DRAFT

Practical Measurement Framework for Software Assurance and Information Security

Version 1.0

October 1, 2008

Acknowledgements

The Department of Defense (DoD), Department of Homeland Security (DHS), and National Institute of Standards and Technology Software Assurance (SwA) Measurement Working Group is composed of government, industry, and academic members. This document was developed by:

Editor and Principle Author:

Nadya Bartol, Booz Allen Hamilton

Additional contributors included:

- Bob Martin, Mitre Corp
- Sean Barnum, Cigital Inc.
- Susan Burgess, Keane Federal Systems
- Richard Gill, Codenomicon
- Michael Kass, NIST
- Ajoy Kumar, DTCC
- Eric Maurice, Oracle
- Mary Ann Davidson, Oracle
- Don O'Neill, Center for National Software Studies
- Dan Reddy, EMC
- Michael Garcia, Boeing
- Michele Moss, Booz Allen Hamilton
- Brian Bates, Booz Allen Hamilton
- Stephanie Shankles, Booz Allen Hamilton
- Ashok Gurumurthy, Hewlett Packard
- Tom McCabe, McCabe Software
- Thomas Neff, Mentorz, LLC
- Paul Kurtz, SAFECode
- Carol Woody, SEI
- Tom Rhodes, NIST
- Dr. Vehbi Tasar, ISC2
- Alexis Tchoumak, JITC/DISA
- Robert Trapp, Booz Allen Hamilton
- Jeff Voas, SAIC
- John Murdoch, University of York
- Jim McCurley, Software Engineering Institute
- Carlos Shaeffers, Defense Contract Management Agency
- Tom Conrad, Department of Navy
- Denis Ahern, Northrop Grumman
- Dan Ferens, Data and Analysis Center for Software
- Mike Denny, Defense Acquisition University
- Mehmet Sahinoglu, Troy University
- Joe Jarzombek, DHS National Cyber Security Division

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

EXECUTIVE SUMMARY

The Practical Measurement Framework for Software Assurance and Information Security provides an approach for measuring the effectiveness of achieving Software Assurance (SwA) goals and objectives at an organizational, program or project level. It addresses how to assess the degree of assurance provided by software, using quantitative and qualitative methodologies and techniques. This framework incorporates existing measurement methodologies and is intended to help organizations and projects integrate SwA measurement into their existing programs.

The framework provides practical guidance for measuring progress toward SwA goals and objectives applied at various levels in an organization. However, the SwA discipline is still an evolving field, and this document does not attempt to answer all questions about SwA measurement. Rather, it provides an initial approach to begin measuring SwA. Further research is required into specific measurement methods and techniques to mature existing SwA measurement approaches and tools.

SwA is interdisciplinary and relies on methods and techniques produced by other disciplines including project management, process improvement, quality assurance, training, information security/information assurance, system engineering, safety, test and evaluation, software acquisition, reliability, and dependability. This framework focuses principally, though not exclusively, on the information security viewpoint of SwA. Many of the contributing disciplines of SwA enjoy an established process improvement and measurement body of knowledge, such as quality assurance, project management, process improvement, and safety. SwA measurement can leverage measurement methods and techniques that are already established in those disciplines, and adapt them to SwA. The information assurance/information security discipline is less mature in the area of measurement. This document focuses on information assurance/information security aspects of SwA to help mature that aspect of SwA measurement.[1]

The common measurement framework provides information on creating SwA measures but is not prescriptive in nature. It does not prescribe any specific measures nor does it prescribe a specific measurement process. It is intended to guide the reader in identifying the essential stakeholder goals and information needs to begin a measurement program. It identifies various stakeholders and provides example goals or information needs for them. A number of representative key measures for different stakeholder groups such as executives, developers, vendors, suppliers, program managers, acquirers, buyers, and practitioners are included to help organizations assess the state of their SwA efforts during any stage of a project.

This framework provides an integrated measurement approach which leverages five existing industry approaches that use similar processes to develop and implement measurement. These methodologies were selected because of their widespread use among the software and systems development community and the information security community. Included is a common measure specification table that illustrates the similarities among these approaches.

---

[1] For the purposes of this document information assurance/information security will be referred to as "information security" or "security."

The document discusses use of enumerations, such as Common Vulnerabilities and Exposures (CVE), Common Control Enumeration (CCE), Common Weakness Enumeration (CWE), and Common Attack Pattern Enumeration and Classification (CAPEC), and provides corresponding measures examples. Enumerations help identify specific software-related items that can be counted, aggregated, evaluated over time and used for the assessment of a variety of aspects of SwA. Measures examples provided in the document include specific measures, information needs, and benefits to assurance that these measures can produce.

Organizations can use common measurement framework to implement SwA and security measurement at the desired organizational level, tailor it to the organizational stakeholders, and integrate into existing measurement and risk management activities.

1. INTRODUCTION

Dependency on information technology makes Software Assurance (SwA) a key element of national security and homeland security. 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 application products. *Software enables and controls the nation's critical infrastructure, and in order to ensure the integrity of key assets within that infrastructure, the software must be reliable and secure*. While methods exist that guide organizations in assessing SwA of the code which they are developing or acquiring; quantifying this assurance has been a challenge.

Poor quality manifested in security vulnerabilities has a wide range of undesirable operational and economic effects. There are the obvious, well publicized costs of system and data breaches, and there are also hidden economic impacts. For example, security vulnerabilities, unlike other types of vulnerabilities, almost always have to be patched. The scarce resources both vendors and customers apply to issue, test, and apply patches could be used on something else which yields a better return. The opportunity cost of applying security patches is that those doing so are not performing other valuable security activities such as reviewing activity logs or hardening configurations. Fixing security vulnerabilities is an unbudgeted cost to many organizations.

Traditional measurement approaches for systems and software do not include SwA and security measurement. While they address quality, they fail to address SwA and the security aspects of quality. Therefore the traditional approaches miss a large source of poor quality, poor software, and system performance issues which are their results. This document focuses on bridging that gap. It aims to assist practitioners in deploying SwA and security measurement, to improve understanding of performance management, and to help create more secure and reliable systems. This document concentrates mainly, though not exclusively, on the security viewpoint of SwA measurement.

A well-known management proverb states that "what is measured is managed." Measurement can help organizations understand how well the software or a system provides assurance and point out opportunities for further improvement. SwA measurement can assist projects and organizations in the following ways:

- Provide quantifiable information about SwA to support enterprise risk management and risk-based decision making
- Articulate progress towards goals and objectives in a consistent way
- Provide a repeatable, quantifiable way to assess, compare, and track improvements in assurance
- Focus SwA activities on risk mitigation in order of priority and severity
- Facilitate adoption and improvement of secure software design and development processes
- Provide quantifiable inputs into software and system assurance cases
- Respond to threats as identified throughout the System Development Lifecycle (SDLC) and ultimately reduce the numbers of vulnerabilities introduced into software code during development

- Assess trustworthiness of the system by verifying, validating, and documenting if the system or software does what it was intended to do and is not exploitable for other uses
- Make informed decisions in the SDLC related to information security compliance, performance, and functional requirements/controls
- Determine if security related functional and performance trade-offs have been defined and accepted
- Provide an objective context and the means of comparing and benchmarking projects, divisions, organizations, and vendor products
- Identify, document, and monitor fulfillment of roles and responsibilities related to implementing and monitoring SwA practices.

1.1 Background

The Department of Homeland Security (DHS), Department of Defense (DoD), and National Institute of Standards and Technology (NIST) are joint co-sponsors of the SwA Program whose objective is to address the concerns of poor-quality, unreliable, and non-secure software. The SwA Program adopted a multi-dimensional approach that encompasses people, process, technology, and acquisition. The SwA Measurement Working Group focuses on identifying and tailoring methods and techniques helpful for assessing the degree of assurance provided by software, using quantitative and qualitative methodologies.

The Measurement Working Group consists of representatives from government, industry, and academia. This document culminates the efforts of the working group to create a SwA measurement framework.

1.2 Purpose and Scope

This document describes a SwA and security measurement framework that:

- Provides organizing principles for measurement of progress toward SwA goals and objectives
- Is flexible and scalable for varying levels of organizational context (e.g. individual projects, programs, or enterprises)
- Emphasizes practical implementation
- Leverages existing measurement and risk management frameworks where possible.

The document does not answer all questions about SwA measurement; rather it provides an initial approach to begin measuring SwA. Further research is required into specific measurement methods and techniques to mature existing SwA measurement approaches and tools.

This document targets a variety of audiences interested in the subject of SwA including executives, developers, vendors, suppliers, program managers, acquirers, and buyers. The processes, methods, and techniques described in this document are suggestions for how to

establish a SwA measurement program or how to integrate it into an existing measurement program.

The content of this document is strictly informative, in the sense that the document does not contain requirements and does not provide a standard with which one complies.

The common measurement framework leverages existing measurement methodologies and applies them to SwA measurement. It is intended to help projects and organizations integrate SwA measurement into their existing measurement efforts, rather than to establish a standalone SwA measurement effort within an organization. This document references these methodologies, demonstrates commonalities among them, and proposes some broadly applicable SwA measures to be considered for use.

The common measurement framework provides information on creating SwA measures but is not prescriptive in nature. It does not provide specific descriptions of existing measurement methodologies nor does it propose an exhaustive list of SwA measures. Implementers and users of SwA measures are encouraged to review and study the "base" methodologies leveraged in this document from the respective sources to ensure they have selected the most appropriate ones for their individual programs. The framework described in this document is applicable to a variety of scopes and operating environments – each unique environment will require a tailored set of measures and approaches, some of which can be gleaned from this document. The results will need to be interpreted for each individual environment including the intended manner in which software is implemented and used.

SwA is interdisciplinary and relies on methods and techniques produced by other disciplines. This concept is depicted in Figure 1.

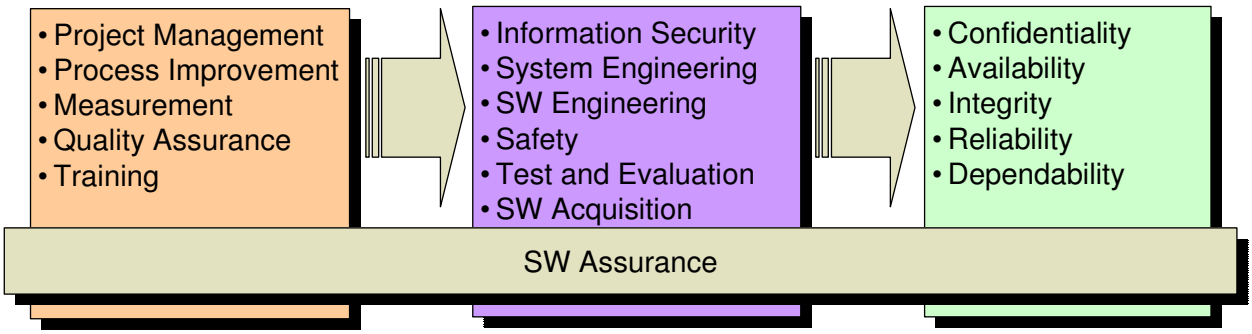

Figure 1. Cross-disciplinary Nature of SwA

Many of the composite disciplines of SwA enjoy an established process improvement and measurement body of knowledge, such as quality assurance, project management, process improvement, and safety. SwA measurement can leverage measures and measurement methods and techniques already established in those disciplines and adapt them to SwA. By comparison, the discipline of information assurance/information security measurement is less mature. This

document focuses on information assurance/information security aspects of SwA to help mature that aspect of SwA measurement.[2]

The measurement framework described in this document can be applied to both SwA and information security measurement efforts. Its use can help facilitate risk-based decision making by providing quantitative information on an organization's performance in the areas of SwA and information security.

