Software Assurance in Acquisition: Mitigating Risks to the Enterprise

---

A Reference Guide for Security-Enhanced Software Acquisition and Outsourcing

October 22, 2008

This document is offered for informative use; it is not intended as a policy or standard.

When referring to, quoting, or excerpting from this document, please always ensure proper acknowledgement is given.

The Software Assurance (SwA) Acquisition Working Group is seeking additional participation in refining this document.

NO WARRANTY

This material is furnished on an “as-is” basis. The authors, contributors, members of the SwA Acquisition Working Group, their employers, the U.S. Government, other sponsoring organizations, all other entities associated with this report, and entities and products mentioned within this report make no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. No warranty of any kind is made with respect to freedom from patent, trademark, or copyright infringement.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder. No warranty is made that use of the information in this document will result in software that is secure. Examples are for illustrative purposes and are not intended to be used as is or without undergoing analysis.

TABLE OF CONTENTS

Acknowledgments

The Department of Defense (DOD) and Department of Homeland Security (DHS) Software Assurance (SwA) Acquisition Working Group is composed of government, industry, and academic members. The working group produced this document as its first step in raising acquisition official awareness on how to incorporate SwA considerations throughout the acquisition process. This document was developed by:

Editor:

- Mary Linda Polydys, National Defense University (NDU) Information Resources Management College (IRMC)

Authors:

- Mary Linda Polydys, NDU IRMC
- Stan Wisseman, Booz Allen Hamilton (contract with Department of Homeland Security [DHS] National Cyber Security Division [NCSD])

Additional contributors included:

- Nadya Bartol, Booz Allen Hamilton
- Brad Doohan, Australian Defence Materiel Organisation (working with Software Engineering Institute [SEI] and Defense Contract Management Agency [DCMA])
- Greg Gogates, Fasor Inc. (in support of Federal Drug Administration [FDA])
- Karen Goertzel, Booz Allen Hamilton
- Joe Jarzombek, DHS NCSD
- Steven Lavenhar, Booz Allen Hamilton
- Michael Leichtman, Booz Allen Hamilton
- Tom O'Flaherty, INPUT
- Jeff Williams, Aspect Security

Detailed reviews of the document were provided by:

- Mary Ann Davidson, Oracle
- Lauren Eisenberg Davis, Johns Hopkins University Applied Physics Laboratory
- Annabelle Lee, National Institute of Standards and Technology (NIST)
- Sandra Ludwig, Booz Allen Hamilton
- Paul Nicholas, Microsoft

During most working group meetings, participants provided feedback on draft materials for the document. Working group participants have included:

- Lawrence Baker, Defense Acquisition University
- Ed Barger, Boeing
- Sean Barnum, Cigital, Inc.
- Redge Bartholomew, Rockwell Collins
- Nadya Bartol, Booz Allen Hamilton
- Joseph Bergmann, The Open Group
- Paul E. Black, NIST
- Andy Bochman, Ounce Labs
- Sharon Bowen, Northrop Grumman

- John Campbell, National Security Agency (NSA)
- Marlene Chandler, Department of State
- Penny Chase, The MITRE Corporation
- Chuck Chrissis, SureLogic, Inc.
- Carl Clavadetscher, NDU
- Kyle Crawford, Georgia Tech
- Rita Creel, SEI/CMU
- Paul Croll, CSC
- Peter Cybuck, Sharp Electronics
- Mary Ann Davidson, Oracle
- Lauren Eisenberg Davis, Johns Hopkins University Applied Physics Laboratory
- Lori Davis, Mobile Armor
- Ross Dence, Booz Allen Hamilton
- Avinash Deolalikar, CSC
- Terry Devine, The MITRE Corporation
- Michelle Dickey, Fortify Software
- Robert Dix, Citadel Security Software, Inc.
- Sally Dixon, U.S. Army, Chief Information Office, G6
- Bradley Doohan, Australian DMO (on exchange with DCMA, assigned with SEI/CMU)
- Bob Ellison, Software Engineering Institute
- Jeremy Epstein, Cigital, Inc.
- Daniel Fisher, ADA Core
- Greg Foley, OPS Consulting
- Kevin Foley, U.S. Air Force
- Steve Foote, The MITRE Corporation
- Michael Garcia, Boeing
- Gene Gerard, CTC
- Greg Gogates, Fasor Inc.
- Becky Grant, DCMA
- Chris Gunderson, Naval Postgraduate School
- Fred Hall, Assurance Engineering, Inc.
- Deanne Harwood, DHS CSIRC
- Dede Haskins, Independent Management and Business Strategy Consultant
- Frank Herman, Fraunhofer Center for Experimental Software Engineering Maryland
- George Huber, SRI International
- Wesley Higaki, Symantec Corporation
- William Janosky, Army
- Joe Jarzombek, DHS NCSD
- Bruce Jenkins, U.S. Air Force
- Lisa Kamae, DCMA
- Susanna Kass, Palamida
- Clint Kreitner, CISecurity
- Rick Kuhn, NIST
- Michael Leichtman, Booz Allen Hamilton
- Glen Logan, Open Systems Joint Task Force
- Sandra Ludwig, Booz Allen Hamilton
- Ernest R. Lucier, Federal Aviation Administration
- Steven Mackie, Wyle Laboratories, Inc.
- Surneet Malhotra, Unisys and OMG

- Robert A. Martin, The MITRE Corporation
- Rich Matias, Lockheed Martin
- Joe McManus, U.S. Computer Emergency Readiness Team
- Julie Mehan, Hartha Systems
- James Moore, The MITRE Corporation
- Rama Moorthy, Hartha Systems
- Kristy Mosteller, Booz Allen Hamilton
- Janet Mostow, IBM
- Haleh Nematollahy, Fortify Software
- Paul Nicholas, Microsoft
- Tom O'Flaherty, INPUT
- Mary Polydys, NDU
- Sam Redwine, James Madison University
- Bob Reynolds, Institute for Defense Analyses
- Jim Robbins, EWA Canada
- Skip Romero, Booz Allen Hamilton
- Warren Russell, DOD Under Secretary of Defense for Intelligence
- Thomas Santaniello, CompTIA
- Richard Struse, VOXEM
- Andras Szakal, IBM
- Jeffrey Tebbe, Idaho National Lab
- Jan Morgan Vargas, Software Engineering Institute, Carnegie Mellon University
- Rafael Vasquez, ESYCOM Cybernet, Inc.
- Larry Wagoner, NSA
- John Weiler, Interoperability Clearinghouse
- David A. Wheeler, Institute for Defense Analyses
- Stan Wisseman, Booz Allen Hamilton
- Dan Wolf, Cyber Pack Ventures
- Robert Wyatt, DOD

Special thanks go to:

- Joe Jarzombek, Director of Software Assurance, DHS NCSD, whose leadership and support enabled the successful completion of this document.

The editor and authors want to thank their co-authors for mutual help, the working group members for the review comments received and the constructive discussions that occurred, and reviewers who contributed their valuable time to improve this document. The working group's life extends beyond the production of this document, and its goals remain the same—to incorporate SwA considerations in the acquisition process relative to potential risk exposures that could be introduced by the supply chain—but its specific activities may vary. The working group welcomes participation in its ongoing activities because it is expected that this information and related SwA Acquisition material will continue to evolve based on use and changing needs to mitigate risks to the enterprises dependent on software.

Executive Summary

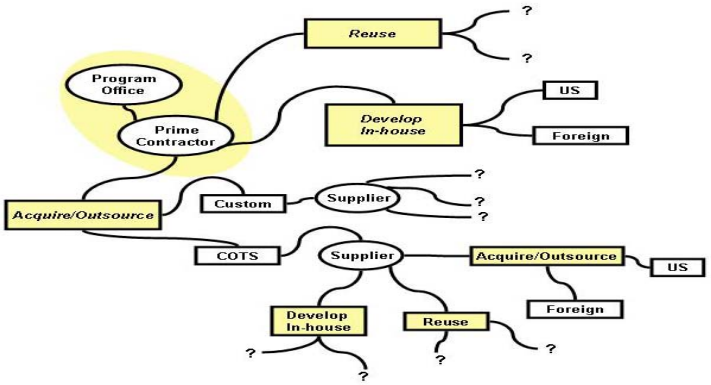
Software vulnerabilities, malicious code, and software that does not function as promised pose a substantial risk to the Nation's software-intensive critical infrastructure that provide essential information and services to citizens. Minimizing these risks is the function of software assurance (SwA). Software assurance is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that it functions in the intended manner [CNSSI No. 4009].

SwA is a key element of national security and homeland security. It is critical because dramatic increases in business and mission risks are attributable to software that does not perform as intended and is exploitable.[1] Exploitable software is vulnerable to attack. Software vulnerabilities jeopardize intellectual property, consumer trust, business operations and services, and a broad spectrum of critical infrastructure, including everything from process control systems to commercial software products.

To ensure the integrity of business operations and key assets within critical infrastructure, software must be reliable and secure. A Chief Information Officer Executive Council™ poll[2] found that the top two most important attributes of software are "reliable software that functions as promised" and "software free from security vulnerabilities and malicious code."

A broad range of stakeholders now needs justifiable confidence that the software that enables their core business operations can be trusted to function as expected (even with attempted exploitation) and can contribute to more resilient operations.[4] Therefore, the responsibility for SwA must be shared not only by software suppliers in the supply chain but also by the acquirer in the supply chain who purchase the software. There is a concern, however, that acquirers are not aware of this responsibility and are inadequately prepared to support SwA in the acquisition process.

Figure ES–1. Potential Software Supply Chain Paths[3]



In 2003, the U.S. Department of Defense (DOD), joined by the Department of Homeland Security (DHS), launched a SwA initiative[5] to address SwA

1 See the President's Information Technology Advisory Committee, "Cyber Security: A Crisis of Prioritization," February 2005, available at <http://www.nitrd.gov/pitac/reports/index.html>.

2 CIO Executive Council News Bureau, "New CIO Executive Council™ Poll Reveals CIOs Lack Confidence in Software," October 11, 2006, available at <https://www.cioexecutivecouncil.com/nb/>.

3 Modified version of [DACS-Walker].

4 Ability to rapidly recover and resume normal operations.

5 Then Deputy Director for SwA, Information Assurance Directorate, Office of Assistant Secretary of Defense (Networks and Information Integration) Joe Jarzombek led the DOD SwA initiative that submitted an interim report. The DOD CIO forwarded that report with findings and recommendations to the Committee on National Security Systems Joint Working Group on the Globalization of IT.

concerns of poor quality, unreliable, and nonsecure software. The acquisition working group, consisting of representatives from government, industry, and academia, was established to address how to leverage the acquisition process to influence SwA in the supply chain. To that end, the SwA Acquisition Working Group created this document to inform acquisition officials on how to influence SwA in software supply chain management by leveraging and including SwA considerations in the acquisition process.

This document provides information on incorporating SwA throughout the acquisition process from the acquisition planning phase to contracting, monitoring and acceptance, and follow-on phases. For each phase, the material covers SwA concepts, recommended strategies, and acquisition management tips. The document also includes recommended request for proposal and/or contract language and due diligence questionnaires that may be tailored by acquisition officials to facilitate the contract evaluation process.

This document's recommendations are noted in the September 2007 Report of the Defense Science Board Task Force on "Mission Impact of Foreign Influence on DOD Software":

- [T]he mere fact of asking what vendors do to engineer security and quality into their lifecycle puts the vendor community on notice that it is important to DOD.
- The DOD/DHS software assurance forum has been working on a procurement guide focused on software assurance, which helps procurement officers glean (through a series of questions) what vendors have (and have not) done as part of their secure development process, how they handle vulnerabilities, and so on. Such a document, when reviewed by a larger audience and finalized, could be used as part of IT [information technology] procurement cycles to help DOD better evaluate risk.
- As long as this is sensible, the questions are phrased to allow expository answers, and the benefit derived is commensurate with the cost of vendors completing it, this is one way for DOD both to know what they are getting and to put vendors on notice that quality and security-worthiness have become purchasing criteria for DOD.

1 Introduction

1.1 Background

Software vulnerabilities, malicious code, and software that does not function as promised pose a substantial risk to the Nation's software-intensive critical infrastructure that provide essential information and services to citizens. Minimizing these risks is the function of software assurance (SwA). Software assurance is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that it functions in the intended manner [CNSSI No. 4009]. Software assurance focuses on the management of risk and assurance of safety, security, and dependability within the context of system and software life cycles [ISO/IEC 15026].

Software assurance is a key element of national security and homeland security. It is critical because dramatic increases in business and mission risks are attributable to exploitable software.[6] Software vulnerabilities jeopardize intellectual property, consumer trust, business operations and services, and a broad spectrum of critical infrastructures, including everything from process control systems to commercial software products. To ensure the integrity of business operations and key assets within critical infrastructures, software must be reliable and secure. A Chief Information Officer Executive Council™ poll[7] found that the top two most important attributes of software are "reliable software that functions as promised" and "software free from security vulnerabilities and malicious code."

In 2003, the U.S. Department of Defense (DOD) launched a SwA initiative,[8] and the Department of Homeland Security (DHS) joined this initiative to address SwA concerns of poor quality, unreliable, and nonsecure exploitable software. Several working groups were established to address SwA concerns. The acquisition working group, consisting of representatives from government, industry, and academia, was established to address how to leverage the acquisition process to influence SwA and reduce risks in software supply chain management.

The software supply chain consists of (but is not exclusive to) the following: the acquirers in industry and government, information assurance personnel supporting acquisition managers, decisionmakers for software procurements (including program/project managers and requirements personnel), prime contractors and subcontractors in their supply chain, and software suppliers. Figure 1–1 illustrates a few potential paths that software can take.

---

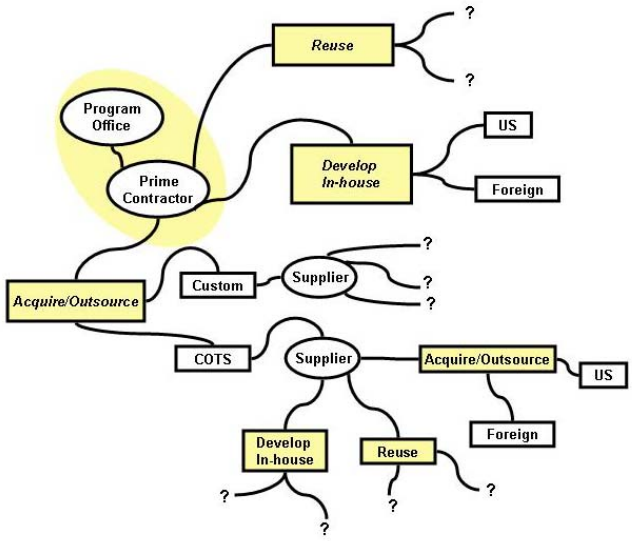
6 See the U.S. President's Information Technology Advisory Committee, *Cyber Security: A Crisis of Prioritization*, February 2005, available at <http://www.nitrd.gov/pitac/reports/index.html>.

7 CIO Executive Council News Bureau.

8 Then Deputy Director for SwA, Information Assurance Directorate, Office of Assistant Secretary of Defense (Networks and Information Integration), Joe Jarzombek led the DOD SwA initiative.

A broad range of stakeholders now needs justifiable confidence that the software that enables their core business operations can be trusted to function as expected (even with attempted exploitation) and can contribute to more resilient operations.[10] Therefore, the responsibility for SwA must be shared by acquirers[11] in the software supply chain. To that end, acquirers involved in purchasing software products or services have a responsibility to factor in SwA to minimize software risks. However, there is growing concern that acquirers are not aware of this responsibility and are inadequately prepared to support SwA in the acquisition process.[12] The SwA Acquisition Working Group was asked to create a document that would provide information to acquirers on how to include SwA considerations in the acquisition process.

Figure 1–1. Potential Software Supply Chain Paths[9]



1.2 Purpose and Scope

The ***purpose*** of this document is to provide information on how to incorporate SwA considerations in key decisions when acquiring software products and services by contract. The bottom line is to "build security in" and incorporate SwA considerations throughout the software acquisition process. This document may also be used as a foundation for training and education.

Figure 1-2 depicts the scope of this document. The scope of the document addresses SwA considerations when acquiring software products and services by contract (also called the acquisition process). This document is written from an acquisition process perspective (activities leading to the award and monitoring of contracts) versus the software development lifecycle process perspective (technical activities involving requirements analysis, construction of the software solution, testing, etc.). As noted in Figure 1-2, these processes interact during the life of a contract because technical activities are normally addressed in a contract work statement.

In addition, as noted in Figure 1-2, the document addresses the SwA perspective versus a system assurance perspective, although, at times, SwA considerations may overlap with system assurance considerations. For a system assurance perspective, refer to the National Defense Industrial Association [NDIA], System Assurance Committee efforts. This document is NOT an exhaustive coverage of SwA considerations when acquiring software products and services by contract.

---

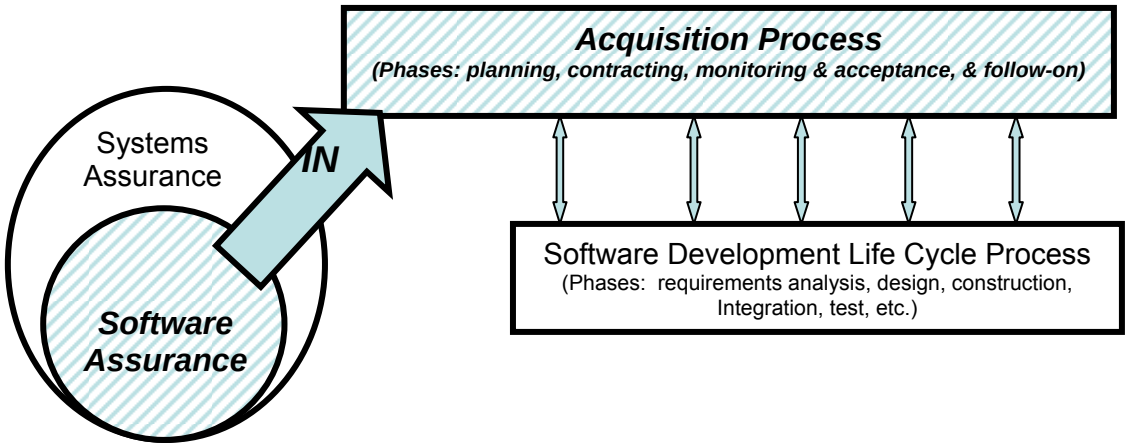
9 Modified version of [DACS-Walker] Walker, E. "Software Development Security: A Risk Management Perspective," in *The DOD Software Tech News—Secure Software Engineering* 8, no. 2 (Rome, NY: Data and Analysis Center for Software, July 2005).

10 Ability to rapidly recover and resume normal operations.

11 One of the recommendations in a report (CNSS–145–06 dated November 2006) by the Global Information Technology Working Group, Committee on National Security Systems, is to: "Clarify acquisition authorities and the enunciation of strategies that permit federal contracting officers to refuse to award contracts to suppliers that are identified as national security risks and/or do not use established security standards and best practices."

12 OMB has specified "Software Acquisition Management" as a Core Competency for the Federal IT workforce.

Figure 1–2. Scope



1.3 Audience—Acquirers

This document is for anyone, both government and private sector, involved in acquiring software products or services by contract[13], including work that is outsourced or sub-contracted. The generic term the "acquirers" is used throughout this document to mean members of the acquisition team. Members of the acquisition team perform a wide variety of functions including:

- developing requirements, plans, and strategies for contract (s)
- developing and issuing Requests for Proposals (RFPs)
- evaluating proposals,
- negotiating and awarding contracts
- monitoring (administrative and technical) contract performance
- accepting delivery of the software product or service

Members of the team may hold positions such as

- developers of requirements (may be systems/software engineers)
- contracting officer representatives (CORs) and contracting officer technical representatives (COTRs)
- contracting officers and specialists
- procurement personnel
- program/project managers
- supervisors of the above

[13] Although this document is primarily written for the acquisition of software products and services by contract, the user may adapt the suggestions in this document for the acquisition of software products and services by other means (e.g., acquisition of open source software that is not by contract, etc.).

Lastly, although this document is for acquirers, it may also be used by the supplier team (e.g., prime contractors, integrators, and subcontractors in the supply chain) of software products or services to facilitate their understanding of SwA requirements that acquirers may request.

1.4 Document Structure

This document is organized around major phases of a generic acquisition process. Figure 1–3 depicts the relationship of these phases to those of several other processes. The major phases are:

- **Planning Phase**. Section 2 covers the planning phase. This phase begins with (1) needs determination for acquiring software services or products, identifying potential alternative software approaches, and identifying risks associated with those alternatives. This set of activities is followed by (2) developing software requirements to be included in work statements; (3) creating an acquisition strategy and/or plan that includes identifying risks associated with various software acquisition strategies; and (4) developing evaluation criteria and an evaluation plan. SwA considerations are discussed for each of the major activities. In the last part of this section (2.5), the development and use of SwA due diligence questionnaires are discussed.
- **Contracting Phase.** Section 3 covers the contracting phase. This phase includes three major activities: (1) creating/issuing the solicitation or RFP with a work statement, instructions to offerors, terms and conditions (including conditions for acceptance), prequalification considerations, and certifications; (2) evaluating supplier proposals submitted in response to the solicitation or RFP; (3) and finalizing contract negotiation to include changes in terms and conditions and awarding the contract. Software risks are addressed and mitigated through terms and conditions, certifications, evaluation factors for award, and risk mitigation requirements in the work statement. The assurance case is introduced in section 3.
- **Monitoring and Acceptance Phase.** Section 4 covers the monitoring (also called administration) and acceptance phase. This phase involves monitoring the supplier's work and accepting the final service or product delivered under a contract. This phase includes three major activities: (1) establishing and consenting to the contract work schedule; (2) implementing change (or configuration) control procedures; and (3) reviewing and accepting software deliverables. During the monitoring and acceptance phase, software risk management and assurance case deliverables must be evaluated to determine compliance in accepted risk mitigation strategies as stated in the requirements of the contract. Assurance case management is also covered in section 4.
- **Follow-on.** Section 5 covers the follow-on phase. This phase involves maintaining the software (the process is often called sustainment). This phase includes two major activities: (1) sustainment (includes risk management, assurance case management, and change management) and (2) disposal or decommissioning. During the follow-on phase, software risks must be managed through continued analysis of the assurance case and should be adjusted to mitigate changing risks.

Figure 1–3. Notional Comparison of Acquisition Processes

<table>
<tr><th></th><th colspan="2">Planning</th><th colspan="2">Contracting</th><th colspan="3">Monitoring
& Acceptance</th><th>Follow-on</th></tr>
<tr><td>IEEE 1062
1998</td><td colspan="2">Planning</td><td colspan="2">Contracting</td><td colspan="3">Product
Implementation
Product
Acceptance</td><td>Follow-on</td></tr>
<tr><td>PMBOK
3.0</td><td colspan="4">Initiating</td><td colspan="2">1. Planning
3. Monitoring & Controlling
2. Executing</td><td>Closing</td><td></td></tr>
<tr><td>NIST SP
800-64
Rev. 1
2004</td><td>Mission & Business Planning</td><td>Acquisition Planning</td><td colspan="2">Acquisition</td><td colspan="2">Contract
Performance</td><td>Contract
Closeout</td><td>Follow-on
Contracts
&
Disposal</td></tr>
<tr><td>DoD
Instruction
5000.2
2003</td><td colspan="2">Pre-Systems
Acquisition</td><td colspan="5">Systems Acquisition</td><td>Sustainment</td></tr>
<tr><td>ISO/IEC
12207
2008(E)</td><td colspan="2">Acquisition
Preparation</td><td>Acquisition
Advertisement</td><td>Supplier
Selection
& Contract
Agreement</td><td>Agreement
Monitoring</td><td colspan="2">Acquirer
Acceptance
& Closure</td><td></td></tr>
</table>

The document also contains several appendices with supporting material. Appendices A and B contain an acronym list and glossary, respectively. Appendix C contains an imperative for implementing SwA in acquisition. Appendix D contains sample software due diligence questionnaires. Appendix E contains other examples of due diligence questionnaires. Appendix F contains sample language for the RFP and/or contract. Appendix G lists useful references to gain a better understanding of SwA.

PAGE INTENTIALLY LEFT BLANK

2 Planning Phase

The planning phase includes four major activities. It begins with (1) a needs determination for acquiring software services or products, identifying potential alternative software approaches, and identifying risks associated with those alternatives. This set of activities is followed by (2) developing software requirements to be included in work statements; (3) creating an acquisition strategy and/or plan that includes identifying risks associated with various software acquisition strategies; and (4) developing evaluation criteria and an evaluation plan. SwA considerations are discussed for each of the major activities. In the last part of this section, the development and use of SwA due diligence questionnaires are discussed.

2.1 Needs Determination, Initial Risk Assessment, and Solution Alternatives

2.1.1 Needs Determination

During the needs determination process, an organization assesses its mission to determine if there are problems in mission performance that could be solved by a software solution. This is followed by an assessment of alternative software-based solutions. Determining the need to acquire software products or services (including software-intensive systems) is the first step in laying the groundwork for full development of software requirements, including SwA requirements.

2.1.2 Initial Risk Assessment

The National Institute of Standards and Technology's (NIST'S) Special Publication 800–30, *Risk Management Guide for Information Technology Systems*, states the following [NIST SP 800-30]:

- Risk is the net negative impact of the exercise of a vulnerability considering both the probability and the impact of occurrence. Risk management is the process of identifying risk, assessing risk, and taking steps to reduce risk to an acceptable level.
- Risk management is the process that allows information technology (IT) managers to balance the operational and economic costs of protective measures and achieve gains in mission capability by protecting the IT systems and data that support their organizations' missions.
- Risk assessment is the first process in the risk management methodology.
- Risk assessment (synonymous with risk analysis) is the process of identifying the risks to system security and determining the probability of occurrence, the resulting impact, and additional safeguards that would mitigate this impact.

An initial risk assessment helps determine the security category, baseline security controls and assurance case[14] required for the acquired software. The acquirer should ask and have answered (by the software application owner) the following questions [Allen]:

- What is the value we need to protect?

[14] Assurance case is addressed in section 4 of this document.

- To sustain this value, what software and information assets need to be protected? Why do they need to be protected? What happens if they are not protected?
- What is the impact if the software behaves unpredictably? What is the *potential impact* on organizations or individuals should there be a breach of security (that is, a loss of confidentiality, integrity, or availability)?
- What potential adverse conditions and consequences need to be prevented and managed? At what cost? How much disruption can we stand before we take action?
- How is residual risk (the risk remaining after mitigation actions are taken) determined and effectively managed?
- How are the answers to these questions integrated into an effective, implementable, enforceable security strategy and plan?
- How do software security controls work together with their operating environment to control and mitigate risk?

The answer to these questions provides a basis for determining a security category. For the Federal Government, Federal Information Processing Standard Publication (FIPS Pub) 199, as mandated by the Federal Information Security Management Act (FISMA) of 2002, requires that a security category be designated for each system based on a range of risk levels. Department of Defense Instruction (DODI) 8500.2[15] provides security categorization rules for DOD systems.

Security categorization includes an assessment of three security objectives defined in the FISMA of 2002: confidentiality, integrity, and availability. This security category is used in conjunction with vulnerability and threat information in assessing the risk to an organization and for determining the security control baseline for a system. Three examples follow:

> *EXAMPLE 1 is quoted from FIPS Pub 199*: A law enforcement organization managing extremely sensitive investigative information determines that the potential impact from a loss of confidentiality is high, the potential impact from a loss of integrity is moderate, and the potential impact from a loss of availability is moderate. The resulting security category, SC, of this information type is expressed as: **SC** investigative information = {(**confidentiality**, HIGH), (**integrity**, MODERATE), (**availability**, MODERATE)}.
>
> *EXAMPLE 2 is quoted from FIPS Pub 199*: A financial organization managing routine administrative information (not privacy-related information) determines that the potential impact from a loss of confidentiality is low, the potential impact from a loss of integrity is low, and the potential impact from a loss of availability is low. The resulting security category, SC, of this information type is expressed as: **SC** administrative information = {(**confidentiality**, LOW), (**integrity**, LOW), (**availability**, LOW)}.
>
> *EXAMPLE 3 [NOTIONAL] Mission Assurance Category (MAC) and Confidentiality Level*: A system must provide access to sensitive and classified combat support data. There must be uninterrupted service and data availability. The loss of confidentiality and integrity are unacceptable and could include the immediate and sustained loss of mission effectiveness. The resulting MAC and confidentiality level is expressed as: **Confidentiality:** TOP SECRET; **MAC I:** Requires the most stringent of protection measures.

If the software supports a critical infrastructure and key resources, the National Infrastructure Protection Plan's Risk Management Framework applies. The risks determined for the operating environment (system, network)

[15] DOD has three defined mission assurance categories that form the basis for availability and integrity requirements. Confidentiality requirements are based on the security classification of information.

and resulting protections needed to mitigate identified risks may impact the security requirements for the software.

2.1.3 Alternative Software Approaches

When considering alternative software approaches, acquisition officials and application owners should seek to reduce or manage the risks identified in the initial risk assessment. The steps are to:

- evaluate alternatives for treatment of risks (accept, mitigate, avoid, transfer, share with a third party [such as a supplier])
- identify protection strategies (security control objectives and controls) that reduce risks to levels that are within acceptable tolerances. Controls can be deployed to reduce likelihood and impact.
- identify potential tradeoffs between reducing risk, increased costs, and decreased operational effectiveness
- identify approaches for managing residual risks that remain after protection strategies are adopted.

