulnerabilities effectively.	Software applications Infrastructure Systems Databases
11.	Facilities and Equipment	Facilities and equipment are sufficient to support vulnerability management.	Facilities and equipment are not sufficient to support vulnerability management.	Building Physical work spaces Support equipment Supplies Other resources
12.	Organizational Conditions	Enterprise, organizational, and political conditions are facilitating completion of vulnerability management tasks and activities.	Enterprise, organizational, and political conditions are hindering completion of vulnerability management tasks and activities.	Stakeholder sponsorship Actions of upper management Effects of laws, regulations, and policies
13.	Compliance	The vulnerability management program complies with all relevant policies, laws, and regulations.	The vulnerability management program does not comply with all relevant policies, laws, and regulations.	Policies Laws Regulations Standards of care
14.	Event Management	The vulnerability management program has sufficient capacity and capability to identify and manage future events and changing circumstances.	The vulnerability management program does not have sufficient capacity and capability to identify and manage future events and changing circumstances.	Risk mitigation plans, reach-back capability Business continuity plans Disaster-recovery plans Contingency plans
15.	Requirements	Vulnerability management requirements are well understood.	Vulnerability management requirements are not well understood.	Customer, user, and stakeholder requirements and needs System and network requirements

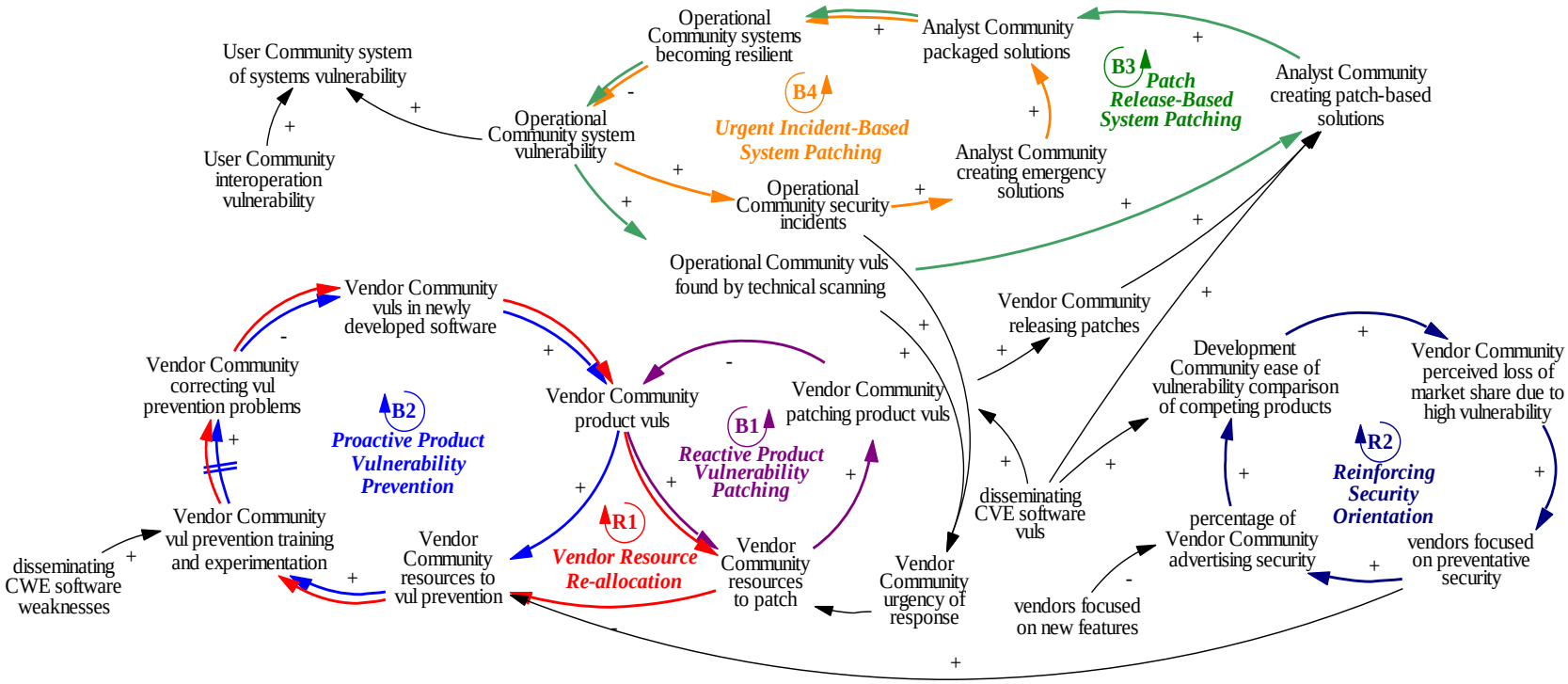
Driver Name		Success State	Failure State	Considerations
16.	Solution Tracking	The application of patches and solutions is tracked for all systems.	The application of patches and solutions is not tracked for all systems.	IT systems Desktop computers Laptops Mobile devices Networking components
17.	Risk Tolerance	Risks to the network environment are maintained within an acceptable tolerance over time.	Risks to the network environment are not maintained within an acceptable tolerance over time.	Development and documentation of risk mitigation plans Independent verification and validation of security posture Definition and documentation of risk tolerance ranges for systems and networks
18.	Unintended Consequences	The adverse or unintended effects of vulnerability solutions and patches are minimized.	The adverse or unintended effects of vulnerability solutions and patches are not minimized.	Effect on users and operations Performance of systems and networks Confidentiality, integrity, and availability requirements Balance between security and performance

