DRAFT

Practical Measurement Framework for Software Assurance and Information Security

Version 0.91

July 18, 2008

Deleted: July 12, 2008

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

EXECUTIVE SUMMARY

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

The common measurement framework provides information on creating SwA measures but 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 stakeholder roles and provides example goals or information needs for them. A number of key measures for different stakeholder groups such as executives, developers, vendors, suppliers, program managers, acquirers, and buyers are included to help organizations asses the state of their SwA efforts during any stage of a project.

The framework provides an integrated measurement approach which leverages five industry approaches that use similar measures development and implementation processes. The methodologies were selected because of their widespread use among software and systems development community and the information community. Included is a table of abbreviated common measure specifications to illustrate the similarities as well as a detailed table in the appendix to provide additional information.

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 include specific measures, information needs and the benefits to assurance that these measures can produce. The document also lists several automated tool examples to facilitate the measurement process and reduce its overall cost.

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 software assurance of the code which they are developing or acquiring; quantifying this assurance has been a challenge.

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 points 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
- Provide a repeatable quantifiable way to assess, compare, and track improvements in assurance
- Focus SwA activities on risk mitigation in order of priority and exploitability
- 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
- Verify, validate, and document whether the system or software does what it was intended to do and more importantly not be exploited for other uses to assess the trustworthiness of a system
- Make informed decisions in the system development lifecycle (SDLC) related to information security compliance, performance, and functional requirements/controls
- Determine if security performances and trade-offs have been defined and accepted
- Provide an objective 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

In 2003, the US Department of Defense (DoD) launched a SwA Initiative,[1] and this was joined in 2004 by the Department of Homeland Security (DHS) to address concerns of poor-quality,

[1] Then Deputy Director for SwA, Information Assurance Directorate, Office of Assistant Secretary of Defense (Networks and Information Integration), Joe Jarzombek led the SwA initiative which submitted an interim report

unreliable, and non-secure software.[2] Working groups were established to address SwA efforts that encompass people, process, technology, and acquisition.[3]

The measurement working group consists of representatives from government, industry, and academia. They addressed how to assess the degree of assurance provided by software, using quantitative and qualitative methodologies and techniques. The working group recommended the creation of a measurement framework that would leverage existing measurement methodologies for SwA measurement for developing and implementing SwA measures as a part of overall organizational risk management framework.

1.2 Purpose and Scope

This document proposes a practical framework for measuring achievement of SwA goals and objectives within the context of individual projects, programs, or enterprises. It targets a variety of audiences interested in the subject of SwA measurement including executives, developers, vendors, suppliers, program managers, acquirers, and buyers.

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.

This document 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 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.

SwA is a cross-functionary discipline that relies on methods and techniques produced by other disciplines. This concept is depicted in Figure 1.

---

[2] Subsequently, the DoD established the SwA Tiger Team and commissioned the Defense Science Board (DSB) Task Force on SwA.

[3] Software Assurance in Acquisition: Mitigating Risks to the Enterprise. Version 0.9, February 2007

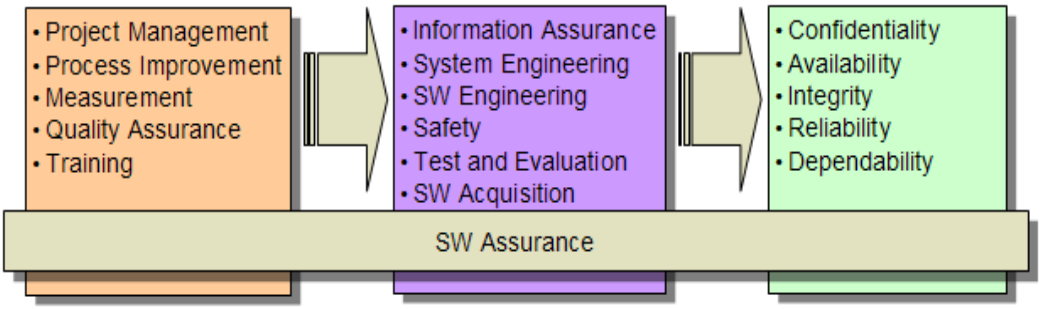

Figure 1. Cross-disciplinary Nature of Software Assurance

The measurement framework can be applied beyond SwA to a variety of security-related measurement efforts to help facilitate risk-based decision, providing quantitative information on a variety of organization's security related performance aspects.