Alternative software approaches may include one or more software types or services. Each software type or service can introduce its own risks. The due diligence questionnaires in appendix D are broken up into software types/services since SwA concerns can vary by type or service. When considering alternatives, be cognizant of organizational policies on their use. For example, some organizations prohibit the use of open source software. The following is a nonexhaustive list of software types/services:

- *Commercial-off-the-shelf (COTS)* is a term for proprietary software products (including software appliances) that are ready-made and available for sale to the general public.

  NOTE: *National Security Agency (NSA) COTS Strategy*. The NSA is currently defining a new process that is intended to increase ability to leverage commercial products and services in national security operational environments. This process has a focus on identifying and providing solutions and leveraging commercial capabilities to satisfy the most pressing IT security issues. It is intended to deliver complete solutions to those problems that include, as appropriate to a given problem, product specifications, a description of how the products comprising the solutions should be interconnected and configured, any required policy statements, information that systems engineers can use to tailor the solution for a specific use, and descriptions that the certification and accreditation community will find useful in fulfilling their roles. This COTS strategy is still in the formative stages and includes pilot activities that will serve as input to a decision regarding the future of the proposed process.

- *Modifiable-off-the-shelf (MOTS)* software is typically a COTS product whose source code can be modified. The product may be customized by the purchaser, by the vendor, or by another party to meet the requirements of the customer. In the military context, MOTS refers to an off-the-shelf product that is developed or customized by a commercial vendor to respond to specific military requirements.
- *Government-off-the-shelf (GOTS)* software is a term for software products that are typically developed by the technical staff of the government agency for which they are created. They are sometimes developed by an external entity, but with funding and specification from the agency.
- *Freeware* is copyrighted software that is available for use free of charge for an unlimited time.
- *Shareware* is a marketing method for commercial software, whereby a trial version is distributed in advance and without payment, as is common for proprietary software. Shareware software is typically

obtained free of charge. Shareware is also known as "try before you buy," demo ware, trialware, and other names. Payment is often required once a set period of time has elapsed after installation.

- *Custom Software* is developed for either a specific organization or function that differs from other already available software. It is generally not targeted to the mass market but is usually created for companies, business entities, and organizations.
- *Mobile Code* describes any software that is mobile, being passed from one system to another. In particular, it is used to describe applets within web browsers based upon Microsoft's ActiveX, Sun's Java, or Netscape's JavaScript technologies.
- *Open-Source Software* is computer software whose source code is available under a copyright license that permits users to study, change, and improve the software, as well as redistribute it in modified or unmodified form.
- *Embedded Software* is part of a larger system and performs some of the requirements of that system (for example, software used in an automobile, traffic control system, or aircraft) and does not provide an interface with the user.
- *Integration Services* usually call for a prime contractor with multiple subcontractors. Each subcontractor provides software products and/or services for part of the software-intensive system. The prime contractor is responsible for integrating the parts into a whole software-intensive system.

2.2 SwA Requirements

The security category provides a basis for SwA requirements. In the Federal Government, the security category facilitates the selection of security controls (requirements) and other assurance requirements. The security controls mandated in Federal regulation[16] are a minimum or baseline and are not exhaustive list to address SwA. Other security and assurance requirements should be identified as required to reduce risk to an acceptable level. The following are examples of areas that should be considered when developing SwA requirements (appendix F provides sample SwA requirements language):

- definitions to provide a common understanding (such as definitions of *confidentiality*, *integrity*, *availability*, *assurance case*, *SwA*, and so forth)
- definitions for a common understanding of software security weaknesses in architecture, design, or implementation that can lead to exploitable vulnerabilities (examples: Absolute Path Traversal, Cross-Site Scripting, PHP File Inclusion, Race Condition, Structured Query Language [SQL] Injection, Unbounded Transfer, Operating System [OS] Common Injection, Format String Vulnerability, Integer Overflow)
- a full explanation of the security category that includes the details for assigning security levels for confidentiality, integrity, and availability and how this relates to the software being acquired
- an assurance case that addresses the SwA requirements and the arguments and evidence needed to prove they are met. This may also include a plan for testing that SwA requirements are met. The [NDIA] and [ISO/IEC 15026] provide details on structure and content of assurance cases for systems and software. See sections 4 and 5 for further explanation on assurance cases.

[16] Security controls mandated by FIPS Publication 200, *Minimum Security Requirements for Federal Information and Information Systems*, describes the minimum security requirements associated with security categorization. The Federal Information Processing Standards, NIST Special Publication 800–53, *Recommended Security Controls for Federal Information* Systems, and DODI 8500.2, *Information Assurance (IA) Implementation*, provide a specific security control for requirements.

- test plan that defines the SwA requirements to be tested.
- SwA acceptance criteria (associated with the assurance case)
- risk management that specifically addresses the mitigation of SwA risks
- consideration for auditing the code by an independent body using methods shown to be effective in locating security functions of interest and the known types of security weaknesses of interest to determine the security posture of the code
- software architecture that includes SwA or other description to provide a structure for the SwA case. The software architecture includes an initial description of software components and connectors and SwA requirements for those components and connectors.
- configuration guidelines for all security configuration options. As an example, Office of Management and Budget (OMB) Memorandum M–07–18, *Ensuring New Acquisitions Include Common Security Configurations*, 1 June 2007, provides recommended language for configuring Window XP and Vista operating systems and FAR Subpart 30.101(d) states: "In acquiring information technology, agencies shall include the appropriate information technology security policies and requirements, including use of common security configurations available from the National Institute of Standards and Technology's website at http://checklists.nist.gov. Agency contracting officers should consult with the requiring official to ensure the appropriate standards are incorporated."
- qualifications and required SwA training of software personnel and identification of key security personnel
- required information relative to foreign ownership, control, or influence and how this information relates to SwA risk management
- organization or agency specific requirements or mandates. Those found in the following are a few examples:
    - OMB Memorandum M–07–18, *Ensuring New Acquisitions Include Common Security Configurations*
    - OMB Memorandum M–04–16, *Software Acquisition*
    - National Security Telecommunications and Information Systems Security Policy No. 11, *National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology Products*.

2.3 Acquisition Strategy and/or Plan

Acquirers may develop an acquisition strategy, acquisition plan, or both. They should refer to their organization's policy on developing strategies and plans. Acquisition strategies and plans precede the actual purchase. These strategies and plans provide a description of roles and responsibilities, a roadmap for completing actions and milestones, and a discussion for including special considerations in the purchase and implementation of products and/or services. Software assurance should be addressed in those strategies or plans. As a Federal Government example, Federal Acquisition Regulation (FAR) 7.105(b) (17) requires that plans discuss how agency information security requirements are to be met. In DOD, the Defense Acquisition Guidebook, Part 2, requires program managers to develop an acquisition information assurance (IA) strategy describing how IA/security requirements are to be incorporated.

How SwA requirements are to be met should be included as part of how the IA/security requirements are met. The following are examples of SwA considerations that acquirers should includes in strategies and plans:

- *SwA Expertise*. Acquirers should require that personnel who possess significant SwA expertise be part of the acquisition process from planning, requirements development, source selection, and contract award through contract administration and project management. This expertise is not only essential in establishing appropriate SwA requirements but also in evaluating potential contractors and ensuring that secure software is delivered. Acquisition strategies and plans should state the SwA expertise required as well as specific statements of involvement. As an example: "This acquisition requires support from a SwA subject matter expert (SME). This individual will develop the SwA requirements, evaluate the SwA aspect of proposals, and monitor the assurance case proving the delivery of SwA requirements that is delivered by the contractor during contract performance."
- *Initial Security Category*. The initial security category should be included in the plan. This step is essential because this category is fundamental in selecting and developing SwA requirements. See section 2.1.2 for further explanation.
- *SwA Requirements*. The acquisition strategy or plan should include statements of critical high-level SwA considerations. These high-level statements help establish the ultimate detailed statement of requirements. Acquirers developing acquisition strategies and plans should rely heavily on the SwA personnel assigned to the acquisition. Two examples follow:

  EXAMPLE 1—COTS Software: To ensure that COTS software is consistent with the overall security requirements of the software-intensive system, SwA personnel assigned to this acquisition will provide requirements to ensure delivery of COTS software that has specified preset security settings. In addition, requirements will mandate that testing of the specified preset software be accomplished on the operating system and platform proposed for production.

  EXAMPLE 2—Software Development or Systems Integration: To manage the development and delivery of SwA requirements, a SwA case shall be developed that presents a convincing argument the software will operate in an acceptably secure manner. To support the SwA case, definitive evidence (for example, process, procedures, test results) shall be produced to present a convincing argument that the software will be acceptably secure throughout its life cycle, including termination. The security stakeholders (for example, accreditors) will evaluate the SwA case in determining that the software will function as expected and be as free as possible of the known types of security weaknesses that can lead to exploitable vulnerabilities.
- *SwA Considerations in Contractor Selection.* High-level statements should be included in acquisition strategies and plans on how SwA will be considered in the selection of contractors. For example, "due-diligence questionnaires will be used to solicit answers from offerors on their SwA practices." The due-diligence questionnaires should be part of the evaluation plan.
- *SwA Considerations in Contract Administration and Project Management.* High-level statements should be included in acquisition strategies and plans on how the SwA requirements will be monitored during contract performance including information on measuring the contractor's performance. For example, SwA personnel will monitor the delivery of SwA requirements through applying quantitative and qualitative measures.
- *Plans for Independent Testing*. Include a statement on the type of testing that will be done. Independent testing of the software product can be used to ensure its construction, safety, and functionality. When relying on existing software testing certifications or attestations, acquirers should be aware that each evaluation is performed at a point prior to the specific acquisition. Therefore, those certifications or attestations may not reflect the quality of the supplier or the acquired version of

the software product.

Acquirers should be aware of the benefits that can be obtained through testing of COTS software against customer, government, or vendor-developed specifications. In general, third-party testing and evaluation provide a significantly greater basis for customer confidence than many other assurance techniques. Yet it is important to note that purchasing an evaluated product simply because it is evaluated and without due consideration of applicable functional and assurance requirements may be neither useful nor cost effective. Acquirers need to consider their overall requirements and select the best software accordingly.

Two government programs are of particular interest here: the National Information Assurance Partnership's (NIAP's) Common Criteria Evaluation and Validation Program and NIST's Cryptographic Module Validation Program (CMVP). The NIAP program focuses on evaluating products against a set of security specifications. The CMVP focuses on security conformance testing of a cryptographic module against FIPS Pub 140–2, *Security Requirements for Cryptographic Modules*, and related Federal cryptographic algorithm standards.

The internationally recognized laboratory accreditation organizations (working under International Standards Organization [ISO]/International Electrotechnical Commission [IEC] 17011) and accrediting Information Technology Testing Laboratories (working under ISO/IEC 17025) maintain a cadre of commercial software testing laboratories. The two major accrediting bodies in the United States are the National Volunteer Laboratory Accreditation Program (NVLAP) and the American Association for Laboratory Accreditation (A2LA). They have mutual recognition agreements providing a global network of competent sources.

2.4 Evaluation Plan and Criteria

2.4.1 Evaluation Plan

This activity involves creating a plan for evaluating proposals submitted in response to a solicitation and the criteria that will be used to evaluate the proposals. The evaluation plan describes the process by which proposals are secured and evaluated.

When acquiring software products and services (to include software-intensive systems), SwA criteria should be included in the solicitation, and the evaluation plan must describe how to evaluate the products and services against the criteria. This includes discussing the timing of the evaluation and any measures that can be used to support the evaluation process. For example, acquirers should require potential suppliers to be evaluated for software quality *prior to* contract negotiations. The results can then be used to help determine the best candidate for the acquisition.

The plan should also include the names of the evaluators. It is imperative that qualified SwA and/or security professional(s) are included to evaluate the SwA criteria.

2.4.2 Evaluation Criteria

SwA depends on people and organization, process maturity, and technology working together to be effective. Therefore, evaluation criteria should be developed to include the following broad categories:

- *People and Organization*. Software developers who are not knowledgeable about SwA cannot recognize when some aspect of their software's architecture, design, or implementation weaknesses may lead to exploitable vulnerabilities; nor, in fact, can they recognize the importance of not introducing design and implementation defects in the first place. The organization may have ill-

defined SwA responsibilities or poorly funded SwA capability. Therefore, SwA related to people and organization should be evaluated to determine that the software developers are SwA savvy and to ensure organizations provide adequate SwA resources.

- *Process Maturity*. Mature SwA processes increase the likelihood of avoiding and discovering unintentional weaknesses that can lead to exploitable vulnerabilities. Lack of mature processes may result in underestimated risks, missed requirements, inadequate testing and reviews, lack of measures, or little or no monitoring. Therefore, processes for SwA should be evaluated to determine their capability of developing and maintaining trustworthy software.
- *Technology*. Testing tools, especially code vulnerability scanners, can be employed to detect known types of software weaknesses that can lead to exploitable vulnerabilities. The technology being used to properly develop and configure software is also key.

Criteria may be qualitative, quantitative, and/or "go/no-go." Qualitative and quantitative criteria are often evaluated using a scorecard method. The evaluation plan should describe how to determine the scores. The "go/no-go" criteria are evaluated based on whether the proposal satisfies the criteria. In the case where a proposal does not meet the criteria, the proposal is normally eliminated from future consideration for awarding a contract.

The SwA due diligence questionnaire provides a means for gathering information to evaluate quantitative, qualitative, and/or "go/no-go" SwA criteria. The questions that prospective suppliers must answer are carefully crafted to suit the particular type of software purchase. Proposal evaluators score the answers against a predetermined acceptable range of responses. The range of responses may carry a rating using a scorecard method (for example, the answer rates a 4 out of a possible 5) or may be accepted based on a "go/no-go" determination (the answer is either acceptable or not).

2.5 SwA Due Diligence Questionnaires

As discussed in section 2.4.2, "Evaluation Criteria," the software assurance due diligence questionnaires can assist acquirers in obtaining additional information about the software and its supplier. In this context, due diligence involves taking all "reasonable steps" necessary to ensure that a software-intensive system not only meets business and technical requirements, but also addresses SwA concerns. The intent is to inform acquirers of potential risks associated with the software they are considering for purchase. The questionnaires support acquirers in implementing due diligence and providing a means to gather, in advance, some of the information needed to make a decision about the assurance of the software. Objectives are to:

- enhance the process for acquiring trustworthy software, thereby enhancing the IT security posture
- "build security in" and maintain SwA throughout the acquisition process
- assist acquirers in assessing the assurance of software before acquisition (or purchase/procurement) from a supplier (or seller or developer).

2.5.1 Using SwA Due Diligence Questionnaires

Whether buying a single copy of a single software application or a multimillion-dollar software-intensive system, the questionnaires are useful tools. They can be used in whole or in part. Some questions may apply, some may not; some may be added, others may be tailored. The questionnaires are a means for gathering relevant information to support decisionmaking versus being a decisionmaking tool.

Expertise in software, acquisition, and IA—as well as common sense—is critical in making smart decisions on acquiring trustworthy software. Questions should be posed by, and responses assessed by, knowledgeable SwA experts or other appropriate functional experts. In addition, when using the questionnaires, acquirers should request evidence or may coordinate an on-site follow-up that reviews objective evidence of the provided answers.

The questionnaires support the exercise of SwA due diligence by acquirers. Questionnaires help to identify potential risks and red flags. ***The questionnaires are tools. They are not checklists or complete listings of all possible software security concerns.*** Several examples when questionnaires may be effectively applied include:

- conducting market research
- developing a request for information to gather information for a major software development program
- developing an RFP for building a critical software-intensive system, including an information system or weapons system platform
    - Some questions can be transformed into SwA requirements that are then included in a work statement.
    - Some questions can be transformed into other contractual language such as terms and conditions.
    - A questionnaire can be incorporated as part of evaluation factors for award (see section 2.4.2).
- developing vendor surveys for trade-off studies
- gathering information on given software products or suppliers to determine which COTS software application to procure under a General Services Administration Schedule or Indefinite Delivery/Indefinite Quantity contract.

Appendix D contains several sample questionnaires tailored by type of software product or service (see section 2.1.3 for a description of software types). It is recommended that they be tailored for the product or service that is being acquired.

2.5.2 SwA Concern Categories

The SwA due diligence questionnaires in appendix D are organized into categories that represent a logical grouping of SwA concerns. The table below relates SwA concerns to the broader people and organization, process maturity, and technology category and includes a risk description and purpose for the data gathered. The identified threats are specific examples and are not intended to be a complete list of all threats.

SwA Concern Categories	People and Organization	Process Maturity	Technology	Risks	Purpose for Questions
Organizational History	√			There may be conflicting circumstances or competing interests within the organization that may lead to increased risk in the software development.	To understand the background, role, and relationships.

SwA Concern Categories	People and Organization	Process Maturity	Technology	Risks	Purpose for Questions
Foreign Interests and Influences	√			There may be controlling foreign interests (among organization officers or from countries) with malicious intent to the United States.	To help identify supplier companies that may have a malicious intent to a U.S. buyer.
Security "Track Record"	√			A software supplier that is unresponsive to known software vulnerabilities may not mitigate/patch vulnerabilities in a timely manner.	To establish insight into whether the supplier places a high priority on security issues and will be responsive to vulnerabilities they will need to mitigate.
Financial History and Status	√			A software supplier that goes out of business will be unable to provide support or mitigate product defects and vulnerabilities.	To identify documented financial conditions or actions of the supplier that may impact its viability and stability, such as mergers, sell-offs, lawsuits, and financial losses. While difficult to predict future viability of a company, this is a risk factor that should be considered.
Individual Malicious Behavior	√			A developer purposely inserts malicious code.	To determine whether the supplier has and enforces policies that minimize individual malicious behavior.
Software Security Training and Awareness	√			Developers unaware of software assurance best practices are likely to implement software susceptible to attack.	To determine whether training of developers in SwA best practices is a supplier policy and practice.
Software History and Licensing		√		The software supplier's development practice in using code of unknown origin may be unable to produce trustworthy software.	To identify specific risks pertaining to the history/pedigree of the software during any and all phases of its life cycle that should have been considered by the supplier. This point addresses supply chain concerns.
Development Process Management		√		If supplier project management does not perceive the value of SwA and enforce best practices, they will not be consistently implemented.	To determine whether project management enforces software assurance–related best practices.

SwA Concern Categories	People and Organization	Process Maturity	Technology	Risks	Purpose for Questions
Software Development Facility		√	√	An example is personnel inappropriately accessing or changing configuration items in the development environment.	To ascertain that the supplier has and enforces policies that support software development environments that minimize risk exposures.
Concept and Planning		√	√	If nonfunctional requirements (security, quality, safety) are not specified, developers will not implement them.	To determine whether the supplier's requirements analysis process explicitly addresses SwA requirements.
Design		√		The software may be designed without considering security or minimization of exploitable defects.	To determine how security is considered during the design phase.
Software Development	√	√	√	Software development activities are sources of a significant number of vulnerabilities.	To determine whether the supplier has and enforces good SwA practices in the development of software.
Component Assembly		√		Insufficient analysis of software components used to assemble larger software packages may introduce vulnerabilities to the overall package.	To ensure that the software components are thoroughly vetted for their security properties, secure behaviors, and known types of weaknesses that can lead to exploitable vulnerabilities.
Testing (supply-side)		√	√	Software released with insufficient testing may contain an unacceptable number of defects.	To determine whether the appropriate set of analyses, reviews, and tests are performed on the software throughout the life cycle.
Installation and Acceptance		√	√	The software may not install as advertised and the acquirer may not get the software to function as expected.	To ensure the supplier provides an acceptable level of support during the installation process.

SwA Concern Categories	People and Organization	Process Maturity	Technology	Risks	Purpose for Questions
Software Change Management		√	√	Weak change control procedures can corrupt software and introduce new security vulnerabilities.	To determine whether software changes are adequately assessed and verified by supplier management.
Built-in Software Defenses			√	The software may lack preventive measures to help it resist attack effectively and proactively.	To ensure that capabilities are designed to minimize the exposure of the software's vulnerabilities to external threats and to keep the software in a secure state regardless of the input and parameters it receives from its users or environment.
Assurance Claims and Evidence		√		Supplier assurance claims may be insufficiently verified.	To determine how suppliers communicate their claims of assurance, ascertain what the claims have been measured against, and identify at what levels they will be verified.
Software Manufacture and Packaging			√	Vulnerabilities or even malicious software can be introduced in the manufacturing or packaging process.	To determine how the software goes through the manufacturing process, how it is packaged, and how it remains secure.
Support		√		Supplier ceases to supply patches and new releases prior to the acquirer ending use of software. Vulnerabilities may go unmitigated.	To ensure understanding of supplier policy for security fixes and when products are no longer supported.
Operating Environment for Services		√	√	Operating environment for the services may not be hardened or otherwise secure.	To understand the controls the supplier has established to operate the software securely.
Security Monitoring		√	√	Insufficient security monitoring may allow attacks to impact services.	To ensure software and its operating environment are regularly reviewed for compliance in SwA requirements through periodic testing and evaluation.

SwA Concern Categories	People and Organization	Process Maturity	Technology	Risks	Purpose for Questions
Timeliness of Vulnerability Mitigation		√		Sometimes it can be extremely difficult to make a software supplier take notice and repair software to mitigate reported vulnerabilities.	To ensure security defects and configuration errors are fixed properly and in a timely fashion.
Service Confidentiality Policies		√	√	Without policies to enforce client data confidentiality/privacy, acquirer's data could be at risk without service supplier liability.	To determine the service provider's confidentiality and privacy policies and ensure their enforcement.

The SwA concern categories are explained further below. Some of these concerns go beyond the acquisition process and into the software development life cycle. Acquirers should use these explanations to help develop questions and interpret responses.

Organizational History. Asking for background information on the software development organization can help acquirers understand how well integrated it is with the overall supplier organization. For example, if the development organization has been recently acquired, it may not follow the same standard security/secure development practices as the larger (acquiring) entity.

Foreign Interests and Influences. The ability of a company to build trustworthy software starts at the top—the chairman of the board, the board, the president, and the "three-letters" (the CXOs). In some cases, there could be malicious intent against U.S. interests. It is helpful to know whether there are any controlling foreign interests among the officers or "Cs" of the company (for example, from what country and to what extent control is exercised). For a publicly held company, limited information of this nature is available. If foreign ownership, control, or influence (FOCI) is an area of particular interest, the questions in OMB Standard Form 328[17] should be used.

Security "Track Record." To increase confidence in the security of software, acquirers can ask about the performance history or "track record" of a software supplier. Though past performance is not a guarantee of future performance, the established record of a supplier can provide factual information concerning the supplier's financial stability, innovation, support services, and reliability. It can also provide insight into whether the supplier places a high priority on security throughout its company and throughout the life cycle of its products.

Financial History and Status. The questions in this category relate to the supplier's financial viability and stability. If the company is publicly held, some of the answers are a matter of public record. The intent of the questions is to identify documented financial conditions or actions of the supplier that may impact its viability and stability, such as mergers, sell-offs, lawsuits, and financial losses. It is a best practice to run a financial report (for example, three brokerage reports or a Dun and Bradstreet report) to gain better insight into the company's financial standing. Such conditions or actions are identified to determine whether and what risk they may pose to the supplier's software development security environment or to the supplier's ability to provide security support services to its customers.

[17] See <http://www.dtic.mil/whs/directives/infomgt/forms/eforms/sf0328.pdf>.

Individual Malicious Behavior. Individual malicious behavior cannot be prevented or easily detected. However, questions can be asked to determine if the supplier has thought about the threat and implemented reasonable practices and deterrents. There are multiple controls that can reduce but not eliminate the risk of individual malicious behavior. Access controls and configuration management are two controls. Design and code reviews provide some deterrence. It is harder to identify malicious behavior with outsourced development. However, suppliers who outsource all or part of software development should have policies in place and enforcement mechanisms available that minimize the impact of malicious behavior of their outsourced developer.

Software Security Training and Awareness. Reviewing training policies that a supplier has in place provides a better understanding of the company's ability and desire to develop secure code. A large percentage of security defects in software can be avoided if development managers and programmers are aware of the common weaknesses that can lead to exploitable vulnerabilities and consciously work at bypassing them. Acquirers should seek software suppliers that educate their development teams in software security best practices. Although internalizing the importance of secure software principles will not solve all security problems related to software, it does instill in developers the discipline of thinking through possible errors, of checking for corrupt inputs and parameters, and of programming defensively to handle unexpected problems. The result is developing programs that are better thought out, better structured, and more secure. Defensive programming teaches developers how to look for hidden assumptions in their programs and how to defend against attempts to exploit those assumptions to cause their programs to behave in unexpected or undesirable ways. Software organizations should ensure that their developers obtain and maintain their security knowledge and reward the application of that knowledge by consistently using secure development practices, producing secure software, and remediating the inadequate practices of developers who consistently produce insecure software.

Software History and Licensimg. The primary objective of this category is to identify specific risks pertaining to the background/lineage of the software during any and all phases of its life cycle. This includes such considerations as how the version of the software under consideration at a given point in time was originally conceived and implemented, and by whom. While the software's pedigree is extended, and thus changed each time the software is modified in some way by its developer (at any given point in time), the software as it exists in that point in time, can be said to have a fixed pedigree. The challenge has been how to attest to the quality and security of software that may contain millions of lines of code whose origin may come from many different sources other than the supplier. Not knowing the origin of code increases the risk of malicious or poor quality code.

Development Process Management. Development process management includes responsibility for software assurance planning, software risk management, assurance case management, and acceptance of the software product or service. Acquirers should rely on software assurance experts to ensure that requirements are implemented appropriately. The purpose of the questions is to determine whether the supplier's project management includes these activities and whether appropriate SwA expertise is on hand for assistance.

Software Development Facility. The questions on the software development facility establish that the supplier has and enforces policies that support a secure development facility. The policies that support a secure development facility depend on a number of factors that include one or multiple physically separated sites; use of outsourced development; and incorporation of open source development. At a minimum, policies should be in place to enforce access controls, audit software to prevent unauthorized access to software, and ascertain whether unauthorized access has occurred. In addition, policies should be clear on the actions that are taken when access controls have been violated and unauthorized access has occurred. The access controls should support separation of roles so that a developer cannot bypass the controls. The check-in of critical software might require the concurrence of two or more individuals. Often, the access and auditing are enforced by a configuration management (CM) system.

Concept and Planning. The suppliers should use this development phase to consider how security/quality is integrated into the development process, identify key assurance objectives, and otherwise maximize software assurance while minimizing disruption to plans and schedules. As part of this process, the team should consider how the security features and assurance measures of its software will integrate with other software likely to be used with it.

Architecture and Design. Attention to security issues in the architecture and design of software can decrease the risk of software (based on that design) vulnerable to attack. If the supplier leverages general principles for secure design, it decreases the probability that exploitable software will exist. Some of the security-related activities in this phase include threat modeling, use of security design patterns, common attack pattern analysis, use of formal methods to verify design (for high assurance software), design reviews, architectural risk assessments, and assurance case inputs in the architecture and design phase [Goertzel, 2007].

Software Development. The questions help determine whether the supplier has and enforces good SwA practices in developing software and whether the development process mitigates the introduction of software weaknesses that can lead to exploitable vulnerabilities. The answers to these questions depend on what is being developed: a general-purpose product, computing infrastructure services, a closed system, or a system or application that must be integrated into an existing collection of systems. Software development policies should reflect differences in scale and scope among these kinds of development contexts. In addition, software development processes should incorporate best practices in the development of trustworthy software. Software development activities can be sources of a significant number of vulnerabilities. While a developer may claim that its software development process incorporates security, critical questions involve the choice of development practices, whether the developer effectively and consistently uses those practices, whether the software risks that are essential for the proposed usage of the software are addressed by the developer's software process, and whether the people involved in the software development life cycle are trained in good software security practices, such as reviewing their architecture, design, and implementation decisions in light of the known types of software weaknesses that can lead to exploitable vulnerabilities.

Component Assembly. The composition of software-intensive systems (at least partially) from existing components presents challenges to secure software architecture design. A reused software component may be exposed to inputs with which it has not been previously tested. Thus, it may introduce vulnerabilities into the new system [Goertzel, 2007]. For secure integration/assembly of acquired or reused software components to be possible, the components selected must be thoroughly vetted for their security properties, secure behaviors, and known types of software weaknesses that can lead to exploitable vulnerabilities. In addition, software/applications are executed and supported by an operating system that provides the interfaces to the network, other applications, or data. It is important that the operating system be hardened to minimize the exploitable vulnerabilities. Changes to operating system configurations may degrade the security of applications that rely on the operating system. Generally, management should implement an operating system change control process similar to the one used for application changes. In addition, management should review application systems following operating system changes to protect against a potential compromise of security or operational integrity.