Appendix C – System Dynamics

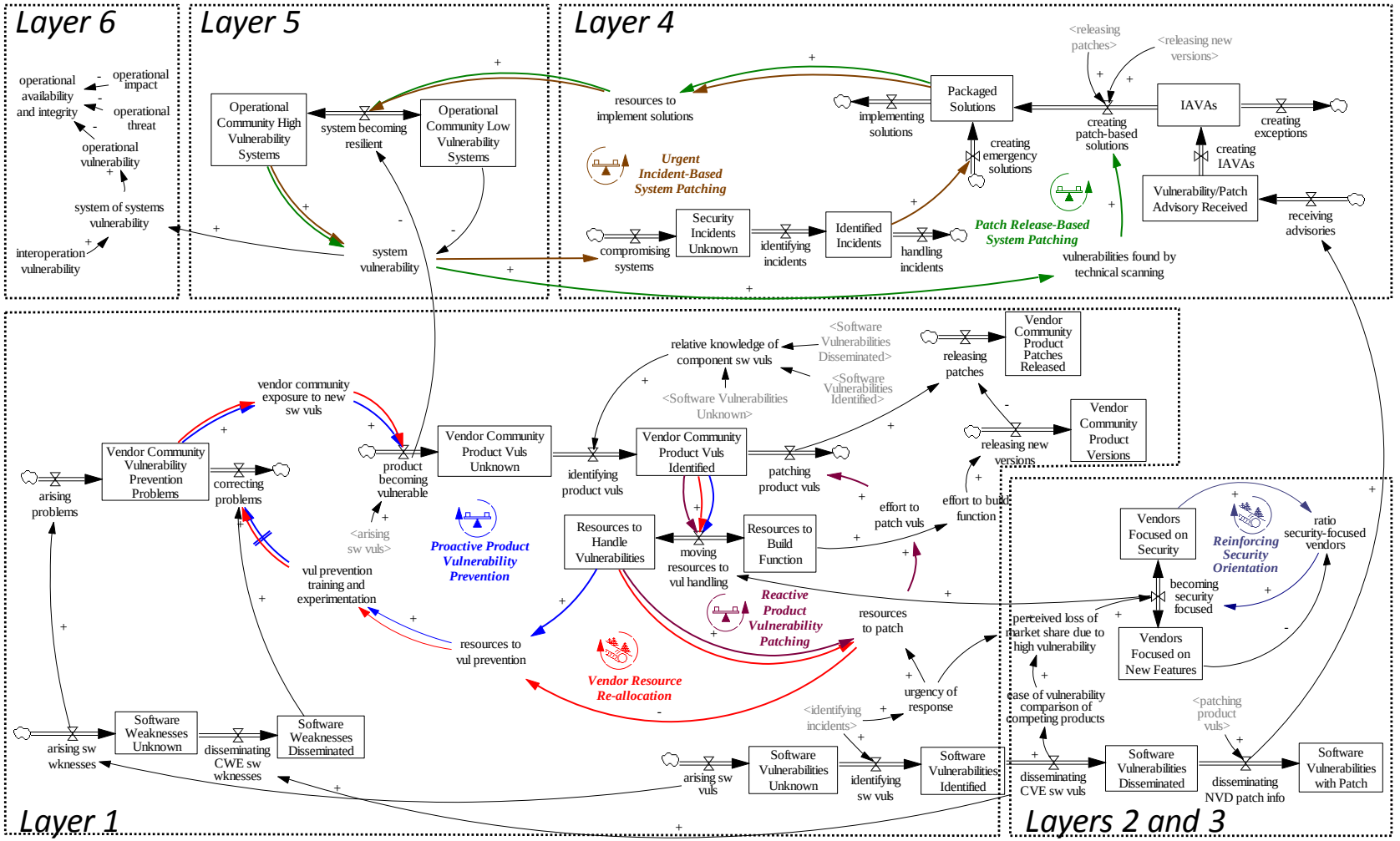
Appendix C contains the following items:

- an abstracted version of the system dynamics model, which is based on a restricted notation and is scoped to emphasize the main points to be made
- detailed system dynamics model on which the abstracted version is based

Abstract System Dynamics Model



Detailed System Dynamics Model



Appendix D – Technology Transition Analysis

Appendix D contains the following items:

- detailed success factors for CVE
- CWE indicators of success
- CWE success factors

Detailed Success Factors for CVE

Success Factor	Evidence or Further Characterization
MITRE identified a clear market need (from a community perspective).	The operational community was feeling pain.
	Vulnerability information was compiled by multiple groups (producing multiple lists).
	The number of vulnerabilities was rapidly increasing.
	Use if the internet was growing (greater interconnectivity).
	The timeline for responding to vulnerabilities was decreasing.
	Vulnerability information about specific products was widely dispersed among many groups.
Vendors were motivated to participate.	Vendors were unable to gage the severity of the vulnerabilities being reported.
	Researchers were generating a tremendous amount of noise related to the number of vulnerabilities.
MITRE's strategy allowed it to partner with researchers and content providers/list makers.	
A growing amount of vulnerability information was distributed across multiple databases (operated by competing groups).	The need for indexing and cross referencing began to emerge.
MITRE filled an unmet community need with CVE.	MITRE's traditional business focus required it to understand a range of technologies (across many acquisition communities).
	MITRE explored the vulnerability market looking for opportunities.
	MITRE saw the market differently from content providers/list makers and vendors.
	MITRE was not in a position to compete with content providers/list makers and researchers.
	MITRE's solution compiled information across all technologies.

Success Factor	Evidence or Further Characterization
MITRE signed agreements with vendors to get information earlier.	
MITRE's proof of concept using public data convinced vendors of the value of the CVE approach.	
MITRE identified the right stakeholders and did a good job of getting them involved in building the solution.	To continue to get funding, MITRE had to show applicability and acceptance within the community.
MITRE explicitly focused on reducing the barriers to adoption.	MITRE did not refer to CVE as a standard until the community began referring to it as a standard.
MITRE's solution did not force adopters to change the way they did business.	Barriers to adopting CVE are low.
Government policy – DoD IAVA was rewritten to include CVE.	
MITRE continues CVE "marketing" and product evolution.	Evolve product to keep it current.
	Keep control of "brand."
There is continued investment in infrastructure.	Efficiency of infrastructure (e.g., automatic push to NVD)
Community articulated "standard" before MITRE used the term.	Resistance to adoption is reduced.
Focus on building collaborations.	Leverage for further maturation.
	Mine collaborations.

CWE Success Indicators

Indicators of Maturation and Adoption Success of CWE
Reduced vulnerabilities in deployed system
Broad motivation of researchers and vendors to define and build corrective actions
Uniform way to identify and characterize weaknesses
Number of tools that identify known weaknesses
Growing range of weaknesses addressed
Increasing percentage of static analysis tools that are CVE compatible
Mapping the identification of weaknesses to appropriate vulnerabilities
Developer of government-delivered systems describes code by number and kind of CVEs prevented
Training of CWE and how to address weaknesses routinely used by architects, designers, and developers