1.4 Assumptions

This document assumes that the audience has knowledge of information security/information assurance and system and software engineering disciplines; therefore it does not intend to explain the founding principles of either discipline. It also assumes that the readers understand the basics of measurement and does not intend to fully explain the measurement methodologies that were leveraged here.[4] The report targets a variety of audiences, including federal, state and local governments and commercial organizations.

1.5 Key Definitions

"Software assurance," "Measure," and "Measurement" are the key terms used in this document. In recent years, many standards and industry organizations have been adopting the terms "Measure" and "Measurement" to describe the result and 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." The following are definitions of the three key terms, defined 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

[4] Information on system and software measurement can be found through 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).

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

1.6 Principles

The SwA measurement approach adopts the following key principles:

- SwA measurement is a composite discipline and can be implemented by including SwA goals and objectives in a project or organizational measures development and implementation regardless of what specific measurement methodology is being used.
- SwA measurement must satisfy information needs of a variety of stakeholders/audiences, including executives, developers, vendors, suppliers and buyers.
- 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 cost effective and practical to help focus resources on improving secure design and coding practices.
- Implementation of SwA measures should facilitate automation of data collection 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.
- For the purposes of this document, the term "measurement" applies to both quantitative and qualitative measurement methodologies.

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 provides guidance for an integrated measurement approach including common measure specification and implementing measures.
- Section 3, Data Sources for SWA Measurement**,** provides enumerations and their use for reducing weaknesses, assessing development activities, measuring vulnerability mitigation, assessing deployed configurations, and assessing skills. The section also provides a brief overview on automated tools that could be used to implement SwA measurement.

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 lists 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, and finally, process and controls standards and guidance. Appendix E, Common Measure

Specification includes detailed common measure specification with the definitions used within those methodologies that comprise Common Measurement Framework.

2 COMMON MEASUREMENT FRAMEWORK

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 a completely new approach for measuring SwA is counterproductive. Rather then creating yet one more approach that is slightly different from others, this report leverages five prominent existing approaches used within software and system engineering and information security industries to propose a common framework.

Measurement practitioners can leverage this framework to integrate SwA measurement into existing measurement efforts, by expanding the content of their organization's measurement activities to include SwA while using processes and methodologies already established by their respective organizations. If an organization is not using any measurement processes nor an organization should select a measurement process that would be appropriate, for example, provide a competitive edge, within a particular industry context. Users of the framework should ensure that the content is appropriate and that specific software assurance measures are used in consort with other measures, regardless of which measurement approach is used.

The common measurement framework guides organizations in creating their measures but does not prescribe any specific measures nor does it prescribe a specific measurement process (e.g., measures creation, collection, analysis, reporting, and using the 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.

This chapter discusses primary SwA stakeholders, suggests the items that the stakeholders may be interested in learning from measurement, lists some examples of SwA measures, and provides an overview of the Framework.

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 uses the following broad groupings of SwA stakeholders:

- Executive decision maker – an individual in a leadership position who has authority to make decisions and may require quantifiable information to acquire an understanding of the level of risk associated with software to support decision-making processes.

- Developer/Vendor/Supplier[5] – an individual or an organization that supports other organizations by providing software and system-related products and services

- Buyer/Acquirer[6] – an individual or an organization that seeks support from other organizations to provide software and system-related products and services.

It is important to note that in diverse organizations all three stakeholder groups may be found internally. Also, individuals within each generic stakeholder group may have different interests and needs based on their individual roles and responsibilities and job descriptions.

Stakeholder "Goals" or "Objectives," sometimes expressed as "Information Needs" that define information that a stakeholder is attempting to gain from the measurement activity will drive which measures are selected, developed and eventually implemented. Table 1 provides example goals and information needs for generic SwA assurance stakeholders.

Table 1. Software Assurance Measurement Stakeholder Example Goals and Information Needs

Stakeholder	Goals/Information Needs
Executive	• Gain insights into risk exposure and liability from acquired/integrated product • Minimize risks created by packaged and custom built vendor and in-house developed software • Establish costs of breaches (e.g., loss of revenue, opportunity costs, loss of credibility, legal consequences) • Compare costs of building SwA in vs. correcting it after the fact • Compare risks across different vendor or custom products • Gain insights into overall security posture of the organization or its component(s)
Developer/Vendor/Supplier	• Ensure understanding of operational environment and integration of use, misuse, abuse, and threat considerations into the SDLC activities • Identify errors in the design, architecture, and code and reduce risks of future exploitation of software • Understand the level of risk associated with making decisions at each phase of the SDLC

[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 owners, information owners and other staff who are working for an organization that is acquiring software from other organizations.