1.4 Assumptions

This document assumes that the audience has knowledge of information security/information assurance as well as system and software engineering disciplines; therefore it does not intend to explain the founding principles of those disciplines. It also assumes that the readers understand the basics of measurement so it does not fully explain the measurement methodologies leveraged herein.[3] The document targets a variety of audiences, including federal, state and local governments and commercial organizations.

1.5 Key Definitions

"Software assurance," "measure," and "measurement" are key terms used in this document. In recent years, many standards and industry organizations have been adopting the terms "measure" to describe the result and "measurement" to describe the process of using quantifiable data to support decision making and accountability, while some use the term "metric." The Measurement Working Group decided to follow many industry examples and adopt the terms "measure" and "measurement" as defined below by authoritative sources. Appendix C lists definitions for other terms used in this document.

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 lifecycle, and the software functions in the intended manner. [CNSS Instruction No. 4009]
Measure	Variable to which a value is assigned as the result of measurement [ISO/IEC 15939]
Measurement	Set of operations having the object of determining a value of a measure [ISO/IEC 15939]

[2] For the purposes of this document information assurance/information security will be referred to as "information security" or "security."

[3] Information on system and software measurement can be found through the Practical Systems and Software Measurement (PSM) and Software Engineering Institute (SEI). Information about information security measurement can be found through the National Institute of Standards and Technology (NIST).

1.6 Principles

The SwA measurement approach adopts certain key principles. These principles should guide an organization as it tailors, introduces, and evolves its SwA measurement program:

- SwA measurement is a composite discipline which, to be most effective, should be integrated into an organization's existing measurement and risk management practices.
- SwA can be implemented by integrating SwA goals and objectives at varying levels within the organization (e.g. project, program, or enterprise-wide).
- A SwA measures development and implementation initiative can be incorporated into whatever measurement methodology is already being used.
- SwA measurement must satisfy information needs for a variety of stakeholders/audiences, including executives, developers, vendors, suppliers and acquirers.
- Each stakeholder group will require tailoring of specific measures based on each group's information needs.
- Different measures targeting different stakeholders may use the same information originating from the same data sources to facilitate multiple uses of the same set of data.
- SwA measures must be effective, practical, and worth the investment of resources in the long term.
- Implementation of SwA measurement should incorporate automation to assist analysts in data collection, analysis, and reporting.
- Each phase of the SDLC, acquisition life cycle, or any other life cycle introduces an opportunity to measure SwA and improve its results.[4]
- For the purposes of this document, the term "measurement" applies to both quantitative and qualitative measurement methodologies.
- SwA measurement principles can be expanded to encompass system assurance
- All considerations applicable to any measurement process apply to SwA measurement. For example, the quality of SwA measures and measurement processes is dependent upon the quality of information and execution of SwA activities and is subject to the "garbage in – garbage out" rule.

[4] The SwA measurement framework can be used with any lifecycle.

1.7 Document Structure

The remaining sections of this document discuss the following:

- Section 2, Common Measurement Framework, describes the stakeholder goals and information needs, key measures, and presents an abbreviated Common Measurement Framework. Those interested in understanding what measurement can reveal and what measures to use should review this section.

- Section 3, Implementing SwA Measures, provides high level guidance for implementing SwA measurement program. Those interested in a summary of how to create or improve a measurement program should review this section.

- Section 4, Data Sources for SWA Measurement**,** summarizes enumerations and their use for reducing weaknesses, assessing development activities, measuring vulnerability mitigation, assessing deployed configurations, and assessing skills. Those interested in finding out how to use a variety of data to support measurement should review this section.

This document contains five appendices.

- Appendix A list references used in this document.

- Appendix B provides a list of acronyms used in this document.

- Appendix C lists definitions.

- Appendix D summarizes several different types of information security measurement methodologies, system and software development measurement methodologies, measurement frameworks, frameworks that provide a foundation for measurement, qualitative assessment methods, process and controls standards and guidance, and other resources.

- Appendix E, Common Measure Specification, includes details of the common measure specification with the definitions used within those methodologies that comprise the Common Measurement Framework.

2. COMMON MEASUREMENT FRAMEWORK

The common measurement framework provides an organizing approach for SwA measurement. The principal elements of this framework are:

- A generic set of common stakeholder types, identifying their perspectives, concerns and SwA questions;
- Typical goals and information needs of those stakeholders;
- Example measures appropriate for those goals;
- Example benefits which convey how the measures should support the goals;
- Definition of a Common Measure Specification to integrate separate measurement methodologies;
- Mapping the suggested measures against existing, well known measurement methodologies to demonstrate compatibility of these methodologies.

The framework suggests measures that can be selectively applied as a useful starting point for a SwA measurement program. These SwA measures should augment existing measurement programs to support stakeholder insight and management of SwA.

Many of today's organizations use measures to quantify some aspects of their performance. Several established measurement approaches exist in the system, software and information security industries, along with additional approaches emerging with broad industry support. In this approach-heavy environment, introducing a completely new one just for SwA is counterproductive. Rather than creating yet one more way that is slightly different from the others, the common measurement framework leverages five prominent existing measurement approaches used within software and system engineering and information security industries.

Practitioners can leverage this framework to integrate SwA measurement into existing measurement efforts. This is done by expanding the content of their organization's measurement activities to include SwA while using established processes and methodologies. If an organization is not currently using any measurement processes, that organization should select one for implementation that is the most appropriate for the organization. For example, selecting a specific approach may provide a competitive edge within a particular industry context. Users of the framework should ensure that the content is appropriate and that specific SwA measures are used in concert with other measures, regardless of which measurement approach is used.

The common measurement framework guides organizations toward creating their measures but does not prescribe any specific ones nor does it prescribe a specific measurement process (e.g., measures creation, collection, analysis, reporting, and using those measures as an input into decision making). Any of the approaches leveraged by the Framework can be used to guide the measurement process as long as the measures are based on organizational/business goals and objectives and are used to facilitate improvement. Stakeholders should be involved in the process of measures development and implementation as early in the process as possible.

2.1 Stakeholder Goals and Information Needs

Different stakeholder groups may be interested in gaining a variety of insights from measurement. Stakeholders differ by their organization's role within the software supply chain, their position within an organization, and their specific job description. This document assumes the following two types of organizations are interested in SwA measurement:

- Supplier[5] – an individual or an organization that offers software and system-related products and services to other organizations.

- Acquirer[6] – an individual or an organization that acquires software and system-related products and services from other organizations.

It is important to note that in diverse organizations, both Supplier and Acquirer groups may be found internally. Furthermore, a broad set of individual stakeholders is expected to exist within each Supplier and Acquirer organizations. At a minimum, those will encompass the following:

- Executive Decision Maker – a leadership individual who has authority to make decisions and may require quantifiable information to understand the level of risk associated with software to support decision-making processes.

- Practitioner – an individual responsible for implementing SwA as a part of their job.

Individuals within each generic stakeholder group may have different interests and needs based on their individual roles, responsibilities, and job descriptions. This document refers to Supplier, Acquirer, Executive Decision Maker, and Practitioner as "generic SwA stakeholders."

Stakeholder "Goals" or "Objectives," sometimes expressed as "Information Needs" define the information a stakeholder wishes to gain from the measurement activity. Those needs will drive which measures are selected, developed and eventually implemented. Table 2-1 provides example goals and information needs for generic SwA stakeholders. While the table assigns a specific generic stakeholder group to each example, it is entirely possible for multiple stakeholder groups to be interested in the same items. On those occasions, goals/information needs are listed once under one of the applicable stakeholders. ***The readers of this document are encouraged to review the entire table to identify goals/information needs that speak best to their individual environments and needs.***

---

[5] This includes software developers, program managers, and other staff working for an organization that develops and supplies software to other organizations.

[6] This includes acquisition officials, program managers, system integrators, system owners, information owners, operators, DAAs, certifying authorities, independent verification and validation (IV&V), and other individuals who are working for an organization that is acquiring software from other organizations.

Table 2-1. SwA Measurement Stakeholder Example Goals and Information Needs

Stakeholder	Goals/Information Needs
Supplier	• Identify and prioritize defects in the design, architecture, and code to reduce risks of future exploitation of software • Understand the level of assurance vs. residual risk associated with making decisions at each phase of the SDLC • Identify software defects that may be exploited in the future • Determine if SwA and security requirements are being planned and implemented • Reduce opportunity for malicious software and undesirable behaviors • Ascertain that defects have been appropriately addressed in a timely manner • Monitor planning and implementation of SwA and security activities in SDLC • Understand organization's strengths and weaknesses in SwA • Ascertain that security is integrated into the SDLC as early as possible • Identify appropriate staffing required to guarantee on time delivery that would appropriately address SwA needs • Enable quantifiable comparison with competitors to enhance organization's reputation and achieve product and service differentiation from competition • Ascertain understanding of operational environment and integration of accidental and intentional use, misuse, abuse, and threat considerations into the SDLC activities • Identify causes of poor design and coding practices that may be introducing vulnerabilities into software • Demonstrate effectiveness and repeatability of assurance processes
Acquirer	• Cost effectively integrate SwA considerations into the acquisition and development lifecycle • Ascertain that contracting officers have a good understanding of information security requirements of the Federal Acquisition Regulation (FAR) • Validate that contracting officers request assistance from information security specialists when required • Validate that requirements for compliance with FISMA, OMB A-130, Appendix III, and NIST standards and guidelines have been integrated into procurement language • Gain insight into how the software to be acquired will impact the organization's SwA and security posture • Validate that SwA considerations are included in the procurement • Validate that SwA requirements defined in the RFP and in the contract have been satisfied throughout product and service delivery

Stakeholder	Goals/Information Needs
	• Ascertain that the supplier has a process for testing and reviewing software for vulnerabilities that has been and will be applied throughout the life of the contract • Ascertain that the delivered product arrives and performs as expected • Ascertain that the supplier has imposed appropriate SwA practices on its own suppliers • Ascertain the integrity of COTS and open source packages • Ascertain SwA requirements are explicitly addressed in a solicitation and considered during the evaluation process • Verify that SwA requirements are integrated into SwA Requirements Document and implemented in the system • Assure that the staff delivering products and services are qualified to implement SwA practices • Monitor impact of SwA and security on business and mission support
Executive	• Understand and manage risks created by software development, acquisition, and operation • Establish costs and likelihood of breaches (e.g., loss of revenue, opportunity costs, loss of credibility, legal consequences) • Establish cost of remediation activities • Establish benefits of SwA • Compare costs of building SwA in vs. correcting it after the fact • Compare risks across different vendor or custom products • Gain insights into aspects of overall SwA and security posture of the organization or its component(s) • Understand the impact of SwA on regulatory compliance
Practitioner	• Understand impact of SwA and security on business and mission support • Identify vulnerabilities exploitation of which would have an unacceptable impact on the mission • Gain insight into the potential and actual cost of vulnerabilities in software (e.g., costs of leaving vulnerabilities or removing them) • Provide inputs into risk management

2.2 Example Measures[7]

To be useful to projects and organizations, measurement should help answer stakeholder goals/information needs or questions that provide insights into an organization's performance.