CWE Success Factors

Success Factors for CWE
PLOVER ’05 – list of vulnerability weaknesses and their source by researchers
Huge numbers of new vulnerabilities (doubling each year)
2004 research that showed applications are biggest source of potential vulnerabilities
Catalyzed government and other major commercial domains awareness of the problem (e.g., Choice Point [2005] confidentiality breach [stock price plummets, customers lost])
Gartner study finds 75% of hacks occur in the software rather than in infrastructure (e.g., networks, servers) Catalyzed important of identifying the source of weaknesses Provides push for PLOVAR work
Ready identification of weaknesses and remediation patterns
Overcoming developer biases and current practices
Code generation environments that use higher order languages that auto-generate source code to C, C++, Java
DoD standard desktop configuration reduces vulnerabilities by 75% (source: head of SANS in congressional testimony) May slow drive for solutions May provide motivation for vendors to demonstrate “goodness” to be included in “authorized” desktop configurations

Appendix E – Strategic Alternatives Analysis

Appendix E contains the following item:

- generated scenario headlines from Team 1

Team 1 Headlines – "Fast and Furious"

Thread	2009	2011	2013	2015	2017	2018-20
Telepresence	Software icon Grady Booch stops business travel -- his avatar gives all his presentations online	Global satellite grid available for public internet access			The average age to graduate from a 4 year college is reduced from 21 to 17 as a result of "learn at your own pace" education programs that were started in 2010	The Museum of Desktop Computers was visited by 100 million people virtually in the first hour it was open -- avatar tours were the most popular mode of touring the facility
Economic Failures or Successes		Global recession hits new low in terms of GDP	Power grid goes down due to failure in cyber control systems Collected financial loss of U.S.-based companies over an eight-year period is estimated at $50 Billion US			

Thread	2009	2011	2013	2015	2017	2018-20
Access to Technology		Schools expand technology course offering by 5X: Business recruiting heavily from graduates of new programs	Expansion of technology user base: techies take over cyber infrastructure Factory opens in Brazil to produce $50 mobile computer Low cost computers made in Brazil provided to every citizen: training of children worldwide is subsidized by the World Trade Organization (WTO)	Collaborative Grid Technologies (CGT) announces availability of icon-based user tools to link services and applications		The Geographic Placement Application (GPA) developed on iPhone software heralded as the turning point in wiping all nuclear bases from the Earth Fisher Price announced icon-based applications integration toy for five-year-olds U.S. passes legislation that relieves security requirements -- personal data is no longer private

Thread	2009	2011	2013	2015	2017	2018-20
Work at Home			Commuters and Global Warming: Government announces competition to develop technology supporting work-at-home for knowledge workers	Collaborative Grid Technologies, Inc announces new SoS assurance capabilities IRS changes tax structure to de-incentivize brick/mortar buildings to house knowledge work IBM Global Services announces closure of 10 out of 12 office complexes: employees to operate remotely via IBM Grid		Ivy League and other major universities around the world agree to take all their classes online. Buildings are put up for auction for recreational use The SEI building was claimed by "Doctors for the World" to offer free of charge health care -- CMU students will volunteer their computing environments for health care application use
Move to Africa				Knowledge workers routinely work from "home" -- a beach in Software Assuranceziland		Major North American cities are deserted due to constant water shortages East African economy flourishes with the influx of high-tech immigrants from across the world The world is down to 5 major airline carriers: Air Africa, Singapore Airlines, Southwest, Kenya Airlines and ?????

Glossary

Activity category
An activity necessary to provide insights for one or more key questions that the Assurance Modeling Framework must address.

Assurance capability area
A set of related activities used to achieve an assurance property. Vulnerability management, incident management, and threat analysis are assurance capability areas for the security assurance property.

Assurance capability area profile
A set of views that collectively describe the relevant elements of the assurance ecosystem landscape for a selected assurance capability area.

Assurance ecosystem
The broad range of interrelated elements that influence operational assurance, including organizations, decision makers, policies, practices, technologies, and people.

Assurance property
A property of assurance such as security, safety, reliability, and robustness, that an operational system or system of systems (SoS) may need to provide.

Assurance solutions
Policies, practices, and technologies related to software assurance.

Attack
An intentional exploitation of a vulnerability.