Stakeholder	Goals/Information Needs
	• Measure accomplishment of internal deadlines • Identify software defects that may be exploited in the future • Determine if security requirements are being planned and implemented • Reduce opportunity for malicious software • Proactively address the security defects prior to testing and deployment • Monitor planning and implementation of 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 number of staff 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 • Identify developers who may be the source of poor design and coding practices that may be introducing vulnerabilities into software
Buyer/Acquirer	• Integrate SwA considerations into the acquisition lifecycle • Improve cost-effectiveness of SwA integration into the SDLC • Ascertain that contracting officers have 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 guidance have been integrated into procurement language • Gain insight into how the software to be acquired will impact organization's security posture • Validate that SwA requirements defined in the RFP and in the contract have been satisfied throughout product and service delivery • Ensure that developer/vendor/supplier has a process for

Stakeholder	Goals/Information Needs
	testing and reviewing software for vulnerabilities that has been and will be applied throughout the life of the contract • Validate that SwA considerations are included in the procurement • Ensure SwA requirements are explicitly addressed in solicitation and considered during the evaluation process • Validate that SwA requirements are integrated into SwA Requirements Document and implemented in the system • Ensure that project is staffed and structured to implement SwA • Monitor impact of security on business and mission support

2.2 Key Measures[7]

To help projects and organizations measurement should help answer key questions that provide insights into organization's performance. Different stakeholders may have different key questions that need answers and may gain different information from the same measures.

The SwA Measurement Working Group identified a number of example measures for the three stakeholder groups defined in section 2.1. These measures are generic in that they can be tailored for and used by a variety of projects and organizations. Table 2 lists examples that are mostly applicable to the Developer/Vendor/Supplier stakeholder group. Some of the measures listed in Table 2 may also be relevant to the Buyer/Acquirer stakeholder group. These measures can be used to assess the state of SwA efforts for a system or software development project. These measures are generic in that they can be used for a variety of projects. The overarching information need that these measures are intended to help satisfy can be summarized in three questions:

- Where are the errors in the design and code and can they be exploited
- How did they get there
- How can they be avoided in the future.

Table 2 displays the measures per project activity and provides corresponding information needs and benefits.

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

Table 2. Example Developer/Vendor/Supplier Measures

Project Activity	Measures	Information Need	Benefit
Project Management	• Percent of defects that negatively impact the security posture of the application (of the total number of defects)	• Identify software defects that may be exploited in the future	• Provides insight into the effectiveness of lifecycle processes and SwA training for developers • Indicates a need for additional security controls in implemented system
Requirements Management	• Percent of non-functional security requirements that are mapped to design • Percent of data entities with full validation constraints defined	• Determine if non-functional security requirements are being implemented in addition to being planned • Assert that all data entities have full data validation criteria defined	• Provides insight into inclusion of security requirements in early releases and into security requirements traceability • Provides insight into complexity of IA implementation • Provides insight into the degree of predictable behavior • Indicates the degree to which SwA can be tested • Indicator of short and long-term need for additional security controls in operations
Design	• Number of entry points for a module (should be as low as possible) • Percent of data input components that positively validate all data input • Number of defects and the area of the code in which they were found (it is a higher risk to have the defects in between components, unit seams, or other interfaces)	• Reduce opportunity for back doors • Determine if data validation is handled as required	• Low number of entry points reduces opportunities for back doors • Ensure that future application handles data inputs as required • Reduce opportunity for exploits

Project Activity	Measures	Information Need	Benefit
Development	• Number of discovered defects that are known as software vulnerabilities (e.g. buffer overflows and cross-site scripting) • Number of user-controllable inputs • Number of deviations between design, code and requirements • Number of times high risk statements (e.g., commands, APIs) are used • Percent of code coverage for which appropriate exception handling has been created • Percent of discovered defects that were fixed	• Proactively address the security defects prior to testing and deployment • Assure that the application performs exception handling as required	• Minimizes development and maintenance rework costs • Reduces the chances of introducing vulnerabilities Increases predictability of software behavior
Test	• Percent of modules that contain vulnerabilities that negatively impact the security posture of the system • Percent of failed security requirements • Percent of tests that evaluate application response to misuse, abuse, or threats • Percent of tests that attempt to subvert execution or work around security controls • Percent of security controls covered by tests • Percent of external messages with complete input validation • Percent of untested source code related to security control requirements	• Identify software defects that may be exploited in the future • Assess test coverage of security control requirements coverage	• Provides insight into risk of the system being exploited when in production • Provides a gauge for the degree to which the application behaves in a predictable manner • Provides a basis for understanding the degree of code coverage and how extensive is the security portion of the test • Indicates a need for additional security controls in implemented system
Entire SDLC	• Cost/Schedule variance in information security activities	• Monitor planning and implementation of security activities	• Provide insight into cost and schedule risks to project success • Increased accuracy in planning of future projects

Table 3 lists examples that are mostly applicable to the Buyer/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. The overarching

information need that these measures are intended to contribute to answering is, "Have SwA considerations been integrated into the SDLC by the Developer/Vendor/Supplier?" Table 3 displays the measures per acquisition activity and provides corresponding information needs and benefits.

Table 3. Example Buyer/Acquirer Measures

Acquisition Activity	Measures	Information Need	Benefit
Planning	• Percent of acquisition discussions that include SwA representative • Percent of contracting officers who received training in the security provisions of the FAR	• Validate 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 • Percent of positions filled with personnel possessing required qualifications and certifications • SwA requirements for sub-contractors are stated in the Subcontracting Plan and are addressed in Subcontracting Agreements • Contract language for validating that SwA requirements have been met is included in the solicitation	• Ensure SwA requirements are explicitly addressed in solicitation • Ensure SwA requirements are considered during the evaluation process	• Facilitates effective selection of Developer/Vendor/Supplier capable of delivering required level of SwA
Implementation and Acceptance	• Percent of documented Supplier claims validated through testing • Security role is included in the configuration management process • Percent of project staff trained on the principles of SwA	• Validate that SwA requirements are integrated into SwA Requirements Document and implemented in the system • Ensure that project is staffed and structured to implement SwA	• Risks associated with the software are identified and documented • Project staff are aware of SwA considerations and cognizant of associated requirements

Acquisition Activity	Measures	Information Need	Benefit
Follow on	• Cost of maintaining security after implementation • Percent of specific vulnerabilities discovered post-implementation caused by known vulnerabilities that could have been remediated before implementation • Percent of reported vulnerabilities that have been determined to have an unacceptable impact	• Monitor impact of security on business and mission support	• Provides insight into cost and impact of SDLC implementation on business and mission

Table 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's associated with software. The overarching information need that these measures are intended to contribute to answering is, "Is the risk generated by software acceptable for the organization?"

Table 4. Example Executive Measures

Measures	Information Need	Benefit

• Percent of applicable[8] vulnerabilities remediated before the system is operational
• Percent of data compromises traced to a specific vendor product
• Speed of response for each data compromise
• Cost to correct vulnerabilities in operational applications
  • Costs to fix known vulnerabilities discovered through code analysis
  • Cost to correct known security control deficiencies in operational applications
	• Gain insights into risk exposure from acquired/integrated product
	• Understand instances of data compromises caused by vendor products
		• Understand the level of risk and potential liability generated by acquired/integrated product
		• Insights into internal processes that need to change to reduce risks
		• Minimize risks created by vendor software

[8] "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

• Costs of individual data breaches
  • Discovery, notification, and response
  • Regulatory fines
  • Lost productivity
  • Liabilities
  • Brand damage/lost customers
	• Establish costs of breaches	• Provide a business case for devoting resources to SwA early within the SDLC

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

2.4 Integrated Measurement Approach

Software assurance measurement has to interact and be interoperable with system and software measurement and information security measurement. Common measurement framework integrates five industry approaches that propose similar measures development and implementation processes. These approaches are comparable and interoperable in terms that any of the approaches can be used to develop measures for SwA. Organizations should either integrate SwA into their current approach or select one of these approaches to implement 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*

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

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

- **CMMI®[9] (***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 due to their widespread use among 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

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

will 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 facilitate 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;
- Facilitating reuse of existing measures originating from other measurements approaches;
- Allowing stakeholder communities to continue using their methods and expand their view into measurement; and,
- Identifying gaps for further development.

The Framework can be used to develop measures and to design and implement measurement programs. Subsequent sub-sections address Common Measure Specification and implementation of measurement within a project or an organization.

2.4.1 Common Measure Specification

SwA measures can be integrated into an existing measurement program by leveraging common measure specification and by ensuring 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
- Analyzing the relationship between those two groupings of concepts to create a series of measures that describe this relationship.

Common Measure Specification documents individual elements of specifying a measure through documenting and mapping the selected methodologies in a single matrix. Table 5 provides an abbreviated version of the Common Measure Specification the current full version of which is provided in Appendix E. The readers of this document can use this specification to explore commonalities and differences between measurement approaches that they use within their respective domains and to translate measures from other domains into the methodology they currently use. The table illustrates that there are many similarities among the selected methodologies, where different terms may be used to communicate similar concepts. Light turquoise cells indicate a lack of corresponding item in the crosswalked methodologies. ISO/IEC 15939 and ISO/IEC 27004 provide the most detailed and comprehensive specifications among the methodologies.

Table 5. 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**		**Data Elements**	**Object of Measurement**	
	Attributes		**Data Elements**	**Attributes**	
Base Measure Specification	**Base Measure**		**Data Elements**	**Base Measure**	**Measure**
	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/Visual Display**	**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; Period of Analysis**	
	Tools Used in Data Collection	**Specify Data Collection and**	**Data Collection - Forms**	**Tools Used in Data Collection**	**Data Source**

	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
		Storage Procedures			
	Verification and Validation:	Collect Measurement Data	Data Storage - How	Collection Date; Reviewer; Information Owner	
	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 Communicato	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	

2.4.2 Implementing measures

To incorporate security measures into an existing measurement program, projects should start with a manageable set of measures. Basic measures like cost, schedule, quality, and growth can be expanded to explicitly include software assurance activities to provide insights into specific software assurance aspects of project management. As a project evolves, the project can add, refine or retire measures and implement new measures, as appropriate.[10]

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
- Analyzing collected data and compiling it into SwA measures

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

- Reporting SwA measures to appropriate stakeholders
- Implementing changes to address issues identified through measures
- Training and continuous improvement of measures to ensure measures are relevant to the project or organization.

Each of the approaches that comprise the Framework provides a process for implementing measures. While they may use different terms, all of them contain the basic process outlined above. As with the Common Measure Specification, organizations should pick a way of implementing measures and ensure that SwA considerations are integrated into the process.

3 DATA SOURCES FOR SWA MEASUREMENT

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

Enumerations, described in this section, provide a common language that describes aspects of software assurance, such as weaknesses, vulnerabilities, attacks, and configurations, and by doing so enable consistent and comparable measures. This section also provides an overview of automated tools that can analyze the data on weaknesses, vulnerabilities, attacks, configurations, accomplishment of milestones, financial indicators, and other items needed to make SwA measurement meaningful.

3.1 Enumerations

The Common Vulnerabilities and Exposures (CVE), Common Control Enumeration (CCE), Common Weakness Enumeration (CWE), and Common Attack Pattern Enumeration and Classification (CAPEC) are information security focused enumerations that allow people, processes, and products from different information security activities to be coordinated and connected, while also decoupling the various activities from each other, which lowers confusion between activities, speeds response times and reduces duplication.[11] 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 collection of metrics across different vendor tools and enables more advanced measurement.

Enumerations are useful throughout the SDLC for a variety of purposes, 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.

The following is a brief summary of the enumerations:

- **CVE** is a list of identifiers (ID) for publicly known vulnerabilities including 30,000+ separate bugs 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.

- **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

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

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.

- **CWE** is an enumeration of the architecture, design, and implementation weaknesses that can lead to exploitable security problems in software. It defines the application security related concepts that developers, testers, project managers, and customers should understand, avoid, and validate against. 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.

- **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.

Using enumerations as a basis for measurement helps accomplish the following goals:

1. Reduce weaknesses introduced during the development process

2. Understand completeness of attack methods that have been considered throughout development

3. Ascertain that publicly known vulnerabilities have had appropriate mitigations applied

4. Assess whether the system has been deployed and configured correctly

As with any measures, enumerations should be used appropriately to develop and collect measures. 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 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 current status of the 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.

3.1.1 Reducing Weaknesses During Development

Use of CWE for SwA measurement throughout the SDLC for both packaged and custom-built software helps 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. Table 4 lists examples of CWE-based measures.

Table 4. CWE-based Measures Examples.

Measures	Information Need	Benefit
• Number of weaknesses present in the system that would make it vulnerable to specific attacks *(ex, unauthorized access)* or a group of exploits • Number of weaknesses determined relevant for the given system configuration • Percent of relevant weaknesses found in application (of the total number of relevant weaknesses)	• Prioritize weaknesses for mitigation based on the weakness type (CWE) and the specific configurations that the system will run on	• Provides assurance that weaknesses are mitigated in order of exploitability based on the specific system configuration and therefore introduction of corresponding vulnerabilities is avoided • Provides information for prioritizing mitigating controls
• Number of instances of applicable CWEs found in software • Are they present (yes/no) • Number of present publicly known weaknesses • Density of a weakness against a context-specific measure of code, such as lines of code • Number/lines of code • Number/number of APIs • Number/interaction with a database	• Understand extent to which weaknesses are found in code and help identify mitigating strategies	

3.1.2 Using Attack Methods to Assess Development

Use of CAPEC throughout the SDLC for both packaged and custom-built software helps narrow down the set of relevant weaknesses by identifying relevant attack patterns that may target them. CAPEC can be useful for a number of purposes including:

- Scope the set of relevant weaknesses by identifying likely attacks
- Identify appropriate tests based on relevant attack patterns[12]
- Evaluate test coverage
- Evaluate penetration testing provider and their approach
- Evaluate tools
- Identify mitigating scenarios and security controls as an analytical tool to help risk mitigation

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

- Prioritize weakness mitigation.

Table 5 lists examples of CAPEC-based measures.

Table 5. CAPEC-based Measures Examples.

Measures	Information Need	Benefit
• Number of relevant attack patterns • List of relevant attack patterns	• Understand the breadth of attacks that the system could experience	• To enable threat modeling during requirements
• 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

3.1.3 Measuring Vulnerability Mitigation

Use of CVE during testing of packaged software installed on operational systems helps identify specific vulnerabilities that require mitigation. 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. Table 6 lists examples of CVE-based measures.

Table 6. CVE-based Measures Examples.

Measures	Information Need	Benefit
• Number of CVEs found in the system • Number of exploitable CVEs in the system • Number of mitigated CVEs through various types of mitigating strategies, such as patches and service packs and mitigating controls	• Ascertain that all appropriate mitigating strategies have been collectively applied	• Focus vulnerability mitigation to exploitable vulnerabilities vs. all vulnerabilities regardless of their applicability

• Solution volatility
  • Number of vulnerabilities discovered over predefined time frame (month, 6 months, year, etc)
  • Number of people who discovered vulnerabilities
  • Number of discovered vulnerabilities by type
	• Understand solution history in terms of publishing vulnerable code	• Ability to evaluate volatility of solutions

Measures	Information Need	Benefit
• Elapsed time between vulnerability discovery and application of mitigation	• Understand time of exposure caused by newly discovered vulnerabilities	• Insight into risk exposure and vendor responsiveness

3.1.4 Assessing Deployed Configurations

Use of CCE across the SDLC for packaged software helps 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. Table 7 lists examples of CCE-based measures.

Table 7. CCE-based Measures Examples.

Measures	Information Need	Benefit
• Percent of compliant configurations/system components/etc.	• Establish that software is configured according to specific minimum configuration requirements or stronger	• Measurable proof of compliance or non-compliance with specific configuration requirements or technical security controls

3.1.5 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. Table 8 lists examples of such measures.

Table 8. Examples of Enumerations-Based Measures for Assessing Skills and Knowledge.

Measures	Information Need	Benefit
• Coverage of enumerations-related material by relevant training programs	• Understand relevancy of a specific training program against latest attack, weakness, vulnerability, and configurations knowledge base	• Evaluate currency and robustness of training

3.2 Automated Tools

Comment [NB1]: Question for the group – should we be mentioning specific tools in the document or move this section out to the web site and point?

A variety of automated tools can be used to facilitate measurement process and to reduce its costs. The following is a list of tools that have been successfully used as part of a measurement program.

- PSM Insight (PSMI) from Practical System and Software Measurement (PSM) (http://www.psmsc.com/PSMI.asp)
- Data Drill from Distributive Management (http://www.distributive.com/index.htm)
- Fortify 360 (http://www.fortify.com/products/governance.jsp)
- Ounce Labs, Software Risk Analysis (http://www.ouncelabs.com/solutions/manage_risk_across_enterprise_portfolio.asp)
- HP Assessment Management Platform software to manage web application security testing (https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200^9580_4000_100__)
- Cenzic Hailstorm® Enterprise ARC™, Test All Web Applications—Developed and Production (http://www.cenzic.com/pdfs/HailstormEntARC_DS_120707.pdf)
- AppScan Reporting Console: Centralized Web Application Security Reporting (http://www.watchfire.com/products/appscan/appscanconsole.aspx)

APPENDIX A — REFERENCES

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

NIST Special Publication (SP) 800-53 Revision 1, *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/)

Data Drill from Distributive Management (http://www.distributive.com/index.htm)

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/)

CorporateInformation 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.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
CMMI®	Capability Maturity Model Integration
CNSS	Committee on National Security Systems
COBIT	Control Objectives for Information and related Technology
CVE	Common Vulnerabilities and Exposures
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
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 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 for the specific development, selection, and implementation of IT system-level metrics to be used to measure the performance of information security controls and techniques[13]

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 measurements in order to assess the effectiveness of information security management system, control objectives, and controls used to implement and manage information security, as specified in ISO/IEC 27001, Information Security Management System – Requirements. The use of this standard will guide the organization to establish and maintain a program to monitor implementation and effectiveness of the information security management program.[14]

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.[15]

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.

---

[13] NIST SP 800-55, *Security Metrics Guide for Information Technology Systems*

[14] ISO/IEC 27004 Information technology - Security techniques - Information security management measurements

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

Measurement Frameworks

Balanced Scorecard – The Executive Branch Management Scorecard tracks how well the departments and major agencies are executing the five government-wide management initiatives. Scores for "status" are based on the scorecard standards for success. Each initiative is rated red, yellow, or green based on how well the department or agency performs in the standards for success. The standards for success were developed by the President's Management Council and have subsequently been refined by incorporating lessons learned through experience in implementing the President's Management Agenda. Under each of these standards, an agency is "green" or "yellow" if it meets all of the standards for success, and "red" if it has any one of a number of serious flaws listed in the "red" column. The standards developed are objective in nature, although metrics could conceivably be developed to support the ratings.[16]

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.[17]

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.[18]

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. [19]

iCMM[20] – describes the essential elements of an organization's acquisition, engineering, and management process that must exist to ensure good acquisition of software intensive systems.

---

[16] http://www.whitehouse.gov/results/agenda/scorecard.html

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

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

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

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

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.

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.[21]

OMG Common Measurement Specification – a software metrics metamodel which facilitates the inoperability of measurements of software artifacts.[22]

Qualitative Assessment Methods

ISO/IEC 15504, *Information Technology – Software Process Assessment* –

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

System Security Engineering Capability Maturity Model (SSE-CMM) Appraisal Method (SSAM) - provides organizations with insight into the processes being practiced within the organization or project

Federal Aviation Administration (FAA) Integrated Capability Maturity Mode (iCMM) Appraisal method – 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.[23]

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

---

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

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

[23] http://www.sans.org/reading_room/whitepapers/standards/545.php The Common Criteria Iso/Iec 15408– The Insight, Some Thoughts, Questions And Issues By Ariffuddin Aizuddin SANS Institute This is not a valid link…not sure where you are trying to take the reader….

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.[24]

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 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[25]

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. [26]

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*

---

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

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

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

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.

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. Categories are defined in Chapter 2 of the PSM book.			**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 behaviour 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. **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.	**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 Format: IIndication 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).	
				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. **Reviewer:** Person or organizational unit who reviews that the measure evaluation criteria are appropriate to verify the control effectiveness. **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.	
	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.	**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.

	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 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).	
	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.	**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.
				Reviewer: Person or organizational unit who reviews that the measure evaluation criteria are appropriate to verify the control effectiveness.	
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.	

	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
	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.	