Testing (supply-side). Testing questions provide insight into the types and level of SwA testing. Quality control mechanisms must also carry through to the supplier's software testing practices. The existence of a well-defined version control and bug tracking system, for example, can suggest a well-contained and organized development process, resulting in a well-contained and organized end product. Depending on the variety and type of internal functional tests performed, a certain degree of quality can logically be assumed. For example, performing software component–level tests versus fully integrated functional system tests allows more granularity in assessing software correctness and quality. In addition, the existence of Total Quality Management (TQM) byproducts, such as decision trees, may indicate how the supplier manages and orchestrates design changes from one organizational tier to another. Another consideration in this phase is the

notion of "consistency of design," whereby software is verified to operate clearly within the bounds of its underlying requirements and design specifications. Somewhat akin to "transparency of function," consistency of design can also be assessed by comparing design artifacts from the previous phase against one another to ensure that design objectives and requirements are properly promulgated and consistently represented across these artifacts.

Installation/Acceptance. These questions help determine whether the supplier continues to address SwA during installation/acceptance. Continuing the end-to-end TQM program, software quality should also be maintained throughout the installation and acceptance phases. This may also require an analysis of entities affiliated with the shipping and/or distribution of the software, providing a middleman between the supplier and the integrator. The supplier should also be committed to providing an acceptable level of support during the installation process. Specific measures such as installation and configuration times may be relevant in this phase. Finally, by evaluating the extent, quality, and depth of supplier-provided training and certification programs, acquirers can also determine the supplier's commitment to acceptance once the software is acquired.

Software Change Management. Acquirers should ask whether there is a change management procedure or document that identifies the type, security impact, and extent of changes conducted on the software throughout the life cycle. The primary objective is to determine whether the changes were adequately assessed by management. This should include such considerations as how the software is licensed, how it is installed and configured in its execution environment, and how it is modified through patching and updating, and by whom. In addition, this includes changes in responsibility for the ongoing development such as, responsibility shifts from the software's original developer to an integrator or a new development organization (as when one software firm buys another). Questions should also be asked on whether a baseline for applications/software is established and if there is a procedure for baseline modification, how version control is organized, what kind of testing procedures should be followed based on a change, and what is the procedure for approving a software change (including current software modification/update, implementation of additional software, replacement the old software, and security patching implementation).

Built-in Software Defenses. These questions provide insight into the self-defense capabilities built into the software. A minimum set of self-defense capabilities is desirable in all software but is absolutely requisite in software that is likely to be used in high-consequence applications (that is, when an application failure would threaten human life, safety, health, freedom, or financial well-being, or when compensating users from loss would be impossible or extremely expensive). These capabilities are designed to minimize the exposure of the software's vulnerabilities to external threats and to keep the software in a secure state regardless of the input and parameters it receives from its users or environment. Software self-defenses include such capabilities as:

- validation to detect and filter out or reject all inputs or parameters that are incorrect or malformed
- recognition and defense against attempts to exploit incorrect and hidden assumptions to force the software's components to behave insecurely
- ability to handle unexpected, unlikely, and even presumably "impossible" events in ways that do not leave the software, the data it "touches," or its environment vulnerable to compromise or subversion
- ability to isolate and contain the damage resulting from a successful attack so that the damage does not affect other parts of the software, the data it "touches," or its execution environment
- ability to recover quickly after a failure, either through its own built-in fault-tolerance capabilities or by being designed to rapidly notify and support recovery actions by the administrator.

Assurance Claims and Evidence. Suppliers should be able to describe the assurance case for their software and explain how claims can be validated. Acquirers should ask the supplier to identify the types and extent of measurements and assessments conducted on the software. This query helps determine whether weaknesses

and flaws are adequately assessed and measure the levels of verification activities imposed on the software. This can include a complete range of validation/verifications—internal to external, penetration testing to regression testing to a complete certification and accreditation in accordance with the applicable authority (for example, NIST, DOD, and Director of Central Intelligence Directive 6/3). It is not possible to "test security in" to a product, and assurance is not merely the absence of defects. However, the supplier should be able to provide evidence to back up its assurance claims.

Software Manufacture and Packaging. Vulnerabilities can be introduced even in the manufacturing and packaging processes for the software. These questions help determine how controls are in place to mitigate those risks.

Support. It is normal within the industry to provide support for n-1 to n-2 versions of software. If acquirers know that they will not keep up with the supplier's release schedule, they need to determine whether there is a point at which the nonupdated version of the software will no longer be supported by that supplier; some form of source revision control is necessary to manage security bug fixes to avoid regression errors. The risks associated with using unsupported software should be weighed against the risks of adopting a new version of the software or replacing it with an alternative product. The supplier's support policy for security fixes should also clearly communicate which versions are supported and when products are no longer supported. The supplier's willingness to support older versions for a fee may be something worth negotiating during acquisition.

Operating Environment for Services. If a supplier is offering to provide software as a service (Application Service Provider—ASP), instead of a standalone software package, acquirers should consider the governance of these services. *Governance* refers to the programs and processes that an organization puts in place to ensure that things are done right, meaning in accordance with best practices, architectural principles, Federal regulations, and other determining factors. Good governance of services is required to ensure that the acquirer's sensitive data is adequately protected and available when needed. Control structures should not only ensure the appropriate operation of access controls, but they should also carry out security obligations such as audit, monitoring, and alerting. The types of threats listed below are associated with vulnerable service acquisitions:

- failure of available service due to (a) failure of a component, (b) denial of service attack, (c) inadequate capacity or performance, and (d) attacks against infrastructure
- failure of integrity due to (a) unauthorized or uncontrolled modification of content of a transaction, (b) unauthorized or uncontrolled modification of system or application software, (c) unauthorized or uncontrolled access to Web site content, and (d) nonmalicious error
- failure to preserve confidentiality of sensitive customer information due to (a) unauthorized or uncontrolled access to confidential data, (b) unauthorized or uncontrolled modification of software, and (c) nonmalicious error.

Security Monitoring. Questions about security monitoring provide insight into whether the software and its operating environment are regularly reviewed through periodic testing and evaluation. This continuous monitoring helps ensure that controls continue to be effective and that they have not unintentionally been relaxed when making patches and upgrades. Patches and upgrades make direct changes to the software and potentially to the configuration of the operating system where they are applied. The changes may degrade performance, introduce new vulnerabilities, or reintroduce old vulnerabilities. To understand the patch risks, the patch process should be examined in some detail. Many companies apply patches to test systems before releasing them throughout the enterprise. However, as the scale of the patch management problem grows, this becomes problematic. Unless the software package comes with a validation test suite, it is difficult to determine whether the software is operating properly after a patch. In addition, most validation suites do not

test software to their failure points. Instead, they perform simple operational tests to determine whether normal operations work in simple conditions. In addition to the patches themselves, hardware, drivers, and detailed configuration settings for the operating system and each of the applications on the system impact the proper functioning of patches.

Timeliness of Vulnerability Mitigation. A large number of skilled attackers are discovering vulnerabilities at a significant rate. Software suppliers with a good record of security fixes often gain early insight into security vulnerabilities, such as Zero-Day exploits that are included on message boards and blogs. Software support should incorporate a process to update and patch software to mitigate newly discovered vulnerabilities.

Service Confidentiality Policies. Suppliers that offer software through services should provide their policies on how customer data will be protected.

3 Contracting Phase

The contracting phase includes three major activities: (1) creating/issuing the solicitation or RFPs with a work statement, instructions to offerors, terms and conditions (including conditions for acceptance), prequalification considerations, and certifications; (2) evaluating proposals submitted in response to the solicitation or RFP; and (3) finalizing contract negotiation to include changes in terms and conditions and awarding the contract. Software risks are addressed and mitigated through terms and conditions, certifications, evaluation factors for award, and risk mitigation requirements in the work statement.

3.1 Request for Proposals

3.1.1 Work Statement

Acquirers usually prepare the work statement. The FAR states that "agencies shall include the appropriate information technology security policies and requirements in all acquisitions for information technology" (FAR Subpart 39.101(d)). See section 2.2, SwA Requirements, for sources of requirements to be included in work statements. Acquirers should consider including the following software assurance requirements in a work statement:

- definitions related to trustworthy software that provides a common understanding
- description of the security category [see FIPS Pub 199 and DODI 8500.2] that provides a common framework and understanding of security needs
- an assurance plan that addresses the development and maintenance of an assurance case for software
- an assurance case that addresses the necessary security requirements (functions and properties) and the arguments and evidence needed to prove the requirements are met. The purpose of an assurance case is to provide convincing justification to stakeholders that critical SwA requirements are met in a system's expected environment(s). An assurance case is the set of claims of critical SwA properties, arguments that justify the claims (including assumptions and context), and evidence supporting the arguments [NDIA]. The [NDIA] and [ISO/IEC 15026] provide details on structure and content of assurance cases for systems and software. Also see Section 4.3.2 for some additional language that may be included in the work statement.
- software assurance risk management that includes a formal program for managing safety and security risks associated with the implementation of software
- consideration for auditing the code for the desired security functionality and known types of weaknesses that can lead to exploitable vulnerabilities by an independent body to determine the security posture of the code
- software description that includes a software architecture and other descriptions as needed to provide a structure for the assurance case. The software architecture includes an initial description of the software components and connector, including software security–related aspects.
- a security test plan that defines the approach for testing each of the SwA requirements
- suggestions for configuring all security configuration options
- patch and upgrade processes that ensure security requirements continue to be met.

3.1.2 Terms and Conditions

Additional SwA requirements may be included in terms and conditions. Some items described above for inclusion in the work statement may be more appropriate as terms and conditions. Whether to include an item in the work statement or as a term or condition depends on the policies and structure of the acquisition organization. Selecting terms and conditions would depend on the type of software to be acquired. Because prime suppliers often subcontract software services, terms and conditions should be worded in such a way to ensure that they flow down to all levels of subcontracts. Terms and conditions include but are not limited to:

- legal responsibilities of supplier and acquirer relative to SwA
- quality of software development processes
- SwA acceptance criteria
- qualifications and training of software personnel and identification of key security personnel
- SwA training program
- quantitative and qualitative measures that articulate expectations about the expected level of service and performance
- required information relative to FOCI
- required preset security features (this is particularly relevant to COTS software)
- penalty clauses for failed SwA.

3.1.3 Instructions to Suppliers

In response to an RFP, suppliers must submit information that provides objective evidence of their ability to perform the SwA aspects of the work statement and terms and conditions. Clear instructions must be included in the RFP on what suppliers must submit for evaluation, including instructions pertaining to onsite evaluation, if required by the RFP. Instructions to suppliers explain how to answer the due diligence questionnaire and what to submit in an initial assurance case and software description.

3.1.4 Certifications

Certifications may also be a way to provide assertions of software trustworthiness when information may be too costly to compile or too voluminous for proposal evaluation. Certifications provide assertions by offerors of existing conditions or compliance in certain requirements. Using certifications shifts the burden of compliance to the suppliers.

3.1.5 Prequalification

Acquirers may want to consider prequalification. Prequalification can be done to evaluate organizational capabilities or other technical management capabilities. As a word of caution, there should always be additional evaluation for the unique SwA requirements of each acquisition.

3.2 Proposal Evaluation

Acquirers should ensure that SwA SMEs are used to evaluate each proposal to determine the level of understanding of the SwA requirements. This includes an evaluation of the evidence provided to support answers to the due diligence questionnaire.

Proposals have multiple components that should be weighed separately and then combined to provide an overall score. An example of three components may be management, technical (includes SwA), and price. All three should have weighted criteria to result in a numerical score.

3.3 Contract Negotiation and Contract Award

The evaluation results in the selection of the best proposals for contract negotiation. During negotiations, the acquirers and suppliers negotiate on requirements, terms, and conditions. It is important that the give-and-take on SwA requirements, terms, and conditions does not compromise the ultimate assurance goals or critical assurance goals. Suppliers may push back on the SwA requirements because they may not be fully competent to do the job or be willing to take the risk. Acquirers may find that suppliers may overbid because of perceived risk and doing something they have never done. Acquirers should consider share-in-savings arrangements (savings as a result of implementing SwA requirements as stated). The sharing includes not only costs and benefits but also the willingness to afford the supplier more time to engage in the education and training that is needed. An alternative would be to consider a contract type that shifts the burden of some of the risk to the acquirer and/or provide additional cost or performance incentives [see FAR Subpart 16.1 and FAR Subpart 16.3 for incentive contracts].

When awarding the contract, acquirers must ensure that all SwA agreements made during negotiation are incorporated into the contract when it is awarded. Negotiated agreements are sometimes overlooked when drafting the final contract award.

PAGE INTENTIONALLY LEFT BLANK

4 Monitoring and Acceptance Phase

The monitoring and acceptance phase (may also be called contract administration phase) involves monitoring of the supplier's work and accepting the final service or product. This phase includes three major activities: (1) establishing and consenting to the contract work schedule, (2) implementing change (or configuration) control procedures, and (3) reviewing and accepting software deliverables. During the monitoring and acceptance phase, software risk management and assurance case deliverables must be evaluated to determine compliance in accepted risk mitigation strategies as stated in the requirements of the contract.

4.1 Contract Work Schedule

The contract work schedule should include very specific scheduled work for delivering SwA requirements. If a work breakdown structure (WBS) is used, acquirers should ensure that SwA deliverables are identified in it. See section 4.3.2 for a discussion on assurance case issues addressed in the WBS.

4.2 Change Control

The change control procedures for a software-intensive system should ensure that SwA requirements are not compromised when changes are requested. Each change control request should include a specific section that addresses the impact of the requested change on SwA requirements. Change or configuration control of SwA requirements is managed as part of assurance case management (see section 4.3.2).

4.3 Reviewing and Accepting Software Deliverables

During this activity, examples of deliverables are the risk management plan for software, assurance case, and test documentation. Acceptance criteria should be explicit, measurable, and included in the assurance case or in the terms and conditions. See appendix F for sample terms and conditions that include acceptance conditions. The SwA SMEs should review each software deliverable and analyze test results produced by the contractor or independent tester to ensure that SwA requirements are met. Acquirers should not accept the service or product until the SwA expert finds the requirements acceptable.

4.3.1 Risk Management

The FAR states: "Contracting and program office officials are jointly responsible for assessing, monitoring and controlling risk . . . during program implementation. . . . Appropriate techniques should be applied to manage and mitigate risks during the acquisition of information technology."

An initial risk assessment (see section 2.1.2) was performed during the acquisition planning process. This risk assessment resulted in the identification of a security category, which may be further refined during this phase of the acquisition process. Acquirers and suppliers who are responsible for implementation should create a plan for managing risks associated with the security category. The plan should include an identification of SwA risks, plans for mitigating those risks, associated measures, and plans for continually assessing those risks.

4.3.2 Assurance Case Management

General Considerations. Acquirers must ensure that the assurance case is implemented in accordance with established requirements and approved project plans, in particular the assurance plan approved for use in the contract. (Note: If an assurance plan is not used, special attention should be paid to supplier processes and products on the project to give users and other stakeholders the confidence that software assurance has been considered in the product development.)

The assurance case must be managed as part of the risk management strategy for the acquisition. If the assurance case is used to demonstrate achievement of the security and dependability properties of the software system, then the acquirers must take appropriate steps to manage the assurance case's development and acceptance into operational service.

All elements of any project management methodology that an acquirers use are affected by development and management of an assurance case. The following paragraphs mention common project elements (or principles) that contribute to the delivery of an assured software–intensive system and explain how they are crucial for a project manager to deliver a robust and complete assurance case for transition to operations.

Project Management Reviews. A key element of assurance case development is periodic management and external stakeholder reviews. Acquirers must maintain sufficient insight into the contractor's assurance case production and commit to necessary actions to facilitate its production as efficiently as possible. These reviews are not technical reviews, but rather a senior acquirers monitoring the effectiveness (for example, meeting stakeholder objectives) and adequacy (for example, appropriate infrastructure and audits) of the management of the assurance case. Some level of independent auditing may be required when high assurance is required. The frequency of the assurance case management reviews depend on the complexity of the program/project and other program/project priorities. At a minimum, assurance case management reviews are needed prior to important acquisition milestones. Acquirers should look for evidence that assurance case documentation and records are being effectively controlled, appropriate audits are taking place, and that all stakeholder issues, including nonconformances, are being resolved. Acquirers should also ensure that planned targets versus expended effort for assurance case budget and schedule are appropriately measured and risk managed.

Risk Management Strategy and the Assurance Case. Acquirers and suppliers should ensure the assurance case is developed in a risk-driven environment where the following is evident:

- the assurance-related attributes articulated well (for example, security, safety, reliability, and user-defined and -derived requirements)
- the acquisition's risk management planning process is integrated with technical control activities
- event-driven technical reviews must ensure assurance-related work products meet stringent assurance criteria and properly documented in the assurance case repository.
- the assurance risk acceptance framework is clearly communicated to all stakeholders (especially end users) and documented on contract
- rigorous initial processes to identify and analyze SwA risks are followed by continuous assessment of SwA risks.

Scope Management. The technical and managerial issues associated with Assurance Case management are difficult even with sound program/project planning processes in place; hence, acquirers need a thorough understanding of the project's assurance case scope of effort. The size and scope of the assurance case developed during the implementation phase varies according to the complexity of the system and level of system dependency required. High assurance software containing ubiquitous and legacy applications, complex enterprise software-intensive systems, and national and international interfaced software-intensive systems all

add complexity and unique challenges for analysis and assessment. The numerous methods for composing assurance evidence for these software-intensive systems have major impacts on program/project workflow, certification requirements, cost, and schedule. Suffice it to say, there are no boilerplate templates for assurance case scope.

The important consideration is that acquirers perform the appropriate coordination, planning, and resource allocation to compile a potentially large and robust assurance case. Acquirers must decide when a complete scope of work (contractually or otherwise) has been sufficiently performed (with due process) to an appropriate level of confidence and rigor of evidence. The types of assurance case deliverables/products are discussed in [NDIA], [ISO/IEC 15026], and [SwA CBK].

To maintain technical control, acquirers should ensure that the Contract WBS contains all the assurance case work products that are to be produced by the supplier and relate the assurance elements of work to be completed to each other and the final product(s). For example, the supplier would include a WBS element that captures the scope of work for developing an architecture that meets the assurance requirements and the architect end products. A similar undertaking is needed for the Acquirer WBS.

Schedule Management. The acquirers must contribute to the development of the program/project schedule and critical path and ensure that sufficient schedule is allocated for assurance case activities by suppliers. Any schedule models used for analysis of the master schedule need to incorporate assurance case assessments and the impact from assurance-related activities such as certification and accreditation. The question must be asked: Is the planned rate of completion for assurance work products realistic? For high assurance requirements, the schedule baseline needs to support the application of rigorous software development and testing processes.

Cost Management. Actual cost data on implementation of assurance is not generally available in the public domain. Furthermore, assurance costs vary for each acquisition. Funding estimation for an assurance case is difficult because there is no public domain data that has compared software assurance case estimates to actual costs expended. To avoid cost escalations, the assurance objectives and requisite planning to meet those objectives must be clearly understood. What is well acknowledged about costs is that "bolting on" (and not "building in") assurance will increase cost and impact the schedule in the end because of potential retrofitting system design requirements. Acquirers should consider the following cost factors that influence the implementation of assurance cases in their project:

- *Compilation methods of assurance evidence*. The assurance case requires storage for voluminous amounts of life-cycle data generated.
- *Technology support used*. Process and product requirements from safety and security standards require various qualified tool support. Assurance case repositories/editors need to be procured.
- Robustness of argument. High requirements need rigorous evidence stemming from detailed specification development and testing. Staff skill levels and training costs will likely increase.
- *Assessment abilities*. Inexperienced staff and inappropriate assessment methods increase the likelihood of "unknown unknowns" and contingency planning. Procuring independent SwA technical expertise, resolving contractual dispute issues, and verifying external certification requirements all require resources.
- *Life cycle maintenance*. Both revalidation of evidence/assumptions as system development changes and recertifications require resources.

Human Resource Management. Developing an assurance case is considered a specialty engineering domain. As such, expertise is required in processes, products, tools, and development environments for security- and safety-related systems. Acquirers should ensure that development of design packages for high-assurance

requirements is performed by SMEs who are appropriately cleared. Furthermore, review of design, as well as the design approval, is performed by personnel other than those who developed the design packages. Similar engineering management and infrastructure should be evident from suppliers who also perform design work for high-assurance requirements. An assurance case that does not show evidence of independence in the design review is less credible. Supplier personnel experience should be commensurate with the experience required for the scope and level of design effort to be performed. Acquirers must have access to material on assessing assurance case artifacts.

Data and Configuration Management. The assurance case may contain a plethora of claims and evidence types not collated or contained together; therefore, the assurance case must be composed and managed in such a fashion that all evidence is preserved, traceable, and accessible. Assurance case data should be considered one of the more critical elements of the program. Regulatory or statutory data need to survive the life of the acquisition (and beyond). Acquirers should develop a data strategy that encapsulates what data (and data rights) are needed for sustaining the assurance case as well as what data may only be needed during the contract. Standardization and data reuse management are crucial in an assurance case argument because good data management will bring together acquisition, technology, and logistic elements in a coherent fashion. Inconsistent, incomplete data produces an indefensible assurance case that is difficult to manage in the follow-on phase. Assurance case management includes a number of issues that the acquirers need to consider, including secure data management tools, contractual digital formats (for example, extensible markup language), data exchange requirements, data protection and security, data retention, and media. Data should be delivered in a format suitable for the user's environment.

Acquirers must ensure that suppliers implement a CM Program that details a SwA CM process. During the implementation, acquirers should observe supplier Configuration Control Board meetings to ensure that SwA is fully considered in software changes.

Quality Management. No assurance case can be produced outside of a certified quality system. Acquirers must begin by assessing suppliers' quality system to ensure that appropriate quality elements are included throughout the software development life cycle. Continued adequacy of quality management is evidenced by auditing, examining, and investigating work suppliers perform in order to substantiate that supplier implementation arrangements and procedures are being complied with by the software engineers/developers. Third-party agencies, such as the Defense Contract Management Agency (DCMA), that are independent of the suppliers' software development team and quality assurance personnel should provide an oversight function and provide a level of assurance to acquirers that the software will meet safety and security targets. Records of decisions and results of design and development reviews must be maintained for the duration of the monitoring and acceptance phase.

Assurance Case Measures. Acquirers maintain communication/oversight to ensure the assurance case is progressing in accordance with the contract requirements. Example measurement issues to consider include:

- *Performance*. Is the assurance case development progressing in accordance with the agreed-to assurance plan? Are project technical milestones incorporating assurance case reviews? Does the assurance case comply with contract requirements to include DOD/Government regulations and certification requirements?
- *Resources*. Have suppliers allocated appropriate qualified personnel to the task? Is the assurance case being developed with appropriate tools? Is the assurance case development budget realistic?
- *Quality*. Are suppliers engaging the right acquirers to review the acceptability of the assurance case? Are corrective actions being followed up adequately? Are suppliers' claims, arguments, and evidence sufficiently robust and commensurate with risk?

- *Time*. Is the assurance case development on schedule and fully integrated with the software system development?
- *Continual improvement*. Are suppliers monitoring their own performance to continuously improve delivery? What are the tangible results of this monitoring in terms of impacting the robustness of the assurance case?

4.3.3 Independent Software Testing

Acquirers should consider independent software testing. The completed software product is provided to an independent accredited software testing organization (ISO/IEC 17025) to verify that not only functional requirements but also SwA requirements are met. This testing organization can test in either a white or black box scenario depending on need. See section 2.3 bullet point on page 2–6, "*Plans for Independent Testing*."

PAGE INTENTIONALLY LEFT BLANK

5 Follow-on Phase

The follow-on phase involves maintaining (often called sustainment) the software. This phase includes two major activities: (1) sustainment (includes risk management, assurance case management, and change management) and (2) disposal or decommissioning. During the follow-on phase, software risks must be managed through continued analysis of the assurance case and should be adjusted to mitigate changing risks.

5.1 Sustainment (or Post-release Support)

Care should be taken to enforce terms and conditions contained in the initial contract that apply to the service or product. A provision for maintaining a specific security configuration of COTS software is a good example. See appendix F for sample language relative to security configurations. Maintenance activities include:

- executing configuration control of executable product baselines
- performing software modification revalidation and integration
- conducting problem analyses, reconstruction, review, and acceptance
- predicting software performance (defect density trending and so forth)
- overseeing the engineering environment to ensure that it is fully documented and validated (or security accredited)
- maintaining organization policies and processes for security and safety fixes
- maintaining and executing software migration, data migration, and decommissioning policies and procedures.

Additional contracts are often awarded to provide support during this phase. Analyses should be ongoing to ensure that security requirements remain adequate. To that end, acquirers should ensure that the assurance/security requirements implemented and accepted in previous contracts flow to the follow-on contract efforts. This includes continuous monitoring of the assurance case (including risks) and making appropriate adjustments in using and maintaining software to update the assurance case and mitigate risks.

A formal assurance case and risk management process should be maintained. This process should include continuous threat analyses and vulnerability assessments. In addition, review and adjustment of mitigation strategies should occur regularly. A full-time assurance case/risk management team should be considered. If this team is contracted as a service, suppliers must be adequately trained and cleared. Because acquirers should not abrogate their security responsibility wholly to suppliers, trained and cleared SwA experts inside acquirers organizations must be a part of that team.

5.1.1 Risk Management

Risk management must continue after the implementation and acceptance phase. This includes updating the risk management plan. In this volatile information environment, new risks inevitably emerge. As a result, the security category may be further refined during this phase. In addition, SwA risks and strategies for mitigating those risks are likely to change as well. Measures should be used to provide insights into the changes in the risk environment and into impacts of risk mitigation strategies.

5.1.2 Assurance Case Management—Transition to Operations

The continual assurance (and certification) of software-intensive systems in the follow-on phase presents some unique challenges:

- Many software systems are not architecturally or detail designed for modifications, and enhancements are made many years after procurement.
- System and software engineering change control mechanisms can lack traceability, rigor, and documentation.
- Adequate assurance case maintenance processes may not be in place before the system transitions to operations.
- Support personnel turnover causes loss of corporate knowledge about maintaining and ensuring integrity of legacy software.
- Many software support agencies are not the original software manufacturer and do not employ the same methods, tools, and processes used in development.
- During previous acquisition phases, the software transition planning is typically poorly executed and "assurance concerns" are "thrown over the fence" for follow-on maintenance.

During implementation, acquirers and suppliers must identify the assurance requirements for the follow-on phase to maintain integrity and dependability in the system. These requirements are defined in greater detail as the product's transition to operations nears and the software risk exposure is clearer. Any claims, evidence, arguments, and assumptions in the assurance case that are neither consistent nor based on a known material state of the in-service software weaken the credibility of the evidence and elevate the safety and security risk in using the system. The authority responsible for the assurance case maintenance must keep an auditable record of his or her decisions to include justification for changes made to the assurance case. Changes to the assurance case during the follow-on phase may be required due to a number of reasons, including:

- changes to the software system itself that may invalidate previous claims/evidence and assumptions (for example, changes in operating system lockdown configurations)
- changes to the operational context or environment (for example, a previously isolated system becomes networked)
- changes to system threats, vulnerabilities, consequences, or new issues previously unknown
- modifications of measures to ensure they are appropriate for this phase of the acquisition process.

5.1.3 Other Change Management Considerations

Information systems are typically in a constant state of migration with upgrades to hardware, software, or firmware, and with possible modifications to the surrounding environment of the system. The schedules for and frequency of new releases, updates, and security (and nonsecurity) patches and response times for technical support by software suppliers are beyond the control of the acquirer. Weak change control procedures can corrupt software and introduce new security vulnerabilities.

Change management invokes revalidation efforts. When any hardware or software component is changed, the extent of revalidation must be evaluated. Generally, when hardware components are replaced like for like, no revalidation is necessary. When new hardware is used, the system must be revalidated to ensure no detrimental effects occur. When software is patched or upgraded, revalidation is always required.

Patches and upgrades make direct changes to software and potentially to the operating system configuration to which they are applied. Changes may degrade performance, introduce new vulnerabilities, or reintroduce old vulnerabilities. To understand patch risks, the patch process must be examined in some detail during the initial acquisition and again when follow-on support contracts are awarded. One of the most common patch failures stems from a lack of encryption and authentication in the implementation and acceptance phase. Suppliers should provide updates in a secure fashion. There should be no doubt that the source is legitimate and the update's integrity is maintained in transit.

5.2 Disposal or Decommissioning

Disposal or decommissioning policies and procedures are often overlooked. Many organizations do not have such policies and procedures. Acquirers' orgaanizations should ensure that policies and procedures are developed and followed to ensure the safe and secure disposal or decommissioning of software, along with ensuring data are destroyed or migrated safely and securely. When a software-intensive system is retired or replaced, the data must be migrated by validated means to the new software-intensive system.

PAGE INTENTIONALLY LEFT BLANK

Appendix A. Acronyms

Acronyms that may be used in this document are defined below.

API	Application Programming Interface
CBK	Common Body of Knowledge
CC	Common Criteria
CCEVS	CC Evaluation & Validation Scheme
CERT	Computer Emergency Response Team
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CMT	Cryptographic Module Testing
CMVP	Cryptographic Module Validation Program
CNSS	Committee on National Security Systems
COTS	Commercial-off-the-shelf
CVE®	Common Vulnerabilities & Exposures
CWE	Common Weakness Enumeration
CXO	Chief (X) Officer where X = information, financial, privacy, etc.
DCID	Director Central Intelligence Directive
DCMA	Defense Contract Management Agency
DHS	Department of Homeland Security
DIACAP	Defense Information Assurance Certification and Accreditation Process
DITSCAP	Defense Information Technology System Certification and Accreditation Process
DoD	Department of Defense
DoDD	DoD Directive
DoDI	DoD Instruction
EAL	Evaluation Assurance Level
FAR	Federal Acquisition Regulation
FIPS	Federal Information Processing Standard
FIPS Pub	FIPS Publication
FISMA	Federal Information Security Management Act
FOCI	Foreign Ownership, Control, or Influence
GOTS	Government off-the-shelf
HIMSS	Healthcare Information and Management Systems Society
IA	Information Assurance
ID/IQ	Indefinite Delivery/Indefinite Quantity
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IPS	Intrusion Protection System
ISO	International Standards Organization
IT	Information Technology

MAC	Mission Assurance Category
MOTS	Modifiable off-the-shelf
NDIA	National Defense Industrial Association
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NIST SP	NIST Special Publication
NSS	National Security System
NSTISSP	National Security Telecommunications & Information Systems Security Policy
NVD	National Vulnerability Database
NVLAP	National Voluntary Laboratory Accreditation Program
OMB	Office of Management and Budget
OS	Operating System
OWASP	Open Web Application Security Project
PDA	Parental Drug Association
PITAC	President's Information Technology Advisory Committee
PGP	Pretty Good Privacy
PM	Program/Project Manager
QA	Quality Assurance
RFI	Request for Information
RFP	Request for Proposals
SANS	The SANS Institute (www.sans.org)
SDLC	Software Development Life Cycle
SME	Subject Matter Expert
SOA	Service Service-Oriented Architecture
SOAP	Simple Object Acess Access Protocol
SOAR	State of the Art Report
SOW	Statement of Work
SQL	Standard Query Language
SSAA	System Security Authorization Agreement
SSL	Secure Socket Layer
SwA	Software Assurance
SwA CBK	SwA Common Body of Knowledge
TQM	Total Quality Management
USC	United States Code
USG	United Stated Government
V&V	Verification and Validation
WBS	Work Breakdown Structure
WSDL	Web Service Definition Language
XML	eXtensible Markup Language

Appendix B. Glossary

accountability ..... The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports non-repudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.

accreditation ..... Formal declaration by a designated accrediting authority that an information system is approved to operate at an acceptable level of risk, based on the implementation of an approved set of technical, managerial, and procedural safeguards [CNSSI 4009].

acquisition ..... The acquiring by contract with appropriated funds of supplies or services (including construction) by and for the use of the Federal Government through purchase or lease, whether the supplies or services are already in existence or must be created, developed, demonstrated, and evaluated. Acquisition begins at the point when agency needs are established and includes the description of requirements to satisfy agency needs, solicitation and selection of sources, award of contracts, contract financing, contract performance, contract administration, and those technical and management functions directly related to the process of fulfilling agency needs by contract [FAR Subpart 2.101].

acquisition life cycle ..... All stages involved in the process of procuring products or services, beginning with the determination of a need for products or services and ending with contract completion or closeout [USCOURTS].

acquisition management ..... Planning, organizing, leading, and controlling the acquisition process. The acquisition process begins with the needs determination and follows with specifying requirements and procurement of supplies or services

acquisition planning ..... The process by which the efforts of all personnel responsible for an acquisition are coordinated and integrated through a comprehensive acquisition plan for fulfilling the organization need in a timely manner and at a reasonable cost. It includes developing the overall strategy for managing the acquisition [adapted from FAR Subpart 2.101].

asset ..... Anything that has value (e.g. data, executing process) to a stakeholder (e.g. organization who owns it) [adapted from ISO/IEC 27005].

assurance ..... Grounds for confidence that an entity meets its security objectives [ISO/IEC 15408-1]. Also see software assurance.

assurance argument ..... A justification that a given assurance claim (or sub-claim) is true or false [NDIA].

assurance case ..... The set of assurance claims of critical system/software assurance properties (requirements of the system), assurance arguments that justify the claims

(including assumptions and context), and assurance evidence supporting the arguments [NDIA].

assurance claim ..... The critical system/software requirements for assurance, including the maximum level of uncertainty permitted [NDIA].

assurance evidence ..... Information that demonstrably substantiate the arguments in an assurance case [adapted from NDIA].

attack ..... Attempt to gain unauthorized access to information resources or to attempt to compromise the integrity, availability, or confidentiality of said resources. For the purposes of this definition information resources include software whether embedded (e.g., mobile phone software, control system software, etc.) or part of a larger information infrastructure or system [adapted from CNSSI 4009].

Attack is the act of carrying out an exploit [Barnum].

availability ..... Ensuring timely and reliable access to and use of information [FISMA 2002].

A loss of availability is the disruption of access to or use of information or an information system [FIPS Pub 199].

buffer overflow ..... A condition at an interface under which more input can be placed into a buffer or data holding area than the capacity allocated, overwriting other information. Attackers exploit such a condition to crash a system or to insert specially crafted code that allows them to gain control of the system [NIST SP 800-28].

A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold or when a program attempts to put data in a memory area past a buffer. In this case, a buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers [CWE-120].

bug ..... A problem that exists in the software's code that may or may not represent a vulnerability [Barnum].

built-in security defenses ..... Capabilities designed to minimize the exposure of the software's vulnerabilities to external threats and to keep the software in a secure state regardless of the input and parameters it receives from its users or environment.

certification ..... Comprehensive evaluation of the technical and non-technical security safeguards of an information system to support the accreditation process that establishes the extent to which a particular design and implementation meets a set of specified security requirements [CNSSI 4009].

certification & accreditation . A comprehensive assessment of the management, operational, and technical security controls in an information system, made in support of security accreditation, to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with

respect to meeting the security requirements for the system. *Accreditation* is the official management decision given by a senior agency official to authorize operation of an information system and to explicitly accept the risk to agency operations (including mission, functions, image, or reputation), agency assets, or individuals, based on the implementation of an agreed-upon set of security controls [NIST SP 800-37].

change management.....A structured approach to change in individuals, teams, organizations and societies that enables the transition from a current state to a desired future state.

commercial off the shelf (COTS).....Commercial software or hardware products, which are ready-made and available for sale to the general public.

component…….. A part or element within a larger system. A component may be constructed of hardware or software and may be divisible into smaller components. In the strictest definition, a component must have a contractually-specified interface(s), explicit context dependencies, the ability to be deployed independently, and the ability to be assembled or composed by someone other than its developer with other components. In a less restrictive definition, a component may also be a code unit (that is, a separately testable element of a software component, a software component that cannot be further decomposed into constituent components, or a logically separable part of a computer program) or a code module (that is, a program unit that is discrete and identifiable with respect to compilation, combination with other units, and loading). Note that the terms code unit and code module are sometimes used interchangeably [Goertzel, 2007].

component assembly .....Process of organizing and configuring components (by the strict definition of that term) to use their built-in interfaces to communicate/interact with each other.

confidentiality.....Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information [FISMA 2002].

A loss of *confidentiality* is the unauthorized disclosure of information [FIPS 199].

configuration management....Management of security features and assurances through control of changes made to hardware, software, firmware, and documentation, test, test fixtures, and test documentation throughout the life cycle of an information system [CNSSI 4009].

continuous security monitoring .....Employment of techniques and procedures for the continuous monitoring of the security state of the software.

contract..... A mutually binding legal relationship obligating the seller to furnish the supplies or services (including construction) and the Acquirer to pay for them. It includes all types of commitments that obligate the organization to an expenditure of appropriated funds and that, except as otherwise authorized, are in writing. In addition to bilateral instruments, contracts include (but are not limited to) awards and notices of awards; job orders or task letters issued under basic ordering agreements; letter contracts; orders, such as purchase orders, under which the contract becomes effective by written acceptance or performance; and bilateral contract modifications. Contracts do not include grants and cooperative agreements covered by 31 U.S.C. 6301, *et seq.* [FAR Subpart 2.101].

contracting..... Purchasing, renting, leasing, or otherwise obtaining supplies or services from nonfederal sources. Contracting includes description (but not determination) of supplies and services required, selection and solicitation of sources, preparation and award of contracts, and all phases of contract administration. It does not include making grants or cooperative agreements [FAR Subpart 2.101].

contract or procurement specialist..... An individual who performs contracting functions usually in support of a contracting officer or other contracting official.

contract administration..... Management of a contract to ensure that organization receives the quality of products and services specified in the contract within established costs and schedules.

contracting officer..... A person with the authority to enter into, administer, and/or terminate contracts and make related determinations and findings [adapted from FAR Subpart 2.101].

contracting officer representative (COR) ..... See contracting officer technical representative

contracting officer technical representative (COTR)..... An individual appointed by the contracting officer to act for the contracting officer in certain contracting situations and administer a contract on a daily basis [FAI].

correctness..... (1) The degree to which software is free from errors or inadequacies in its specification, design, and implementation.
(2) The degree to which software, documentation, or other items satisfy their specified requirements.
(3) The degree to which software, documentation, or other items meet user needs and expectations, whether those needs and expectations are specified or not [adapted from IEEE 610.12].

critical software..... Software the failure of which could have an impact on security, safety, or could cause large financial or social loss. Critical software is also referred to as "high consequence software" [adapted from IEEE Std 1012].

custom software.....Software developed either for a specific organization or function. It is generally not targeted to the mass market, but usually created for a specific customer to satisfy that customer's unique needs.

defense-in-depth .....Security strategy in which people, technology, and operational capabilities are combined and coordinated to establish variable barriers across multiple layers and dimensions of computing environments or networks. This term is synonymous with security-in-depth [adapted from CNSSI 4009].

A principle for building systems stating that multiple defensive mechanisms at different layers of a system are usually more secure than a single layer of defense. For example, when performing input validation, one might validate user data as it comes in and then also validate it before each use — just in case something was not caught, or the underlying components are linked against a different front end, etc. [OWASP Glossary].

denial of service (DoS) .....Prevention of authorized access to a system resource by making that resource unavailable or inaccessible at its expected level of operation capacity and performance, e.g., by delaying system operations and functions, terminating system operations, or interfering with connectivity to/from the system [adapted from ISO/IEC 18028-1].

Any action or series of actions that prevents any part of an IS from functioning [CNSSI 4099].

due care…… The responsibility that managers and their organizations have a duty to provide for information security to ensure that the type of control, the cost of control, and the deployment of control are appropriate for the system being managed [NIST SP 800-30].

embedded software .....Software that is part of a larger physical system and performs some of the requirements of that system, e.g., software used in an aircraft or rapid transit system. Typically, such software does not provide an interface with the user; however, this limitation is changing with some modern embedded software.

error .....The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition [IEEE 610.12].

event .....An occurrence of some specific situation, activity, or data handling [adapted from ISO/IEC TR 15947].

exploit.....A technique, which may be implemented by software code (often in the form of a script), that takes advantage of a vulnerability or security weakness in a piece of target software. If implemented by software code, the code itself (rather than the activity it performs) is sometimes referred to as the exploit [adapted from Barnum].

failure..... The inability of a system or component to perform its required functions within specified requirements [adapted from IEEE 610.12].

flaw..... Error of commission, omission, or oversight in an information system that may allow protection mechanisms to be bypassed [CNSSI 4009].

A flaw is a problem that exists in the software's design. May or may not represent a vulnerability [Barnum].

freeware..... Software that is available for use free of charge for an unlimited time.

government off the shelf (GOTS)..... Software and hardware products that are developed by the technical staff of the government agency for which it is created or by an external entity, but with funding and specification from the agency.

implementation……Of a system, the system development phase at the end of which the hardware, software, and procedures of the system considered become operational [ANSDIT].

incentive contract…….Incentive contracts as described in this subpart are appropriate when a firm-fixed-price contract is not appropriate and the required supplies or services can be acquired at lower costs and, in certain instances, with improved delivery or technical performance, by relating the amount of profit or fee payable under the contract to the contractor's performance.

Incentive contracts are designed to obtain specific acquisition objectives by—(1) Establishing reasonable and attainable targets that are clearly communicated to the contractor; and (2) Including appropriate incentive arrangements designed to—(i) motivate contractor efforts that might not otherwise be emphasized; and (ii) discourage contractor inefficiency and waste.

When predetermined, formula-type incentives on technical performance or delivery are included, increases in profit or fee are provided only for achievement that surpasses the targets, and decreases are provided for to the extent that such targets are not met. The incentive increases or decreases are applied to performance targets rather than minimum performance requirements.

The two basic categories of incentive contracts are fixed-price incentive contracts (see 16.403 and 16.404) and cost-reimbursement incentive contracts (see 16.405). Since it is usually to the Government's advantage for the contractor to assume substantial cost responsibility and an appropriate share of the cost risk, fixed-price incentive contracts are preferred when contract costs and performance requirements are reasonably certain. Cost-reimbursement incentive contracts are subject to the overall limitations in 16.301 that apply to all cost-reimbursement contracts.

Award-fee contracts are a type of incentive contract [FAR Subpart 16.401].

independent testing ..... A common practice of software testing is that it is performed by an independent group of testers after the functionality is developed but before it is shipped to the customer. This practice often results in the testing phase being used as project buffer to compensate for project delays, thereby compromising the time devoted to testing.

information assurance..... Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities [CNSSI 4009].

information resources..... Information and related resources, such as personnel, equipment, funds, and information technology [FISMA 2002].

information security..... The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability [FISMA 2002].

information sensitivity..... A measure of the importance assigned to information by its owner, for the purpose of denoting its need for protection [adapted from "sensitivity" defined in NIST SP 800-60—also known as sensitive information].

information system..... A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposal of information. [44 U.S.C., Sec. 3502].

information technology..... Any equipment or interconnected system or subsystem of equipment that is used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information by the executive agency. For purposes of the preceding sentence, equipment is used by an executive agency if the equipment is used by the executive agency directly or is used by a contractor under a contract with the executive agency which (i) requires the use of such equipment; or (ii) requires the use, to a significant extent, of such equipment in the performance of a service or the furnishing of a product. The term information technology includes computers, ancillary equipment, software, firmware and similar procedures, services (including support services), and related resources [40 U.S.C., Sec. 11101].

input validation ..... The act of determining that data input to a program is sound (e.g., for example, might include: the length, format, physical content of the data do not vary from the acceptable parameters defined for length, format, and physical content) [adapted from OWASP Glossary].

integrity..... Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity [FISMA 2002].

A loss of *integrity* is the unauthorized modification or destruction of information [FIPS 199].

information security ..... Protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide—

1) integrity, which means guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity;

2) confidentiality, which means preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary information; and

3) availability, which means ensuring timely and reliable access to and use of information [FISMA 2002].

information security personnel ..... Individuals who protect information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction to provide confidentiality, integrity, and availability.

justifiable confidence ..... The actions, arguments and evidence that provides a basis for a defensible reduction in uncertainty.

malicious activity ..... An activity by a person or software process that intentionally misuses, misappropriates, damages, or destroys the functionality, resources, or data of the system, or which violates any aspect of its governing usage policies including its security policy.

malicious code ..... Software or firmware intended to perform an unauthorized process that will have adverse impact on the confidentiality, integrity, or availability of an information system [CNSSI 4009].

A virus, worm, Trojan horse, or other code-based malicious entity that successfully infects the host [NIST SP 800-61].

Undocumented software or firmware intended to perform an unauthorized or unanticipated process that will have adverse impact on the dependability of a component or system. Malicious code may be self –contained (as with viruses, worms, malicious bots, and Trojan horses), or may be embedded in another software component (as with logic bombs, time bombs, and some Trojan horses) [Goertzel, 2007].

malware ..... A program that is inserted into a system, usually covertly, with the intention of compromising the specified operation of that system, including its ability to protect the confidentiality, integrity, and availability of the system's data, applications, or operating system or of otherwise annoying or inhibiting the operational abilities of the system's users [adapted from NIST SP 800-83].

measure.....Variable to which a value is assigned as the result of measurement [ISO/IEC 15939]. This definition is the coinage of the measurement community, and is at variance with any standard dictionary definition of the word.

measurement.....Set of operations having the object of determining a value of a measure [ISO/IEC 15939].

mission..... A specific task with which a person or a group is charged [Webster].

mission assurance.....An engineering process performed over the life cycle of a program to identify and mitigate design, production, test, and field support deficiencies that could affect mission success. It requires the application of system engineering, risk management, quality and management principles to achieve mission success. It relies on independent technical assessment throughout the entire design, development, testing, deployment, and operations process [Grimm, 2004].

misuse.....Usage that deviates from what is expected (with expectation usually based on the software's specification). If the misuse is maliciously motivated, it is referred to as *abuse*.[Goertzel, 2007].

mobile code.....Software modules obtained from remote systems, transferred across a network, and then downloaded and executed on local systems without explicit installation or execution by the recipient [CNSSI No. 4009].
In particular, "mobile code" is used to describe applets within web browsers based upon Microsoft's ActiveX, Sun's Java, or Netscape's JavaScript technologies.

national security system.....(A) Any information system (including any telecommunications system) operated by an agency or by a contractor of an agency, or other organization on behalf of an agency—

(i) the function, operation, or use of which—

(I) involves intelligence activities;

(II) involves cryptologic activities related to national security;

(III) involves command and control of military forces;

(IV) involves equipment that is an integral part of a weapon or weapons system; or

(V) subject to subparagraph (B), is critical to the direct fulfillment of military or intelligence missions; or

(ii) is protected at all times by procedures established for information that have been specifically authorized under criteria established by an Executive order or an Act of Congress to be kept classified in the interest of national defense or foreign policy.

(B) Subparagraph (A)(i)(V) does not include a system that is to be used for routine administrative and business applications (including payroll, finance, logistics, and personnel management applications) [FISMA 2002].

non-developmental item ..... (1) Any previously developed item of supply used exclusively for governmental purposes by a Federal agency, a State or local government, or a foreign government with which the United States has a mutual defense cooperation agree;
(2) Any item described in paragraph (1) of this definition that requires only minor modification or modifications of a type customarily available in the commercial marketplace in order to meet the requirements of the procuring department or agency; or
(3) Any item of supply being produced that does not meet the requirements of paragraphs (1) or (2) solely because the item is not yet in use.[FAR Subpart 2.101].

non-repudiation ..... Assurance the sender of data is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the data. [CNSSI 4009] In terms of software's activities, non-repudiation extends to the inability of software to deny having performed a specific action.

open source software ..... Commercial software whose source code is available by license permitting users to study and change (improve) the software, as well as redistribute it in modified or unmodified form.

outsourcing..... The delegation of operations or jobs from internal production within a business to an external entity usually by contract.

patch management..... The process of acquiring, testing, and distributing patches to the appropriate administrators and users throughout the organizations [NIST SP 800-61].

penetration testing ..... Security testing in which evaluators mimic real-world attacks to attempt to identify methods for circumventing the security features of an application, system, or network. Penetration testing often involves issuing real attacks on real systems and data, using common tools and techniques used by attackers. Most penetration tests involve looking for combinations of vulnerabilities on a single system or multiple systems that can be used to gain more access than could be achieved through any single vulnerability [NIST SP 800-115].

program..... The umbrella structure established to manage a series of related projects. The program does not produce any project deliverables. The project teams produce them all. The purpose of the program is to provide overall direction and guidance, to make sure the related projects are communicating effectively, to provide a central point of contact and focus for the client and the project teams, and to determine how individual projects should be defined to ensure all the work gets completed successfully [Mochal].

Program may also be an executable software entity.

quality ..... The degree to which a component, system or process meet its specified requirements and/or stated or implied user, customer, or stakeholder needs and expectations [Goertzel, 2007].

regulations ..... Rules and administrative codes issued by governmental agencies at all levels, municipal, county, state and federal. While not laws they have the force of law, since they are adopted under authority granted by statutes, and often include penalties for violations. [Legal]

regulatory and standards compliance ..... Refers to the application of the principles, policies, and procedures that enable an enterprise to meet applicable information security laws, regulations, standards, and policies to satisfy statutory requirements, perform industry-wide best practices, and achieve its information security program goals [SwA CBK].

reliable software ..... The ability of a software application and its parts to perform its mission without failure, degradation, or demand on the support system.

Software that possesses the characteristic of reliability to the extent that it can be expected to consistently perform its intended functions satisfactorily. This implies a time factor in that reliable software is expected to perform correctly over a period of time. It also encompasses environmental considerations in that the software is required to perform correctly in whichever conditions it finds itself - this is sometimes termed robustness.

request for information ..... A document used to obtain price, delivery, other market information, or capabilities for planning purposes when the Government does not presently intend to issue a solicitation [FAR Subpart 15.202(e)].

request for proposal ..... A solicitation used in negotiated acquisitions to solicit proposals from prospective contractors to communicate the Acquirer's requirements, anticipated terms and conditions that will apply to the contract, information required to be in proposals, and factors and significant subfactors that will be used to evaluate proposals and their relative importance [FAR Subpart 15.303].

requirement ..... A statement that identifies an operational, functional, or design characteristic or constraint of a product or process. Ideally, a requirement should be unambiguous, testable or measurable, and necessary to the acceptability of the process or product (by consumers or those responsible for verifying the product's/process' conformance to internal quality assurance guidelines [adapted from ISO/IEC 26702 IEEE 1220].

residual risk ..... The remaining potential risk after all security measures are applied [NIST SP 800-33].

risk ..... Possibility that a particular threat will adversely impact an information resource (including information systems, information, and software, whether embedded

or part of an information systems) by exploiting a particular vulnerability [adapted from CNSSI 4009].

The level of impact on agency operations (including mission, functions, image, or reputation), agency assets, or individuals resulting from the operation of an information system given the potential impact of a threat and the likelihood of that threat occurring [FIPS 200].

The potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization. It is measured in terms of a combination of the probability of an event and its consequence [ISO/IEC 13335-1].

risk analysis ..... The process of identifying the risks to system security and determining the likelihood of occurrence, the resulting impact, and the additional safeguards that mitigate this impact. Risk analysis is part of risk management and synonymous with risk assessment [NIST SP 800-30].

risk assessment ..... The process of identifying risks to agency operations (including mission, functions, image, or reputation), agency assets, or individuals by determining the probability of occurrence, the resulting impact, and additional security controls that would mitigate this impact. Part of risk management, synonymous with risk analysis, and incorporates threat and vulnerability analyses [NIST SP 800-30 and NIST SP 800-53].

risk-based decision ..... Decision making in which such decisions are made solely based on the results of a probabilistic risk analysis.

risk management ..... The process of managing risks to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation resulting from the operation or use of an information systems, and includes: (i) the conduct of a risk assessment; (ii) the implementation of a risk mitigation strategy, and (iii) employment of techniques and procedures for the continuous monitoring of the security state of the information system. [adapted from NIST SP 800-39 and FIPS 200].

risk mitigation ..... Prioritizing, evaluating, and implementing the appropriate risk-reducing controls recommended from the risk assessment process [NIST SP 800-30].

risk tolerance ..... The level of risk an entity is willing to assume in order to achieve a potential desired result [NIST SP 800-32].

robustness ..... The degree to which a component or system can function correctly in the presence of invalid inputs or stressful environmental conditions, including inputs or conditions that are intentionally and maliciously created [IEEE 610.12].

role ..... An abstract definition of a set of functions performed and work products or deliverables owned. Roles are typically realized by an individual, or a set of

individuals, working together as a team. Roles are not individuals; instead, they describe how individuals behave in the business and what responsibilities these individuals have [IBM].

secure coding ..... Software programming practices that reduce or eliminate software defects/programming errors as well as other programming practices that lead to software vulnerabilities [CERT Secure Coding].

secure coding principles..... A set of philosophical imperatives that collectively govern how coding is done by the programmer so that the resulting software will behave and function as securely as possible.

secure coding tools ..... Tools are that can make work easier, at various stages of the software development life cycle. Categories of such tools include:
1. Static Code Checkers
2. Runtime Code Checkers
3. Profiling Tools
[Graff]

secure design principles..... A set of philosophical imperatives that collective govern how the design is conceived by the developer so that the resulting software will behave and function as securely as possible

secure software ..... Software that realizes, with justifiably high confidence but does not guarantee absolutely a substantial set of explicit security properties and functionality, including all those required for its intended usage [Redwine & Davis].

secure software project management ..... Systematic, disciplined, and quantified" application of management activity that ensures the software being developed conforms to security policies and meets security requirements [Abran].

security ..... Protection against intentional subversion or sabotage (which includes forced failure). Security is a composite of four attributes – confidentiality, integrity, availability, and accountability plus aspects of a fifth, usability, all of which have the related issue of their assurance [SwA CBK].

To be considered secure, software must exhibit three properties:

1. Dependability: Dependable software executes predictably and operates correctly under all conditions, including hostile conditions, including when the software comes under attack or runs on a malicious host.
2. Trustworthiness: Trustworthy software contains few if any vulnerabilities or weaknesses that can be intentionally exploited to subvert or sabotage the software's dependability. In addition, to be considered trustworthy, the software must contain no malicious logic that causes it to behave in a malicious manner.
3. Resilience: Resilient software can resist most known attacks and as many novel attacks as possible. It will also be able to tolerate most of the attacks it

cannot resist. Finally, it will be able to isolate the source of, limit the extent of damage from, and recover quickly from the few attacks it can neither resist nor tolerate.

security architecture ..... Computer security model referring to the underlying computer architectures, protection mechanisms, distributed computing environment security issues, and formal models that provide the framework for information systems security policy.

security attributes ..... A security-related quality of an object. Security attributes may be represented as hierarchical levels, bits in a bit map, or numbers. Compartments, caveats, and release markings are examples of security attributes [FIPS 188].

security category ..... The characterization of information or an information system based on an assessment of the potential impact that a loss of confidentiality, integrity, or availability of such information or information system would have on organizational operations, organizational assets, or individuals [FIPS 199]. [Note that the security category of information or an information system also applies to the software that processes the information in an information system.]

security certification ..... A comprehensive assessment of the management, operational, and technical security controls in an information system, made in support of security accreditation, to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system [NIST SP 800-37].

security change management ..... All activities involved in (1) defining and instilling new values, attitudes, norms, and behaviors within an organization that support new ways of doing work and overcome resistance to change that involve changes to the security configuration; (2) building consensus among customers and stakeholders on specific changes designed to better meet their needs; and (3) planning, testing, and implementing all aspects of the transition from one organizational structure or business process to another [adapted from GAO BPR Glossary].

security control ..... The management, operational, and technical control (i.e., safeguards or countermeasures) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information [FIPS 199].

security control baseline ..... The set of minimum security controls defined for a low-impact, moderate-impact, or high-impact information system.

security objectives ..... Confidentiality, integrity, and availability [FISMA, 2002].

security metrics ..... A system of security measurement to quantitatively assess information and information systems security based on security performance goals and objectives [NIST SP 800-55].

security policy ..... A document or documents that describe the security requirements and their solutions .

security requirements ..... Requirements levied on a system that are intended to ensure that the system exhibits all of the security properties and performs all of the security-related functions required to ensure its own dependable, trustworthy, and resilient operation, and the preservation of the confidentiality, integrity, and availability of the information it processes, stores, and/or transmits. Security requirements may be derived from laws, executive orders, directives, policies, instructions, regulations, organizational (mission), or individual user needs [adapted from NIST SP 800-53 and Goertzel, 2008].

security requirements analysis ..... A process for analysis of security requirements to determine how, when, where, and to what extent planned security controls are needed. The process involves reviewing mandated security requirements, functional security requirements, and assurance requirements [adapted from NIST SP 800-64, p. 28].

security specifications ..... Documented security requirements.

sensitive information ..... A measure of the importance assigned to information by its owner, for the purpose of denoting its need for protection [NIST SP 800-60].

sensitivity determination ..... A graduated system of marking (e.g., low, moderate, high) information and information processing systems based on threats and risks that result if a threat is successfully conducted [FIPS 201-1].

shareware ..... Marketing method for commercial software, whereby a trial version is distributed in advance and without payment, as is common for proprietary software. Shareware software is typically obtained free of charge. Shareware is known as "try before you buy," demoware, trialware, among other names. Payment is often required once a set period of time has elapsed after installation.

A kind of freeware for which the software's author or distributor requests some payment, usually in the accompanying documentation files or in an announcement made by the software itself. Such payment may or may not buy the purchaser additional support of functionality.

software ..... A set of instructions, written in some form of symbolic language (i.e., a "programming language" or "scripting language"), which are ultimately interpreted or compiled into the low-level binary language directly understood by the hardware of the processor on which the software executes, in order for that processor to accomplish the functional tasks specified by the software.

software acceptance

testing..... A formal test defined to check acceptance criteria for software prior to its delivery.

software assurance..... The level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its life cycle, and the software functions in the intended manner [CNSSI 4009].

software pedigree..... Background/lineage of the software being acquired. This includes such considerations as how the version of the software under consideration at a given point in time was originally conceived and implemented, and by whom. While the software's pedigree is extended, and thus changed, each time the software is modified in some way by its developer, at any given point in time, the software as it exists in that point in time, can be said to have a fixed pedigree.

software provenance..... Experience of the software being acquired after it leaves the control of its developer(s) and enters the supply chain. This includes such considerations as how the software is licensed, how it is installed and configured in its execution environment, and how it is modified through patching and updating, and by whom. Provenance also reflects changes in responsibility for the ongoing development of the software (new versions, patches, etc.)----for example, if this responsibility shifts from the software's original developer to an integrator or a new development organization (as when one software firm buys another).

software development process ..... The process by which user needs are translated into a software product. the process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes installing and checking out the software for operational activities. Note: these activities may overlap or be performed iteratively [adapted from IEEE 610.12].

software-intensive system ..... A system in which the majority of components are implemented in/by software, and in which the functional objectives of the system are achieved primarily by its software components [Goertzel, 2007].

software resilience..... Software that can resist most known attacks and as many novel attacks as possible and able to tolerate most of the attacks it cannot resist. Finally, resilient software will be able to isolate the source of, limit the extent of damage from, and recover quickly from the few attacks it can neither resist nor tolerate.

software ..... See the definition for "security."

software security weakness..... An underlying condition or construct in software that has the potential for degrading the security of the software [Barnum].

software supply chain ..... A coordinated system of organizations, people, activities, information and resources involved in moving software in physical or virtual manner from supplier to customer.

solicitation ..... A document that requests proposals, offers, quotes, or information from prospective contractors.

stakeholder ..... An individual or constituencies who have a vested interest in an outcome.

standard ..... An agreement among any number of organizations that defines certain characteristics, specification, or parameters related to a particular aspect of computer technology [IEEE 100].

Statement of Work (SOW) or Work Statement (WS) ..... A document incorporated into a solicitation (and contract upon award) that describes the needs and requirements of work to be done/delivered.

Statement of Objectives (SOO) ..... A document incorporated into the solicitation that states the overall performance objectives. It is used in solicitations when the organization intends to provide the maximum flexibility to each potential supplier to propose an innovative approach [adapted from FAR Subpart 2.101].

strategy ..... A plan of action resulting from a formal process of planning and anticipation of realizing specific goals [Webster].

subversion ..... Changing (process or) product so as to provide a means to compromise a required property, such as security [adapted from Anderson].

supplier relationship management ..... A business strategy designed to optimize profitability, revenue and customer satisfaction by organizing the enterprise around customer segments, fostering customer-centric behavior and implementing customer-centric processes. The application domains of CRM include technology-enabled selling (TES), customer service and support (CSS), and technology-enabled marketing (TEM). CRM optimized through Web channels is known as e-channel CRM (e-CRM) [Gartner].

supply chain ..... The set of organizations, people, activities, information, and resources for creating and moving a product or service, including its subcomponents, from suppliers through to their customers [NDIA].

system ..... A combination of interacting elements organized to achieve one or more stated purposes [ISO/IEC 15288].

testing ..... An activity performed for assessing the conformance of software with any or all of its required properties and/or behaviors, and for improving it, by identifying defects and problems [adapted from Abran].

threat ..... Any circumstance or event with the potential to adversely impact an IS through unauthorized access, destruction, disclosure, modification of data, and/or denial of service [CNSSI 4009].

An actor, agent, circumstance, or event with the potential to cause harm to a software-intensive system or to the data or resources to which it has or enables access. If intentional and malicious, the threat is likely to be realized by an attack that exploits a vulnerability in software [Barnum].

threat model(ing) ..... The analysis, assessment and review of audit trails and other information collected for the purpose of searching out system events that may constitute violations of system security [CNSSI 4009].

total quality management ..... A management strategy aimed at embedding awareness of quality in all organizational processes.

trojan horse ..... Malicious program that masquerades as a benign application [ISO/IEC 18043].

trust ..... The confidence one element has in another that the second element will behave as expected.

trustworthiness ..... Logical basis for assurance (i.e. justifiable confidence) that the system will perform correctly, which includes predictably behaving in conformance with all of its required critical properties, such as security, reliability, safety, survivability, etc, in the face of wide ranges of threats and accidents, and will contain no exploitable vulnerabilities either of malicious or unintentional origin. Software that contains exploitable faults or malicious logic cannot justifiably be trusted to "perform correctly" or to "predictably satisfy all of its critical requirements" because of its compromisable nature and the presence of unspecified malicious logic would make prediction of its correct behavior impossible [Goertzel, 2007].

trustworthy software ..... Computer software that contains few if any vulnerabilities or weaknesses that can be intentionally exploited to subvert or sabotage the software's dependability, and contains no malicious logic that causes it to behave in a malicious manner [Goertzel, 2008].

unauthorized access ..... A person gains logical or physical access without permission to a network, system, application, data, or other information technology resource [NIST SP 800-61,Rev 1].

Occurs when a user, whether legitimate or not, accesses a resource that he/she is not permitted to use [adapted from FIPS 191, p. 11].

validation ..... Process of applying specialized security test and evaluation procedures, tools, and equipment needed to establish acceptance for joint usage of an information system by one or more departments or agencies and their contractors [CNSSI 4009].

The act of determining that data is sound. In security, this term is generally used in the context of validating input [OWASP Glossary].

vulnerability.....Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited [CNSSI 4009].

A software weakness that can be exploited by an attacker. Bugs and flaws collectively form the basis of most software vulnerabilities [Barnum].

weakness .....A flaw, defect, or anomaly in software that has the potential of being exploited as a vulnerability when the software is operational. A weakness may originate from a flaw in the software's security requirements or design, a defect in its implementation, or an inadequacy in its operational and security procedures and controls.

The distinction between "weakness" and "vulnerability" originated with the MITRE Corporation Common Weaknesses and Exposures (CWE) project (http://cve.mitre.org/cwe/about/index.html) [CWE-120].

web service.....A software component or system designed to support interoperable machine- or application-oriented interaction over a network. A Web service has in interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its descriptions using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. [NIST SP 800-95].

PAGE INTENTIONALLY LEFT BLANK

Appendix C. An Imperative for SwA in Acquisition

Current Practice. Common current practice in acquisition is to accept software that satisfies functionality with little regard for achieving and assuring security properties—increasing the danger (risk exposure) to users. Acquisition officials continue to accept software riddled with errors and other security vulnerabilities. This, in part, may be due to acquisition policies and procedures that do not ensure that security is a main concern of software. In addition, acquisition officials may not be aware of the costs and increased risk exposure attributable to software that is not secure.

Purchasing secure software does entail moderate upfront costs to the acquisition manager; however, the price paid in lost time and resources to continually maintain (patch) a vulnerable software component can run as much as three times the initial purchase of secure software [Graff & Van Wyk]. Many organizations fall behind in properly patching vulnerable software due to those exponential costs, leaving them exposed to attack.

Dangers may be attributable to software errors or other vulnerabilities to include the unknowing acceptance of software that contains malicious code. Vulnerable software may permit the following:

- unintentional errors leading to faulty operations that result in destruction of information or major disruption of operations
- intentional insertion of malicious code intent on loss of life, destruction of information, major disruption of operations, or even destruction of critical infrastructure
- theft of vital information that is sensitive or classified
- theft of personal information
- changed product, inserted agents, or corrupted information.

The Imperative. Rampant worldwide increase in exploitation of software vulnerabilities demands that acquisition officials not only check for acceptable functionality but also achieve acceptable software assurance. Although the prevailing practices of software suppliers have failed to produce safe, secure, reliable, and dependable software, some segments of the software industry are moving toward rigorous software development practices to minimize software errors and other vulnerabilities that give our adversaries an open door to exfiltrate data and to deny or degrade services.

The acquisition process can be leveraged to promote good software development practices and to facilitate the delivery of trustworthy software. All final software security requirements decisions are made during the acquisition process, in addition to acceptance and implementation decisions. Security cannot be "bolted on" after the product is delivered.

The Open Web Application Security Project (OWASP) Foundation states, "Ultimately, we believe that there is no alternative to making security a part of the software contracting. Currently, we believe that there are serious misunderstandings about the security of code being delivered under many software development contracts. This can only lead to expensive litigation and a decision made by individuals with little software experience or understanding."[18] The acquisition official is important in the line of defense to ensure that safe, secure, reliable, and dependable software is delivered. To that end, the acquisition official should provide an

18 See http://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex for a full discussion of this problem.

acquisition process that ensures the continuity of essential business operations across a wide range of potential emergencies and/or exploitations.

Information Assurance vis-à-vis SwA. Information assurance relates to measures that protect and defend information and information systems by ensuring their availability,[19] integrity,[20] authentication,[21] confidentiality,[22] and nonrepudiation.[23] These measures include providing for restorations of information systems by incorporating protection, detection, and reaction capabilities. Information systems include the software that controls the system and processes data and information. Therefore, measures must be used to protect the systems from software vulnerabilities and unintended software processing that expose a system to compromises in availability, integrity, and other security properties. SwA provides those measures.

United States National Concerns. A February 2005 report to the President of the United States identified the vulnerability of software as a cyber security issue of national importance. As indicated in the report, software development practices lack the scientific underpinnings and rigorous controls needed to produce high-quality, secure products at acceptable cost. Commonly used software engineering practices permit dangerous error, which enable hundreds of attack programs to compromise millions of computers every year. "Vulnerabilities in software that are introduced by mistake or poor practices are a serious problem today. In the future, the Nation may face even more challenging problems as adversaries—both foreign and domestic—become increasingly sophisticated in their ability to insert malicious code into critical software" [U.S. President's Information Technology Advisory Committee (US PITAC) 2005, 9]. The report also recognizes that software is an underpinning of all critical infrastructure. Software is an essential element of systems that comprise critical infrastructure sectors: public health, banking and finance, agriculture and food, water, emergency services, telecommunications, energy, transportation, chemicals and hazardous materials, postal services, and the defense industrial base. The vulnerabilities that permit adversaries to insert malicious code into these critical systems must be minimized.

In May 2004, the U.S. General Accounting Office (now the Government Accountability Office) issued a report to the U.S. Congress on the vulnerability of software in the acquisition process [GAO–04–678]. In particular, the report found that "malicious code is a threat that is not adequately addressed in current acquisition policies and security procedures." The main thrust of the review revolved around risks inherent in the development of weapons system software by foreign sources. However, the findings and recommendations covered all purchased products or systems that are software intensive. Two recommendations are quoted below:

- "Require program managers, working with software/application security experts, acquisition personnel, and other organizations as necessary to specifically define software security requirements, including those for identifying and managing software suppliers. These requirements should then be communicated as part of the prime development contract, to be used as part of the criteria to select software suppliers."
- "Based on defined software security requirements, require program managers to collect and maintain information on software suppliers, including software from foreign suppliers. This information

---

19 Timely, reliable access to data and information services for authorized users [CNSSI No. 4009].

20 Quality of an information system reflecting the logical correctness and reliability of the operating system; the logical completeness of the hardware and software implementing the protection mechanisms; and the consistency of the data structures and occurrence of the stored data. Note that, in a formal security mode, integrity is interpreted more narrowly to mean protection against unauthorized modification or destruction of information [CNSSI No. 4009].

21 Security measure designed to establish the validity of a transmission, message, or originator, or means of verifying an individual's authorization to receive specific categories of information [CNSSI No. 4009].

22 Assurance that information is not disclosed to unauthorized individuals, processes, or devices [CNSSI No. 4009].

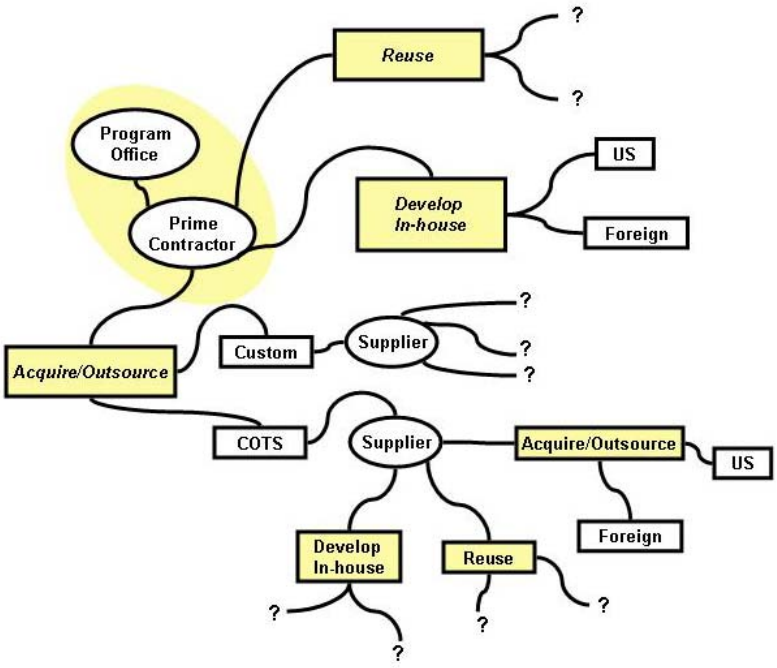
23 Assurance that the sender of data is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the data [CNSSI No. 4009].

should be evaluated periodically to assess changes in the status of suppliers and adjustments to program security requirements."

A report to the President of the United States in February 1999 [US PITAC 1999, 27–31] identified software development practices failing to produce high-quality, reliable, and secure software. An increasing number of software intensive systems fail because of errors imposed into the software.

In September 2005, the Federal Acquisition Regulation was updated to include the Information Technology Security provisions of the Federal Information Security Management Act specifically noting that the "supply chain introduces risks to American society that relies on the Federal Government for essential information and services." Because the majority of network and system exploits target software applications, it is important that acquisitions and procurements factor in risks posed by the entire supply chain,[24] not just the end seller. Figure C–1 shows some of the potential paths software can take before it is acquired.

Figure C–1. Potential Software Supply Chain Paths[25]



[24] Though the program/project manager (PM) is responsible for ensuring the acquisition process is followed, it is important that the PM convey what is required at each level of the process (expectations), not just at the beginning. The supply chain consists of (but is not exclusive to) the following: Software Acquirers/Buyers in industry and government, IA personnel supporting acquisition managers (if available), decision makers for software acquisitions, and the prime contractors and the subs in their supply chain, software suppliers, Program/Project Managers, contracts personnel, and the requirements personnel.

[25] From [DACS-Walker] Walker, E. (2005, July). Software Development Security: A Risk Management Perspective. In *The DOD* Software Tech News—Secure Software Engineering. Vol(8)No(2). Rome, NY: Data & Analysis Center for Software.

PAGE INTENTIONALLY LEFT BLANK

Appendix D. Software Due Diligence Questionnaires (Examples)

This appendix contains software assurance due diligence questionnaire examples for several types of software. Acquirers may use a questionnaire as a means for gathering relevant information to support decisionmaking versus being a decisionmaking tool. **When using the questionnaires, acquirers should change the questions to suit their particular acquisitions since** (1) not all questions are applicable and (2) other acquisition-specific questions may be more appropriate. ***The questionnaire is intended to solicit information from the suppliers. It is not a checklist, nor is it a complete listing of all possible SwA/security concerns.*** Expertise in software, acquisition, and information assurance—as well as common sense —is critical to smart decisions regarding the acquisition of assured software.

Questions should be reviewed prior to submission, and responses assessed, by knowledgeable SwA subject matter experts or other appropriate functional experts. In addition, when using the questionnaire as a tool, acquisition officials should ensure that they solicit evidence to support supplier responses when applicable. The table below could aid in the trade-off analysis.

Categories	Priority	Product Score			Weighted Average			
		Product 1 Score (0–4)	Product 2 Score (0–4)	Product 3 Score (0–4)	Product 1	Product 2	Product 3	Average
					1.6	2.1	2.5	10.6
Software History & Licensing	5	2	2	3	8.4	11.4	13.0	10.9
Development Process Management	3	1	3	4	9.0	9.0	9.0	9.0
Software Security Awareness and Training	3	2	2	3	9.0	9.0	9.0	9.0
Concept and Planning	2				0.0	0.0	0.0	0.0
Architecture and Design	3				0.0	0.0	0.0	0.0
Software Development	3				0.0	0.0	0.0	0.0
Testing	4				0.0	0.0	0.0	0.0
Software Change Management	4				0.0	0.0	0.0	0.0
Built-in Software Defenses	2				0.0	0.0	0.0	0.0
Assurance Claims and Evidence	5				0.0	0.0	0.0	0.0
Timeliness of Vulnerability Mitigation	3				0.0	0.0	0.0	0.0
Security Track Record	4				0.0	0.0	0.0	0.0

PAGE INTENTIONALLY LEFT BLANK

Table D–1. COTS Software Questionnaire

Commercial-off-the-shelf (COTS) software is a term for software products that are ready-made and are readily available for purchase in the commercial market. The table below lists questions to consider asking during a COTS software evaluation.

Related suggestions for products in general can be found in National Institute of Standards and Technology Special Publications 800–23, *Guidelines to Federal Organizations on Security Assurance and Acquisition/Use of Tested/Evaluated Products*, and 800–36, *Guide to Selecting IT Security Products*. Supplemental evaluation for information assurance (IA)-enabled (as defined within National Security Telecommunications and Information Systems Security Policy No. 11), biometric (as defined within Federal Information Processing Standard [FIPS] 201–1), electronic authentication (as defined within Office of Management and Budget M04–04), and cryptographic (as defined by FIPS 140–2) software may exist based on the particular application.

#	Question	Evidence	Priority (1-5)	Score (1-4)
Software History and Licensing				
1	Can the pedigree of the software be established? Briefly explain what is known of the people and processes that created the software.		2	1
2	Explain the change management procedure that identifies the type and extent of changes conducted on the software throughout its lifecycle.		2	2
3	Is there a clear chain of licensing from original author to latest modifier. Describe the chain of licensing.		3	3
4	What assurances are provided that the licensed software does not infringe upon any copyright or patent? Explain		4	4
5	Does your company have corporate policies and management controls in place to ensure that only corporate-approved (licensed and vetted) software components are used during the development process? Provide a brief explanation. Will the supplier indemnify the Acquirer from these issues in the license agreement? Provide a brief explanation.		4	
Development Process Management				
6	What are the processes (e.g., ISO 9000, CMMi), methods, tools (e.g., IDEs, compilers) techniques, etc. used to produce and transform the software (brief summary response)?		3	
7	What security measurement practices and data does your company use to assist product planning?		3	
8	Is software assurance considered in all phases of development? Explain		1	
Software Security Training and Awareness				
9	Describe the training your company offers related to defining security requirements, secure architecture and design, secure coding practices, and security testing.		2	
10	Do you have developers that possess software security related certifications (e.g., the SANS secure coding certifications)?		2	
11	Describe the company's policy and process for professional certifications and ensuring certifications are valid and up-to-date.		3	

#	Question	Evidence	Priority (1-5)	Score (1-4)
Concept and Planning				
12	Are there some requirements for security that are "structured" as part of general releasability of a product and others that are "as needed" or "custom" for a particular release?		1	
13	Are there some requirements for quality that are "structured" as part of general releasability of a product and others that are "as needed" or "custom" for a particular release?		1	
14	What process is utilized by your company to prioritize security-related enhancement requests?		3	
Architecture and Design				
15	What threat assumptions were made, if any, when designing protections for the software and information assets processed?		2	
16	What security design and security architecture documents are prepared as part of the SDLC process?		1	
17	How are design documents for completed software applications archived?		2	
Software Development				
18	What are/were the languages and non-developmental components used to produce the software (brief summary response)?		2	
19	What secure development standards and/or guidelines are provided to developers?		3	
20	Are tools provided to help developers verify that the software they have produced software that is minimized of weaknesses that could lead to exploitable vulnerabilities? What is the breadth of common software weaknesses covered (e.g., specific CWEs)?		3	
21	In preparation for release, are undocumented functions in the software disabled, test/debug code removed, and source code comments sanitized?		2	
Built-in Software Defenses				
22	Does the software validate (e.g., filter with white listing) inputs from untrusted sources before being used?		3	
23	Has the software been designed to execute within a constrained execution environment (e.g., virtual machine, sandbox, chroot jail, single-purpose pseudo-user) and is it designed to isolate and minimize the extent of damage possible by a successful attack?		2	
24	Does the documentation explain how to install, configure, and/or use it securely? Does it identify options that should not normally be used because they create security weaknesses?		3	
25	Where applicable, does the program use run-time infrastructure defenses (such as address space randomization, stack overflow protection, preventing execution from data memory, and taint checking)?		3	
26	How do you minimize the threat of reverse engineering of binaries? Are source code obfuscation techniques used? Are legal agreements in place to protect against potential liablities of nonsecure software?		3	

#	Question	Evidence	Priority (1-5)	Score (1-4)
Component Assembly				
27	What security criteria, if any, are considered when selecting third-party suppliers?		3	
28	Is the software required to conform to coding or API standards in any way? Explain.		1	
Testing				
29	What types of functional tests are/were performed on the software during its development (e.g., spot checking, component-level testing, integrated testing)?		2	
30	Who and when are security tests performed on the product? Are tests performed by an internal test team, by an independent third party, or by both?		1	
31	What degree of code coverage does your testing provide?		2	
32	Are misuse test cases included to exercise potential abuse scenarios of the software?		1	
33	Are security-specific regression tests performed during the development process? If yes, how frequently are the tests performed?		3	
34	What release criteria does your company have for its products with regard to security?		2	
Software Manufacture and Packaging				
35	What security measures are in place for the software packaging facility?		3	
36	What controls are in place to ensure that only the accepted/released software is placed on media for distribution?		3	
37	How is the software packaged (e.g. Zipped , Linux RPM etc) and distributed?		2	
38	How is the integrity of downloaded software (if an option) protected?		2	
39	For the released software "object", how many "files" does it consist of? How are they related?		2	
Installation				
40	Is a validation test suite or diagnostic available to validate that the application software is operating correctly and in a secure configuration following installation? If so, how is it obtained?		3	
41	What training programs, if any, are available or provided through the supplier for the software? Do you offer certification programs for software integrators? Do you offer training materials, books, computer-based training, online educational forums, or sponsor conferences related to the software?		4	
Assurance Claims and Evidence				
42	How has the software been measured/assessed for its resistance to identified, relevant attack patterns? Are Common Vulnerabilities & Exposures (CVE®) or Common Weakness Enumerations (CWEs) used? How have the findings been mitigated?		3	

#	Question	Evidence	Priority (1-5)	Score (1-4)
43	Has the software been evaluated against the Common Criteria, FIPS 140-2, or other formal evaluation process? If the CC, what evaluation assurance level (EAL) was achieved? If the product claims conformance to a protection profile, which one(s)? Are the security target and evaluation report available?		1	
44	Are static or dynamic software security analysis tools used to identify weaknesses in the software that can lead to exploitable vulnerabilities? If yes, which tools are used? What classes of weaknesses are covered? When in the SDLC are these scans performed? Are SwA experts involved in the analysis of the scan results?		1	
45	Does the software contain third-party developed components? If yes, are those components scanned by a static code analysis tool?		3	
46	Has the product undergone any penetration testing? When? By whom? Are the test reports available under a nondisclosure agreement? How have the findings been mitigated?		1	
47	Are there current publicly-known vulnerabilities in the software (e.g., an unrepaired CWE entry)?		2	
Support				
48	Is there a Support Lifecycle Policy within the organization for the software in question? Does it outline and establish a consistent and predictable support timeline?			
49	How will patches and/or Service Packs be distributed to the Acquirer?		3	
50	What services does the help desk, support center, or (if applicable) online support system offer?		3	
Software Change Management				
51	How extensively are patches and Service Packs tested before they are released?		2	
52	Can patches and Service Packs be uninstalled? Are the procedures for uninstalling a patch or Service Pack automated or manual?		2	
53	Will configuration changes (if needed for the installation to be completed) be reset to what was there before the patch was applied in cases where the change was not made explicitly to close a vulnerability?		2	
54	How are reports of defects, vulnerabilities, and security incidents involving the software collected, tracked, and prioritized?		2	
55	Do you determine relative severity of defects and does that drive other things like how fast you fix issues?		2	
56	What are your policies and practices for reviewing design and architecture security impacts in relation to deploying patches?		2	
57	Are your version control and configuration management policies and procedures the same throughout your entire organization and for all your products? How are they enforced? Are third-party developers contractually required to follow these policies and procedures?		2	

#	Question	Evidence	Priority (1-5)	Score (1-4)
58	What policies and processes does your company use to verify that software components do not contain unintended, "dead," or malicious code? What tools are used?		1	
59	How is the software provenance verified (e.g. any checksums or signatures)?		2	
Timeliness of Vulnerability Mitigation				
60	Does your company have a vulnerability management and reporting policy? Is it available for review?		1	
61	Does your company publish a security section on its Web site? If so, do security researchers have the ability to report security issues?		3	
Security "Track Record"				
62	Does your company have an executive-level officer responsible for the security of your company's software products and/or processes?		3	
Financial History and Status				
63	Has your company ever filed for Recompany under U.S. Code Chapter 11? If so, please provide dates for each incident and describe the outcome.		3	
64	Does your company have policies and procedures for periodically reviewing the financial health of the third-party entities with which it contracts for software development, maintenance, or support services?		3	
65	Does your company have established policies and procedures for dealing with the contractual obligations of third-party developers that go out of business?		2	

PAGE INTENTIONALLY LEFT BLANK

Table D–2. Open-source Software Questionnaire

Open-source software is computer software whose source code is available under a copyright license that permits users to study, change, and improve the software, and to redistribute it in modified or unmodified form. The table lists questions to consider asking during an open-source software evaluation.

#	Question	Evidence	Priority (1-5)	Score (1-4)
Software History and Licencing				
1	Can the software pedigree be established? What is known of the people and processes that created the software (brief summary response)?		4	1
2	Is the software in question original source or a modified version?		1	2
3	What type of license(s) are available for the open source software? Is it compatible with other software components in use?		5	3
4	Has the software been reviewed to confirm that it does not infringe upon any copyright or patent?		4	4
5	How long has the software source been available?		3	2
6	Is there an active user community providing peer review and actively evolving the software?		1	1
7	Does software have a positive reputation? Are there reviews that recommend it?		1	1
Built-in Software Defenses				
8	Does the software validate (e.g., filter with whitelisting) inputs from untrusted sources before being used?		3	
9	Has the software been designed to execute within a constrained execution environment (e.g., virtual machine, sandbox, chroot jail, single-purpose pseudo-user) and is it designed to isolate and minimize the extent of damage possible by a successful attack?		2	
10	Does the documentation explain how to install, configure, and/or use it securely? Does it identify options that should not normally be used because they create security weaknesses?		3	
11	Where applicable, does the program use run-time infrastructure defenses (such as address space randomization, stack overflow protection, preventing execution from data memory, and taint checking)?		3	
Assurance Claims and Evidence				
12	Has the software been measured/assessed for its resistance to identified relevant attack patterns?		3	
13	Has security testing been performed on the software with posted results?		2	
14	Has the software been evaluated against the Common Criteria, FIPS 140-2, or other formal evaluation process? If the CC, what evaluation assurance level (EAL) was achieved? Are the security target and evaluation report available?		1	
15	Have static source code analysis tools been used to identify weaknesses that could lead to exploitable vulnerabilities in the software? If yes, what tools are used? What classes of weaknesses were covered?		1	

#	Question	Evidence	Priority (1-5)	Score (1-4)
16	Are there current publicly-known vulnerabilities in the software (e.g., an unrepaired CWE entry)?		2	
Software Change Management				
17	Which open-source repository is used?		2	
18	How are patches distributed?		3	
19	Can patches be uninstalled?		2	
20	How are reports of defects, vulnerabilities, and security incidents involving the software reported and resolved? How rapidly have they been resolved in the past?		2	
21	How frequently are major versions of the software released?		1	
22	How is the software provenance verified (e.g. any checksums or signatures)?		2	

Table D–3. Custom Software Questionnaire

Custom software is software developed either for a specific organization or function that differs from other already available software. It is generally not targeted to the mass market but rather is usually created for specific companies, business entities, and organizations.

#	Question	Evidence	Priority (1-5)	Score (1-4)
Software History and Licenses				
1	Can the software pedigree be established? What is known of the people and processes that created the software (brief summary response)?		2	1
2	Is there a change management procedure or document that will identify the type and extent of changes conducted on the software throughout its lifecycle?		2	2
3	What assurances are provided that the software does not infringe upon any copyright or patent?		3	3
4	Does your company have corporate policies and management controls in place to ensure that only corporate-approved (licensed and vetted) software components are used during the development process? Will the supplier indemnify the Acquirer from these issues in the license agreement?		4	
Development Process Management				
6	What are the processes (e.g., ISO 9000, CMMi), methods, tools (e.g., IDEs, compilers) techniques, etc. used to produce and transform the software (brief summary response)?		1	
6	What security measurement practices and data does your company use to assist project planning?		1	
7	Is software assurance considered in all phases of development?			
8	How is software risk managed? Are anticipated threats identified, assessed, and prioritized?		2	
Software Security Training and Awareness				
9	What training does your company offer related to defining security requirements, secure architecture and design, secure coding practices, and security testing?		2	
10	Do you have developers that possess software security related certifications (e.g., the SANS secure coding certifications)?		2	
11	Describe the company's policy and process for professional certifications and for ensuring certifications are valid and up-to-date.		3	
Concept and Planning				
12	Are there some requirements for security that are "structured" as part of general releasability of an application and others that are "as needed" or "custom" for a particular release?		1	
13	Are there some requirements for quality that are "structured" as part of general releasability of an application and others that are "as needed" or "custom" for a particular release?		1	
14	What review processes are implemented to ensure that nonfunctional requirements are unambiguous, traceabl,e and testable throughout the entire SDLC?		1	

#	Question	Evidence	Priority (1-5)	Score (1-4)
15	Are security requirements developed independently of the rest of the requirements engineering activities, or are they integrated into the mainstream requirements activities?		3	
16	Are misuse/abuse cases derived from the application requirements? Are relevant attack patterns used to identify and document potential threats?		3	
17	What tool(s) does your company use for requirements management?		3	
18	If an agile development method is used, how formally are requirements documented?		3	
Architecture and Design				
19	What threat modeling process, if any, is used when designing the software protections?		2	
20	What analysis, design, and construction tools are used by your software design teams?		2	
21	What security design and security architecture documents are prepared as part of the SDLC process? How are they maintained? Are they available to\\for review?		2	
Software Development				
22	What languages and non-developmental components are used to produce the software (brief summary response)?		2	
23	Does your company have formal coding standards for each language in use? If yes, how are they enforced? How often are these standards and practices reviewed and revised?		2	
24	Does the software development plan include security peer reviews?		1	
25	Are tools provided to help developers verify that the software they have produced software that is minimized of weaknesses that could lead to exploitable vulnerabilities? What is the breadth of common software weaknesses covered (e.g., specific CWEs)?		3	
26	Does your organization incorporate security risk management activities as part of your software development methodology? If yes, please provide a copy of the documentation of this methodology or provide information on how to obtain it from a publicly accessible source.		2	
27	Does your organization establish contractually binding agreements with their own developers and/or with their third-party developers regarding the ownership and/or licensing of intellectual property?		3	
28	Does the software use closed-source Application Programming Interfaces (APIs) that have undocumented functions?		4	
29	Are there contractual recourses that the organization can take if a third-party developer delivers software that contains malicious code?		4	

#	Question	Evidence	Priority (1-5)	Score (1-4)
30	Does the organization ever perform site inspections/policy compliance audits of its U.S. development facilities? Of its non-U.S. facilities? Of the facilities of its third-party developers? If yes, how often do these inspections/audits occur? Are they periodic or triggered by events (or both)? If triggered by events, provide examples of "trigger" events.		2	
31	In preparation for release, are undocumented functions in the software disabled, test/debug code removed, and source code comments sanitized?		2	
Built-in Software Defenses				
32	Does the software validate (e.g., filter with whitelisting) inputs from untrusted sources before being used?		3	
33	Has the software been designed to execute within a constrained execution environment (e.g., virtual machine, sandbox, chroot jail, single-purpose pseudo-user) and is it designed to isolate and minimize the extent of damage possible by a successful attack?		2	
34	How does the software's exception handling mechanism prevent faults from leaving the software, its resources, and its data (in memory and on disk) in a vulnerable state?		2	
35	Does the exception-handling mechanism provide more than one option for responding to a fault? If so, can the exception-handling options be configured by the administrator or overridden?		3	
36	Does the documentation explain how to install, configure, and/or use it securely? Does it identify options that should not normally be used because they create security weaknesses?		3	
37	Where applicable, does the program use run-time infrastructure defenses (such as address space randomization, stack overflow protection, preventing execution from data memory, and taint checking)?		3	
38	How do you minimize the threat of reverse engineering of binaries? Are source code obfuscation techniques used? Are legal agreements in place to protect against?		3	
Component Assembly				
39	Does the software have any security critical dependencies or need additional controls from other software (e.g., operating system, directory service, applications), firmware, or hardware? If yes, please describe.		2	
40	Is the software regularized to conform to coding or API standards in any way?		1	
41	Is delivery of demonstrably secure software a contractual requirement for third-party developed software? If yes, what criteria are used to operationally define "secure software"?		2	
42	Are additional risk management measures in place in the software's design to mitigate risks posed by use of third-party components?		2	
Testing				
43	What types of functional tests are performed on the software during its development (e.g., spot checking, component-level testing, security testing, integrated testing)?		2	

#	Question	Evidence	Priority (1-5)	Score (1-4)
44	Does your company's defect classification schemes include security categories? During testing what proportion of identified defects relate to security?		3	
45	What degree of code coverage does your testing provide?		2	
46	Are misuse test cases included to exercise potential abuse scenarios of the software?		2	
47	Are security-specific regression tests performed during the development process? If yes, how frequently are the tests performed?		3	
48	When does security testing occur during the SDLC (e.g., unit level, subsystem, system, certification and accreditation)?		1	
Installation				
49	If you are responsible for installing the software, is this done by your organization or through third-party consultants?		4	
50	Is a validation test suite or diagnostic available to validate that the application software is operating correctly and in a secure configuration following installation?		3	
51	What training/documentation is available for software installation and maintenance?		2	
Assurance Claims and Evidence				
52	Does your company develop security measurement objectives for phases of the SDLC? Has your company identified specific statistical and/or qualitative analytical techniques for measuring attainment of security measures?		2	
53	Has the software been measured/assessed for its resistance to identified relevant attack patterns? Are Common Vulnerabilities & Exposures (CVE®) or Common Weakness Enumeration (CWEs) used? How have the findings been mitigated?		2	
54	Are static or dynamic software security analysis tools used to identify the weaknesses that can lead to exploitable vulnerabilities in the software? If yes, which tools are used? What classes of weaknesses are covered? When in the SDLC are these scans performed? Are SwA experts involved in the analysis of the scan results?		1	
55	Does the software contain third-party developed components? If yes, are those components scanned by a static code analysis tool?		3	
56	Has the software undergone any penetration testing? When? By whom? Are the test reports available under a nondisclosure agreement? How have the findings been mitigated?		1	
57	Are there current publicly-known vulnerabilities in the software (e.g., an unrepaired CWE entry)?		2	
58	How is the assurance of software produced by third-party developers assessed?		2	
Support				
59	Are multiple tiers of support contracts available? If yes, please describe the support plans available.		3	

#	Question	Evidence	Priority (1-5)	Score (1-4)
60	Is there a Support Lifecycle Policy for the software in question? Does it outline and establish a consistent and predictable support timeline?		3	
61	How will patches and/or Service Packs be distributed to the Acquirer?		2	
62	How are trouble tickets submitted? How are support issues, specifically those that security related, escalated?		3	
63	Are help desk or support center personnel internal company resources or are these services outsourced to third parties?		3	
64	If help desk or support center services are outsourced to third parties, are they located in foreign countries?		3	
Software Change Management				
65	What are your policies and procedures for maintaining development documents, including requirements, design and architecture documents, source code, binaries, and user documentation?		1	
66	Are your version control and configuration management policies and procedures the same throughout your entire organization? How are they enforced? Are third-party developers contractually required to follow these policies and procedures?		2	
67	Are configuration/change controls in place to prevent unauthorized modifications or additions to source code and related documentation? Do these controls detect and report unexpected modifications/additions to source code? Do they aid in rolling back an affected artifact to a pre-modified version?		1	
68	Are there any undocumented features present not intended for use by end users, but available for use by the supplier for technical support and development?		4	
69	How are reports of defects, vulnerabilities, and security incidents involving the software collected, tracked, and prioritized?		2	
70	What are your policies and practices for reviewing design and architecture security impacts in relation to deploying patches?		2	
71	Does your organization have policies and procedures in place to monitor and audit the transmission of its technology-related intellectual property to third parties, and to prevent unauthorized transmission of that intellectual property?		3	
72	What policies and processes does your organization use to verify that software components do not contain unintended, “dead,” or malicious code? What tools are used?		2	
73	Is a process utilized by your company that can be used for documenting and analyzing the security aspects of fielded systems and for steering future improvements and modifications to those systems?		3	
Timeliness of Vulnerability Mitigation				
74	Does your company have a vulnerability management and reporting policy? Is it available for review?		1	
Individual Malicious Behavior				

#	Question	Evidence	Priority (1-5)	Score (1-4)
75	Does your company perform background checks on members of the software development team? If so, are there any additional "vetting" checks done on people who work on critical application components, such as security? Explain.		2	
76	Does your company have formally defined security policies associated with clearly defined roles and responsibilities for personnel working within the software development life cycle, along with management oversight and enforcement? Explain.		2	
77	What training is available to your development staff to help them identify malicious behavior? Are there formal policies for reporting malicious behavior?		3	
78	Has civil legal action ever been filed against your company for delivering or failing to correct defective software? Explain.		3	
Organizational History				
79	Please summarize your company's history of ownership, acquisitions, and mergers (both those performed by your company and those to which your company was subjected).		3	
80	Please provide a list of the names and dates of service of the following executive officers: • Chairman of the Board (COB) • Chief Executive Officer (CEO) • President (if different from CEO) • Vice President(s) • Chief Financial Officer (CFO)		3	
81	How many employees does your company have: • In the U.S.? • Worldwide?		4	
Foreign Interests and Influences				
82	Is the controlling share (51+%) of your company owned by a non-U.S. entity? If so, please complete Standard Form 328, Certificate Pertaining to Foreign Interests.		3	
83	Is your company an entity of a larger "parent" company? If yes" does that "parent" company include any subsidiaries or other sub-entities that are 51+% foreign owned? If so, please identify those subsidiaries/sub-entities.		3	
84	Please provide company names of all third-party entities with whom your firm contracts software development, maintenance, or support services related to this procurement.		3	
Financial History and Status				
85	Has your company ever filed for Recompany under U.S. Code Chapter 11? If so, please provide dates for each incident and describe the outcome.		3	
86	What are your company's policies and procedures for periodically reviewing the financial health of the third-party entities with which it contracts for software development, maintenance, or support services?		3	
87	What are your company's policies and procedures for dealing with the contractual obligations of third party developers that go out of business?		2	

Table D–4. GOTS Software Questionnaire

Government-off-the-shelf (GOTS) software is a term for a software product that is typically developed by the technical staff of the government agency for which it is created. It is sometimes developed by an external entity, but with funding and specification from the agency. GOTS software may be reused by other agencies but should be analyzed to manage the risk of its reuse. The table presents sample questions to ask during a GOTS software evaluation.

#	Questions	Evidence	Priority (1-5)	Score (1-4)
Software History and Licensing				
1	Can the software pedigree be established? What is known of the people and processes that created the software (brief summary response)?		2	1
2	Is there a change management procedure or document that will identify the type and extent of changes conducted on the software throughout its lifecycle?		2	2
3	What assurances are provided that the software does not infringe upon any copyright or patent?		3	3
4	Are licensed software components still valid for the intended use?		4	4
5	Is the level of security where the software was developed the same as where the software will operate?		2	
Development Process Management				
6	What were the processes (e.g., ISO 9000, CMMi), methods, tools (e.g., IDEs, compilers) techniques, etc. used to produce and transform the software (brief summary response)?		2	
7	Was software assurance considered in all phases of development?		1	
8	How is software risk managed?		2	
Concept and Planning				
9	Were security and quality requirements included in the requirements analysis process?		2	
10	If an agile development method was used, how formally are requirements documented?		3	
Architecture and Design				
11	What threat assumptions, if any, were made when software was originally designed? Is the threat model documented and available?		2	
12	Are design documents for the software archived and available?		2	
13	What security design and security architecture documents are available?		1	
14	How are confidentiality, availability, and integrity addressed in the software design?		1	
15	Are software interfaces described in published documentation?		2	
Software Development				
16	What were the languages and non-developmental components used to produce the software (brief summary response)?		2	
17	Were formal coding standards for the application were used during the software development life cycle?		1	

#	Questions	Evidence	Priority (1-5)	Score (1-4)
18	Are configuration/change controls in place to prevent unauthorized modifications or additions to source code and related documentation? Do these controls detect and report unexpected modifications/additions to source code? Do they aid in rolling back an affected artifact to a pre-modified version?		3	
19	Does the software's exception-handling mechanism prevent all faults from leaving the software, its resources, and its data (in memory and on disk) in a vulnerable state? Does the exception-handling mechanism provide more than one option for responding to a fault? If so, can the exception-handling options be configured by the administrator or overridden?		3	
20	Does the available version of the software have undocumented functions disabled, test/debug code removed, and source code comments sanitized?		2	
Built-in Software Defenses				
21	Does the software validate (e.g., filter with white lsting) inputs from untrusted sources before being used?		3	
22	Has the software been designed to execute within a constrained execution environment (e.g., virtual machine, sandbox, chroot jail, single-purpose pseudo-user) and is it designed to isolate and minimize the extent of damage possible by a successful attack?		2	
23	Does the software default to requiring the administrator (or user of a single-user software package) to expressly approve the automatic installation of patches/upgrades, downloading of files, execution of plug-ins or other "helper" applications, and downloading and execution of mobile code?		4	
24	Does the documentation explain how to install, configure, and/or use it securely? Does it identify options that should not normally be used because they create security weaknesses?		3	
25	Where applicable, does the program use run-time infrastructure defenses (such as address space randomization, stack overflow protection, preventing execution from data memory, and taint checking)?		3	
26	How is the threat of reverse engineering of binaries minimized? Are source code obfuscation techniques used?		3	
Component Assembly				
27	Does the software include content produced by suppliers other than the primary developer? If so, who?		2	
28	Is the software regularised to conform to coding or API standards in any way?		2	
29	What are the policies and procedures for verifying the quality and security of non-developmental components used?		2	
Testing				
30	What types of functional tests are/were performed on the software (e.g., spot checking, component-level testing, security testing, integrated testing)?		2	
31	Were misuse test cases included to exercise potential abuse scenarios of the software?		1	
32	What degree of code coverage do the available test cases provide?		2	
33	Are regression test scripts available?		3	

#	Questions	Evidence	Priority (1-5)	Score (1-4)
Installation				
34	Are installation instructions available?		4	
35	Are instructions available to securely configure the application?		3	
36	Is a validation test suite or diagnostic available to validate that the application software is operating correctly and in a secure configuration following installation?		3	
Assurance Claims and Evidence				
37	Has the software been measured/assessed for its resistance to identified relevant attack patterns?		3	
38	Were static software security analysis tools used to identify the weaknesses that can lead to exploitable vulnerabilities in the software? If yes, what tools were used? What classes of weaknesses were covered?		1	
39	Has the software undergone any penetration testing? When? By whom? Are the test reports available under a nondisclosure agreement? How have the findings been mitigated?		1	
40	Are there current publicly-known vulnerabilities in the software (e.g., an unrepaired CWE entry)?		2	
41	Has the software been certified and accredited? When? By whom?		2	
Support				
42	Is there a Support Lifecycle Policy for the software in question? Does it outline and establish a consistent and predictable support timeline?		3	
43	How are patches and/or Service Packs distributed?		3	
44	How are support issues resolved?		3	
Software Change Management				
45	How extensively are patches and Service Packs tested before they are released?		2	
46	Can patches and Service Packs be uninstalled? Are the procedures for uninstalling a patch or Service Pack automated or manual?		2	
47	Will configuration changes (if needed for the installation to be completed) be reset to what was there before the patch was applied in cases where the change was not made explicitly to close a vulnerability?		2	
48	How are reports of defects, vulnerabilities, and security incidents involving the software reported and resolved? How rapidly have they been resolved in the past?		2	
49	What are the policies and practices for reviewing design and architecture security impacts in relation to deploying patches?			
50	What policies and processes were used to verify that software components do not contain unintended or , “dead” code? What tools were used?		1	
51	How can the integrity of update/patches be verified to ensure that they are correct and unaltered (e.g., comparisons of cryptographic hashes)?		2	
52	How is the software provenance verified (e.g. any checksums or signatures)?		2	

PAGE INTENTIONALLY LEFT BLANK

Table D–5. Hosted Applications

Increasingly, software is executed and maintained by someone other than the acquirer and provided as a service to them. Application service providers host the servers that support the applications in a data center and provide different levels of service, including security-related services. Users remotely access the software. Suppliers should also ask software development questions for the appropriate software type to augment the questions below.

#	Questions	Evidence	Priority (1-5)	Score (1-4)
Service Confidentiality Policies				
1	What are your customer confidentiality policies? How are they enforced?		1	1
2	What are your customer privacy policies? How are they enforced?		1	2
3	What are the policies and procedures used to protect sensitive information from unauthorized access? How are the policies enforced?		1	3
4	What are the set of controls to ensure separation of data and security information between different customers that are physically located in the same data center? On the same host server?		1	4
Operating Environment for Services				
5	Who configures and deploys the servers? Are the configuration procedures available for review, including documentation for all registry settings?		2	
6	What are your policies and procedures for hardening servers?		2	
7	What are your data backup policies and procedures? How frequently are your backup procedures verified?		2	
8	What are the procedures for evaluating any vendor security alerts and installing patches and Service Packs?		3	
9	How are vendor patches and Services Packs applied?		3	
10	Is testing done after changes are made to servers? What are your rollback procedures in the event of problems resulting from installing a patch or Service Pack?		2	
11	What are the agents or scripts executing on servers of hosted applications? Are there procedures for reviewing the security of these scripts or agents?		2	
12	What are the procedures and policies used to approve, grant, monitor and revoke access to the servers? Are audit logs maintained?		2	
13	What are your procedures and policies for handling and destroying sensitive data on electronic and printed media?		2	
14	Do you have a formal disaster recovery plan? What actions will be taken to recover from a disaster? Are warm or hot backups available?		2	
15	What are the procedures used to approve, grant, monitor, and revoke file permissions for production data and executable code?		2	

#	Questions	Evidence	Priority (1-5)	Score (1-4)
16	Is two-factor authentication used for administrative control of all security devices and critical information systems?		2	
Security Service Available				
17	What are the types of information security services you provide?		3	
18	How are virus prevention, detection, correction, and updates handled for the products?		2	
19	What type of firewalls (or application gateways) do you use? How are they monitored/managed?		1	
20	What type of Intrusion Detection System/Intrusion Protection Systems (IDS/IPS) do you use? How are they monitored/managed?		1	
21	Is your system and network architecture based on a high availability design that includes redundant firewalls, routers, switches and IDS, and load balanced or clustered servers?		2	
Security Monitoring				
22	Do you perform regular reviews of system and network logs for security issues?		2	
23	Do you have an automated security event management system?		3	
24	What are your procedures for intrusion detection, incident response, and incident investigation/escalation?		1	
25	Will you provide on-site support 24x7 to resolve security incidents?		2	
26	Do you provide write-once technology for storing audit trails and security logs?		3	
27	How do you control physical and electronic access to the log files? Are log files consolidated to single servers?		3	
28	Do you provide security performance measures to the customer at regular intervals?		2	
Assurance Claims and Evidence				
29	Has functional security testing been performed on the services?		1	
30	Do you perform penetration testing of the service? If yes, how frequently are penetration tests performed? Are the tests performed by internal resources or by a third party?		2	
31	Do you provide automated vulnerability testing of the service? If yes, how frequently are the tests performed? Are the tests performed by internal resources or by a third party?		2	

Appendix E. Other Examples of Due Diligence Questionnaires

E.1 Health Community Questionnaires

Medical Devices. The Healthcare Information and Management Systems Society (HIMSS) Information Technology (IT) Systems Security Workgroup has developed a medical devices questionnaire. See <http://www.himss.org/ASP/index.asp> for more on the society. The HIMSS IT Systems Security Workgroup membership consists of representatives from the medical device industry, medical device standards bodies, healthcare organizations, academia, the U.S. Veteran's Health Administration, and the U.S. Federal Drug Administration. See <http://www.himss.org/content/files/MDSWorkgroupRoster.pdf> for a membership list.

Pharmaceutical Operations. The Parental Drug Association (PDA) has developed a process for auditing suppliers that provide computer products and services for regulated pharmaceutical operations. This process is published as PDA Technical Report 32 (October 2004), *Auditing of Suppliers Providing Computer Products and Services for Regulated Pharmaceutical Operations Release 2.0, Volume 58 Number 5*. See <https://store.pda.org/bookstore/ProductDetails.aspx?productabbreviation=01032>.

E.2 U.S. Cyber Security Consequences Unit Cyber Security Checklist

The U.S. Cyber Security Consequences Cyber Unit, a nonprofit research institute, developed a checklist containing a set of questions to help organizations reduce their vulnerability to cyber attack. See <http://www.selfstorage.org/PDF/US-CCU-Cyber-SecurityCheckList2007.pdf>.

This list includes sets of questions organized around six areas. In particular Area 6, *Software Supply Vulnerabilities*, includes questions regarding secure procedures for developing new software, developing security features into new software, performing security testing of new software, and establishing appropriate relationships and managing ongoing relationships with vendors. The area that contains questions on relationships with vendors deals, in part, with supply chain and software pedigree issues.

E.3 Software Assurance Considerations for Suppliers Using Capability Maturity Models

"System and software assurance focuses on the management of risk and assurance of safety, security, and dependability within the context of system and software life cycles" [ISO/IEC JTC1 SC7]. Many suppliers use capability maturity models (CMMs) to guide process improvement and assess capabilities; yet most of the CMMs do not explicitly address safety and security. As such, suppliers claiming mature process capabilities can fail to exercise practices critical to software assurance (SwA). Therefore, questions should be asked to determine how SwA has been factored into suppliers' process capabilities.

How does the supplier ensure that an infrastructure for safety and security is established and maintained? What evidence can be presented to demonstrate that the supplier:

- ensures safety and security competency within the workforce?
- established a qualified work environment (including the use of qualified tools)?
- ensures integrity of safety and security information?
- monitors operations and report incidents (relative to the environment in which the software will be deployed)?
- ensures business continuity?

How does the supplier ensure safety and security risks are identified and managed? What evidence can be presented to demonstrate that the supplier:

- identifies safety and security risks?
- analyzes and prioritizes risks relative to safety and security?
- determines, implements, and monitors the associated risk mitigation plan?

How does the supplier ensure safety and security requirements are satisfied? What evidence can be presented to demonstrate that the supplier:

- determines regulatory requirements, laws, and standards?
- develops and deploys safe and secure products and services?
- objectively evaluates products?
- establishes safety and security assurance arguments?

How does the supplier ensure that activities and products are managed to achieve safety and security requirements and objectives? What evidence can be presented to demonstrate that the supplier:

- establishes independent safety and security reporting?
- establishes a safety and security plan?
- selects and manages suppliers, products, and services using safety and security criteria?
- monitors and controls activities and products relative to safety and security requirements?

[United States Department of Defense and Federal Aviation Administration joint project on Safety and Security Extensions for Integrated Capability Maturity Models, September 2004, <http://www.faa.gov/about/office_org/headquarters_offices/aio/documents/media/SafetyandSecurityExt-FINAL-web.pdf>]

To support the integration of assurance considerations in the development lifecycle, an industry working group created a draft set of assurance goals and practices that harmonize existing practices in the Motorola Secure Software Development Model (MSSDM), System Security Engineering Capability Maturity Model (SSE-CMM) and experience. The goals and practices were mapped to CMMI-DEV v1.2 and can produce a Focus Topic that can assist other organizations with integrating assurance into their continuous process improvement efforts.
https://buildsecurityin.us-cert.gov/swa/downloads/PRM_for_Assurance_to_CMMI.pdf

Appendix F. Sample SwA Requirements Language for the RFP/Contract

The language contained in this appendix is sample language only. The authors and contributors make no warranties about using any of this language. The language should be used by acquirers as suggestions and should be tailored as appropriate in accordance with the acquirers' legal authorities and organizational policies and procedures. Language similar to the following can be used to communicate requirements and terms and conditions in RFP's and/or contracts. As used in this section, the terms contractor and offeror are synonymous with the term supplier.

F.1 Security Controls and Standards

The following is sample language on implementing security controls and standards that may be considered for Federal agency use and may be appropriately modified for other uses. Federal Information Systems and National Security Systems are those defined by the FISMA, NIST standards and publications, and other publications applicable to a particular Federal agency's information systems. In using this language, Federal Information and National Security Systems need to be explicitly defined in accordance with the regulations and publications followed by the organization/agency. Contractor assets may be contractor information technology or other assets that interface with Federal Information and National Security Systems. Paragraph (b) refers to certification and accreditation or other processes that an organization/agency may require. This should be explicitly stated in this paragraph as well.

Language for Security Controls and Standards

(a) When mitigating or remediating risks to confidentiality, integrity, and availability of Federal Information Systems, National Security Systems, contractor assets that enable possession, control, or otherwise enable access to Federal Information or National Security Systems, the Contractor shall implement controls and standards as effective or more effective than those implemented by the Agency for the same or substantially similar risks with the same or substantially similar potential measure of harm.

(b) When selecting appropriate controls and standards for protecting confidentiality, integrity, and availability of Federal Information and National Security Systems, the Contractor shall use the analyses, processes, and standards established for Federal Government systems established by the [current organization/agency and other applicable standards] publications.

F.2 Securely Configuring Proprietary Commercial Software

The following language is quoted from Office of Management and Budget Memorandum M–07–18, *Ensuring New Acquisitions Include Common Security Configurations*, dated 1 June 2007 (effective 1 February 2008). This is recommended language that may be supplemented as necessary. This language should also change when the software and associated regulations and suggestions for the configuration change. The use of common security configurations is included in part 39 of the Federal Acquisition Regulation:

Vista™ and Windows XP™ Standard Secure Configuration

(a) The provided of information technology shall certify applications are fully functional and operate correctly as intended on systems using the Federal Desktop Core Configuration (FDCC). This includes Internet Explorer 7 configured to operate on Windows XP and Vista (in Protected Mode on Vista). For Windows XP settings, see: <http://csrc.nist.gov/itsec/guidance_WinXP.html>, and for the Windows Vista settings, see: <http://csrc.nist.gov/itsec/guidance_vista.html>.

(b) The standard installation, operation, maintenance, update, and/or patching of software shall not alter the configuration settings from the approved FDCC configuration. The information technology should also use the Windows Installer Service for installation to default "program files" directory and should be able to silently install and uninstall.

(c) Applications designed for normal end users shall run in the standard user context without elevated system administration privileges.

F.3 Acceptance Criteria

The following lists suggested generic software security acceptance and measurement criteria to be tailored as appropriate. (This list is similar to F.2.) These would apply at delivery and throughout the software life cycle. The language below may be interpreted in different ways. Therefore, when using the language below or similar language, clarification should be provided on the specific meaning within the context of the software purchased:

Software Acceptance Criteria

(a) The Supplier shall provide all operating system, middleware, and application software to the Acquirer security configured by Supplier in accordance with the FAR requirement based on 44 USC 3544 (b) (2) (D) (iii).

(b) The Supplier shall demonstrate that all application software is fully functional when residing on the operating system and on middleware platforms used by the Acquirer in its production environment, configured as noted above.

(c) The Supplier shall NOT change any configuration settings when providing software updates unless specifically authorized in writing by the Acquirer.

(d) The Supplier shall provide the Acquirer with software tools that the Acquirer can use to continually monitor software updates and the configuration status.

(e) At specified intervals by the Buyer, the Supplier shall provide the Acquirer with a comprehensive vulnerability test report for the suite of applications and associated operating system and middleware platforms used by the Acquirer in its production environment, configured as noted above.

(f) The Acquirer and Supplier agree to work together to establish appropriate measures to quantify and monitor the supplier's performance according to the contract requirements. Specific guidance should include types of measures to be used, measures reporting frequency, measures refresh and retirement, and thresholds of acceptable performance.

(g) The Supplier shall provide all operating system, middleware, and application software to the Acquirer free of common vulnerabilities as specified by the Common Vulnerabilities and Exposures (CVE®)—The Standard for Information Security Vulnerability Names that can be retrieved from http://cve.mitre.org/

(h) The Supplier shall provide all operating system, middleware, and application software to the Acquirer free of common weaknesses as specified in the Common Weakness Enumeration, A Community-Developed Dictionary of Software Weakness Types that can be retrieved from http://cwe.mitre.org/

F.4 Certifications

The following language relates to certification and accreditation processes and should be appropriately tailored for the organization's or agency's particular requirements for certification and accreditation:

> *Contractors must also warrant that proposed system and software product specifications and security and data access architectures have either been addressed in ongoing documentation required by the agency's certification and accreditation process [name the process, regulations governing the process, and specific documentation where this must be addressed] and are ready for evaluation in applicable phases of the process [list the specific phases of the process and specifically what is required in each phase]. Contractors must also address willingness to provide proposed equipment and engineering assistance as required, at no cost to the government, to the specified [name the testing facility] testing facility to obtain required certification of functionality.*

The following language relates to Common Criteria:

> *Contractors must warrant that their products have been satisfactorily validated under common criteria or that products will be satisfactorily validated with the period of time specified in the contract and that such product validation will be maintained for updated versions or modifications by subsequent evaluation as required.*

(Also see F.8 for a sample certification of originality. The due diligence questionnaires in appendix D may also be used in identifying appropriate certifications.)

F.5 Sample Instructions to Potential Suppliers

The following is generic language to include in solicitations. This language provides instructions to potential suppliers on what they must submit with their offer. The information submitted is used to evaluate offers or proposals.

Instructions to Suppliers on Software Assurance

1.0 Foreign ownership, control, or influence (FOCI) is a concern. For any software product that the supplier intends to acquire or develop, the supplier shall answer the following questions: [Note: Insert appropriate questions as shown in the sample questionnaires in appendix D or instruct the offerors to complete the OMB Standard Form 328, "Certificate Pertaining to Foreign Interests."]

2.0 Due Diligence Questionnaire. Offerors shall complete the SwA due diligence questionnaire attached to this RFP.

3.0 Software Assurance Case

3.1 In order for the Acquirer to evaluate the proposed software assurance capabilities, the potential suppliers must submit an initial Software Assurance Case in accordance with ISO/IEC 15026, *Systems and software engineering—Systems and software assurance—Part 2: Assurance Case* . Paragraph 3.2 below identifies the minimum that should be included in the initial assurance case. The initial Software Assurance Case shall subsequently become a part of the contract and be used by the Acquirer as initial acceptance conditions.

3.2 It is understood that the initial Software Assurance Case will be broad in nature because potential suppliers will not know all the details of safety and security until contract performance. However, the assurance case should be comprehensive enough to convey a clear understanding of the safety and security requirement of this RFP. As a minimum, the initial Software Assurance Case shall include the following:

3.2.1 Top-level claims (and sub-claims as appropriate). These claims shall include all the characteristics of claims defined in ISO/IEC 15026.

3.2.2 Arguments for the top-level claims and subclaims. These arguments shall include all the characteristics of arguments defined in ISO/IEC 15026.

3.2.3 Evidence and explicit assumptions supporting the arguments. The evidence shall include all the characteristics of evidence defined in ISO/IEC 15026.

3.2.4 Approving authority for the assurance case. The approving authority resume shall be included. The resume should include evidence of the authority's experience and education in software assurance and developing and managing software assurance cases.

4.0 Initial Software Description. The potential supplier shall submit an initial Software Architecture and such other descriptions as needed to provide a structure for the software. The Software Architecture shall include an initial description of the software components and connector, including software security related aspects. [NOTE: Include additional explanation.]

F.6 Sample Work Statement

1.0 Trustworthy Software

1.1 Key definitions

"Security controls" mean the management, operational, and technical controls (that is, safeguards or countermeasures) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information [NIST SP 800–53]. This definition includes software.

"Security category" means the characterization of information or an information system based on an assessment of the potential impact that a loss of confidentiality, integrity, or availability of such information or information system would have on organizational operations, organizational assets, or individuals [FIPS Pub 199].

"Security objectives" mean confidentiality, integrity, and availability [44 USC, Sec. 3542].

"Assurance" means grounds for justified confidence that a claim has been or will be achieved. (ISO/IEC 15026).

"Assurance Case" means representation of a claim or claims, and support for these claims (ISO/IEC 15026). A Software Assurance Case includes (software assurance) claims and evidence that support those (software assurance) claims.

Include other appropriate definitions.

1.2 Security Category [NOTE: This is an example. Also see [FIPS 199] and [DODI 8500.2, Enclosure 4]

This software system is used for large procurements in a contracting organization and contains both sensitive and proprietary supplier information and routine administrative information. For the sensitive supplier information, the potential impact from a loss of confidentiality is moderate (for example, the loss may result in a significant financial loss), the potential impact from a loss of integrity is moderate (for example, the loss may result in the effectiveness of the contracting mission is significantly reduced and there is significant damage to the information asset), and the potential impact from a loss of availability is low (for example, the loss may result in downtime, but there is backup). For the routine administrative information, the potential impact from a loss of confidentiality is low, the impact from a loss of integrity is low, and the impact from a loss of availability is low.

Based on 1.2, the resulting security category of the software system is {(confidentiality, moderate), (integrity, moderate), (availability, low)}.

1.3 Software Security Requirements. Based on the security category for the software system, the minimum security requirements specified in [NOTE: Reference the external document(s)] are required.

[NOTE: Minimum security controls may be specified in this paragraph or in an external document similar to FIPS Pub 200; National Institute of Standards and Technology (NIST) SP 800–53; and DODI 8500.2, Enclosure 4].

1.4 Software Assurance Case. The Software Assurance Case shall be the primary instrument for refining and monitoring software assurance during the life of this contract. The Software Assurance Case shall be developed and conform to the requirements of ISO/IEC 15026, *Systems and software engineering—Systems and software assurance—Part 2: Assurance Case*. The supplier shall refine the Software Assurance Case throughout the development process and should be based on the software assurance requirements of this contract. The contractor shall submit the case for review. [NOTE: Specify when the case should be reviewed, such as with the submission of the software design] Lastly, the successful execution of the Software Assurance Case shall be a condition for final acceptance of the software product/service.

F.7 Sample Contract Language—US Federal Contracts

Although, the following sample language is tailored for US Federal contracts (task order), other acquirers may tailor parts or all the language for their use, as well.

Sample Contract Language

1.0 GENERAL

All work under this contract shall comply with the latest version of all applicable standards. Individual task orders will reference applicable versions of standards or exceptions as necessary. These may include, but are not limited to, {AGENCY} Manual(s), Acquisition Bulletins [AB], American National Standards Institute [ANSI] standards, and National Institute of Standards and Technology [NIST] standards, including Federal Information Processing Standards [FIPS] publications. Software Development Standards Life Cycle [SDLC] Guidelines contains a list of software development standards for {AGENCY} tasks. The {AGENCY} has developed its own Enterprise Life Cycle. While complying with the latest version of all applicable standards is not a new initiative, it does provide an emphasis of the {AGENCY}'s expectation that the contractor will comply with, and provide verification that these standards are adhered to.

2.0 CORRECTION OF SOFTWARE AND DOCUMENTATION

The contractor shall, over the term of the contract, under any task order issued, correct errors in contractor developed software and applicable documentation that are not commercial off-the-shelf which are discovered by the Government, and any other user of the software, or the contractor. If the system is in production, such corrections shall be completed within one working day of the date the contractor discovers or is notified of the error (or a date mutually agreed upon between the CO and the contractor not to exceed 30 working days). If the system is not in production, such corrections shall be made within five working days of the date the contractor discovers or is notified of the error (or a date mutually agreed upon between the CO and the contractor, not to exceed 30 days). Latent defects will be handled in the same manner, as soon as they are discovered. Inability of the parties to determine the cause of software errors shall be resolved in accordance with the Disputes clause in Section I, FAR 52.233-1, incorporated by reference in the contract, but in no event constitutes grounds for delay of error correction beyond the periods specified.

3.0 SOFTWARE DEVELOPMENT PROCEDURES

3.1 CMMI

3.1.1 All contractors awarded task orders for any activity related to software development for the {AGENCY} shall comply with the {AGENCY} policy for CMMI® compliance. All tasks that fall within the software development life cycle shall at minimum comply with Level {2, 3, 4, or 5 as required} of the staged representation of the CMMI® for Software Engineering (CMMI-SW). There are no exceptions to this {AGENCY} policy. Contractors developing software for the {AGENCY} shall maintain Level {2, 3, 4, or 5 as required} or higher in the staged representation of the CMMI-SW in order to continue to receive software tasking.

3.1.2 The CMM Review Team will monitor contractor process maturity (1) using standard {AGENCY} Process Appraisal Review Methodology (PARM) processes, including execution of Standard CMMI Appraisal Method for Process Improvement (SCAMPISM), as needed, (2) performing annual cycles of review for CMMI-SW, and (3) considering all types of appraisal data and process improvement infrastructure data as standardized by the {AGENCY} PARM process to verify alignment and mapping of the contractor's CMMI processes to the {AGENCY} Enterprise Life Cycle (ELC). The responsible organization is indicated as Contractor (to be delivered under this Task Order), Government (Government will prepare), or Joint (a joint effort with the {AGENCY} in the lead). The Government may waive (indicated as Not Applicable) the requirements for certain deliverables or work products based on the approved Program Tailoring Plan.

3.2 SECURITY CONTROLS

3.2.1 The Contractor shall follow the NIST 800-53, Recommended Security Controls for Federal Information Systems and the {AGENCY} guidance to ensure that the Software will be or has been developed using secure coding practices in a manner that minimizes security flaws within the Software. Prior to the execution of a software development Work Request the Contractor shall provide the {AGENCY} a copy of the Contractor's secure coding best practices policy and upon delivery of the Software to the {AGENCY}, the Contractor shall certify to the {AGENCY} in writing that the Contractor complied with the Policy in the performance of its obligations under the task order.

3.2.2 The contractor will be subject to an annual review that will allow the {AGENCY} to assess the effectiveness of security controls. In addition, the contractor shall ensure that appropriate security management tools are in place to allow for the review of security configurations, user identities, etc.

3.3 REQUIREMENTS TRACEABILITY. Contractor shall provide the requirements traceability matrix at the end of analysis phase, design phase, build phase, deployment phase that designates the security requirements in a separate section so that they can be traced through the development life cycle. The contractor shall also provide the application designs and test plan documentation, and source code to Government for review.

3.4 SOFTWARE CHANGES. Without exception, for changes that may produce an impact on security, the contractor shall follow the Security Change Management procedures.

3.5 MALICIOUS CODE WARRANTY. The Contractor represents and warrants that the Software shall be free from all computer viruses, worms, time-outs, time bombs, back doors, disabling devices and other harmful or malicious code intended to or which may damage, disrupt, inconvenience or permit access to the Software user's or another's software, hardware, networks, data or information.

3.6 ACCEPTANCE—SECURE SOFTWARE

3.6.1 Notwithstanding any other provision of the task order, the {AGENCY} will not accept the software until a Government source code and security analysis has been performed. The Software shall be deemed to be "Non-Secure" if the Software includes any one or more of the security flaws. A detailed listing of security flaws will be provided to the contractor and will be updated based on newly discovered flaws.

3.6.2 If after a security audit the Software is determined to be Non-Secure, then upon written notice of such Non-Secure status, the Contractor, at its cost and expense, shall use its commercially reasonable best efforts to remedy the security flaws by modifying or replacing the Software within 30 days of receipt of such written notice (the "Remedy Period"). Upon receipt of revised Software and notice from the Contractor that the security flaws have been remedied prior to the end of the Remedy Period, the Government, shall again subject the Software to a security audit at the Contractor's expense.

3.6.3 Notwithstanding any other provision of the Agreement, if the Software is determined to be Non-Secure as set forth above and remains Non-Secure at the end of the Remedy Period, the {AGENCY} shall be deemed to have not accepted the Software under the terms of the Contract unless the {AGENCY} in its sole discretion otherwise expressly agrees in writing to accept the Software notwithstanding that it is deemed to be Non-Secure.

3.7 WORK PRODUCTS REQUIRED

(These will be defined within the Work Requests written for this sub-task).

3.8 ACCEPTANCE CRITERIA--OTHER

Acceptance Criteria (These will be defined within the Work Requests written for this sub-task).

3.9 AGENCY SECURITY STANDARD. The security standard for the {AGENCY}, including both the security policies and the security requirements, are pre-defined. In general, the sources of these policies and requirements include:

- In accordance with the requirements of the Office of Federal Procurement Policy Act of 1974 (Pub. L. 93-400), as amended by Pub. L. 96-83.
- The Federal Information Security Management Act of 2002 (H.R. 2458, Title III, Section 301)
- Computer Security Act of 1987 - Section 11332 of Title 40, United States Code,
- OMB A-130, Management of Federal Information Resources, Appendix III -
- OMB A-127, Financial Management Systems -
- OMB A-123, Management's Responsibility for Internal Control
- Publication 1075, Safeguarding Taxpayer Information for Federal, State, and Local Agencies & Entities

- {Appropriate AGENCY or Senior AGENCY Directives and Manuals}

4.0 PREPARATION AND MAINTENANCE OF CERTIFICATION AND ACCREDITATION DOCUMENTS

4.1 The Contractor shall to participate in the {AGENCY} security certification and accreditation (C&A) process by providing all product specific input (electronic) to the Application System Security Plan (SSP) and the Application Information Technology Contingency Plan (ITCP). Refer to NIST Special Publication 800-18, Guide for Developing Security Plans for Information Technology Systems (http://csrc.ncsl.nist.gov/publications/nistpubs), for additional guidelines in writing a security plan. An {AGENCY} Certification and Accreditation checklist, guidance and document templates is available to assist the Contractor. The purpose of the SSP and the ITCP is to provide the {AGENCY} with technical insight on how the Contractor meets the {AGENCY} security requirements.

4.2 The contractor shall follow the {AGENCY} security standard and policies (see the document references) for security. The contractor shall use the policies and other applicable guidance as a framework of for specific security controls, documents, procedures and features when performing security requirements analysis and security design.

4.3 The Contractor shall provide draft updates draft (electronic) version of the Application System Security Plan within 30 calendar days of a major change to the system or at a minimum on an annual basis by June 30 to the PMO.

4.4 Included with the submission of the draft SSP the Contractor shall maintain a current up-to-date version of the Application Information Technology Contingency Plan (ITCP) and related escalation procedures.

4.5 The PMO will review the security documents for 10 calendar days and provide written comments and changes for the Contractor to address. Contractors will have 10 calendar days to address the comments and complete the documents.

4.6 The security standard for the {AGENCY}, including both the security policies and the security requirements, are pre-defined. In general, the sources of these policies and requirements include:

{Appropriate AGENCY or Senior AGENCY Directives and Manuals}

- NIST Draft SP 800-37 “Guidelines for the Security Certification and Accreditation (C&A) of Federal Information Technology Systems,” November 5, 2002, http://csrc.nist.gov/sec-cert/
- The Computer Security Act of 1987 (PL 100-235)
- The Privacy Act (PA)
- The Federal Information Systems Management Act (FISMA).

5.0 CONTIUOUS MONITORING, TESTING, AND REPORTING

5.1 SELF TESTING

5.1.1 SELF TESTING REQUIREMENTS. The contractor shall perform self testing of their implemented security controls. The contractor shall continuously monitor all testing activities and report on the performance and effectiveness of the {AGENCY}security controls (as required by FISMA, OMB Circular A-130, NIST guidance and FIPS publications) to the {AGENCY} project manager assigned to oversee this contract. The

specific assessments procedures as outlined in draft NIST Special Publication 800-53A, shall be used by the contractor in assessing the whether appropriate corrective action was taken on previously closed POA&Ms (Plan of Action and Milestones) and volatile security controls.

5.1.2 SELF TESTING SCHEDULE. The contractor shall work with {AGENCY} project manager to establish schedules, discuss roles and responsibilities, testing requirements, and other general activities to ensure continuous monitoring is well organized and completed in a timely manner and in accordance with the {AGENCY} procedures.

5.2 TESTING OF CONTROLS

5.3 TESTING CONTROLS. The security controls specified in {AGENCY} policy {identify the policies or other documents} shall be tested by the contractor using approved {AGENCY} methods. As part of testing controls, the contractor shall examine existing data sources and metrics, such as self-assessments, incident reporting statistics, risk assessments, third-party evaluations, and other existing data and propose a set of metrics that leverages existing data and is consistent with the compliance evaluation criteria. The contractor shall include verification and validation to ensure that appropriate corrective action was taken on POA&Ms closed in the last quarter.

5.4 REPORTING. The contractor shall provide a determination, in a written form agreed to by the {AGENCY} project management, on whether the implemented corrective action was adequate to resolve the identified information security weaknesses and provide the reasons for any exceptions or risked-based decisions.

5.3 ASSESSMENT METHODOLOGY

5.3.1 ASSESSMENT PROCEDURES. The assessment procedures as outlined in draft NIST 800-53A shall be used by the contractor as guide in the execution of the test procedures as deemed appropriate. These procedures shall be supplemented and augmented by tailored test procedures based on the control objective as it applies to {AGENCY}.

5.3.2 TEST METHODS. The contractor shall us test methods that include interview, examine, and test. The interview method is the process of conducting focused discussions with individuals or groups of individuals within an organization to facilitate the reviewer's understanding, achieve clarification, or obtain evidence. The examine method is the process of reviewing, inspecting, observing, studying, or analyzing one or more assessment objects (i.e., specifications, mechanisms, or activities). Similar to the interview method, the primary purpose of the examine method is to facilitate the reviewer's understanding, achieve clarification, or obtain evidence. The test method is the process of exercising one or more objects (limited to activities or mechanisms) under specified conditions to compare actual with expected behavior. In all three cases (i.e., interview, examine, and test) where the methods are employed, the results are used to support the determination of overall security control effectiveness.

5.3.3 DETERMINATION STATEMENTS. The contractor shall write a determination statement describing the results of each tested control.

5.3.4 SCORING. The contractor shall mark each determination with an S" for "Satisfied" or an "O" for "Other than satisfied." If all of the determination statements and test procedures for a control be marked as "S for Satisfied", then the contractor shall score the control as "In Place". If one or more of the determination statements and test procedures for a control is marked as "O for Other than Satisfied", then the contractor shall score the control as "Partially in Place". If most or all of the determination statements and test procedures for a control are marked as "O for Other than Satisfied", then the contractor shall score the control as "Planned".

5.5 REPORTING

The contractor shall fully document test in accordance with the reporting format prescribed by {AGENCY} procedures. When the results are partially satisfied or other than satisfied condition, the contractor shall document any vulnerabilities indicating which portions of the security control have not been implemented or applied.

6.0 OTHER ACTIVITIES ***The following are some idea for additional activities—some of which may be repeated in previous paragraphs. These additional ideas provide additional language from which to choose. Also not that the language below and in some previous paragraphs apply to the "system" vice specifically to the "software" in the system.***

6.1 PO&M MAINTENANCE. The contractor shall develop and support Web Services in implementing solutions that will provide a means of planning and monitoring corrective actions; define roles and responsibilities for risk mitigation; assist in identifying security funding requirements; track and prioritize resources; and inform decision-makers of progress of open POA&M items. The contractor shall perform verification of IT security weaknesses to ensure that all weaknesses identified through third party (e.g., OIG) audits are included in the POA&Ms that the quarterly reporting to OMB is accurate, as well as actual activities are mirroring planned activities and the reasons for any exceptions or risked-based decisions are reasonable and clearly documented. This verification process will be done in conjunction with the continuous monitoring program and will leverage the knowledge and methodology established through that strategy.

6.2 FISMA COMPLIANCE. The contractor shall plan and execute FISMA testing of controls.

6.3 C&A DOCUMENTATION. The contractor shall update the Application Certification and Accreditation (C&A) documentation to ensure that the C&A artifacts are kept current and contain all information and supporting evidence is documented for the next certification and accreditation is available and complete. This shall include reviewing changes made to the system in order to identify any new data types that may have a Privacy Impact or change the Security Categorization of the system.

6.4 SECURITY RISK ASSESSMENT. The contractor shall work with the {AGENCY} project manager in performing Security Risk Assessment (SRA). This includes identifying risks related to the design and functionality of a new system against compliance with the {AGENCY} risk management process, NIST SP 800-39, FIPS 200 and SP 800-53. Activities performed during this phase shall include analyzing how the security architecture implements the {AGENCY} documented security policy for the system, assessing how management, operational, and technical security control features are implemented by the software and hardware, how the system interconnects to other networks while maintaining security, and lastly analyzing other inherent design features. Procedures including a checklist shall be used to document compliance with baseline security requirements and existing agency guidance.

6.6 SECURITY TESTING AND EVALUATION (ST&E). The contractor shall work with {AGENCY} project manager in performing pre-ST&E activities, including but not limited to, coordinating the ST&E and developing the ST&E Plan and ST&E test cases. The contractor shall also assist project officer in performing dry run assessments to determine the adequacy of functional and non-functional controls in preparation for the ST&E, update the ST&E plan based on dry run assessments, observe and provide assistance to the ST&E testing team to ensure successful and timely completion of the test plans, and prepare the Plan of Action and Milestones (POA&M) resulting from the ST&E results.

7.0 GOVERNMENT INDEPENDENT TESTING

The Government will perform periodic vulnerability testing to evaluate the security of {AGENCY}. The process involves an active analysis of the system for any potential vulnerabilities that may result from poor or

improper system configuration, known and/or unknown hardware or software flaws, or operational weaknesses in process or technical countermeasures. The intent of testing is to determine feasibility of an attack and the amount of business impact of a successful exploit, if discovered. The frequency of the testing will be at a minimum quarterly and on demand based on the risk associated with newly discovered vulnerabilities.

F.8 Open Web Application Security Project

The following contract annex is a product of the Open Web Application Security Project (OWASP). The information was retrieved from <http://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex> on 30 November 2006. The language below in whole, in part, or as tailored may be used to communication requirements in a work statement or stated as terms and conditions. As quoted on the Web site:

WARNING: THIS DOCUMENT SHOULD BE CONSIDERED ADVICE ONLY.
OWASP STRONGLY RECOMMENDS THAT YOU CONSULT A QUALIFIED
ATTORNEY TO HELP YOU NEGOTIATE A SOFTWARE CONTRACT.

The Web site further states: "This contract Annex is intended to help software developers and their clients negotiate and capture important contractual terms and conditions related to the security of the software to be developed or delivered. The reason for this project is that most contracts are silent on these issues, and the parties frequently have dramatically different views on what has actually been agreed to. We believe that clearly articulating these terms is the best way to ensure that both parties can make informed decisions about how to proceed."

1. INTRODUCTION

This Annex is made to _____ ("Agreement") between Client and Developer. Client and Developer agree to maximize the security of the software according to the following terms.

2. PHILOSOPHY

This Annex is intended to clarify the security-related rights and obligations of all the parties to a software development relationship. At the highest level, the parties agree that:

(a) *Security Decisions Will Be Based on Risk.* Decisions about security will be made jointly by both Client and Developer based on a firm understanding of the risks involved.

(b) *Security Activities Will Be Balanced.* Security effort will be roughly evenly distributed across the entire software development lifecycle.

(c) *Security Activities Will Be Integrated.* All the activities and documentation discussed herein can and should be integrated into Developer's software development life cycle and not kept separate from the rest of the project. Nothing in this Annex implies any particular software development process.

(d) *Vulnerabilities Are Expected.* All software has bugs, and some of those will create security issues. Both Client and Developer will strive to identify vulnerabilities as early as possible in the life cycle.

(e) *Security Information Will Be Fully Disclosed.* All security-relevant information will be shared between Client and Developer immediately and completely.

(f) *Only Useful Security Documentation Is Required*. Security documentation does not need to be extensive in order to clearly describe security design, risk analysis, or issues.

3. LIFE CYCLE ACTIVITIES

(a) Risk Understanding. Developer and Client agree to work together to understand and document the risks facing the application. This effort should identify the key risks to the important assets and functions provided by the application. Each of the topics listed in the requirements section should be considered.

(b) Requirements. Based on the risks, Developer and Client agree to work together to create detailed security requirements as a part of the specification of the software to be developed. Each of the topics listed in the requirements section of this Annex should be discussed and evaluated by both Developer and Client. These requirements may be satisfied by custom software, third-party software, or the platform.

(c) Design. Developer agrees to provide documentation that clearly explains the design for achieving each of the security requirements. In most cases, this documentation will describe security mechanisms, where the mechanisms fit into the architecture, and all relevant design patterns to ensure their proper use. The design should clearly specify whether the support comes from custom software, third-party software, or the platform.

(d) Implementation. Developer agrees to provide and follow a set of secure coding guidelines. These guidelines will indicate how code should be formatted, structured, and commented. All security-relevant code shall be thoroughly commented. Specific suggestions on avoiding common security vulnerabilities shall be included. Also, all code shall be reviewed by at least one other Developer against the security requirements and coding guideline before it is considered ready for unit test.

(e) Security Analysis and Testing. Developer agrees to provide and follow a security test plan that defines an approach for testing or otherwise establishing that each of the security requirements has been met. The level of rigor of this activity should be considered and detailed in the plan. Developer will execute the security test plan and provide the test results to Client.

(f) Secure Deployment. Developer agrees to provide secure configuration guidelines that fully describe all security relevant configuration options and their implications for the overall security of the software. The guideline shall include a full description of dependencies on the supporting platform, including operating system, Web server, and application server, and how they should be configured for security. The default configuration of the software shall be secure.

4. SECURITY REQUIREMENT AREAS

The following topic areas must be considered during the risk understanding and requirements definition activities. This effort should produce a set of specific, tailored, and testable requirements Both Developer and Client should be involved in this process and must agree on the final set of requirements.

(a) Input Validation and Encoding. The requirements shall specify the rules for canonicalizing, validating, and encoding each input to the application, whether from users, file systems, databases, directories, or external systems. The default rule shall be that all input is invalid unless it matches a detailed specification of what is allowed. In addition, the requirements shall specify the action to be taken when invalid input is received. Specifically, the application shall not be susceptible to injection, overflow, tampering, or other corrupt input attacks.

(b) Authentication and Session Management. The requirements shall specify how authentication credentials and session identifiers will be protected throughout their life cycle. Requirements for all related functions,

including forgotten passwords, changing passwords, remembering passwords, logout, and multiple logins, shall be included.

(c) Access Control. The requirements shall include a detailed description of all roles (groups, privileges, authorizations) used in the application. The requirements shall also indicate all the assets and functions provided by the application. The requirements shall fully specify the exact access rights to each asset and function for each role. An access control matrix is the suggested format for these rules.

(d) Error Handling. The requirements shall detail how errors occurring during processing will be handled. Some applications should provide best effort results in the event of an error, whereas others should terminate processing immediately.

(e) Logging. The requirements shall specify what events are security-relevant and need to be logged, such as detected attacks, failed login attempts, and attempts to exceed authorization. The requirements shall also specify what information to log with each event, including time and date, event description, application details, and other information useful in forensic efforts.

(f) Connections to External Systems. The requirements shall specify how authentication and encryption will be handled for all external systems such as databases, directories, and Web services. All credentials required for communication with external systems shall be stored outside the code in a configuration file in encrypted form.

(g) Encryption. The requirements shall specify what data must be encrypted, how it is to be encrypted, and how all certificates and other credentials must be handled. The application shall use a standard algorithm implemented in a widely used and tested encryption library.

(h) Availability. The requirements shall specify how it will protect against denial of service attacks. All likely attacks on the application should be considered, including authentication lockout, connection exhaustion, and other resource exhaustion attacks.

(i) Secure Configuration. The requirements shall specify that the default values for all security-relevant configuration options must be secure. For audit purposes, the software should be able to produce an easily readable report showing all the security-relevant configuration details.

(j) Specific Vulnerabilities. The requirements shall include a set of specific vulnerabilities that shall not be found in the software. If not otherwise specified, then the software shall not include any of the flaws described in the current "OWASP Top Ten Most Critical Web Application Vulnerabilities."

5. PERSONNEL AND ORGANIZATION

(a) Security Architect. Developer will assign responsibility for security to a single senior technical resource, to be known as the project Security Architect. The Security Architect will certify the security of each deliverable.

(b) Security Training. Developer will be responsible for verifying that all members of the developer team have been trained in secure programming techniques.

(c) Trustworthy Developers. Developer agrees to perform appropriate background investigations of all development team members.

6. DEVELOPMENT ENVIRONMENT

(a) Secure Coding. Developer shall disclose what tools are used in the software development environment to encourage secure coding.

(b) Configuration Management. Developer shall use a source code control system that authenticates and logs the team member associated with all changes to the software baseline and all related configuration and build files.

(c) Distribution. Developer shall use a build process that reliably builds a complete distribution from source. This process shall include a method for verifying the integrity of the software delivered to Client.

7. LIBRARIES, FRAMEWORKS, AND PRODUCTS

(a) Disclosure. Developer shall disclose all third-party software used in the software, including all libraries, frameworks, components, and other products, whether commercial, free, open source, or closed source.

(b) Evaluation. Developer shall make reasonable efforts to ensure that third-party software meets all the terms of this agreement and is as secure as custom-developed code developed under this agreement.

8. SECURITY REVIEWS

(a) Right to Review. Client has the right to have the software reviewed for security flaws at their expense at any time within 60 days of delivery. Developer agrees to provide reasonable support to the review team by providing source code and access to test environments.

(b) Review Coverage. Security reviews shall cover all aspects of the software delivered, including custom code, components, products, and system configuration.

(c) Scope of Review. At a minimum, the review shall cover all of the security requirements and should search for other common vulnerabilities. The review may include a combination of vulnerability scanning, penetration testing, static analysis of the source code, and expert code review.

(d) Issues Discovered. Security issues uncovered will be reported to both Client and Developer. All issues will be tracked and remediated as specified in the Security Issue Tracking section of this Annex.

9. SECURITY ISSUE MANAGEMENT

(a) Identification. Developer will track all security issues uncovered during the entire life cycle, whether a requirements, design, implementation, testing, deployment, or operational issue. The risk associated with each security issue will be evaluated, documented, and reported to Client as soon as possible after discovery.

(b) Protection. Developer will appropriately protect information regarding security issues and associated documentation to help limit the likelihood that vulnerabilities in operational Client software are exposed.

(c) Remediation. Security issues that are identified before delivery shall be fixed by Developer. Security issues discovered after delivery shall be handled in the same manner as other bugs and issues as specified in this Agreement.

10. ASSURANCE

(a) Assurance. Developer will provide a "certification package" consisting of the security documentation created throughout the development process. The package should establish that the security requirements, design, implementation, and test results were properly completed and all security issues were resolved appropriately.

(b) Self-Certification. The Security Architect will certify that the software meets the security requirements, all security activities have been performed, and all identified security issues have been documented and resolved. Any exceptions to the certification status shall be fully documented with the delivery.

(c) No Malicious Code. Developer warrants that the software shall not contain any code that does not support a software requirement and weakens the security of the application, including computer viruses, worms, time bombs, back doors, Trojan horses, Easter eggs, and all other forms of malicious code.

11. SECURITY ACCEPTANCE AND MAINTENANCE

(a) Acceptance. The software shall not be considered accepted until the certification package is complete and all security issues have been resolved.

(b) Investigating Security Issues. After acceptance, if security issues are discovered or reasonably suspected, Developer shall assist Client in performing an investigation to determine the nature of the issue. The issue shall be considered "novel" if it is not covered by the security requirements and is outside the reasonable scope of security testing.

(c) Novel Security Issues. Developer and Client agree to scope the effort required to resolve novel security issues and to negotiate in good faith to achieve an agreement to perform the required work to address them.

(d) Other Security Issues. Developer shall use all commercially reasonable efforts consistent with sound software development practices, taking into account the severity of the risk, to resolve all security issues not considered novel as quickly as possible.

F.9 Certification of Originality

The following is an IBM example on how certifications can be used [IBM CO]. Permission for use is granted by IBM. This certification covers more than SwA and should be used as an idea generator for ultimate requirements and terms and conditions that the acquisition official creates for the RFP and/or contract.

“Certificate of Originality”

This questionnaire must be completed by a vendor (“You”) furnishing copyrightable material, such as software, audio/visual works, written materials, etc. (“Material”), to IBM. The acceptance of this questionnaire by IBM is a prerequisite for the IBM final payment for the furnished Material.

Depending on Your agreement with IBM, You may have an obligation to communicate additional information to IBM that IBM may require for copyright registration and/or enforcement of legal rights relating to the furnished material.

Please leave no questions blank. Write "not applicable" or "N/A" if a question is not relevant to the furnished material.

Summary Information

Your name and address: _____
Name of the Material: _____
IBM Contract No: _____
IBM Contract Administrator: _____

A—Material Identification

1. Category of the material (Please check only one):

   ☐ a) Software (including its related documentation)

   ☐ b) Audiovisual Works

   ☐ c) Mask Works

   ☐ d) Written Materials excluding related documentation of a) through c)

   ☐ e) Other (please identify)_____

If You selected either "Software" or "Audio/Visual Works", please provide the names of any software tools (for example, compiler, software development tool, etc.) that were used to create such Material: _____

2. General description of the Material (including the description of any new function that You contributed):

   _____

3. What was the date that the creation of Material was completed? (except for minor error corrections, etc.):_____

B—Newly Created Material

The questions in this section are targeted at any **newly** created portion of the Material ("Newly Created Material"). If the Material includes any preexisting material, please provide detailed information for such preexisting material in section C (Preexisting Material). All developers or creators of the Newly Created Material must be specified in one of the following Categories I, II or III. Unless otherwise indicated, Your employees include temporary and supplemental employees who created or contributed to the creation of the Material under contract or other agreement with You.

I. Was any portion of the Newly Created Material created by Your employee(s) *within the scope* of their work assignment or job function ("Category I") assignment? ☐ Yes ☐ No

If You checked Yes, please provide a copy of any relevant employee agreement governing the creation of intellectual property for Your company by the employee and provide below the requested

information for each employee. It is not necessary to provide copies of the agreements actually signed by each employee as long as You provide the terms of each agreement. For example, it would be sufficient to provide blank employee agreement forms of the type actually completed by the employee.

Name of employee: _____
Title: _____

Name of employee: _____
Title: _____

Name of employee: _____
Title: _____

(If there is insufficient space to list all contributors, please attach an additional page with the required information).

II. Was any portion of the Newly Created Material created by Your employee(s) *outside the scope* of their work assignment or job function ("Category II")? ☐ Yes ☐ No

If You checked Yes, please provide a copy of any relevant employee agreement governing the creation of intellectual property for Your company by the employee and provide below the requested information for each employee. It is not necessary to provide copies of the agreements actually signed by each employee as long as You provide the terms of each agreement. For example, it would be sufficient to provide blank employee agreement forms of the type actually completed by the employee.

Name of employee: _____
Title: _____

Name of employee: _____
Title: _____

Name of employee: _____
Title: _____

(If there is insufficient space to list all contributors, please attach an additional page with the required information).

III. Was any portion of the Newly Created Material created for You by anyone other than Your employees, including another vendor company, an independent contractor, a subcontractor, a consortium or university ("Category III")? ☐ Yes ☐ No

If You checked Yes, please provide a copy of any relevant agreement that You may have governing the creation and/or license of the intellectual property for this Material and the names and title of the individuals who contributed the material. If the third party was a company, please provide the name and address for the company.

Name of employee: _____
Title: _____

Name of employee: _____
Title: _____

Name of employee: _____
Title: _____

(If there is insufficient space to list all contributors, please attach an additional page with the required information).

1. Does any portion of the Newly Created Material link to any libraries or other software that is characterized as freeware, shareware, or open source software ("OSS Material")? For the purposes of this Certificate of Originality, open source software is computer software programs whose source code is available for inspection and use by anyone and is made available under a license that permits recipients to copy, modify and distribute the program's source code without payment of royalty. Common examples of such licenses, include, but are not limited to, the GNU GPL and LGPL licenses, the Mozilla Public License, Apache license, BSD License, MIT License, Common Public License, etc.

   ☐ Yes ☐ No

If you checked No, please go to section C.

If you checked Yes, please, provide the following OSS Material information.

Is the linking static or dynamic? ☐ static ☐ dynamic

OSS Material Name: _____

Source of the OSS Material (URL, company address, etc.): _____

License Information (please attach a copy of the license): _____

Any information that would be helpful to identify the ownership of the OSS Material (Copyright notice, author's name, contact information, etc.):

_____

C—Preexisting Material

The target of this section is any material that had been created by You or others, before You entered into an agreement with IBM to create the Material ("Preexisting Material"). Preexisting Material includes, but is not limited to, software, software libraries, textbooks, and publications that were used in the creation of the Material provided by You to IBM.

1. Was any portion of the Material composed of or derived from Preexisting Material?

☐ Yes ☐ No

If you checked No, please go to section D.

2. Is any portion of the Preexisting Material owned by You? ☐ Yes ☐ No

If you checked Yes, please provide the name of the Preexisting Material.

3. Is any portion of the Preexisting Material owned by a third party (excluding OSS Material)?

☐ Yes ☐ No

If you checked Yes, please provide the following information:

Name of Preexisting Material: _____

Source of the Preexisting Material (URL, company address, etc):

_____

License Information (please attach a copy of the license): _____

Any information that would be helpful to identify the source and ownership of the material (Copyright notice, author's name, contact information, etc.):

_____

4. Have You modified the third-party Preexisting Material? ☐ Yes ☐ No

If you checked Yes, please briefly describe the nature of the modifications:

_____

5. Is any portion of the Preexisting Material OSS Material? ☐ Yes ☐ No

If you checked Yes, please provide the following information:

Name of Preexisting Material: _____
Source of the Preexisting Material (URL, company address, etc): _____
License Information (please attach a copy of the license): _____

Any information that would be helpful to identify the source and ownership of the material (Copyright notice, author's name, contact information, etc.): _____

6. Have You modified the third party OSS Material? ☐ Yes ☐ No

If you checked Yes, please briefly describe the nature of the modifications:

_____

7. Does any portion of the Preexisting Material link to any OSS Material, including, for example, by using an OSS Material source software development kit? ☐Yes ☐No

If you checked Yes, please, provide the following OSS Material information.

8. Is the linking static or dynamic? ☐static ☐dynamic

OSS Material Name: _____

Source of the OSS Material (URL, company address, etc): _____

License Information (please attach a copy of the license): _____

Any information that would be helpful to identify the source and ownership of the material (Copyright notice, author's name, contact information, etc.): _____

D—External Characteristics including Icons

"External Characteristics" include display screens, data formats, instruction or command formats, operator messages, interfaces, images video, sound recordings, icons, etc.

1. Were the "External Characteristics" of the Material or any portion thereof copied or derived from the preexisting "external characteristics" of other software or copyrightable material ("Preexisting Externals")?

☐Yes ☐No

If You checked No, please go to section E.

If You checked Yes, please provide the following information:

a) Type of External Characteristic: _____

b) Name of External Characteristic: _____

c) Source of External Characteristic: _____

d) Author (if known):_____

e) Owner: _____

f) License information (if applicable): _____

g) Please identify or describe any preexisting External Characteristics are known to you that are similar in appearance to the External Characteristic(s) that you are providing in the Material. _____

E—Miscellaneous

1. Does the Material conform to any particular technology standards? ☐Yes ☐No

If You checked Yes, please identify the name of such standard and standards body.

Name of Standard:_____

Standards body: _____

2. Identify below, or in an attachment, any other circumstance which might affect IBM's ability to reproduce and market this material, including:

a) Confidentiality or trade secrecy of Preexisting Materials included in the Material:

_____

b) Known or possible royalty obligations to others arising out of the Material:

_____

c) Other circumstances:_____

Certification

By submitting this form, You acknowledge that You have responsibility for and direct knowledge of development or creation of this Material and hereby certify that:

a) All statements made in this form are true;

b) This Material does not contain any materials copied or derived from other code, designs, document or other materials, except as listed herein; and

c) All newly written parts of this material are original work of Your employees and/or third party under contract as specified herein.

Yes, I certify to the above statements.

_____
Signature
Name: _____
Title: _____
Date: _____

[IBM CO]

F.10 Other Sources of SwA Requirements

Common Weakness Enumeration (CWE): A list of the known software security weaknesses in architecture, design, or implementation that can lead to exploitable vulnerabilities developed under funding from DHS/NCSD Strategic Initiatives, available at <http://cwe.mitre.org/data/index.html>.

Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software (SwA CBK) developed by the DHS/DOD SwA Workforce, Training and Education Working Group. Available at <https://buildsecurityin.us-cert.gov>.

Open Web Application Security Project (OWASP) Secure Software Development Contract Annex, available at <http://www.owasp.org/documentation/legal.html>.

OWASP Guide to Building Secure Web Applications and Web Services (Version 3.0), available at <http://www.owasp.org/index.php/OWASP_Guide_Project>.

ISO/IEC 15408–3. *Information technology—Security techniques—Evaluation criteria for IT security—Part 3: Security assurance requirements*, available at <http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm>.

Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, available at <http://www.commoncriteriaportal.org/public/consumer/index.php?menu=2>

ISO/IEC 17799. *Information Technology—Security techniques—Code practice for information security management*, available at <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=39612&ICS1=35&ICS2=40&ICS3=>.

ISO/IEC 27001. *Information technology—Security techniques—Information security management systems – Requirements*, available at <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=42103>.

Appendix G. References

[40 U.S.C., Sec. 11101] Information Technology Management. *Definitions.*

[44 U.S.C., Sec. 3502] Federal Information Policy. *Definitions.*

[Abran] Abran, Alain, & Moore, J.W. (Executive editors); Borque, P.,Dupuis, R. & Tripp, L. (Editors). (2004). *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society.

[Allen] Allen, J.H. (2006). *How Much Security Is Enough.* Pittsburg, PA: Carnegie Mellon University. Retrieved October 3, 2008, from https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/management/566-BSI.html

[Anderson] Anderson, E. A., Irvine, C. E., & Schell, R. R. (2004). "Subversion as a threat in information warfare." In *Journal of Information Warfare*, 3:51 – 64.

[ANSDIT] *American National Standard Dictionary of Information Technology.* Retrieved on 15 August 2008 from http://www.incits.org/tc_home/k5htm/Ansdit.htm.

[Barnum] Barnum, S. & Sethi, A. (2006, November 11). *Attack Pattern Glossary*. Cigital, Inc. Retrieved 15 August 2008 from https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/attack/590-BSI.html

[CCA] Clinger-Cohen Act of 1996, Public Law 104-106

[CC1] *Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model*. http://www.commoncriteriaportal.org/public/consumer/index.php?menu=2

[CC2] *Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components*. http://www.commoncriteriaportal.org/public/consumer/index.php?menu=2

[CC3] *Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components*. http://www.commoncriteriaportal.org/public/consumer/index.php?menu=2

[CERT Secure Coding] *CERT—Secure Coding.* Retrieved 15 August 2008 from http://www.cert.org/secure-coding.

[CNSSI 4009] Committee on National Security Systems. (2003, May). *National Information Assurance (IA) Glossary (CNSSI No. 4009).* Ft Meade, MD: National Security Agency.

[CNSS-145-06] Global Information Technology Working Group, Committee on National Security Systems. (2006, November). *Framework for Lifecycle Risk Mitigation for National Security Systems in the Era of Globalization*. Ft. Meade, MD: National Security Agency.

[COHEN] Cohen, F. (2004, January 9). *Enterprise Patch Management: Strategies, Tools, and Limitations*. http://www.burtongroup.com

[CWE-120] *Common Weakness Enumeration—A Community-Developed Dictionary of Software Weakness Types*—CWE 120, Individual Dictionary Definition (Draft 9). Retrieved 14 August, 2008 from http://cwe.mitre.org/data/definitions/120.html

[DACS-Walker] Walker, E. (2005, July). Software Development Security: A Risk Management Perspective. In *The DoD Software Tech News—Secure Software Engineering.* Vol*(8)*No*(2)*. Rome, NY: Data & Analysis Center for Software. http://www.softwaretechnews.com/stn8-2/

[DCID 6/3] DCID. (1999, June 5). *Protecting Sensitive Compartmented Information Within Information Systems*. http://www.fas.org/irp/offdocs/DCID_6-3_20Policy.htm

[DoD 2004] Department of Defense. (2004, September). *Interim Report on SwA: Mitigating Software Risks in the DoD IT and National Security Systems*

[DoD Guidebook] Department of Defense. *Defense Acquisition Guidebook*. Fort Belvoir, VA: Defense Acquisition University. http://akss.dau.mil/dag

[DoDI 5000.2] Department of Defense Instruction. (May 2003, May 12). *Operation of the Defense Acquisition System*. Washington, DC: U. S. Department of Defense. http://www.dtic.mil/whs/directives/corres/pdf/500002p.pdf

[DoDD 8500.1] Department of Defense Directive 8500.01E. (2002, October 24-certified current as of 23 April 2007). *Information Assurance (IA)*. Washington, DC: U.S. Department of Defense. http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf

[DoDI 8500.2] Department of Defense Instruction 8500.2. (2003, February 6). *Information Assurance (IA) Implementation*. Washington, DC: U.S. Department of Defense. http://www.dtic.mil/whs/directives/corres/pdf/850002p.pdf

[FAI] *Federal Acquisition Institute—Contracting Officer Technical Representative (COTR).* Retrieved 15 August 2008 from http://www.fai.gov/certification/techrep.asp.

[FAR] *Federal Acquisition Regulation*. http://www.arnet.gov/far/index.html

[FIPS 140-2] Federal Information Processing Standard (FIPS) Publication 140-2. (2001, May). *Security Requirements for Cryptographic Modules.* Gaithersburg, MD: National Institute of Standards and Technology (NIST), U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[FIPS 188] Federal Information Processing Standard (FIPS) Publication 188. (1994, September). *Standard Security Label for Information Transfer*. Gaithersburg, MD: National Institute of Standards and Technology (NIST), U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[FIPS 191] Federal Information Processing Standard (FIPS) Publication 191. (1994, November). *Guidelines for the Analysis of Local Area Network Security*. Gaithersburg, MD: National Institute of Standards and Technology (NIST), U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[FIPS 199] Federal Information Processing Standard (FIPS) Publication 199. (2004, February). *Standards for Security Categorization of Federal Information and Information Systems*. Gaithersburg, MD: National Institute of Standards and Technology (NIST), U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[FIPS 200] Federal Information Processing Standard (FIPS) Publication 200. (2005, July). *Minimum Security Standard for Federal Information Systems*. Gaithersburg, MD: National Institute of

Standards and Technology (NIST), U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[FIPS 201-1] Federal Information Processing Standard (FIPS) Publication 201-1. (2006, Jun 26). *Personal Identity Verification (PIV) for Federal Employees and Contractors*. Gaithersburg, MD: National Institute of Standards and Technology (NIST), U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[FISMA 2002] Federal Information Security Management Act of 2002, 44 U.S.C. § 3541 et seq. http://csrc.nist.gov/policies/FISMA-final.pdf

[Goertzel, 2007] Goertzel, K., et al (2007, July). *Software Security Assurance: A State-of-the-Art Report (SOAR).* Herndon, VA: Information Assurance Technology Analysis Center (IATAC) and Defense Technical Information Center (DTIC).

[Graff & Van Wyk]. Graff, M.G., & Van Wyk, K.R. (2003). *Secure Coding Principles and Practice.* O'Reilly Media, Inc.

[Goertzel, 2008] Goertzel, K.M. & Winograd, T., et al. (2008, July-Final Draft Version 2.0). *Enhancing the Development Life Cycle to Produce Secure Software.* Herndon, VA: Information Assurance Technology Analysis Center (IATAC) and Defense Technical Information Center (DTIC).

[GAO-04-393] Defense Acquisitions. (2004, March). *Stronger Management Practices are Needed to Improve DoD's Software-Intensive Weapons Acquisitions.* Washington, DC: General Accountability Office. http://www.gao.gov/new.items/d04393.pdf

[GAO-04-678] Defense Acquisitions. (2004, May). *Knowledge of Software Suppliers Needed to Manage Risks.* Washington, DC: General Accountability Office. http://www.gao.gov/new.items/d04678.pdf

[GAO BPR Glossary] *GAO BPR Glossary.* Retrieved 15 August 2008 from http ://www.gao.gov/special.pubs/bprag/bprgloss.htm#sectC.

[Graff] Graff, M.G. & van Wyk, K.R. (2003, June). *Secure Coding Principles and Practices.* 1st Ed. Sebastopol, CA: O'Reilly & Associates, Inc.

[GRIMM, 2004] Grimm, J., (2004, November 16). *The Role of CMMI in Mission Assurance.* Retrieved April 20, 2008 from http://www.dtic.mil/ndia/2004cmmi/CMMIGS/NDIARoleofCMMIinMA_final.pdf.

[IBM CO] No Author. (2006). *Certificate of Originality.* Poughkeepsie, NY: IBM Corporation.

[IBM CO] No Author. (ND). *IBM Rational Unified Process.* Poughkeepsie, NY: IBM Corporation.

[IBRAHIM] Ibrahim, L, Jarzombek, J., et al. (2004, September). *Safety and Security Extensions for Integrated Capability Maturity Models.* Washington, DC: Federal Aviation Administration.

[IEEE 100] Institute of Electrical and Electronics Engineers. (date unknown). *The Authoritative Dictionary of IEEE Standards Terms*,

[IEEE 610.12] Institute of Electrical and Electronics Engineers. (1990). *IEEE Standard Glossary of Software Engineering Terminology.*

[IEEE 1012] Institute of Electrical and Electronics Engineers. (2004). *IEEE Standard for Software Verification and Validation.*

[IEEE 1062] Institute of Electrical and Electronics Engineers. (1998). *Recommended for Software Acquisition.* http://standards.ieee.org/reading/ieee/std_public/description/se/index.html

[INPUT 2005] O'Flaherty, T. (2005, December). *The Impact of SwA on the Procurement Process*. INPUT, TargetVIEW, Volume 1, Issue 10. Reston, VA: INPUT.

[ISO/IEC JTC1 SC7] (ND). *Software & System Engineering.* Retrieved October 3, 2008, from http://www.jtc1-sc7.org.

[ISO/IEC 13335-1] ISO/IEC 13335-1:2004. *Information technology -- Security techniques – Management of Information and Communications Technology Security—Part 1: Concepts and Models for Information and Communications Technology Security Management..*

[ISO/IEC 15026] ISO/IEC 15026:1998. *Information Technology—System and Software Integrity Levels.*

[ISO/IEC 15288] ISO/IEC 15288:2008. *Systems Engineering: System Life Cycle Processes.*

[ISO/IEC 15408-1] ISO/IEC 15408-1:2005. *Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model.* http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm

[ISO/IEC 15408-2] ISO/IEC 15408-2:2005. *Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional requirements.* http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm

[ISO/IEC 15408-3] ISO/IEC 15408-3:2005. *Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance requirement.* http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm

[ISO/IEC 15939] ISO/IEC 15939:2007. *System and Software Engineering—Measurement Process.*

[ISO/IEC TR 15947] ISO/IEC TR 15947:2002. *Information Technology—Security Techniques—IT Intrusion Detection Framework.*

[ISO/IEC 17011] ISO/IEC 17011:2004 *Conformity assessment—General Requirements for Accreditation Bodies Accrediting Conformity Assessment Bodies.*

[ISO/IEC 17025] ISO/IEC 17025:2005. *General Requirements for the Competence of Testing and Calibration.*

[ISO/IEC 17799] ISO/IEC 17799:2005. *Information Technology—Security Techniques—Code practice for information security management.*

[ISO/IEC 18028-1] ISO/IEC 18028-1:2006. *Information Technology—Security Techniques—IT Network Security—Part1: Network Security Management.*

[ISO/IEC 18043] ISO/IEC 18043:2006. *Information Technology—Security Techniques—Selection, Deployment and Operations of Intrusion Detection Systems.*

[ISO/IEC 26702 IEEE 1220] ISO/IEC 26702 IEEE Std 1220-2005. (2007). *Systems Engineering - Application and Management of the Systems Engineering Process.*

[ISO/IEC 27001] ISO/IEC 27001:2005. *Information technology -- Security techniques -- Information security management systems – Requirements.*

[ISO/IEC 27005] ISO/IEC 27005:2008. *Information technology -- Security techniques -- Information Security Risk Management.*

[Legal] Legal Definitions. Retrieved October 3, 2008, from http://www.definitions.uslegal.com.

[Martin] Martin, R.A. (2007, March). *Being Explicit About Security Weaknesses.* CrossTalk—The Journal of Defense Software Engineering, Vol. 20, No. 3.

[McGraw] McGraw, G. (2006, February). *Software Security and SOA: Danger, Will Robinson!* http://www.cigital.com/papers/download/bsi12-soa.doc.pdf

[Mochal] Mochal, T. (2005). *The Complete Book of Project-Related Terms and Definitions—Mysteries Explained.* TenStep, Inc: http://www.tenstep.com.

[NDIA] National Defense Industrial Association. (2008). *Engineering for System Assurance*. Washington, DC: NDIA

[NIST IR 7298] National Institute of Standards and Technology (NIST) IR 7298. (2006, April 25). *Glossary of Key Information Security Terms.* Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistir/NISTIR-7298_Glossary_Key_Infor_Security_Terms.pdf

[NIST SP 800-23] National Institute of Standards and Technology (NIST) Special Publication 800-23. (2000, August). *Guidelines to Federal Organizations on Security Assurance and Acquisition/Use of Tested/Evaluated Products*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-28] National Institute of Standards and Technology (NIST) Special Publication 800-28-Version 2. (2008, May). *Guidelines on Active Content and Mobile Code*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-30] National Institute of Standards and Technology (NIST) Special Publication 800-30. (2002, July). *Risk Management Guide for Information Technology Systems*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-32] National Institute of Standards and Technology (NIST) Special Publication 800-32. (2001, February). *Introduction to Public Key Technology and the Federal PKI Infrastructure.* Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-33] National Institute of Standards and Technology (NIST) Special Publication 800-33 (2001, December). *Underlying Technical Models for Information Technology Security.* Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-37] Ross, Ron, Marianne Swanson, Gary Stoneburner, Stu Katzke, and Arnold Johnson. (2004, May). NIST Special Publication 800-37, *Guide for the Security Certification and Accreditation of Federal Information Systems*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-39] National Institute of Standards and Technology (NIST) Special Publication 800-39. (2008, April 3). *DRAFT Managing Risk from Information Systems: An Organizational Perspective*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-53] National Institute of Standards and Technology (NIST) Special Publication 800-53. (2005, February). *Recommended Security Controls for Federal Information Systems*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-55] National Institute of Standards and Technology (NIST) Special Publication 800-55. (2005, July). *Security Metrics Guide for Information Technology Systems*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-60] National Institute of Standards and Technology (NIST) Special Publication 800-60-Rev 1. (2008, August). *Guide for Mapping Types of Information and Information Systems to Security Categories: (2 Volumes) - Volume 1: Guide Volume 2: Appendices*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-61] National Institute of Standards and Technology (NIST) Special Publication 800-61-Rev 1. (2008, March). *Computer Security Incident Handling Guide.* Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-64] National Institute of Standards and Technology (NIST) Special Publication 800-64, Rev 2. (2008, March). *Security Considerations in the System Development Life Cycle*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-67] National Institute of Standards of Technology (NIST) Special Publication 800-67--Rev 1. (2004, June). *Security Considerations in the Information System Development Life Cycle*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-83] National Institute of Standards of Technology (NIST) Special Publication 800-83. (2005, November). *Guide to Malware Incident Prevention and Handling*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-84] National Institute of Standards of Technology (NIST) Special Publication 800-67--Rev 1. (2004, June). *Security Considerations in the Information System Development Life Cycle*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-95] National Institute of Standards of Technology (NIST) Special Publication 800-95. (2007, August). *Guide to Secure Web Services*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NIST SP 800-115] National Institute of Standards of Technology (NIST) Special Publication 800-115. (2007, November). *DRAFT Technical Guide to Information Security Testing*. Gaithersburg, MD: U.S. Department of Commerce. http://csrc.nist.gov/publications/nistpubs/index.html

[NSTISSAM INFOSEC/2-00] National Security Telecommunications and Information Systems Security Advisory Memorandum (NSTISSAM)/2-00. (2000, February 8). *Advisory Memorandum on the Strategy for Using the National Information Assurance Partnership (NIAP) for the Evaluation of Commercial Off-The-Shelf (COTS) Security Enabled Information Technology Products*. Fort Meade, MD: U.S. National Security Agency/.

[NSTISSP No. 11] National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11. (2003, July). *National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology Products*. Fort Meade, MD: U.S. National Security Agency.

[NUPIC] Nuclear Procurement Issues Committee (NUPIC). (2001, July 5). *Document No. 6, Nuclear Procurement Issues Committee Joint Audit Program.* http://www.nupic.com/nupicdoc/NUPIC06.PDF

[OMB 328] Office of Management and Budget. *Standard Form 328, Certificate Pertaining to Foreign Interests.* http://www.dss.mil/isec/sf328.pdf

[OMB M-04-04] Office of Management and Budget. (2003, December 16). *E-Authentication for Federal Agencies*. Washington, DC: Office of the President. http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf

[OMB M-04-16] Office of Management and Budget (2004, July 1). *Software Acquisition.* Washington, DC: Office of the President.

[OMB M-07-18] Office of Management and Budget (2007, June 1). *Ensuring New Acquisitions Include Common Security Configurations.* Washington, DC: Office of the President.

[OWASP Glossary] *Glossary-OWASP*. Columbia, MD: Aspect Security, Inc. Retrieved 15 August 2008 from http://www.owasp.org/index.php/Category:Glossary

[OWASP--SSDC] *OWASP Secure Software Development Contract Annex.* The Open Web Application Security Project. Columbia, MD: Aspect Security, Inc. Retrieved 15 August 2008 from http://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex

[OWASP-SWAS] *OWASP Guide to Building Secure Web Applications and Web Services* (Version 3.0). Columbia, MD: Aspect Security, Inc. Retrieved on 15 August 2008 from http://www.owasp.org/index.php/OWASP_Guide_Project

[PDA-TR] PDA. (Oct 2004, October). *Technical Report 32, Auditing of Suppliers Providing Computer Products and Services for Regulated Pharmaceutical Operations.* Release 2.0, Volume 58, Number 5. https://store.pda.org/bookstore/ProductDetails.aspx?productabbreviation=01032

[Redwine & Davis] Redwine, S.T., Jr. & Davis, N. (2004). *Processes to Produce Secure Software—Toward More Secure Software.* Vol 1. Software Process Subgroup of the Task Force on Security Across the Software Development Lifecycle, National Cyber Security Summit.

[SEI-CMU/SEI-2003-TR-011] SEI. (2003, May). *SEI Case Study: Computer Supplier Evaluation Practices of the Parental Drug Association (PDA).* Technical Report CMU/SEI-2003-TR-011 Software Engineering Institute. http://www.sei.cmu.edu/pub/documents/03.reports/pdf/03tr011.pdf

[SwA CBK] Redwine, S. T.; Baldwin, R. O.; Polydys, M. L.; Shoemaker, D. P.; Ingalsbe, J.A.; Wagoner, L.D. (2006). (2006). *Software Assurance*: *A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software.* Washington, DC: Department of Homeland Security. https://buildsecurityin.us-cert.gov/bsi/docs/Secure%20Software%20Assurance%20CBK%20DRAFT%20v09%20010906.pdf

[US-CCU] Bumgarner, J. & Scott, B. (2006). *The US-CCU Cyber-Security Check List.* Unknown: U.S. Cyber Consequences Unit.

[USCOURTS] U.S. Courts, The Federal Judiciary. (2004). *Judiciary Procurement Program Procedures.* Washington, DC: Administrative Office of the US Courts. http://www.uscourts.gov/procurement/Glossary.pdf

[US PITAC 1999] U.S. President's Information Technology Advisory Committee. (1999, February). *Information Technology Research: Investing in Our Future.* Arlington, VA: National Coordination Office for Information Technology Research and Development. http://www.nitrd.gov/pitac/report/

[US PITAC 2005] U.S. President's Information Technology Advisory Committee. (2005, February). *Cyber Securiy: A Crisis of Prioritization.* Arlington, VA: National Coordination Office for Information Technology Research and Development. http://www.nitrd.gov/pitac/report/

[Webster] *Merriam-Webster OnLine*. Retrieved October 3, 2008, from http://www.merriam-webster.com/cgi-bin/mwwod.pl