Common Vulnerabilities and Exposures (CVE) List
From http://cve.mitre.org/
"CVE is a dictionary of publicly known information security vulnerabilities and exposures. CVE's common identifiers enable data exchange between security products and provide a baseline index point for evaluating coverage of tools and services."

Common Weakness Enumeration (CWE) Dictionary
From http://cwe.mitre.org/
"CWE provides a unified, measurable set of software weaknesses that is enabling more effective discussion, description, selection, and use of software security tools and services that can find these weaknesses in source code and operational systems as well as better understanding and management of software weaknesses related to architecture and design."

Critical Context Analysis
A method that provides an understanding of the broad context of software vulnerability management, including two principal operational scenarios, and a characterization of the critical stakeholders and primary relationships.

Developer community
SoS developer community (henceforth, called developer community) develops custom software systems from which an SoS is built. Often, custom software is built from vendor products or expects to operate on an IT infrastructure composed of IT commercial, off-the-shelf (COTS) products.

Exploit
A particular means of using a vulnerability in an attack.

Information Assurance Vulnerability Alert (IAVA)
From http://en.wikipedia.org/wiki/Information_Assurance_Vulnerability_Alert
"An Information Assurance Vulnerability Alert (IAVA) is an announcement of a computer application software or operating system vulnerability notification in the form of alerts, bulletins, and technical advisories identified by DoD-CERT, a division of the Joint Task Force-Global Network Operations. These selected vulnerabilities are the mandated baseline, or minimum configuration of all hosts residing on the GIG. JTF GNO analyzes each vulnerability and determines if is necessary or beneficial to the Department of Defense to release it as an IAVA. Implementation of IAVA policy will help ensure that DoD Components take appropriate mitigating actions against vulnerabilities to avoid serious compromises to DoD computer system assets that would potentially degrade mission performance."

National Vulnerability Database (NVD)
From http://nvd.nist.gov/
"NVD is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. NVD includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics."

Operator community
The operator community operates a system or SoS built from vendor products as the IT infrastructure and custom software built by developers (and may include COTS products.

Software assurance
Software's contribution system and SoS assurance. System assurance is the justified confidence that a system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle. Justified confidence requires establishing a rational for defining readiness for use. Functioning as intended involves confirmation that a system meets user expectations for some specified level of confidence. Evaluation of a system's intended functionality must be considered within the environment of actual use and not considered in relation to a projected or idealized environment of use.

Software Assurance Metrics and Tool Evaluation (SAMATE)
From http://samate.nist.gov
"The NIST SAMATE (Software Assurance Metrics And Tool Evaluation) project is dedicated to improving software assurance by developing methods to enable software tool evaluations, measuring the effectiveness of tools and techniques, and identifying gaps in tools and methods. …

The scope of the SAMATE project is broad: ranging from operating systems to firewalls, SCADA to web applications, source code security analyzers to correct-by-construction methods."

SoS Focus Analysis
A method that provides an understanding and characterization of the gaps and alignment in the ways suppliers of assurance-related technologies or practices provide capabilities or services and what operational users do to achieve operational assurance results.

Strategic Alternatives Analysis
A method that provides an understanding of an applicable range of future trends, influences, and uncertainties that may shape new operational demands and assurance practices.

System Dynamics
A method that provides an understanding of the causal relationships among collections of participants and identification of the primary variables of interest and their influences that drive critical behaviors.

Technology Development and Transition Analysis
A method that provides an understanding of the maturation and adoption mechanisms used and their effectiveness for collections of related technologies, practices, and products at varying states of maturity and adoption.

Value Mapping
A method that provides an understanding and capture of the interrelationships and high-level value exchanged among pairs of participants associated with assurance technologies.

Vendor community
IT vendor/supplier community develops commercial software products that forms the IT infrastructure used in operations.

View
A model or data formed from an activity category using one or more methods within the Assurance Modeling Framework to build an assurance capability area profile.

Vulnerability (vul)
A defect in software that allows someone to gain unauthorized access to a system, software, or a network.

Vulnerability management
A process of prevention, discovery, and correction of vulnerabilities. Discovery and correction of vulnerabilities is often referred to as patching. Prevention usually involves improving the software development process so that vulnerabilities are not introduced into software artifacts in the first place.