[7] Some of the example measures were developed in collaboration with NIST.

Different stakeholders may have different goals or questions and may gain different information from the same measures.

The SwA Measurement Working Group identified a number of ***example*** measures for the stakeholder groups defined in section 2.1. These are ***generic*** in that they can be tailored for and used by a variety of projects and organizations. Tables 2-2, 2-3, 2-4, and 2-5 list ***examples*** of measures that are ***generally*** applicable to the Supplier, Acquirer, Executive, and Practitioner stakeholder group. However, any of the measures listed under individual tables may prove useful to other stakeholders. The placement of measures in specific tables is notional and does not preclude use of measures by other stakeholders. The ***example*** measures provided in this section can be used to assess the state of SwA efforts for a system or software development project. It should be noted that these measures do not provide an all-inclusive list of measures for SwA, may not be applicable to every situation, and are not prescriptive as far as implementation. Each organization or project should consider their specific goals and develop or adopt measures that correspond to the goals. The measures provided in this section can be used as a resource in that process. Further information about measures implementation is provided in Section 3.

To be useful and actionable, these measures are intended to help answer the following five questions:

- What are the defects in the design and code that have a potential to be exploited
- Where are they
- How did they get there
- Have they been mitigated
- How can they be avoided in the future.

Table 2-2 displays the ***examples*** that apply ***mainly*** to the Supplier stakeholder, organized by project activity and provides corresponding information needs and benefits.[8]

[8] This list is not intended to be comprehensive.

Table 2-2. Example Supplier Measures

Project Activity	Measures	Information Need	Benefit
Requirements Management	• Number or percent of functional and non-functional security and SwA requirements mapped to design	• Determine if functional and non-functional security and SwA requirements are being implemented in addition to being planned	• Provides insight into inclusion of security and SwA requirements in early releases and into security and SwA requirements traceability
	• Number of threats identified in the threat model • Number of relevant attack patterns[9] (attack surface)	• Understand the breadth of attacks that the system could experience to inform functional and non-functional security and SwA requirements	• Provides a structured approach for threat modeling • Facilitates reduction of attack surface
	• Percent of SwA requirements added/modified/deleted relative to the total number of baseline requirements	• Assess SwA requirements stability	• Provides insight into complexity of SwA implementation • Provides insight into the degree of predictable behavior • Indicates the degree to which SwA can be tested
	• Percent of data entities with full validation constraints defined	• Assert that all data entities have full data validation criteria defined	
Design	• Number of entry points for a module (should be as low as appropriate	• Reduce opportunity for back doors	• Ascertains that future application handles data inputs as required • Reduces opportunity for exploits • Reduces attack surface
	• Percent of data input components that positively validate all data input	• Determine if data validation is handled as required	

[9] "Relevant" attack patterns are those attack patterns that target specific platforms, infrastructure environment, or other technology through which the application can be exploited.

Project Activity	Measures	Information Need	Benefit
	• Number of defects ranked by severity[10], the area of code in which they were found, and their origin[11] (it is a higher risk to have the defects in between components, unit seams, or other interfaces)	• Identify origins of defects	
Development	• Number of discovered defects (e.g., CWEs or CVEs) present in the system that would make it vulnerable to specific attacks (e.g. buffer overflows and cross-site scripting)[12] • Percent of discovered defects that were fixed • Number of changes between design, code and requirements	• Ascertain that defects are fixed when found and not left open until testing and deployment	• Minimizes development and maintenance rework costs • Reduces the chances of introducing vulnerabilities • Increases predictability of software behavior
	• Number of user-controllable inputs • Number of times high risk statements (e.g., commands, APIs) are used • Percent of code coverage for which appropriate exception handling has not been created	• Assure that the application performs exception handling as required	
Test	• Number and percent of modules that contain vulnerabilities that may be exploited in the future	• Identify software defects that may be exploited in the future	• Provides insight into risk of the system being exploited when in production • Increased resiliency and survivability

[10] CVSS can be used to determine severity.

[11] Defect origin may represent an injection point during the SDLC.

[12] CAPEC can be used to identify and categorize attack patterns to further expand this measure.

Project Activity	Measures	Information Need	Benefit
	• Number and percent of tests that evaluate application response to misuse, abuse, or threats • Number and percent of tests that attempt to subvert execution or work around security controls • Percent of security controls and SwA requirements covered by tests • Number and percent of external messages with complete input validation • Percent of untested source code related to security controls and SwA requirements[13]	• Assess test coverage of security controls and SwA requirements coverage	• Increases predictability of software behavior • Provides insights into how extensive is the security and SwA portion of the test • Indicates a need for additional security controls in implemented system
	• Number of relevant attack patterns covered by executed test cases • Density of test cases identified and executed per relevant attack pattern • Number of relevant misuse/abuse case requirements covered by test cases using attack patterns	• Ascertain that testing is conducted against all relevant attack patterns	• To ensure that testing has been conducted against all attacks relevant to the system, including all relevant steps, techniques, and varieties • Provides a basis for understanding the degree of code coverage during test
Entire SDLC	• Results of a capability-based appraisal that included assurance practices • Percent of defects discovered during the previous lifecycle phase that remain open	• Monitor planning and implementation of SwA • Assess capabilities to deliver secure products and services • Assess effectiveness of correcting defects when found	• To communicate capabilities to potential acquirers • To assure closure of defects when found

[13] This measure should be used with consideration of the criticality of the security controls and SwA requirements.

Project Activity	Measures	Information Need	Benefit
	• Number of milestone meetings with security and SwA experts participating per phase of SDLC	• Monitor planning and implementation of SwA and security activities in SDLC	• To ascertain that appropriate experts are involved in decision-making activities throughout the SDLC

Table 2-3 lists ***examples*** that are ***mostly*** applicable to the Acquirer stakeholder group. These measures can be used to assess the state of SwA efforts as a part of an acquisition. These measures are generic in that they can be used for a variety of projects. These measures are intended to help answer the following questions: "Have SwA activities been adequately integrated into the organization's acquisition process", and from an external perspective "Have SwA considerations been integrated into the SDLC and resulting product by the Supplier?" Table 2-3 displays the measures per acquisition activity and provides corresponding information needs and benefits.

Table 2-3. Example Acquirer Measures

Acquisition Activity	Measures	Information Need	Benefit
Planning	• Number and percent of acquisition discussions that include SwA representative • Number and percent of contracting officers who received training in the security provisions of the FAR	• Ascertain that SwA considerations are included in the procurement	• Provide for the procurement to include appropriate SwA considerations and requirements
Contracting	• Applicable SwA requirements are included in the solicitation (yes/no) • Contract language for validating SwA requirements have been met is included in the solicitation (yes/no)	• Ascertain that SwA requirements are explicitly addressed in solicitation • Ascertain that SwA requirements are considered during the evaluation process	• Facilitates effective selection of Supplier capable of delivering required level of SwA
	• SwA requirements for sub-contractors are stated in the Subcontracting Plan and are addressed in Subcontracting Agreements (yes/no)	• Ascertain that supplier manages supply chain risk from software	

Acquisition Activity	Measures	Information Need	Benefit
Implementation and Acceptance	• Percent of documented Supplier assurance claims outlined in the RFP response verified through testing, inspection, or other methods	• Verify that the product functions as claimed	• Risks associated with the software are identified and documented
	• Number and percent of Supplier positions filled with personnel possessing required qualifications and certifications • Security role is included in the configuration management process (yes/no) • Number and percent of Supplier project staff trained on the principles of SwA	• Ascertain that Supplier project is staffed and structured to implement SwA	• Supplier project staff are aware of SwA considerations and cognizant of associated requirements
	• Number and percent of accepted supplier deliverables	• Gauge the amount of rework that supplier is engaging in to satisfy customer requirements	• Provides a high-level measure of quality of supplier deliverables

Table 2-4 lists ***examples*** that are ***mostly*** applicable to the Executive stakeholder group. These measures can be used to provide information to Executives about the risks to their organization associated with software. These measures are intended to help answer the following question: "Is the risk generated by software acceptable to the organization?"

Table 2-4. Example Executive Measures

Measures	Information Need	Benefit
• Number and percent of patches published on announced date • Number and percent of patch reloads	• Determine if a vendor is meeting customers' planning cycles and expectations while delivering quality	• Understand the level of risk and potential liability generated by acquired/integrated product • Insight into risk exposure

Measures	Information Need	Benefit
• Time elapsed for Supplier to fix defects in operational software based on severity of vulnerability and its actual or anticipated impact[14] • Number of past system breaches or data compromises traced to a specific vendor product	• Understand the impacts of data compromises caused by Supplier products	and vendor responsiveness • Gain insights into internal processes that require change(s) to reduce risks • Minimize risks created by acquired/integrated software
• Number of known defects by type and impact[15] • Number and percent of applicable[16] defects (weaknesses - known CWEs and vulnerabilities – known CVEs) remediated before the system is operational of total universe of applicable defects that could have been introduced throughout development	• Understanding of SwA that the system provides • Gain insights into risk exposure from acquired/integrated product	
• Cost to correct vulnerabilities in operational applications o Cost to fix known vulnerabilities discovered through code analysis o Cost to correct known security control deficiencies in operational applications • Cost of fixing defects before system becomes operational • Cost of individual data breaches o Discovery, notification, and response o Regulatory fines o Lost productivity o Liabilities o Brand damage/lost customers	• Establish cost of fixes and breaches	• Provide a business case for devoting resources to SwA early within the SDLC

[14] This measure is more appropriate for custom-built rather then packaged software.

[15] Common Vulnerability Scoring System (CVSS) can be used to articulate the impact of exploitation for known vulnerabilities.

[16] "Applicable" vulnerabilities are those vulnerabilities of specific platforms, infrastructure environment, or other technology through which the application can be exploited. The level of risk caused by individual vulnerabilities also may be taken into account when deciding which vulnerabilities are "applicable."

Measures	Information Need	Benefit

• Cost of SwA throughout SDLC phases
  o SwA/security engineer LOE
  o Cost per individual fix
  o Time/schedule delays[17]
	• Compare cost of building SwA in vs. correcting it after the fact	

Table 2-5 lists ***examples*** that are ***mostly*** applicable to the Practitioner stakeholder group. These measures are intended to help answer the following question: "How well are current SwA processes and techniques mitigating software-related risks?"

Table 2-5. Example Practitioner Measures

Measures	Information Need	Benefit
• Number and percent of known vulnerabilities (CVEs) discovered post-implementation that could have been remediated before implementation, arranged by impact of exploitation[18] • Number and percent of relevant high impact vulnerabilities (CVEs) present in the system • Number of patches installed and other mitigating measures implemented since the last architecture and design review • Percent of exploitable CVEs that were addressed through various types of mitigating strategies, such as patches and service packs and mitigating controls	• Identify vulnerabilities, exploitation of which would have an unacceptable impact on the organization's mission • Ascertain that all appropriate mitigating strategies have been collectively applied • Provide input into risk management	• Better ability to prioritize resources for fixing vulnerabilities • Focus vulnerability mitigation to exploitable vulnerabilities vs. all vulnerabilities regardless of their applicability

[17] These measures are useful for those SwA and security practitioners who are seeking help in convincing their leadership to integrate SwA into SDLC. This measure should not be used by those who have already succeeded at this integration.

[18] CVSS can be used to assess and describe impact of exploitation.

Measures	Information Need	Benefit
• Ratio of actual and planned costs of maintaining SwA after implementation including mitigating vulnerabilities	• Gain insight into the potential and actual cost of vulnerabilities in software (leaving them in or removing them)	• Provides insight into cost and impact of SDLC implementation on business and mission
• Number of vulnerabilities (CVEs and new) discovered over predefined time frame (month, 6 months, year, etc) • Number of people who discovered vulnerabilities (both over time and the total number) • Number of discovered vulnerabilities by type and severity (both over time and absolute)[19]	• Gain an understanding of the vulnerability discovery in software	• Provides an input into the determination of assurance
• Timeliness and quality of vendor patches o Number and percent of patches published on announced date o Number and percent of patch reloads • Time elapsed for Supplier to fix defects in operational software based on severity of vulnerability and its actual or anticipated impact[20]	• Determine if a vendor is meeting customers' planning cycles and expectations while delivering quality • Understand the impacts of data compromises caused by Supplier products	• Insight into risk exposure and vendor responsiveness
• Number of weaknesses (e.g., CWEs) determined applicable for the given system configuration • Number and percent of instances of applicable CWEs found in software o Number of present publicly known weaknesses o Density of a weakness against a context-specific measure of code, such as lines of code o Number/lines of code o Number/number of APIs o Number/interaction with a database	• Prioritize weaknesses for mitigation based on the weakness type and the applicable system configuration • Understand the extent to which weaknesses are found in code and help identify what must be mitigated	• Provides assurance that weaknesses are mitigated in order of exploitability based on the specific system configuration to ensure the introduction of corresponding vulnerabilities is avoided • Provides information for prioritizing mitigating controls

[19] CVSS can be used to prioritize by severity.

[20] This measure is more appropriate for custom-built rather then packaged software.

Measures	Information Need	Benefit
• Percent of compliant configurations/system components • Percent of non-compliant configuration/system components ordered by impact of non-compliance[21]	• Establish that software is configured according to specific minimum configuration requirements or stronger • Determine impact of non-compliance with system configuration requirements	• Measurable proof of compliance or non-compliance with specific configuration requirements or technical security controls
• Percent of programmers who have had their secure programming skills assessed	• Understand SwA qualifications of your labor pool	• Increased awareness of the need to educate programmers on security and SwA aspects of their jobs

2.3 Integrated Measurement Approach

SwA measurement has to interact, and be interoperable with, the measurement methods used for other disciplines that comprise SwA. Many of those disciplines use widely known software and system measurement methodologies, such as Practical Software and System Measurement (PSM) and the Capability Maturity Model Integration (CMMI). Emerging measurement approaches now exist for information security measurement that are compatible with system and software measurement methodologies. The common measurement framework described in this document integrates five broadly used and supported measurement approaches that enable SwA measurement integration into other measurement programs rather than creating an individual stove-piped method specifically for SwA. These approaches are comparable and interoperable so that any of them can be used to develop SwA measures and integrate them into an existing measurement program or to implement a new measurement program. Organizations should either integrate SwA into their current approach or select one of these approaches to create an overarching measurement program with a SwA component. The following are the five industry approaches integrated into the Common Framework:

- **Draft National Institute of Standards and Technology (NIST) Special Publication (SP) 800-55, Revision 1,** *Performance Measurement Guide for Information Security*

- **International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 27004** *Information technology - Security techniques - Information security management measurement*

[21] Common Configuration Scoring System (CCSS) could be useful for this measure.

- **ISO/IEC 15939,** *System and Software Engineering - Measurement Process*, also known as Practical Software and System Measurement (PSM)

- **CMMI®[22] (***Capability Maturity Model Integration)* Measurement and Analysis Process Area

- **CMMI® GQ(I)M** – *Capability Maturity Model Integration Goal Question Indicator Measure.*

These methodologies were selected because of their widespread use within the software and system development community (PSM and CMMI®) and information security community (NIST). The ISO/IEC standard was selected due to the broad industry use of the corresponding requirements standard – ISO/IEC 27001, Information Security Management System - Requirements which should facilitate swift acceptance of ISO/IEC 27004. A high level summary of these and other existing measurement methodologies and related sources is provided in Appendix D.

Use of existing methodologies to implement SwA measures is intended to promote continued collaboration across domains that contribute to SwA without creating yet another measurement approach exclusively for SwA. This approach will facilitate interaction among software and information security professionals to identify and implement measures that address SwA by:

- Providing a translation mechanism for different stakeholder communities to understand each others' measurement approaches and results;
- Supporting reuse of existing measures originating from other measurement approaches;
- Allowing stakeholder communities to continue using their methods and expand their view into measurement; and,
- Identifying gaps for further development.

This Framework can be used to develop measures and design and implement measurement programs.

2.4 Common Measure Specification

SwA measures can be integrated into an existing measurement program by leveraging the Common Measure Specification to ensure that SwA information needs and questions are addressed. The basic process for developing each individual measure consists of:

- Stating goals/information needs/questions

- Identifying data sources (entities) and individual data (attributes) that will support measurement

[22] Capability Maturity Model, Capability Maturity Modeling, and CMM are registered in the U.S. Patent & Trademark Office.

- Analyzing the relationship between those two groupings of concepts to create a series of measures that describe this relationship.

The Common Measure Specification is a crosswalk of specifications, templates, forms and other means of documenting individual measures provided by the five industry approaches that were leveraged to create the Framework. It correlates individual elements which specify a measure, defined in these industry approaches. The Common Measure Specification maps the ways in which each approach documents an individual measure within a single matrix. Table 2-6 provides an abbreviated version of the Common Measure Specification. The full version is provided in Appendix E.

Readers of this document can use this specification to explore commonalities and differences between measurement approaches they use within their respective domains and to translate measures from other domains into the methodology they currently use. Table 2-6 illustrates that there are many similarities among the selected methodologies, and where different terms may be used to communicate similar concepts.

Light shaded cells indicate a lack of corresponding item in the mapped methodologies. ISO/IEC 15939 and ISO/IEC 27004 provide the most detailed and comprehensive specifications among the methodologies.

Table 2-6. Abbreviated Common Measure Specification

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Goal/ Objective/ Information Need Description	**Information Need**	**SG 1: SP 1.1** Establish measurement objectives.	**Objective**	**Purpose of measure**	**Goal and Objective**
	Information Category			**Control or Control Objective**	
Measurable Concept/ Question	**Measurable Concept**		**Question**		
Entities/ Attributes	**Relevant Entities**		**Inputs - Data Elements**	**Object of Measurement**	
	Attributes		**Inputs - Data Elements**	**Attributes**	
Base Measure Specification	**Base Measure**		**Inputs - Data Elements**	**Base Measure; Numerical Identifier; Measure Name**	**Measure Measure ID**
	Measurement Method		**Data Collection - How**	**Measurement Method**	
	Type of Method	**Specify Measures**	**Data Collection - How**		
	Scale	**Specify Measures**	**Inputs - Definition**	**Scale**	
	Type of Scale	**Specify Measures**	**Inputs - Definition**	**Scale**	
	Unit of Measurement	**Specify Measures**	**Inputs - Definition:**		
Derived Measure Specification	**Derived Measure**	**Specify Measures; Collect Measurement Data**	**Inputs - Data Elements**	**Derived Measure**	**Measure**
	Measurement Function	**Specify Measures**	**Algorithm**	**Measurement Function**	**Formula**
Indicator Specification	**Indicator Description and Sample**	**Specify Measures; Analyze Measurement Data**	**Indicator**	**Indicator Description and Sample**	
	Analysis Model	**Specify Measures; Analyze Measurement Data**	**Analysis**	**Analytical Model**	**Implementation Evidence**
	Decision Criteria	**Specify Analysis Procedures**		**Decision Criteria**	**Implementation Evidence**
	Indicator Interpretation	**Analyze Measurement Data; Communicate Results**	**Interpretation**	**Indicator Interpretation; Effects/Impact; Causes of deviation; Positive values; Reporting formats**	**Target; Type; Reporting Format**
Data Collection and Storage Procedures	**Frequency of Data Collection**	**Specify Data Collection and Storage Procedures**	**Data Collection - When/How Often**	**Frequency of collection**	**Frequency**
	Responsible Individual	**Specify Data Collection and Storage Procedures**	**Data Collection - By Whom**	**Information Collector**	**Responsible Parties**
	Phase or Activity in which Collected	**Specify Data Collection and Storage Procedures**	**Data Collection - When/How Often**	**Measure valid up to; Data-record procedure; Period of Analysis**	

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
	Tools Used in Data Collection	Specify Data Collection and Storage Procedures	Data Collection - Forms	Tools Used in Data Collection	Data Source
	Verification and Validation:	Collect Measurement Data	Data Storage - How	Collection Date	
	Repository for Collected Data	Specify Data Collection and Storage Procedures	Data Storage - Where; How, Security	Repository for Collected Data	
Analysis and Reporting Procedures	Frequency of Data Reporting	Specify Analysis Procedures	Data Reporting - How Often	Frequency of Data Reporting	Frequency
	Responsible Individual	Specify Analysis Procedures	Data Reporting - Responsibility of Reporting; By/To Whom	Information Communicator; Information Owner	Responsible Parties
	Phase or Activity in which Analyzed	Specify Analysis Procedures	Assumptions	Measure valid up to; Period of Analysis	
	Source of Data for Analysis	Specify Analysis Procedures	Data Elements	Source of Data for Analysis	Data Source
	Tools Used in Analysis	Specify Analysis Procedures	Data Collection - Forms	Tools Used in Analysis	
	Review, Report, or User	Store Data and Results; Communicate Results	Data Reporting - By/To Whom; Perspective	Information Client; Reviewer	Responsible Parties
Additional Information	Additional Analysis Guidance	Analyze Measurement Data	Evolution	Additional Analysis Guidance	
	Implementation Considerations	Analyze Measurement Data	X-references	Implementation Considerations	

3. IMPLEMENTING SWA MEASURES

Incorporating security measures into an existing measurement program should start with a manageable set of measures. Basic measures like cost, schedule, quality, and growth can be expanded to explicitly include SwA activities to provide insights into specific SwA aspects of project management. PSM provides a variety of measures that could be expanded to explicitly include SwA, including staff experience, staff turnover, change request workload, function points, problem reports, defect density, failure interval, cyclomatic complexity, rework size, rework effort, and achieved accuracy in software performance. As a project evolves, it can add, refine or retire measures and implement new measures, where appropriate.[23]

3.1 Basic Measures Implementation Process

The basic process for implementing SwA measures consists of:

- Creating SwA measures or updating existing measures to include SwA
- Collecting data to support SwA measures
- Storing collected data in a measures repository
- Analyzing collected data and compiling it into SwA measures
- Normalizing and triangulating the measures to determine causes of observed SwA performance[24]
- Documenting and reporting SwA measures to appropriate stakeholders
- Using measures to support decision making and resource allocation
- Training measurement staff coupled with continuous improvement of measures to ensure measures are relevant to the project or organization.

The corrective actions identified through measures-based decision making are implemented by appropriate stakeholders within a project or an organization[25].

This process is common among the base methodologies comprising the Framework with some variations in terminology. As with the Common Measure Specification, organizations should

---

[23] Michele Moss, Riley Rice, *Getting Started with Measuring Your Security*, PSM Conference July 2006.

[24] An example of a measure that must be normalized and triangulated is "number of reported vulnerabilities in vendor software." To be meaningful the analyst needs to evaluate this measure in light of vendor disclosure policy regarding self reporting of vulnerabilities, including whether such reporting includes severity.

[25] More information on this topic can be found in NIST SP 800-55 Rev1 Section 6.

pick an approach for implementing measures and ensure that SwA considerations are integrated into the process.

The basic measures implementation process is depicted in Figure 5-1.

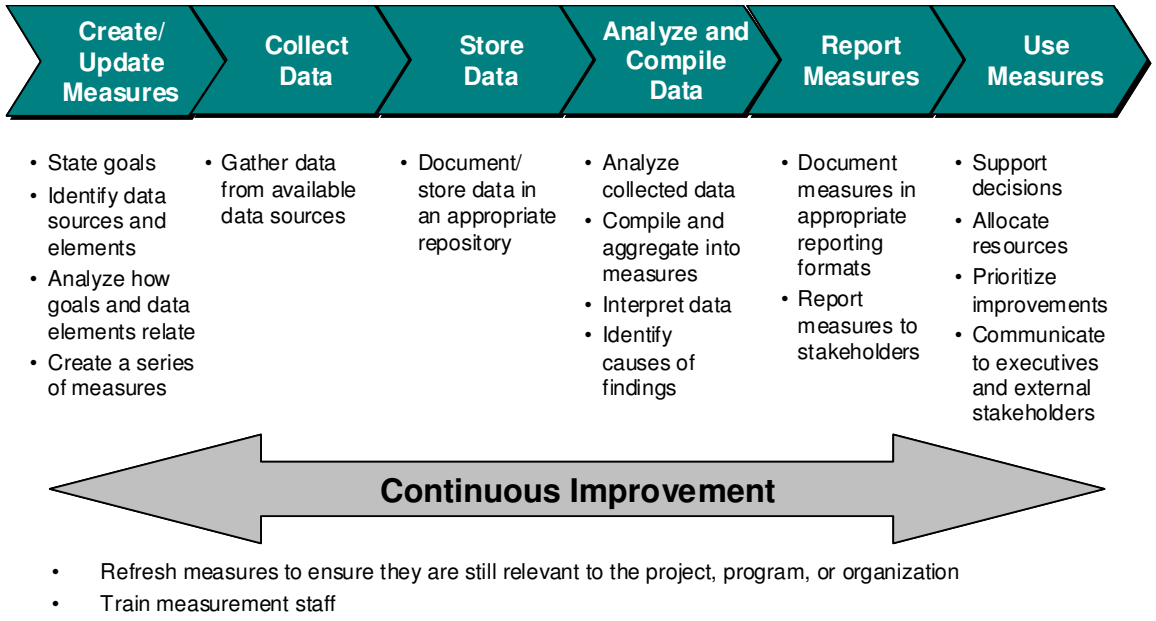

Figure 5-1. Basic Measures Implementation Process.

3.2 Implementation Considerations

Users of this document should be aware of important implementation considerations that can help make their program a success, including:

- SwA measures program should be manageable and cost-effective:
    - Recommend no more than 1 to 3 goals, with associate measures (no more than 5 to 10), per stakeholder at a time, based on current priorities
    - Ensure that the cost of measurement activities does not exceed the benefit that these activities provide
- Data quality is important to ensure that measures are objective, measurable, and reliable:
    - Standardize data collection methods and tools, as well as data repositories to ensure comparability of collected data
    - Count activities and events in a standard manner throughout the organization, and store the results in a standard data repository to facilitate “apples to apples” comparisons

- Ensure that data cleansing and redaction is a part of the process and be aware that quality of data may limit the degree of automation that is feasible at any given point in time

- Measures must be useful and relevant:
  - Use a measures repository to conduct trend analysis to enhance evaluation and effect improvement
  - If measures are not found useful after 2 cycles of use, retire them and try other measures to get to the same information need
  - Review, revise, or phase out old measures, and phase in new measures, when targeted level of performance is achieved or when organization's requirements change
  - Measurement should help determine general trends such as improvement or degradation, and help in determining causes of good or poor performance
  - Information about performance trends and causes of such trends should be used in decision making about improvement actions and resource allocation

- Measurement should motivate appropriate behaviors:
  - Design the measurement program to help motivate desired behavior aimed at improved management and better performance, rather than motivating people to make the numbers look good[26]
  - Use measurement to increase accountability and responsibility and help individuals implement changes required to improve performance, rather then to punish individuals for poor results
  - Identify which measures are to be reported and to whom to ensure that only appropriate information reaches each external and internal stakeholder.[27]

---

[26] To provide a real-life example of a metric that motivates wrong behavior, in one development organization, management ran "open defect reports" every Friday at noon, in order to "measure" how quickly the defects were being closed and absolute numbers of defects by module. Product managers would routinely "close" open defects around 11AM on Friday, then reopen them Friday afternoon to achieve good statistics that had nothing to do with the actual situation. (testimonial from an industry player)

[27] A more extensive discussion on structuring and prioritizing measurement programs can be found in NIST SP 800-55 Revision 1 Sections 3.3, 3.4, and 6.

3.3 How to Begin

Like any other process, the measurement process must start small and develop over time. However, many times it is expected to deliver high value on a relatively low budget without a realization that it takes time, effort, and maturation to do so. To ensure its survival and continued success, it is critical to start on a small scale with a manageable and economical program that is able to demonstrate some level of success relatively early during implementation. Organizations should carefully plan and prioritize measurement implementation, and work with senior leadership to set appropriate expectations. All base methodologies or their compendium documents provide information on this subject. The following are high-level points to help organizations get started successfully:

- Start with a small, manageable set of SwA measures and expand to achieve small successes:
    - Pick an individual project or a small group of projects to pilot SwA measures
    - Leverage existing measurement capabilities
    - Expand project cost, schedule, quality, and growth measures to integrate SwA
    - Develop or identify measures that correspond to your stakeholder goals/information needs and prioritize them by feasibility of implementation and cost, rank them by short-, medium-, and long-term measures, and document in an implementation plan
    - Leverage existing industry lists, select applicable measures, and use the Framework to translate measures from industry lists into the organization's approach[28]
    - Identify existing data and maximize its use
    - Add more SwA measures as the project learns
    - Train existing data collectors to apply their knowledge to SwA or train SwA/security staff rather then hiring and training new staff.
- Be aware of the law of unintended consequences:
    - Assess process behaviors as well as results to gauge whether the system is generating undesirable behaviors (e.g., gaming the numbers rather then using them to understand and improve performance)

[28] Example measures are available at www.psmsc,com, from NIST SP 800-55 Rev1, and from other sources.

- Take advantage of unintended consequences produced by process measurement to effect positive change
- Identify and measure best and worst practice behaviors to help projects and organizations determine which behaviors should start, stop, or continue.

- Get management support
  - Obtain tangible support for SwA measures development and use at every management level
  - Maintain support through regular reporting to stakeholders, tailored to their levels to address their goals/information needs and reduce detail further up the management chain.

4. DATA SOURCES FOR SWA MEASUREMENT

To enable comprehensive SwA measurement, required data must be identified, collected, analyzed, and reported. Organizations need to identify attributes and the data sources that will produce those attributes and use automated data collection and analysis tools to the maximum extent possible to make measurement efficient.

Useful sources of data across a wide set of systems and technology include the openly available enumerations. These are a useful source of well structured and comprehensive SwA data that are generically applicable. They can be used as a complement to organization-specific data which an organization will generate specific to its systems and environment.

Enumerations provide a common language that describes aspects of SwA, such as weaknesses, vulnerabilities, attacks, and configurations, and by doing so enable consistent and comparable measures. Enumerations-based measures do not provide all-inclusive information and should be used in conjunction with organizational and project-specific measures for increased effectiveness.

4.1 Enumerations Overview

The Common Vulnerabilities and Exposures (CVE), Common Control Enumeration (CCE), Common Weakness Enumeration (CWE), and Common Attack Pattern Enumeration and Classification (CAPEC) focused on information security and SwA. They allow people, processes, and products from different information security and SwA activities to be coordinated and compared, while also decoupling the various activities from each other, to decrease confusion between activities, improve response times, and reduce duplication.[29] Enumerations provide commonly accepted descriptions of vulnerabilities, configurations, weaknesses, and attack patterns that allow for comparison among different IT solutions and applications. Increasing vendor adoption of enumerations simplifies the collection of measures across different vendor tools and enables more advanced measurement.

Enumerations are useful throughout the SDLC for a variety of reasons, including shaping requirements, assessing design, and evaluating test coverage. Enumerations are also useful for measurement purposes because they identify specific individual software-related items that can be counted, aggregated, evaluated over time and used for assessment of a variety of aspects of SwA.

4.2 Use of Enumerations for Measurement

As with any set of measures, enumerations must be used in a way that is appropriate to the business or mission context. The measurement process will provide an overall framework for answering pertinent questions and support overall assurance claims. The measures themselves will provide a path for conducting a "what if" analysis and to diagnose potential exploits, weaknesses, vulnerabilities, configuration errors, or other potential issues. Interpretation of

[29] More information on CVE, CCE, CWE, and CAPEC including specific examples is available at measurablesecurity.mitre.org.

measurement results will always depend on the context of the system, its functional requirements, as well as security and SwA requirements. Same results may be interpreted differently depending on the operating system or other packaged software present on the system or network that carries the application that is being assessed. New threat information will not be useful for measurement until the current status of a system is well understood, including current configuration and present vulnerabilities (if applicable), to enable a realistic assessment of what the new threat might mean for a specific system. Measures based on enumerations can be used throughout the SDLC unless otherwise noted in the subsequent sections.

4.2.1 CWE

CWE is an enumeration of the architecture, design, and implementation weaknesses that can lead to exploitable security problems in software. It helps gain insights into potential application security risks that developers, testers, project managers, and customers should understand and manage. It also provides a means for assessment tool vendors and service suppliers to clearly articulate what security-related issues they look for and which ones they are effective at locating.

CWE can be used for SwA measurement for both packaged and custom-built software to reduce weaknesses during development. CWE can help determine which weaknesses are important to mitigate and prioritize them for mitigation. The set of weaknesses should be limited to those applicable to specific configurations that the system will run on during development and operation.

4.2.2 CAPEC

CAPEC is an enumeration of the fundamental patterns of attack used by adversaries to go after information technology. It helps analysts, architects, designers, developers and testers think about how their systems can be attacked, ways of preventing those attacks from succeeding, and identifying those attacks when attempted. Additionally, the breadth and depth of particular tools and services can describe their attack-centric testing methods and approaches with CAPEC to improve consistency, cross correlation and comparison.

CAPEC can be used for SwA measurement for both packaged and custom-built software to assess development. CAPEC can help narrow down the set of relevant weaknesses by identifying relevant attack patterns that may target them. Specifically, CAPEC can be used for a number of purposes including:

- Scope the set of relevant weaknesses by identifying likely attacks
- Identify appropriate tests based on relevant attack patterns
- Evaluate test coverage
- Evaluate penetration testing providers and their approach
- Evaluate tools
- Identify mitigating scenarios and security controls as an analytical tool to help risk mitigation

- Prioritize weakness mitigation.

4.2.3 CVE

CVE is a list of identifiers (ID) for publicly known vulnerabilities including 30,000+ separate vulnerabilities and used by nearly 300 products globally. By leveraging CVE-IDs in an organization's vulnerability alerting services, vulnerability triage and analysis, patch deployment, vulnerability assessment and intrusion detection, an organization can achieve faster response times, greater communication accuracy and reduced rework.

CVE can be used for SwA measurement during testing of packaged software installed on operational systems assess vulnerability mitigation. CVE can help identify specific vulnerabilities that require mitigation, and help ascertain that publicly known vulnerabilities have had appropriate mitigations applied. Because CVEs are assigned to issues applicable to publicly available packaged software (commercial or open source) they are used within the context of testing a fix to a vulnerability present in a shipped product. Usually the CVE is assigned right before the patch or fix is announced and/or shipped.

4.2.4 CCE

CCE is a list of IDs for security related configuration controls for most OS platforms including Microsoft Windows, Solaris, and Red Hat. By utilizing CCE-IDs in system design documentation, system testing activities, configuration management, configuration audit, change management and regulatory and policy compliance reporting, an organization can improve communication accuracy and alignment with a resulting reduction of effort.

CCE can be used for SwA measurement for packaged software to assess whether the system has been deployed and configured correctly. CCE can help identify specific configuration deficiencies that require mitigation and articulate the controls that should be considered during requirements allocation and design, and tested and monitored later in SDLC.

4.2.5 Use of Enumeration Schemas to Assess Skills

In addition to providing useful tools for SwA measurement throughout SDLC, enumerations can be used to assess skills and knowledge of software developers, security analysts, and other similar roles.

APPENDIX A — REFERENCES

FIPS 200, *Minimum Security Requirements for Federal Information and Information Systems*

NIST Special Publication (SP) 800-53 Revision 2, *Recommended Security Controls for Federal Information Systems*

NIST SP 800-55 Revision 1, *Performance Measurement Guide for Information Security*

NIST SP 800-65, *Integrating Security into the Capital Planning and Investment Control Process*

ISO/IEC 15939*, Software engineering — Software measurement process*

ISO/IEC 15408, *Evaluation criteria for IT security*

ISO/IEC 15443, *A framework for IT security assurance*

ISO/IEC 15026*, Software and Systems Integrity Levels*

ISO/IEC 27001, *Information Security Management System (ISMS) Requirements*

ISO/IEC 27002, *Code of Practice for Information Security Management*

ISO/IEC 21827, System *Security Engineering Capability Maturity Model (SSE CMM)*

ISO/IEC 27004, *Information technology - Security techniques - Information security management measurement*

International Systems Security Engineering Association (http://www.issea.org/))

Practical Software and System Measurement (PSM,) (http://www.psmsc.com/))

NIST Computer Security Division (http://csrc.nist.gov)

System Engineering Institute, Carnegie Mellon University, *Capability Maturity Model Integration (CMMI®)* (http://www.sei.cmu.edu/cmmi/)

White House Scorecard (http://www.whitehouse.gov/results/agenda/scorecard.html)

The Object Management Group (http://www.omg.org)

Project Management Body of Knowledge (PMBOK) (http://www.pmi.org/info/default.asp)

Making Security Measurable (http://makingsecuritymeasurable.mitre.org/)

Federal Aviation Administration, *Integrated Capability Maturity Model* (http://www.faa.gov/about/office_org/headquarters_offices/aio/documents/media/SafetyandSecurityExt-FINAL-web.pdf)

Control Objectives for Information Technology (COBIT) (http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=7981)

The White House, Office of Management and Budget, Assessing Program Performance, Program Assessment Rating Tool (PART) (http://www.whitehouse.gov/omb/part/)

Corporate Information Security Working Group Report Of The Best Practices and Metrics Teams (http://www.cisecurity.org/Documents/BPMetricsTeamReportFinal111704Rev11005.pdf)

President's Management Agenda (http://www.whitehouse.gov/omb/budintegration/pma_index.html)

Measures and Measurement for Secure Software Development Article (https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/measurement/227-BSI.html)

Monitor Security Metrics Wiki (http://www.owasp.org/index.php/Monitor_security_metrics)

Measuring Security Presentation (http://geer.tinho.net/usenix/measuringsecurity.tutorialv2.pdf)

Community Website for Security Practitioners (http://securitymetrics.org/content/Wiki.jsp)

Build Security In (https://buildsecurityin.us-cert.gov/daisy/bsi/home.html)

APPENDIX B — ACRONYMS

CAPEC	Common Attack Pattern Enumeration and Classification
CCE	Common Control Enumeration
CCSS	Common Configuration Scoring System
CMMI®	Capability Maturity Model Integration
CNSS	Committee on National Security Systems
COBIT	Control Objectives for Information and related Technology
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DHS	Department of Homeland Security
DoD	Department of Defense
GQ(I)M	Goal Question (Indicator) Measure
GPRA	Government Performance and Results Act
FAR	Federal Acquisition Regulation
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act
iCMM	Integrated Capability Maturity Model
IEC	International Electrotechnical Commission
ISMS	Information Security Management System
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library
NIST	National Institute of Standards and Technology
OMB	Object Management Group
PART	Program Assessment Rating Tool

PMA	Performance Management Association
PSM	Practical Software and Systems Measurement
SDLC	Software Development Lifecycle
SwA	Software Assurance
SP	Special Publication

APPENDIX C — GLOSSARY[30]

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]

---

[30] The entire glossary was borrowed from Software Assurance (SwA) in Acquisition: Mitigating Risks to the Enterprise, Draft Version 1.01 February 26, 2008.

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..... Means 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. 1401]

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. [Merriam Webster's Online Dictionary]

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. One problem is that regulations are not generally included in volumes containing state statutes or federal laws, but often must be obtained from the agency or located volumes in law libraries and not widely distributed. The regulation-making process involves hearings, publication in governmental journals which

supposedly give public notice and adoption by the agency. The process is best known to industries and special interests concerned with the subject matter, but only occasionally to the general public. Federal regulations are adopted in the manner designated in the Administrative Procedure Act (A.P.A.) and states usually have similar procedures. [Gerald N. Hill and Kathleen T. Hill. http://legal-dictionary.thefreedictionary.com/regulation]

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. [DAU]

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 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/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 Rational Unified Process]

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. [IEEE]

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's II New College Dictionary]

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). [http://www.gartner.com]

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).

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]

APPENDIX D — MEASUREMENT METHODOLOGIES AND RESOURCES

Information Security Measurement Methodologies

National Institute of Standards and Technology (NIST) Special Publication (SP) 800-55, Revision 1, *Performance Measurement Guide for Information Security* – a guide to assist in the development, selection, and implementation of measures to be used at the information system and program levels.[31]

International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 27004, *Information technology - Security techniques - Information security management measurement* – provides guidance on the development and use of measures and measurement in order to assess the effectiveness of information security management system (ISMS), including the ISMS policy and objectives and security controls used to implement and manage information security, as specified in ISO/IEC 27001, Information Security Management System – Requirements.[32]

System and Software Development Measurement Methodologies

ISO/IEC 15939, *Software Engineering - Software Measurement Process*, also known as Practical Software and System Measurement (PSM) – identifies the activities and tasks that are necessary to successfully identify, define, select, apply, and improve software measurement within an overall project or organizational measurement structure. It also provides definitions for measurement terms commonly used within the software industry. Although this International Standard does not catalogue software measures, nor does it provide a recommended set of measures to apply on software projects, it does identify a process that supports defining a suitable set of measures that address specific information needs.[33]

CMMI® (*Capability Maturity Model Integration)* Measurement and Analysis Process Area – CMMI® process area intended to develop and sustain a measurable capability that is used to support management information needs.

CMMI® GQ(I)M – *Capability Maturity Model Integration Goal Question Indicator Metric* - a method for identifying and defining indicators (graphical displays) and measures that directly support an organization's business goals related to product development, process improvement, and project management.

---

[31] NIST SP 800-55, Revision 1, *Performance Measurement Guide for Information Security*

[32] ISO/IEC 27004 Information technology - Security techniques - Information security management measurement

[33] PSM ISO/IEC 15939, Software Engineering- Software Measurement Process

Measurement Frameworks

Balanced Scorecard – The balanced scorecard is a strategic planning and management system used to align business activities to the vision and strategy of the organization, improve internal and external communications, and monitor organizational performance against strategic goals.[34].

PART – The Office of Management and Budget (OMB) introduced the Program Assessment Rating Tool (PART) as the methodology for Departments and Agencies to measure their progress under the Government Performance Results Act (GPRA). PART was developed to assess and improve program performance so that the Federal government can achieve better results.[35]

The President's Management Agenda (PMA) – announced in 2001, establishes the President's strategy for improving the management and performance of the Federal government. It establishes five government-wide initiatives: strategic management of human capital, competitive sourcing, improved financial reporting, expanded electronic government, and budget and performance integration.[36]

Frameworks that Provide Foundation for Measurement

CMMI® – is a process improvement approach that provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization. CMMI® helps integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes. [37]

Federal Aviation Administration (FAA) Integrated Capability Maturity Mode (iCMM)[38] – describes the essential elements of an organization's acquisition, engineering, and management process that must exist to ensure good acquisition of software intensive systems.

ISO/IEC 16085, *Software Engineering, Software Life Cycle Processes, Risk Management* - defines a process for the management of risk during software acquisition, supply, development, operations and maintenance.

---

[34] http://www.balancedscorecard.org/

[35] http://www.whitehouse.gov/omb/part/

[36] http://www.whitehouse.gov/omb/budintegration/pma_index.html

[37] http://www.sei.cmu.edu/cmmi

[38] http://www.faa.gov/about/office_org/headquarters_offices/aio/documents/media/SafetyandSecurityExt-FINAL-web.pdf

ISO/IEC 21827, *System Security Engineering Capability Maturity Model (SSE CMM)* - addresses security engineering activities that span the entire trusted product or secure system life cycle, including concept definition, requirements analysis, design, development, integration, installation, operations, maintenance, and decommissioning.

Project Management Body of Knowledge (PMBOK) – The Project Management Body of Knowledge is the sum of knowledge within the profession of project management.[39]

OMG Common Measurement Specification – a software metrics metamodel which facilitates the inoperability of measurements of software artifacts.[40]

Qualitative Assessment Methods

ISO/IEC 15504, *Information Technology – Software Process Assessment* – provides a framework for the assessment of processes. This framework can be used by organizations involved in planning, managing, monitoring, controlling and improving the acquisition, supply, development, operation, evolution and support of products and services.

CMMI® Appraisal Method for Process Improvement (SCAMPI) – provides organizations with insight into the processes being practiced within the organization or project

ISO/IEC 15408, *Evaluation criteria for IT security (a.k.a. Common Criteria)* – represents the outcome of series of efforts to develop criteria for evaluation of IT Security that are broadly useful within the international community.

ISO/IEC 15443, *A Framework for IT Security Assurance* – a multi-part Technical Report to guide the IT security professional in the selection of an appropriate assurance method when specifying, selecting, or deploying a security service, product, or environmental factor such as an organization or personnel (known as a *deliverable*). The aim is to understand the assurance type and amount required to achieve confidence that the deliverable satisfies the stated IT security assurance requirements and consequently its security policy.[41]

Process and Controls Standards and Guidance

Control Objectives for Information Technology (COBIT) – Control Objectives for Information and related Technology is a set of IT governance and security guidelines that was

---

[39] http://www.pmi.org/Pages/default.aspx

[40] http://www.omg.org/

[41] http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=39733 International Standards Organization

first published in 1996. COBIT, issued by the IT Governance Institute, is increasingly internationally accepted as good practice for control over information, IT and related risks[42]

eSourcing Capability Model for Service Providers (eSCM-SP) – a model that codifies proven best practices among e-enabled service providers worldwide. This model is composed of 84 practices that define critical capabilities needed to remain competitive among IT-enabled service providers. [43]

NIST Federal Information Processing Standards (FIPS) 200, Minimum Security Requirements for Federal Information and Information Systems – specifies minimum security requirements for information and information systems supporting the executive agencies of the federal government and a risk-based process for selecting the security controls necessary to satisfy the minimum security requirements.

ISO/IEC 15026, *Software and Systems Integrity Levels* – provides a way for developing assurance argument and assurance evidence for a variety of software and systems projects.

Other Measurement Resources

NIST Interagency Report 7435, *The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems*

NIST Interagency Report 7502, *The Common Configuration Scoring System (CCSS) (Draft)*

Enumerations - *measurablesecurity.mitre.org*

L. Wang, A. Singhal, S. Jajodia, *Measuring the Overall Security of Network Configurations Using Attack Graphs*

O'Neill, Don, *Calculating Security Return on Investment*, Build Security In, 2007

Sahinoglu, Mehmet, *Security Meter: A Practical Decision-Tree Model to Quantify Risk,* IEEE Security & Privacy Vol. 3, No. 3 (May/June 2005), pp. 18-24.

---

[42] http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=7981

[43] http://itsqc.cmu.edu/

APPENDIX E – COMMON MEASURE SPECIFICATION

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Approach	**Methodology:** Information Need driven. **Purpose:** To align Information Needs with Indicators and Measures.	**Purpose:** To develop and sustain a measurement capability that is used to support management information needs.	**Methodology:** Goal driven. **Purpose:** To align Goals with Indicators and Measures.	**Purpose:** To guide an organization through the use of information security measurements, identifies the adequacy of an existing ISMS, including policy, risk management, control objectives, controls, processes and procedures.	**Purpose:** To guide for the specific development, selection, and implementation of information system-level and program-level measures to indicate the implementation, efficiency/effectiveness, and impact of security controls, and other security related activities.
Goal/Objective/Information Need Description	**Information Need:** What the measurement user (e.g., manager or project team member) needs to know in order to make informed decisions.	**SG 1: SP 1.1** Establish measurement objectives.	**Objective:** Describe the objective or purpose of the indicator.	**Purpose of measure:** Describes the reasons for introducing the measure.	**Goal and Objective:** Statement of information security goal and objective. For system-level security control measures, the goal would guide security control implementation for that information system. For program-level measures, both strategic goals and information security goals can be included. For example, information security goals can be derived from enterprise-level goals in support of the organization's mission. These goals are usually articulated in strategic and performance plans. When possible, include both the enterprise-level goal and the specific information security goal extracted from agency documentation, or identify an information security program goal that would contribute to the accomplishment of the selected strategic goal or objective.

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
	Information Category: A logical grouping of information needs that are defined in the PSM to provide structure for the Information Model. PSM categories include schedule and progress, resources and cost, product size and stability, product quality, process performance, technology effectiveness, and customer satisfaction.			**Control or Control Objective:** Control or control objective under measurement.	
Measurable Concept/Question	**Measurable Concept:** An abstract relationship between attributes of entities and information needs.		**Question: List the question(s) the indicator user is trying to answer. Probing Questions: List questions that delve into the possible reasons for the value of an indicator, whether performance is meeting expectations or whether appropriate action is being taken.**		
Entities/Attributes	**Relevant Entities:** The object that is to be measured. Entities include process or product elements of a project such as project tasks, plans/estimates, resources, and deliverables.		**Inputs - Data Elements:** List all data elements in the production of the indicator. **Inputs - Definition:** Precisely define the data element used or point to where the definition can be found.	**Object of Measurement:** The object that is to be measured. Objects may include processes, systems, or system components.	
	Attributes: The property or characteristic of any entity that is quantified to obtain a base measure.		**Inputs - Data Elements:** List all data elements in the production of the indicator.	**Attributes:** Property or characteristic of an object of measurement that can be distinguished quantitatively or qualitatively by human or automated means.	

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Base Measure Specification	**Base Measure:** A base measure is a measure of a single attribute defined by a specified measurement method (e.g., planned number of lines of code, cumulative cost to date). As data is collected, a value is assigned to a base measure.		**Inputs - Data Elements:** List all data elements in the production of the indicator.	**Base Measure:** A base measure is a measure of a single attribute defined by a specified measurement method (e.g., number of trained personnel, number of sites, cumulative cost to date). As data is collected, a value is assigned to a base measure.	**Measure:** Statement of measurement. Use a numeric statement that begins with the word "percentage," "number," "frequency," "average," or a similar term. If applicable, list the NIST SP 800-53 security control(s) being measured. Security controls that provide supporting data should be stated in Implementation Evidence. If the measure is applicable to a specific FIPS 199 impact level (high, moderate, or low), state this level within the measure.
				Numerical identifier: Unique organization-specific numerical identifier. **Measure Name:** Measure Name	**Measure ID:** State the unique identifier used for measure tracking and sorting. The unique identifier can be from an organization-specific naming convention or can directly reference another source.
	Measurement Method: The logical sequence of operations that define the counting rule to calculate each base measure.		**Data Collection - How:** Describe how the data will be collected.	**Measurement Method:** The logical sequence of operations that define the counting rule to calculate each base measure. For base measures, measurement method by which the data for measurement will be obtained, including the precision, scale and units of measure.	
	Type of Method: The type of method used to quantify an attribute, either (1) subjective, involving human judgment, or (2) objective, using only established rules to determine numerical values.	**SG 1: SP 1.2** Specify Measures.	**Data Collection - How:** Describe how the data will be collected.		

Software & Systems			Information Security	
PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Scale: The ordered set of values or categories that are used in the base measure.	**SG 1: SP 1.2** Specify Measures.	**Inputs - Definition:** Precisely define the data element used or point to where the definition can be found.	**Scale:** The ordered set of values or categories that are used in the base measure.	
Type of Scale: The type of relationship between values on the scale, either: - Nominal: the measurement values are categorical, as in defects by their type. - Ordinal: the measurement values are rankings, as in assignment of defects to a severity level. - Interval: the measurement values have equal increments for equal quantities of the attribute, such as an additional cyclomatic complexity value for each additional logic path in the software unit. - Ratio: the measurement values have equal increments, beginning at zero, for equal quantities of the attribute, such as size measurement in terms of LOC.	**SG 1: SP 1.2** Specify Measures.	**Inputs - Definition:** Precisely define the data element used or point to where the definition can be found.	**Scale:** The ordered set of values or categories that are used in the base measure.	
Unit of Measurement: The standardized quantitative amount that will be counted to derive the value of the base measure, such as an hour or a line of code.	**SG 1: SP 1.2** Specify Measures.	**Inputs - Definition:** Precisely define the data element used or point to where the definition can be found.		

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Derived Measure Specification	**Derived Measure:** A measure that is derived as a function of two or more base measures.	**SG 1: SP 1.2** Specify Measures. **SG 2: SP 2.1** Collect Measurement Data.	**Inputs - Data Elements:** List all data elements in the production of the indicator.	**Derived Measure:** A measure that is derived as a function of two or more base measures.	**Measure**: Statement of measurement. Use a numeric statement that begins with the word "percentage," "number," "frequency," "average," or a similar term. If applicable, list the NIST SP 800-53 security control(s) being measured. Security controls that provide supporting data should be stated in Implementation Evidence. If the measure is applicable to a specific FIPS 199 impact level (high, moderate, or low), state this level within the measure.
	Measurement Function: The formula that is used to calculate the derived measure.	**SG 1: SP 1.2** Specify Measures.	**Algorithm:** Specify the algorithm or formula required to combine data elements to create input values for the indicator. It should also include how the data is plotted on the graph.	**Measurement Function:** The formula that is used to calculate the derived measure. For derived measures, measurement function by which the derived measures are aggregated based on corresponding base measures and resulting cumulative precision.	**Formula:** Calculation to be performed that results in a numeric expression of a measure. The information gathered through listing implementation evidence serves as an input into the formula for calculating the measure.
Indicator Specification	**Indicator Description and Sample:** A display of one or more measures (base and derived) to support the user in deriving information for analysis and decision making. An indicator is often displayed as a graph or a chart. Include a sketch of the indicator.	**SG 1: SP 1.2** Specify Measures. S**G 2: SP 2.2** Analyze Measurement Data.	**Indicator:** An indicator is defined as a measure or a combination of measures that provides insight into a process, a project, or a product. An indicator is usually a graph or table that you define for the organization's needs. **Visual Display:** Provide a graphical view of the indicator.	**Indicator Description and Sample:** A display of one or more measures (base and derived) to support the user in deriving information for analysis and decision making. An indicator is often displayed as a graph or chart. Include a sketch of the indicator.	
	Analysis Model: A process that applies decision criteria to define the behavior responses to the quantitative results of the indicator.	**SG 1: SP 1.2** Specify Measures. **SG 2: SP 2.2** Analyze Measurement Data.	**Analysis:** Specify what type of analysis can be done with the information.	**Analytical Model:** A process that applies decision criteria to define the behavior responses to the quantitative results of indicators.	**Implementation Evidence:** Implementation evidence is used to calculate the measure, to validate that the activity is performed, and to identify probable causes of unsatisfactory results for a specific measure.

Software & Systems			Information Security	
PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Decision Criteria: A defined set of actions that will be taken in response to achieved quantitative values of the model.	**SG 1: SP 1.4** Specify Analysis Procedures. **SG 1: SP 1.4** Specify Analysis Procedures.		**Decision Criteria:** A defined set of actions that will be taken in response to achieved quantitative values of the model.	**Implementation Evidence:** Implementation evidence is used to calculate the measure, to validate that the activity is performed, and to identify probable causes of unsatisfactory results for a specific measure.
Indicator Interpretation: A description of how the sample indicator (see sample figure in indicator description) was interpreted.	**SG 2: SP 2.2** Analyze Measurement Data. **SG 2: SP 2.4** Communicate Results	**Interpretation:** Describe what different values of the indicator mean. Make it clear how the indicator answers the "Questions" section above. Provide any important cautions about how the data could be misinterpreted and measures to take to avoid misinterpretation.	**Indicator Interpretation:** A description of how the sample indicator (see sample figure in indicator description) was interpreted. **Effects/Impact:** Definition of the effects and impact derived as a consequence of the results obtained by the measure. **Causes of deviation:** Definition of possible causes of deviations in the results obtained. **Positive values:** Statement explaining whether increasing values indicate positive values (good result) or whether decreasing values are to be taken to indicate positive values.	**Target:** Threshold for a satisfactory rating for the measure, such as milestone completion or a statistical measure. Target can be expressed in percentages, time, dollars, or other appropriate units of measure. Target may be tied to a required completion timeframe. Select final and interim target to enable tracking of progress toward stated goal. **Type:** Statement of whether the measure is implementation, effectiveness/efficiency, or impact.
			Reporting formats: Reporting format should be identified and documented. Describes the observations that the organization or owner of the information may want on record. Reporting formats will visually depict the measures and provide a verbal explanation of the indicators. Reporting formats should be customized to the information customer.	**Reporting Format:** Indication of how the measure will be reported, such as a pie chart, line chart, bar graph, or other format. State the type of format or provide a sample.

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Data Collection and Storage Procedures	**Frequency of Data Collection:** How often data is collected.	**SG 1: SP 1.3** Specify Data Collection and Storage Procedures.	**Data Collection - When/How Often:** Describe when the data will be collected and how often.	**Frequency of collection:** How often data is collected.	**Frequency:** Indication of how often the data is collected and analyzed, and how often the data is reported. Select the frequency of data collection based on a rate of change in a particular security control that is being evaluated. Select the frequency of data reporting based on external reporting requirements and internal customer preferences.
	Responsible Individual: The person who is assigned to collect the data.	**SG 1: SP 1.3** Specify Data Collection and Storage Procedures.	**Data Collection - By Whom:** Specify who will collect the data.	**Information Collector:** The person or organizational unit responsible for collecting, recording, and storing the data.	**Responsible Parties:** Indicate the following key stakeholders: • Information Owner: Identify organizational component and individual who owns required pieces of information; • Information Collector: Identify the organizational component and individual responsible for collecting the data. (Note: If possible, Information Collector should be a different individual or even a representative of a different organizational unit than the Information Owner, to avoid the possibility of conflict of interest and ensure separation of duties. Smaller organizations will need to determine whether it is feasible to separate these two responsibilities); and • Information Customer: Identify the organizational component and individual who will receive the data.
	Phase or Activity in which Collected: The phase or activity when the data is collected.	**SG 1: SP 1.3** Specify Data Collection and Storage Procedures.	**Data Collection - When/How Often:** Describe when data will be collected and how often.	**Measure valid up to:** Date of revision (expiry or renovation of measure validity). **Data-record Procedure:** Defines the data record procedure (link to procedure). **Period of Analysis:** Defines the period being measured.	

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
	Tools Used in Data Collection: List any tools used to collect the data.	**SG 1: SP 1.3** Specify Data Collection and Storage Procedures.	**Data Collection - Forms:** Reference any standard forms for data collection and provide information about where to obtain them.	**Tools Used in Data Collection:** List any tools used to collect the data (e.g., vulnerability scanner).	**Data Source:** Location of the data to be used in calculating the measure. Include databases, tracking tools, organizations, or specific roles within organizations that can provide required information.
	Verification and Validation: List and V&V tests that will be run to ensure the data is complete and accurate.	**SG 2: SP 2.1** Collect Measurement Data.	**Data Storage - How:** Indicate the storage media, procedures, and tools for the configuration control.	**Collection Date:** Date the data was obtained.	
	Repository for Collected Data: List any tools where data is stored after it is collected.	**SG 1: SP 1.3** Specify Data Collection and Storage Procedures.	**Data Storage - Where:** Indicate where the data is to be stored. **Data Storage - How:** Indicate the storage media, procedures, and tools for the configuration control. **Data Storage - Security:** Specify access to this data will be controlled.	**Repository for Collected Data:** List any tools where data is stored after it is collected (e.g., database).	
Analysis and Reporting Procedures	**Frequency of Data Reporting:** How often data is reported.	**SG 1: SP 1.4** Specify Analysis Procedures.	**Data Reporting - How Often:** Specify how often the data will be reported.	**Frequency of Data Reporting:** How often data is collected.	**Frequency:** Indication of how often the data is collected and analyzed, and how often the data is reported. Select the frequency of data collection based on a rate of change in a particular security control that is being evaluated. Select the frequency of data reporting based on external reporting requirements and internal customer preferences.

Software & Systems			Information Security	
PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
Responsible Individual: The person who is assigned to analyze data and report the results.	**SG 1: SP 1.4** Specify Analysis Procedures.	**Data Reporting - Responsibility of Reporting:** Indicate who has responsibility for reporting the data. **Data Reporting - By/To Whom:** Indicate who will do the reporting and to whom the report is going to. This may be individual or an organizational entity.	**Information Communicator:** The person or organizational unit responsible for analyzing data and reporting the results. **Information Owner:** The person or organization who owns the information about objects of measurement and attributes used to create base measures and who is responsible for measurement.	**Responsible Parties:** Indicate the following key stakeholders: • Information Owner: Identify organizational component and individual who owns required pieces of information; • Information Collector: Identify the organizational component and individual responsible for collecting the data. (Note: If possible, Information Collector should be a different individual or even a representative of a different organizational unit than the Information Owner, to avoid the possibility of conflict of interest and ensure separation of duties. Smaller organizations will need to determine whether it is feasible to separate these two responsibilities); and • Information Customer: Identify the organizational component and individual who will receive the data.
Phase or Activity in which Analyzed: The phase or activity when the data is analyzed.	**SG 1: SP 1.4** Specify Analysis Procedures.	**Assumptions:** Identify any assumptions about the organization, its processes, life cycle models, and so on that are important conditions for collecting and using this indicator.	**Measure valid up to:** Date of revision (expiry or renovation of measure validity). **Period of Analysis:** Defines the period being measured.	
Source of Data for Analysis: List any sources of data for this analysis.	**SG 1: SP 1.4** Specify Analysis Procedures.	**Data Elements:** List all the data elements in the production of the indicator.	**Source of Data for Analysis:** List any sources of data for this analysis.	**Data Source:** Location of the data to be used in calculating the measure. Include databases, tracking tools, organizations, or specific roles within organizations that can provide required information.
Tools Used in Analysis: List any tools used for analysis.	**SG 1: SP 1.4** Specify Analysis Procedures.	**Data Collection - Forms:** Reference any standard forms for data collection and provide information about where to obtain them.	**Tools Used in Analysis:** List any tools used for analysis (e.g., statistical tools).	

	Software & Systems			Information Security	
	PSM ISO/IEC 15939	CMMI® (Measurement and Analysis Process Area)	CMMI® GQ(I)M	ISO/IEC 27004	NIST SP 800-55 Revision 1
	Review, Report, or User: Document when results are reviewed and reported, along with the intended user of the results.	**SG 2: SP 2.3** Store Data and Results. **SG 2: SP 2.4** Communicate Results.	**Data Reporting - By/To Whom:** Indicate who will do the reporting and to whom the report is going to. **Perspective:** Describe the audience (for whom is this display intended) for the visual display.	**Information Client:** The person or organizational unit requesting and requiring the measures in support of their business functions. **Reviewer:** Person or organizational unit who reviews that the measure evaluation criteria are appropriate to verify the control effectiveness.	**Responsible Parties:** Indicate the following key stakeholders: • Information Owner: Identify organizational component and individual who owns required pieces of information; • Information Collector: Identify the organizational component and individual responsible for collecting the data. (Note: If possible, Information Collector should be a different individual or even a representative of a different organizational unit than the Information Owner, to avoid the possibility of conflict of interest and ensure separation of duties. Smaller organizations will need to determine whether it is feasible to separate these two responsibilities); and • Information Customer: Identify the organizational component and individual who will receive the data.
Additional Information	**Additional Analysis Guidance:** Provide any additional guidance on variations of this measure.	**SG 2: SP 2.2 Analyze Measurement Data.**	**Evolution:** Specify how the indicator can be improved over time, especially as more historical data accumulates.	**Additional Analysis Guidance:** Provide any additional guidance on variations of this measure.	
	Implementation Considerations: List any process or implementation requirements that are necessary for successful implementation.	**SG 2: SP 2.2** Analyze Measurement Data.	**X-references:** If the values of other defined indicators influence the appropriate interpretation of the current indicator.	**Implementation Considerations:** List any process or implementation requirements that are necessary for successful implementation.	