References

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

[Alberts 2009]
Alberts, C. & Dorofee, A. *A Framework for Categorizing Key Drivers of Risk* (CMU/SEI-2009-TR-007). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2009. http://www.sei.cmu.edu/library/abstracts/reports/09tr007.cfm

[Forrester 2003]
Forrester, E. "A Life-Cycle Approach to Technology Tranistion." *News at SEI*, September, 2003. Software Engineering Institute, Carnegie Mellon University, 2003.

[Green 2004]
Green, A. & Jack, A. "Creating Stakeholder Value by Consistently Aligning the Support Environment with Stakeholder Needs." *Facilities Journal 22*, 13/14 (2004): 359-363.

[Jack 2002]
Jack, A. *Value Mapping–A Second Generation Performance Measurement and Performance Management*. http://www.valuebasedmanagement.net/articles_jack_value_mapping_second_generation_performance_management.pdf (2002).

[Jolly 1997]
Jolly, V. *Commercializing New Technologies: Getting from Mind to Market*. Harvard Business School Press, 1997.

[Kelly 2005]
Kelly, E. *Powerful Times: Rising to the Challenge of Our Uncertain World*. Wharton School, 2005.

[Meadows 2008]
Meadows, D. *Thinking in Systems: A Primer*, Chelsea Green Publishing, 2008.

[MITRE 2009a]
The MITRE Corporation. CWE-20: Improper Input Validation. http://cwe.mitre.org/data/definitions/20.html (2009)

[MITRE 2009b]
The MITRE Corporation. 2009 CWE/SANS Top 25 Most Dangerous Programming Errors. http://cwe.mitre.org/top25/index.html (2009)

[Moore 1999]
Moore, G. *Crossing the Chasm, Revised Edition*, Harper Business, New York, NY., 1999.

[Prensky 2000]
Prensky, M. *Digital Game-Based Learning.* McGraw-Hill, 2000.

[Schwarz 1996]
Schwarz, P. *Art of the Long View*. Currency Doubleday, 1996.

[Seacord 2005]
Seacord, R. *Secure Coding in C and C++*. Addison-Wesley, 2005.

[Siviy 2009]
Siviy, J.; Moore, A.; Alberts, C.; Woody, C.; & Allen, J. "Value Mapping and Modeling SoS Assurance Technologies and Assurance Supply Chain," 236-240. *IEEE Internatioanl Systems Conference*. Vancouver, Canada, March 2009. IEEE, 2009.

[Sterman 2000]
Sterman, J. *Business Dynamics: System Thinking and Modeling for a Complex World*. McGraw-Hill, 2000.

[Tornatzky 1990]
Tornatzky, L. & Fleischer, M. *The Processes of Technological Innovation.* Lexington Books, 1990.

[Van der Heijden 2005]
Van der Heijden, Kees. *Scenarios: The Art of Strategic Conversation*, 2nd edition. Wiley, 2005.

[Woody 2009]
Woody, C.; Brownsword, L.; Alberts, C.; & Moore, A. "The Landscape of Software Assurance—Participating Organizations and Technologies." *Infotech@Aerospace Conference*, American Institute of Aeronautics and Astronautics, IEEE, Seattle, WA, 2009. AIAA 2009-1919

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	August 2010	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
A Framework for Modeling the Software Assurance Ecosystem: Insights from the Software Assurance Landscape Project	FA8721-05-C-0003

6. AUTHOR(S)

Lisa Brownsword; Carol C. Woody, PhD; Christopher J. Alberts; Andrew P. Moore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2010-TR-028

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	ESC-TR-2010-028

11. SUPPLEMENTARY NOTES

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

13. ABSTRACT (MAXIMUM 200 WORDS)

This report describes the Carnegie Mellon® Software Engineering Institute (SEI) Assurance Modeling Framework. It also discusses an initial piloting of the framework to prove its value and insights gained from that piloting for the adoption of selected assurance solutions. The SEI is developing a way to model key aspects of assurance to accelerate the adoption of assurance solutions within operational settings for the U. S. Department of Defense (DoD) and other government organizations. As part of that undertaking, SEI researchers have developed an Assurance Modeling Framework to build a profile for an assurance capability area such as vulnerability management within an assurance quality such as security. The profile consists of many views developed using selected methods and models. From the analysis of these views, inefficiencies and candidate improvements for assurance adoption can be identified.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Software assurance, assurance modeling, vulnerability management, system of systems	106

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL