Assured Software Development 1 (ASD1)

INSTRUCTOR MATERIALS

Assured Software Development 1

CERT® Division

<http://www.sei.cmu.edu>

**http://www.cert.org**

Copyright 2014 Carnegie Mellon University  
  
This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.  
  
Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Department of Homeland Security or the United States Department of Defense.  
  
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.  
  
This material has been approved for public release and unlimited distribution except as restricted below.  
  
Internal use:\* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.  
  
External use:\* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.  
  
\* These restrictions do not apply to U.S. government entities.  
  
Architecture Tradeoff Analysis Method®, ATAM®, Carnegie Mellon® and CERT® are registered marks of Carnegie Mellon University.  
  
Personal Software ProcessSM, PSPSM, Team Software ProcessSM and TSPSM are service marks of Carnegie Mellon University.  
  
DM-0001343

Table of Contents

[Overview 1](#_Toc388001204)

[Schedule 5](#_Toc388001205)

[1 Week 1 7](#_Toc388001206)

[Homework Assignment One 8](#_Toc388001207)

[2 Week 2 9](#_Toc388001208)

[Case Study Team Formation 10](#_Toc388001209)

[3 Week 3 11](#_Toc388001210)

[Case Study Assignment One 13](#_Toc388001211)

[4 Week 4 14](#_Toc388001212)

[5 Week 5 15](#_Toc388001213)

[Homework Assignment Two 17](#_Toc388001214)

[Case Study Assignment Two 18](#_Toc388001215)

[6 Week 6 19](#_Toc388001216)

[Case Study Assignment Three 21](#_Toc388001217)

[7 Week 7 22](#_Toc388001218)

[8 Week 8 23](#_Toc388001219)

[9 Week 9 24](#_Toc388001220)

[Case Study Assignment Four 26](#_Toc388001221)

[10 Week 10 27](#_Toc388001222)

[Homework Assignment Three 28](#_Toc388001223)

[11 Week 11 29](#_Toc388001224)

[Homework Assignment Four 32](#_Toc388001225)

[12 Week 12 33](#_Toc388001226)

[13 Week 13 34](#_Toc388001227)

[Case Study Assignment Five 35](#_Toc388001228)

[Case Study Assignment Final Report 36](#_Toc388001229)

[14 Week 14 37](#_Toc388001230)

[Bibliography 38](#_Toc388001231)

Overview

*Assured Software Development 1* (ASD1) is one of the eight courses (plus capstone experience) that comprise the Master of Software Assurance Reference Curriculum as detailed in *Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum* [Mead 2010].

This document is meant to provide additional information to the instructor. See also the notes portion of the PowerPoint slides and the slides themselves, as in some cases exercise and additional information are given after the end of the formal lecture. In some instances, some suggested answers are also given in the slide set end matter or in slide notes.

This document also references additional files/material. Additional references may be cited in this guide or in the slides/notes, but are not necessarily listed in the Bibliography (e.g. the additional elicitation techniques in Lecture 7, and the additional threat modeling learning resources in Lecture 9). A subset of this document can be used to construct a syllabus for the student.

The report *Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum* [Mead 2010] defines software assurance as

“application of technologies and processes to achieve a required level of confidence that software systems and services function in the intended manner, are free from accidental or intentional vulnerabilities, provide security capabilities appropriate to the threat environment, and recover from intrusions and failures.”

Catalog Description

The *Assured Software Development 1* course covers the fundamentals of incorporating assurance practices, methods, and technologies into software development and acquisition lifecycle processes and models. With this foundation, the course provides students with rigorous methods for eliciting software and system assurance requirements; using threat identification, characterization, and modeling; assurance risk assessment; and misuse/abuse cases. Students will also learn how to evaluate methods and environments for creating software and systems that meet their functionality and security requirements.

Prerequisites

This course assumes that students have a software engineering background, such as knowledge of common lifecycle models.

There are two prerequisites for this course

* Undergraduate software engineering course (or equivalent)
* Undergraduate information security course (or equivalent)

Learning Outcomes

After completing this course, students will be able to

* Understand lifecycle models and processes for newly developed software systems
* Understand lifecycle models and processes for the acquisition, supply, and service of a software system
* Use methods and techniques to assess the applicability of assurance processes and practices for typical lifecycle phases
* Elicit and analyze requirements for assured software
* Apply security requirements engineering methods in developing assurance requirements

Course Topics

Security models and methods in the areas of

* Lifecycle process models
* Risk management
* Requirements engineering
* Architecture and design
* Threat modeling
* Strategy for implementing a software security initiative

As time permits, the acquisition of newly developed and commercial-off-the-shelf (COTS) software will also be discussed.

Approach to Learning

This course consists of a series of slide sets for lectures and readings. The lectures lead the way through a series of assignments and classroom discussions. Readings are from the text book, reports, and papers. Assignments include homework assignments and a project. The project includes selected software development activities:

* Lifecycle security management plan
* Selection of a process model (Agile, Spiral, etc.) and rationale
* Security risk analysis
* Development of misuse cases/attack trees
* Security requirements elicitation
* Architectural trade-off analysis/Quality Attribute Workshop (QAW)
* Design of security features (e.g. access control mechanisms)
* Inspection

Students are encouraged to

* Do the assigned readings.
* Attend and participate in the lecture.
* Complete the assignments.

Readings

Students are required to complete a series of readings related to each lecture. These readings are used to stimulate discussion and expose the students to course topics not covered directly in the lectures. A complete Bibliography is provided in this document.

The required text for this course is *Software Security Engineering: A Guide for Project Managers* [Allen 2008]. The text is available from Pearson Education, Inc. and Amazon.com.

Abstract from the publisher

Software that is developed from the beginning with security in mind will resist, tolerate, and recover from attacks more effectively than would otherwise be possible. While there may be no silver bullet for security, there are practices that project managers will find beneficial. With this management guide, you can select from a number of sound practices likely to increase the security and dependability of your software, both during its development and subsequently in its operation.

Software Security Engineeringdraws extensively on the systematic approach developed for the Build Security In (BSI) Web site. Sponsored by the Department of Homeland Security Software Assurance Program, the BSI site offers a host of tools, guidelines, rules, principles, and other resources to help project managers address security issues in every phase of the software development life cycle (SDLC). The book’s expert authors, themselves frequent contributors to the BSI site, represent two well-known resources in the security world: the CERT Program at the Software Engineering Institute (SEI) and Cigital, Inc., a consulting firm specializing in software security.

This book will help you understand why

* Software security is about more than just eliminating vulnerabilities and conducting penetration tests
* Network security mechanisms and IT infrastructure security services do not sufficiently protect application software from security risks
* Software security initiatives should follow a risk-management approach to identify priorities and to define what is “good enough”—understanding that software security risks will change throughout the SDLC
* Project managers and software engineers need to learn to think like an attacker in order to address the range of functions that software should not do, and how software can better resist, tolerate, and recover when under attack

There are additional readings, videos, etc., which include materials from the U.S. Department of Homeland Security Build Security In site (<https://buildsecurityin.us-cert.gov/>).

Unless otherwise directed, the students should read the entire paper or article assigned. Sometimes the student will need to read only one or two chapters from a book, or to skim a website. Directions can be found in the margin next to the reading list.

Evaluation

We wish to give students the maximum flexibility in completing the course, but it is in the student’s best interest to spread work evenly, so that feedback can be given regularly.

Grading will take into consideration

* Completeness
* Creativity
* Deep insights
* Thinking outside the box

Grading criteria include three factors: individual homework assignments, a team case study, and class participation:

* Homework assignments (45%) 9% for each homework assignment
* Case study assignments (50%) 10% for each assignment (This is a TEAM grade.)
* Class participation (5%): constructive participation in class discussions

The result of the distribution of these factors is that 50% of the grade is based on individual work and 50% is based on group assignments.

Homework and case study assignments should be turned in on the day they are due,***before*** class begins. Homework and case studies submitted after the deadline lose 10% for every day late, and this starts immediately after the deadline.

Cooperation Policy, Citations, and Quotations

Final grades may be adjusted up or down based on the instructor’s judgment, taking into account factors such as evidence of effort and participation in class. Materials that are referenced from other sources must be correctly cited. Materials that are directly lifted must be quoted.

The instructor will conduct chat sessions every week, at a specific time determined by the instructor. Other times, the instructor will be available by email. In addition, students should feel free to post questions and comments on the course electronic bulletin board at any time to discuss the readings, the course, and issues related to software security engineering with other members of the class.

Schedule

The following section contains a weekly schedule for a semester course. The due dates are notional. The instructor can make modifications to the schedule as they see fit.

Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| WEEK | Topic | Readings | Assignments |
| 1 | Introduction and Software Lifecycle Models | [Allen 2008] Chapters 1 & 2, [Saltzer 1974], [Woody 2012], [Montalbano 2011], [IEEE 2011] | Homework 1 distributed |
| 2 | Process Frameworks |  | Form Case Study Teams  Homework 1 DUE |
| 3 | Software Assurance Lifecycle and Maturity Models: Overview of lifecycle and maturity models; CLASP | [Allen 2008] Chapters 1 & 2, [Lipner 2013] See also additional resources listed with lecture | Case Study Team Membership DUE  Case Study Assignment 1 distributed |
| 4 | Selected Maturity Models in Detail:  OpenSAMM; BSIMM4 | [OWASPa], [BSIMM] | Case Study Assignment 1 DUE |
| 5 | SQUARE Overview and Steps 1-2 | [Allen 2008] Chapter 3, [Beckers 2012], [Khan 2009], [BSI site], [Mead 2005] | Homework 2 distributed  Case Study Assignment 2 distributed. (Note this assignment spans multiple weeks through Week 8). |
| 6 | Risk Analysis and SQUARE Steps 3-4 | [Alberts 2011], [CERT 2012] See also additional resources listed with lecture | Case Study Assignment 3 distributed  Homework 2 DUE |
| 7 | SQUARE Steps 5-7 | Additional resources: See [Mead 2005] for more elicitation technique references (mentioned on slide 7). | Case Study Assignment 3 DUE |
| 8 | Prioritization, Inspections, and SQUARE Steps 8-9 |  |  |
| 9 | Threat Modeling | [Allen 2008] Chapter 5, [Ingalsbe 2008a], [Ingalsbe 2008b], [OWASPb], [Microsoft 2009], including the lab Additional resources include a threat modeling video and learning resources. | Case Study Assignment 2 DUE  Case Study Assignment 4 distributed |
| 10 | Threat Modeling and Secure Tropos | [Mead 2009a], [Mead 209b] See also additional resources | Case Study Assignment 4 DUE  Homework Assignment 3 distributed |
| 11 | Quality Attribute Workshop and Attack Surfaces | [Allen 2008] Chapter 6, [Mead 2012], [Manadhata 2011] See also additional resources | Homework Assignment 3 DUE  Homework Assignment 4 distributed |
| 12 | Supply Chain Risk Management | See also additional resources | Homework Assignment 4 DUE |
| 13 | SQUARE for Acquisition | [Mead] ASQUARE white paper  See also additional resources | Case Study Assignment 5 distributed |
| 14 | Student Presentations |  | Case Study Assignment 5 DUE  Final Case Study Report DUE |

# Week 1

Introduction and Software Lifecycle Models

This first lecture is an overview of the course and an introduction to the topics that will be covered. We start to explore the challenges of software assurance and address why it is important. An overview of software development lifecycle models is also presented.

Unless otherwise directed, the students should read the entire paper or article assigned. Sometimes the student will need to read only one or two chapters from a book, or to skim a website. Directions can be found in the margin next to the reading list.

Readings

|  |  |
| --- | --- |
| **[Allen 2008]** Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers*. Boston, MA: Addison-Wesley, 2008. | Read 1&2 |
| **[Saltzer 1974]** Saltzer, Jerome H. & Schroeder, Michael D. “The Protection of Information in Computer Systems.” *Communications of the ACM* 17, 7 (July 1974): 388-402. |  |
| **[Woody 2012]** Woody, Carol; Mead, Nancy; & Shoemaker, Dan. “Foundations for Software Assurance,” 5368-5374. *Proceedings of the 45th Hawaii International Conference on Systems Science*, Maui, Hawaii, January 4-7, 2012. New York: IEEE Computer Society Press, 2012. |  |
| **[Montalbano 2011]** Montalbano, Elizabeth. “Air Force Says Drone Virus is No Threat: an attack on the network that controls U.S. military unmanned aerial vehicles was only a ‘nuisance’ military arm claims.” *Information Week Dark Reading* (October 2011). [http://www.](http://www/) informationweek.com/government/security/air-force-says-drone-virus-is-no-threat/231900741 |  |
| **[IEEE 2011]** Institute of Electrical and Electronics Engineering. “Malware Infects US Military Drone System.” *Computer* 44, 11 (November 2011): 16. |  |

Objectives

* Understand various types of attacks and avoidance techniques.
* Learn about historic and modern assurance principles and how they differ.
* Learn about software development lifecycles.

Homework Assignment One

1. [80%] Surf the web and find four different actual examples of successful intrusion:
2. One that resulted from human error (e.g., such as giving out a password or downloading a virus)
3. One that resulted from a system configuration error
4. One that resulted from software providing an intrusion opportunity because of a flawed development process
5. One that resulted from a vulnerability in a COTS product

Describe how each of these attacks could have been avoided. Consider changes in policy, configuration management, software development practice, and COTS acquisition practices.

1. [20%] Compare and contrast the HICSS Principles paper with the Saltzer and Schroeder Principles paper.

Turn this assignment in BEFORE the next class.

# Week 2

Process Frameworks

An overview of process is presented, followed by overviews of the Personal Software Process (PSP); the Team Software Process (TSP); the Architecture-Centric Development Method (ACDM); Agile processes: eXtreme Programming (XP) and Scrum; and Rational, Agile, and Open Unified Processes (RUP, AUP, OUP).

Readings

None

Objectives

* Learn about process frameworks

Case Study Team Formation

Form teams of four to five people.

Each team should have one or more students working on a software development project that can be used as a software security case study.

The team members should have reasonably compatible schedules in order to accomplish the team work.

Turn this assignment in BEFORE the next class.

# Week 3

Software Assurance Lifecycle and Maturity Models: Overview of Lifecycle and Maturity Models; CLASP

We talk about software assurance practices with motivation, similar to the motivating scenarios that were discussed previously. We discuss lifecycle models that are specific to software assurance, and then start to talk about some maturity models that are specific to software assurance.

You will see that the lifecycle models in certain regards look rather different than lifecycle models you may have seen before. The same can be said of the maturity models. You will find that once you introduce software assurance, it is a different view altogether.

Readings

|  |  |
| --- | --- |
| **[Allen 2008]** Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers.*Boston, MA: Addison-Wesley, 2008. | Read 1&2 |

Additional Resources

|  |  |
| --- | --- |
| **[Lipner 2013]** Lipner, Steve. “The Security Development Lifecycle.” Three-part video. <http://www.cert.org/curricula/sae-course-materials-by-type.cfm> (Read and accept the click-thru license agreement then scroll down to “The Security Development Lifecycle.”) | *Either show part of the Lipner video in class or assign viewing for homework and discuss in the next class.* |
| **[BSIMM]** *Building Security in Maturity Model*[. http://bsimm.com/download/](http://bsimm.com/download/) (no registration required).  (note this is now BSIMM V) |  |
| **[Davis 2013]** Davis, Noopur. “Secure Software Development Life Cycle Processes,” *Build Security In.* (July 2013). <https://buildsecurityin.us-cert.gov/articles/knowledge/sdlc-process/secure-software-development-life-cycle-processes>  **[Lipner 2005]** Lipner, Steve & Howard, Michael. “The Trustworthy Computing Security Development Lifecycle.” (March 2005). <http://msdn.microsoft.com/> en-us/library/ms995349.aspx |  |
| **[OWASPa]** Open Web Application Security Project[. https://www.owasp.org/](file:///C:/Users/cas/Desktop/June%202009%20NEED%20TO%20FILE/140505%20xfer%20to%20laptop/.%20https:/www.owasp.org/) (For CLASP see <https://www.owasp.org/index.php/CLASP>) |  |
| **[van Wyk 2005]** van Wyk, Kenneth R. & McGraw, Gary. “Bridging the Gap Between Software Development and Information Security.” *IEEE Security & Privacy* (September/ October 2005): 64-68. |  |

Steve Lipner’s video, “The Security Development Lifecycle,” has three parts:

1. [Creating and Selling the Security Development Lifecycle (SDL)](http://www.cert.org/curricula/sae-course-materials-by-type.cfm?stream=SAE_D02_S01_T01).

<http://www.cert.org/curricula/sae-course-materials-by-type.cfm?stream=SAE_D02_S01_T01>

1. [Managing the Process](http://www.cert.org/curricula/sae-course-materials-by-type.cfm?stream=SAE_D02_S01_T02).

<http://www.cert.org/curricula/sae-course-materials-by-type.cfm?stream=SAE_D02_S01_T02>

1. [Making a Difference](http://www.cert.org/curricula/sae-course-materials-by-type.cfm?stream=SAE_D02_S02_T01).

<http://www.cert.org/curricula/sae-course-materials-by-type.cfm?stream=SAE_D02_S02_T01>

Objectives

* Learn to select a security lifecycle approach.
* Identify the activities supported by the selected approach.
* Overview the Open Web Application Security Project (OWASP) and Comprehensive, Lightweight, Application Security Process (CLASP) methods.

Resources for an In-Class Exercise

“Lifecycle and MM exercise Study Case.docx” provides a richly detailed background of the organization, its business environment, its software engineering process aspects, etc. leading up to a specific case: the organization being contracted by the Army to upgrade the target acquisition and display fire control system for the AH64-D Apache Longbow attack helicopter. This background and specific case can be used as a basis for both a secure lifecycle model exercise and, in the next lecture, a maturity model exercise.

Since this study targets both organizational and software engineering processes, additional information and descriptions of how it relates to the ISO 12207:2008 standard is provided in the “Solution basis for Lifecycle and MM Study Case.docx” materials.

The last page/section of “Lifecycle and MM exercise Study Case.docx” describes a larger assignment/exercise that can be tailored as desired. The lifecycle exercise is the first described. As noted earlier the “Solution basis for Lifecycle and MM Study Case.docx” provides additional information as well as some suggestions regarding an answer (not the only one).

Finally, as we will see in the next lecture, all the secure lifecycle models are good for experts to use as a guide, but for difficult for non-security personnel to use off the shelf. Microsoft’s SDL is heavyweight, and good for large, independent software vendors. Touchpoints are high-level, but there are not enough details to execute against, while CLASP is a large collection of activities, but there is no priority ordering among the activities.

Case Study Assignment One

1. [15%] Describe the project and why it is a good software security project OR the changes that you had to make to get it to be a good software security project.
2. [20%] Describe the security lifecycle approach that your team intends to use and the rationale behind it. Why is it better than other approaches?
3. [15%] What are the activities that this lifecycle approach supports?
4. [20%] What is the underlying development model (e.g., Waterfall, Spiral, Agile)? Why is it a good model for this project?
5. [15%] How well do the security activities fit with the selected development model?
6. [15%] Compare your activities to the activities described in BSIMM4. Describe the similarities and differences. Are there important differences from a software security viewpoint?

Turn this assignment in BEFORE the next class.

# Week 4

Selected Maturity Models in Detail: OpenSAMM; BSIMM4

After reviewing other secure software development lifecycle efforts, we explore in greater detail the OWASP Software Assurance Maturity Model (OpenSAMM) and the Building Security In Maturity Model (BSIMM).

Readings

|  |  |
| --- | --- |
| **[OWASPa]**  Open Web Application Security Project[. https://www.owasp.org/](file:///C:/Users/cas/Desktop/June%202009%20NEED%20TO%20FILE/140505%20xfer%20to%20laptop/.%20https:/www.owasp.org/). (For SAMM see <https://www.owasp.org/index.php/SAMM>.) |  |
| **[BSIMM]**  Building Security in Maturity Model[. http://bsimm.com/download/](http://bsimm.com/download/) (no registration required). (Note this is now BSIMM V.) |  |

Objectives

Develop the ability to

* Evaluate an organization’s existing software security practices.
* Build a balanced software security assurance program in well-defined iterations.
* Demonstrate concrete improvements to a security assurance program.
* Define and measure security-related activities throughout an organization.

Resources for an In-Class Exercise

As discussed in the Week 3 Lifecycle exercise notes, “Lifecycle and MM exercise Study Case.docx” provides a richly detailed background of the organization, its business environment, its software engineering process aspects, etc. leading up to a specific case, which can be used as a basis for a maturity model exercise. The last page/section of “Lifecycle and MM exercise Study Case.docx” describes a larger assignment/exercise, which can be tailored as desired (the maturity model exercise is the last described). The “Solution basis for Lifecycle and MM Study Case.docx” provides some additional information and some suggestions regarding an answer to the maturity model exercise, again as the last item described.

# Week 5

SQUARE Overview and Steps 1-2

In the previous week we discussed maturity models. We begin this week with a discussion of requirements engineering and security requirements methods, including Security Quality Requirements Engineering (SQUARE). An overview of the nine-step SQUARE Methodology is given, concluding with a more detailed discussion of Steps 1 and 2 (Agree on definitions. Identify assets and security goals). For the instructor there are also additional materials available at <http://www.cert.org/curricula/square-instructional-materials.cfm>.

Readings

|  |  |
| --- | --- |
| **[Allen 2008]** Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers.*Boston, MA: Addison-Wesley, 2008. | Read 3 |
| **[Beckers 2012]** Beckers, Kristian; Fassbender, Stephen; Heisel, Maritta, Küster; Jan-Christoph; & Schmidt, Holger. “Supporting the Development and Documentation of ISO 27001 Information Security Management Systems through Security Requirements Engineering Approaches,” 14-21. *Engineering Secure Software and Systems 2012*, Lecture Notes in Computer Science 7159. Berlin: Spring-Verlag, 2012. <http://link.springer.com/chapter/10.1007/978-3-642-28166-2_2> |  |
| **[Khan 2009]** Khan, Muhammad Umair Ahmed & Zulkernine, Mohammed. “On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software,”353-358. *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference.* Seattle, WA, July 20-24, 2009. New York: IEEE Computer Society Press, 2009. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5254051&tag=1 |  |
| **[Mead 2005]** Mead, Nancy R.; Hough, Eric; & Stehney II, Ted. *Security Quality Requirements Engineering* (CMU/SEI-2005-TR-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. [www.sei.cmu.edu/pub/documents/05.reports/pdf/05tr009.pdf](http://www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm) |  |
| **[Build Security In]** *Build Security In*. <https://buildsecurityin.us-cert.gov/> (Requirements Engineering articles: https://buildsecurityin.us-cert.gov/articles/best-practices/requirements-engineering) | Skim BSI site for content on requirements engineering. |

Additional Resources for Exercises

|  |  |
| --- | --- |
| *SQUARE Workshop Guide*: Client Team materials |  |
| *SQUARE Workshop Guide*: Requirements Engineering Team materials |  |
| These two sets of materials will be used by the students to conduct the in-class SQUARE exercises. Note that there is Supplemental Material on Brainstorming and an extensive reference list at the back of each of the Client and Requirements Engineering Team materials. |  |

Objectives

1. Understand aspects of security requirements engineering in general.
2. Learn more about SQUARE Steps 1 and 2.

Additional Information Regarding Case Study 2 Assignment

Unlike other assignments, this one spans multiple weeks (through the discussion of all of the SQUARE steps). Recommend to the students that the team complete the steps as they are discussed in class; they should not wait until Week 8 to begin this assignment, which is due before the start of the Week 9 lecture.

Homework Assignment Two

1. [25%] You are working on a project where you can select a security requirements engineering process. First you want to decide on some criteria for selection. What criteria do you pick (refer to the Khan/Zulkernine and Beckers papers for a start)?
2. [50%] Using those criteria, which existing process is the best fit (you can use the list of processes from this lecture as a start)?
3. [25%] Does the selected process need to be modified for your project?

Turn this assignment in BEFORE the next class.

Case Study Assignment Two

1. Using the SQUARE Technical Report as a guide, apply SQUARE Steps 1, 2, 3, 4, 5, 6, 7, and 8 to your case study project.

Note: You do not need to interview your actual stakeholders for purposes of this exercise.

1. Develop attack trees and selected corresponding misuse cases as part of this exercise.

This is a TEAM exercise.

Turn this assignment in BEFORE the class in Week 9. Since this assignment spans multiple weeks, it is recommended that the team complete the steps as they are discussed in class–do not wait until Week 8 to begin this assignment.

# Week 6

Risk Analysis and SQUARE Steps 3-4

The basic goal of risk analysis is to provide decision makers with the information they need, when they need it, and in the right form. In this lecture, a brief overview of risk management is followed by a discussion of two different approaches for managing risk—mission and event risk analysis. The Security Engineering Risk Analysis (SERA) is described in detail, followed by SQUARE Steps 3 and 4 (Develop artifacts to support security requirements definition. Assess Risks.).

Readings

|  |  |
| --- | --- |
| **[Alberts 2011]** Alberts, Christopher; Dorofee, Audrey J.; Creel, Rita; Ellison, Robert J.; & Woody, Carol. “A Systematic Approach for Assessing Software Supply- Chain Risk,” 1-8. *Proceedings of the 44th Hawaii International Conference on Systems Science,* Kauai, Hawaii, January 4-7,2011*.* New York: IEEE Computer Society Press, 2011. |  |

Additional Resources

|  |  |
| --- | --- |
| **[Alberts 2012a]** Alberts, Christopher; Allen, Julia; & Stoddard, Robert. *Risk-Based Measurement and Analysis: Application to Software Security* (CMU/SEI-2012-TN-004), Software Engineering Institute, Carnegie Mellon University, 2012. <http://www.sei.cmu.edu/reports/12tn004.pdf> |  |
| **[Alberts 2012b]** Alberts, Christopher & Dorofee, Audrey. *Mission Risk Diagnostic (MRD) Method Description* (CMU/SEI-2012-TN-005). Software Engineering Institute, Carnegie Mellon University, 2012. <http://www.sei.cmu.edu/reports/12tn005.pdf> |  |
| **[Alberts 2009]** Alberts, Christopher & Dorofee, Audrey. *A Framework for Categorizing Key Drivers of Risk* (CMU/SEI-2009-TR-007). Software Engineering Institute, Carnegie Mellon University, 2009. <http://www.sei.cmu.edu/library/abstracts/reports/09tr007.cfm> |  |
| **[CERT CSE]** Cyber Security Engineering (CSE) Team Web Page. <http://www.cert.org/cybersecurity-engineering/index.cfm> |  |
| **[MSCE]** SEI Mission Success in Complex Environments (MCSE) Special Project. <http://www.sei.cmu.edu/risk> |  |

Additional Resources for Exercises

|  |  |
| --- | --- |
| *SQUARE Workshop Guide*: Client Team materials |  |
| *SQUARE Workshop Guide*: Requirements Engineering Team materials |  |
| These two sets of materials will be used by the students to conduct the in-class SQUARE exercises. Note that there is Supplemental Material on Brainstorming and an extensive reference list at the back of each of the Client and Requirements Engineering Team materials. |  |

Objectives

* Understand risk analysis.
* Understand the assessment of operational security risks early in the software lifecycle through the SERA method.
* Learn more about SQUARE Steps 3 and 4.

Case Study Assignment Three

1. [15%] Document the target of the analysis (i.e., the software or system being developed and deployed). Describe how the software or system will be used during operations.
2. [35%] Consider a range of scenarios that are putting the target at risk. Select one scenario that you intend to analyze. Document the following for the selected scenario:

* Threats
* Consequences
* Enablers

1. [10%] Document a risk statement (in if-then format) for the risk scenario that you selected.
2. [35%] For the purpose of this assignment, it is assumed that you will mitigate the selected risk. Document mitigation actions for the risk. Remember to consider requirements from the following categories:

* Monitor and respond
* Protect
* Recover

Identify which of these mitigation actions can be addressed during development.

1. [5%] Describe what insights you gained (if any) by applying the method.

This is a TEAM exercise.

Turn this assignment in BEFORE the next class.

# Week 7

SQUARE Steps 5-7

SQUARE Steps 5-7 are described in more detail (Select elicitation technique(s). Elicit security requirements. Categorize requirements.).

Readings

|  |  |
| --- | --- |
| None |  |

Additional Resources

|  |  |
| --- | --- |
| See [Mead 2005] for more elicitation technique references (cited on slide 7). |  |

Additional Resources for Exercises

|  |  |
| --- | --- |
| *SQUARE Workshop Guide*: Client Team materials |  |
| *SQUARE Workshop Guide*: Requirements Engineering Team materials |  |
| These two sets of materials will be used by the students to conduct the in-class SQUARE exercises. Note that there is Supplemental Material on Brainstorming and an extensive reference list at the back of each of the Client and Requirements Engineering Team materials. |  |

Objectives

* Learn more about SQUARE Steps 5, 6, and 7.

# Week 8

Prioritization, Inspections, and SQUARE Steps 8-9

SQUARE Steps 8 and 9 are described in more detail (Prioritize requirements. Inspect requirements.).

Readings

|  |  |
| --- | --- |
| None. |  |

Additional Resources for Exercises

|  |  |
| --- | --- |
| *SQUARE Workshop Guide*: Client Team materials |  |
| *SQUARE Workshop Guide*: Requirements Engineering Team materials |  |
| These two sets of materials will be used by the students to conduct the in-class SQUARE exercises. Note that there is Supplemental Material on Brainstorming and an extensive reference list at the back of each of the Client and Requirements Engineering Team materials. |  |

Objectives

* Learn more about SQUARE Steps 8 and 9.

# Week 9

Threat Modeling

In this lecture we introduce threat modeling and look at how to do threat modeling, the STRIDE method used by Microsoft, and rules of thumb for validating diagrams.

Readings

|  |  |
| --- | --- |
| **[Allen 2008]** Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers.*Boston, MA: Addison-Wesley, 2008. | Read 5 |
| **[Ingalsbe 2008a]** Ingalsbe, Jeffrey A.; Mead, Nancy R. “Threat Modeling: Diving into the Deep End.” *IEEE Software* (January/February 2008): 2-8. |  |
| **[Ingalsbe 2008b]** Ingalsbe, Jeffrey A.; Shoemaker, Dan; Mead, Nancy R.; & Drommi, Antonio. “Threat Modeling the Enterprise.” *Proceedings of the 14th Americas Conference on Information Systems*, Toronto, Canada, August 14-17, 2008. Atlanta: AIS, 2008. |  |
| **[OWASPb]** OWASP Material on Threat Modeling. <https://www.owasp.org/index.php/Threat_Risk_Modeling> |  |
| **[Microsoft 2009]** “IT Infrastructure Threat Modeling Guide” Release 1.0. See “IT Infrastructure Threat Modeling Guide.docx” and its associated “IT Infrastructure Threat Modeling Guide.pptx” materials. There is also an “Introduction to Threat Modeling Lab.” (See “Threat\_Modeling\_Lab\_01 .90.docx” materials.) The Lab document contains an exercise to test students’ knowledge. |  |

Additional Resources

|  |  |
| --- | --- |
| This lecture includes a threat modeling video. Go to <http://www.microsoft.com/security/sdl/video/default.aspx> . Scroll down to the SDL videos section and go to page 3 of the 5 pages. There is a 10 minute SDL Threat Modeling Tool video, dated 12/7/2010. |  |

Threat Modeling Learning Resources

|  |  |
| --- | --- |
| **MSDN Magazine**  “[Reinvigorate your Threat Modeling Process](http://msdn.microsoft.com/en-us/magazine/cc700352.aspx)” <http://msdn.microsoft.com/en-us/magazine/cc700352.aspx>  “[Threat Modeling: Uncover Security Design Flaws Using The STRIDE Approach](http://msdn.microsoft.com/msdnmag/issues/06/11/ThreatModeling/default.aspx)” <http://msdn.microsoft.com/en-us/magazine/cc163519.aspx> |  |
| **Article**  “[Experiences Threat Modeling at Microsoft](http://download.microsoft.com/download/9/D/3/9D389274-F770-4737-9F1A-8EA2720EE92A/Shostack-ModSec08-Experiences-Threat-Modeling-At-Microsoft.pdf)” PDF available on <http://blogs.msdn.com/b/sdl/archive/2008/10/08/experiences-threat-modeling-at-microsoft.aspx> |  |
| **SDL Blog**  [All threat modeling posts](http://blogs.msdn.com/sdl/archive/tags/threat%20modeling/default.aspx): <http://blogs.msdn.com/b/sdl/archive/tags/threat%20modeling/default.aspx> |  |
| **Books**  *The Security Development Lifecycle (SDL): A Process for Developing Demonstrably More Secure Software* (Howard, Lipner, 2006) “Threat Modeling” Chapter. |  |

SDL Resources

|  |  |
| --- | --- |
| **Security Development Lifecycle Portal**  <http://www.microsoft.com/sdl> |  |
| **Security Development Lifecycle Blog**  <http://blogs.msdn.com/sdl/> |  |
| **Security Development Lifecycle Process on MSDN (Web)**  <http://msdn.microsoft.com/en-us/library/cc307748.aspx> |  |
| **Security Development Lifecycle Process on MSDN (MS Word)**  <http://www.microsoft.com/downloads/details.aspx?FamilyID=d045a05a-c1fc-48c3-b4d5-b20353f97122&displaylang=en> |  |

Objectives

* Learn how to do threat modeling.
* Assess whether threat modeling provides new information

Note to Instructor

After the “Questions” slide separator (in the Powerpoint slides for the CERT Module 9 Threat Modeling Lecture), information for an exercise and suggested exercise answers are provided in the slides and notes.

Case Study Assignment Four

1. [90%] Develop a threat model for your case study project. You may optionally download and use the Microsoft tool to support this activity.
2. [10%] Did the threat model provide any new insights beyond the work you had already done?

This is a TEAM exercise.

Turn this assignment in BEFORE the next class.

# Week 10

Threat Modeling and Secure Tropos

The lecture starts with an industry case study exploring the use of threat modeling at Ford Motor Company, followed by a discussion of using threat modeling to assist in prioritization of security requirements. These topics are followed by a discussion of Secure Tropos, a self-contained life-cycle approach. It is very specific in terms of how to go about specifying requirements.

Readings

|  |  |
| --- | --- |
| **[Mead 2009a]** Mead, Nancy R.; Shoemaker, Dan; Ingalsbe, Jeffrey A. “Ensuring Cost- Efficient and Secure Software through Student Case Studies in Risk and Requirements Prioritization,” 1-8. *Proceedings of the 42nd Hawaii International Conference on Systems Science,* Big Island, Hawaii, January 5-8, 2009. New York: IEEE Computer Society Press, 2009. |  |
| **[Mead 2009b]** Mead, Nancy R.; Shoemaker, Dan; & Ingalsbe, Jeffrey A. “Software Assurance Practice at Ford: A Case Study.” *CrossTalk: The Journal of Defense Software Engineering* (March/April 2009), pp 4-7. <http://www.crosstalkonline.org/storage/issue-archives/2009/200903/200903-Mead.pdf> |  |

Additional Resources

|  |  |
| --- | --- |
| **[Mead 2005]** Mead, Nancy R.; Hough, Eric; & Stehney II, Ted. *Security Quality Requirements Engineering* (CMU/SEI-2005-TR-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. [www.sei.cmu.edu/pub/documents/05.reports/pdf/05tr009.pdf](http://www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm) |  |

Objectives

* Evaluate risk assessment methods.
* Determine whether risk assessment influences priorities of security requirements.

Homework Assignment Three

1. [45%] You have seen a number of risk assessment methods, some specialized for security and some not. Which of these do you think works the best and why? Consider at least three different methods in your assessment.
2. [45%] Using your selected method from Part 1, revisit SQUARE Step 8 (prioritization) for your case study project. Does the risk assessment cause you to change your priorities? What are the changes and the rationale?
3. [10%] When do you think your selected method should be applied? As part of planning? Requirements engineering? Architecture?

Turn this assignment in BEFORE the next class.

# Week 11

Quality Attribute Workshop and Attack Surface

Quality Attribute Workshop (QAW)

This lecture introduces QAW, a facilitated method that engages system stakeholders early in the lifecycle to discover the driving quality attribute requirements of a software-reliant system. One of the key points about the QAW is that it is scenario based.

Attack Surface

In this lecture we continue our journey through Microsoft with a look at Attack Surface. Initially developed by Jeanette Wing Attack Surface is more detailed and to some extent could be dependent on ideas of an actual implementation. It is used on software that is considered to be highly subject to vulnerabilities and is a very focused technique. We will also look at measurement techniques, how they can be extended to include security, and how to conduct an inspection with security in mind.

Readings

|  |  |
| --- | --- |
| **[Allen 2008]** Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers*. Boston, MA: Addison-Wesley, 2008. | Read 6 |
| **[Mead 2012]** Mead, Nancy R. “Measuring the Software Security Requirements Engineering Process,”583-588. *Proceedings of the Computer Software and Applications Conference Workshops,* Izmir, Turkey, July 16-20, 2012. New York: IEEE Computer Society Press, 2012. |  |
| **[Manadhata 2011]** Manadhata, Pratyusa K. & Wing, Jeannette M. “An Attack Surface Metric.”*IEEE Transactions on Software Engineering 37*, 3 (May/June 2011): 371-386. (A list of papers and talks on attack surface measurement is also available at <http://www.cs.cmu.edu/~pratyus/as.html>) |  |

Additional Resources

|  |  |
| --- | --- |
| QAW-Related  **[Barbacci 2003]** Barbacci, Mario R., et al. *Quality Attribute Workshops (QAWs), Third Edition* (CMU/SEI-2003-TR-016). Software Engineering Institute, Carnegie Mellon University, 2003. http://resources.sei.cmu.edu/asset\_files/TechnicalReport/2003\_005\_001\_14249.pdf |  |
| **[Kazman 2005]** Kazman, Rick; & Len Bass. *Categorizing Business Goals for Software Architectures* (CMU/SEI-2005-TR-021). Software Engineering Institute, Carnegie Mellon University, 2005. http://resources.sei.cmu.edu/asset\_files/TechnicalReport/2005\_005\_001\_14621.pdf |  |
| **[Bass 2013]**  Bass, Len; Clements, Paul; & Kazman, Rick. Software Architecture in Practice, Third Edition. Addison-Wesley SEI Series in Software Engineering, Pearson Education, Inc.: Upper Saddle River, N.J., 2013. <http://www.informit.com/store/software-architecture-in-practice-9780321815736> |  |
| **Attack Surface-Related** Microsoft Webcasts (including Attack Surface) <http://msdn.microsoft.com/en-us/security/aa570424.aspx> |  |
| Attack Surface Tool Download: <http://blogs.msdn.com/b/sdl/archive/2013/08/02/attack-surface-analyzer-1-0-released.aspx> |  |

Objectives

* Describe the Quality Attribute Workshop.
* Learn to do attack surface analysis.
* Learn how to reduce the attack surface.

Additional Resources for QAW Scenarios and an In-Class Exercise

The QAW scenario is a vehicle for communication among stakeholders; the more specific the scenario, the faster it will reveal hidden assumptions, misunderstandings about requirements, conflicting requirements, etc. Watch for vague words or phrases (i.e. “affordable”). What does “affordable” mean?

Additional Instructor Materials

Description of the six component parts of the six-part scenario refinement is given in “Components of six-part scenario.docx.”

Example business goal categories are given in “Business goal categories.docx.”

Example architectural drivers are given in “Architectural Driver examples.docx.”

An example of an expanded six part scenario refinement (with business goals, questions, issues, etc.) is given in “Example Expanded QAW 6-part Scenario Refinement.docx.”

In “Security General Scenario.pdf,” the Security General Scenario (Table 9.1, page 150) from the Bass, et al *Software Architecture in Practice, Third Edition* is reprinted by permission of Pearson Education, Inc. See the Bibliography for the full citation of this publication.

Templates

“QAW 3-part unrefined Scenario template.docx”

“QAW 6-part Scenario Refinement template.docx”

“Expanded QAW 6-part Scenario Refinement template.docx”

More Information on General Six-Part Scenario Refinements

See Bass, Len; Clements, Paul; & Kazman, Rick. *Software Architecture in Practice, Third Edition*. Addison-Wesley SEI Series in Software Engineering, Pearson Education, Inc.: Upper Saddle River, N.J., 2013. There are seven general scenarios, with a concrete example of each as well as additional information:

Availability General Scenario, Table 5.3, page 86

General Interoperability Scenario – Table 6.2, page 108

Modifiability General Scenario – Table 7.1, page 120

Performance General Scenario – Table 8.1, page 134

Security General Scenario – Table 9.1, page 150

Testability General Scenario – Table 10.1, page 163

Usability General Scenario – Table 11.1, page 176

Suggestions for an In-Class QAW Exercise

Students can work individually or in groups [15 minutes].

Example system: Automated Teller Machine Cash Withdrawal

Task

Generate three-part scenarios (stimulus, environment, response) that describe a system interaction with respect to some quality attribute.

Exit Criteria

Generate at least four scenarios, and for each scenario indicate if it is a use case, growth, or exploratory scenario.

--

Report on Exercise [15 minutes]

Homework Assignment Four

1. [40%] Download the Microsoft Attack Surface tool and follow the download instructions to model the attack surface on your computer. If you are unable to perform the download and execution, do as much as possible by hand. Either way, include the results.
2. [20%] What did you learn as a result of the attack surface calculation?
3. [20%] Will you make changes to your configuration OR to a development project on the basis of the attack surface video and your analysis? What are they?

Turn this assignment in BEFORE the next class.

# Week 12

Supply Chain Risk Management

This lecture covers an overview of three areas: (1) understanding Information and Communications Technology (ICT) Supply Chain Risk Management (SCRM), (2) recognizing supply chain hardware and software threats/enablers, and (3) understanding the range of countermeasures within the system development lifecycle.

Readings

|  |  |
| --- | --- |
| None |  |
|  |  |

Additional Resources

|  |  |
| --- | --- |
| **[NIST]** NIST: Supply Chain Risk Management Practices for Federal Information Systems and Organizations, <http://csrc.nist.gov/publications/drafts/800-161/sp800_161_draft.pdf> |  |
| **[OTIPS]** Open Trusted Technology Provider Standard (O-TTPS), “Mitigating Tainted and Counterfeit Products”: https://www2.opengroup.org/ogsys/catalog/C139 |  |
| **[SafeCode]** SafeCode. “Supply chain integrity framework”: <http://www.safecode.org/> |  |
| **[Alberts 2011]** Alberts, Christopher; Dorofee, Audrey J.; Creel, Rita; Ellison, Robert J.; & Woody, Carol. “A Systematic Approach for Assessing Software Supply- Chain Risk,” 1-8. *Proceedings of the 44th Hawaii International Conference on Systems Science*, Kauai, Hawaii, January 4-7, 2011. New York: IEEE Computer Society Press, 2011. |  |

Objectives

* Obtain a basic understanding of supply chain risk management.

# Week 13

SQUARE for Acquisition

This session begins with a definition of acquisition, a brief background of current efforts in software acquisition, a recap of the nine-step SQUARE process, and an introduction to SQUARE for acquisition (A-SQUARE) through three cases. There has been much less work in the acquisition area than in the development area. While research has lagged in this area, it is nonetheless an interesting way of looking at things.

Readings

|  |  |
| --- | --- |
| **[Mead 2010]** Mead, Nancy R. *Adapting the SQUARE Method for Security Requirements Engineering to Acquisition*. White Paper. <https://resources.sei.cmu.edu/asset_files/WhitePaper/2010_019_001_51613.pdf>  **[Mani 2014]**  Mani, Sidhartha; & Mead, Nancy. *An Evaluation of A-SQUARE for COTS Acquisition* (CMU/SEI-2014-TN-003). Software Engineering Institute, Carnegie Mellon University, 2014. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=90536 |  |

Objectives

* Experience security requirements engineering as part of the acquisition process.

Additional Resources for Exercises

SQUARE for Acquisition: Case Study 1

SQUARE for Acquisition: Case Study 2

SQUARE for Acquisition: Case Study 3

These three sets of materials will be used by the students to conduct the in-class A-SQUARE exercises. Case Study 3 is also used in the Case Study Assignment. Note these materials also include an extensive reference list.

Case Study Assignment Five

Using the SQUARE for Acquisition white paper and lecture materials as a guide, apply SQUARE for Acquisition Case 3 (acquisition of COTS software) to your project. You may reuse material from Case Study Assignment 2, such as Steps 1 and 2.

The intent of the exercise is for you to experience security requirements engineering as part of the acquisition process.

This is a TEAM exercise.

Turn this assignment in BEFORE the next class.

[See also the Case Study Final Report requirement.]

Case Study Assignment Final Report

Final Case Study Report

Collate all of the case study assignments into a single report; include a project introduction and all of the assignments following it. Feel free to make changes if you want.

This is a TEAM exercise.

Turn this assignment in BEFORE the next class.

# Week 14

Student Presentations

This is the final week of the course. Students present the results of their case studies.

Bibliography

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

[Alberts 2012a]

Alberts, Christopher; Allen, Julia; & Stoddard, Robert. *Risk-Based Measurement and Analysis: Application to Software Security* (CMU/SEI-2012-TN-004), Software Engineering Institute, Carnegie Mellon University, February 2012. <http://www.sei.cmu.edu/reports/12tn004.pdf>

[Alberts 2012b]

Alberts, Christopher & Dorofee, Audrey. *Mission Risk Diagnostic (MRD) Method Description* (CMU/SEI-2012-TN-005). Software Engineering Institute, Carnegie Mellon University, 2012.   
<http://www.sei.cmu.edu/reports/12tn005.pdf>

[Alberts 2011]

Alberts, Christopher; Dorofee, Audrey J.; Creel, Rita; Ellison, Robert J.; & Woody, Carol. “A Systematic Approach for Assessing Software Supply- Chain Risk,” 1-8. *Proceedings of the 44th Hawaii International Conference on Systems Science*, Kauai, Hawaii, January 4-7, 2011. New York: IEEE Computer Society Press, 2011.

[Alberts 2009]

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

**[Allen 2008]**

Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers*. Boston, MA: Addison-Wesley Professional, 2008.

[Barbacci 2003]

Barbacci, Mario R., et al. *Quality Attribute Workshops (QAWs), Third Edition* (CMU/SEI-2003-TR-016). Software Engineering Institute, Carnegie Mellon University, 2003. <http://resources.sei.cmu.edu/asset_files/TechnicalReport/2003_005_001_14249.pdf>

[Bass 2013]

Bass, Len; Clements, Paul; & Kazman, Rick. *Software Architecture in Practice*, *Third Edition*. Addison-Wesley SEI Series in Software Engineering, Pearson Education, Inc.: Upper Saddle River, N.J., 2013. <http://www.informit.com/store/software-architecture-in-practice-9780321815736>

[Beckers 2012]

Beckers, Kristian; Faßbender, Stephen; Heisel, Maritta, Küster, Jan-Christoph; & Schmidt, Holger. “Supporting the Development and Documentation of ISO 27001 Information Security Management Systems through Security Requirements Engineering Approaches,” 14-21. *Engineering Secure Software and Systems 2012*, Lecture Notes in Computer Science 7159. Berlin: Spring-Verlag, 2012. <http://link.springer.com/chapter/10.1007/978-3-642-28166-2_2>

[BSIMM]

Building Security in Maturity Model[. http://bsimm.com/download/](http://bsimm.com/download/) (no registration required).

[Build Security In]

Build Security In. <https://buildsecurityin.us-cert.gov/>. Requirements Engineering articles:

https://buildsecurityin.us-cert.gov/articles/best-practices/requirements-engineering

[CERT CSE]

Cyber Security Engineering (CSE) Team Web Page. <http://www.cert.org/cybersecurity-engineering/index.cfm>

[Davis 2013]

Davis, Noopur. “Secure Software Development Life Cycle Processes,” Build Security In. [https://buildsecurityin.us-cert.gov/articles/knowledge/sdlc-process/secure-software-development-life-cycle-processes, July 2013](https://buildsecurityin.us-cert.gov/articles/knowledge/sdlc-process/secure-software-development-life-cycle-processes,%20July%202013).

[IEEE 2011]

Institute of Electrical and Electronics Engineering. “Malware Infects US Military Drone System.” *Computer 44*, 11 (November 2011): 16.

[Ingalsbe 2008a]

Ingalsbe, Jeffrey A.; Mead, Nancy R. “Threat Modeling: Diving into the Deep End.” *IEEE Software* (January/February 2008): 2-8.

[Ingalsbe 2008b]

Ingalsbe, Jeffrey A.; Shoemaker, Dan; Mead, Nancy R.; & Drommi, Antonio. “Threat Modeling the Enterprise.” *Proceedings of the 14th Americas Conference on Information Systems*, Toronto, Canada, August 14-17, 2008. Atlanta: AIS, 2008.

[Kazman 2005]

Kazman, Rick; & Len Bass. *Categorizing Business Goals for Software Architectures* (CMU/SEI-2005-TR-021). Software Engineering Institute, Carnegie Mellon University, 2005. <http://resources.sei.cmu.edu/asset_files/TechnicalReport/2005_005_001_14621.pdf>

[Khan 2009]

Khan, Muhammad Umair Ahmed & Zulkernine, Mohammed. “On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software,” pages 353-358. *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference*. Seattle, WA, July 20-24, 2009. New York: IEEE Computer Society Press, 2009. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5254051>. If no access to IEEE Xplore digital library, doi: 10.1109/COMPSAC.2009.206.

[Lipner 2013]

Lipner, Steve. “The Security Development Lifecycle.” Three-part video. <http://www.cert.org/curricula/sae-course-materials-by-type.cfm> (Read and accept the click-thru license agreement then scroll down to “The Security Development Lifecycle”)

[Lipner 2005]

Lipner, Steve & Howard, Michael. *The Trustworthy Computing Security Development Lifecycle*. <http://msdn.microsoft.com/> en-us/library/ms995349.aspx (2005).

[Manadhata 2011]

Manadhata, Pratyusa K. & Wing, Jeannette M. “An Attack Surface Metric.”*IEEE Transactions on Software Engineering 37*, 3 (May/June 2011): 371-386. A list of papers and talks on attack surface measurement is also available at <http://www.cs.cmu.edu/~pratyus/as.html>

[Mani 2014]

Mani, Sidhartha; & Mead, Nancy. *An Evaluation of A-SQUARE for COTS Acquisition* (CMU/SEI-2014-TN-003). Software Engineering Institute, Carnegie Mellon University, 2014. <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=90536>

[Mead]

Mead, Nancy R. *Adapting the SQUARE Method for Security Requirements Engineering to Acquisition*. White Paper. <https://resources.sei.cmu.edu/asset_files/WhitePaper/2010_019_001_51613.pdf>

[Mead 2012]

Mead, Nancy R. “Measuring the Software Security Requirements Engineering Process**,”** 583-588. *Proceedings of the Computer Software and Applications Conference Workshops,* Izmir, Turkey, July 16-20, 2012. New York: IEEE Computer Society Press, 2012.

[Mead 2010]

Mead, Nancy R.; Allen, Julia H.; Ardis, Mark; Hilburn, Thomas B.; Kornecki, Andrew J.; Linger,

Rick; & McDonald, James. *Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum* (CMU/SEI-2010-TR-005). Software Engineering Institute, Carnegie Mellon University, 2010. <http://www.sei.cmu.edu/library/abstracts/reports/10tr005.cfm>

[Mead 2009a]

Mead, Nancy R.; Shoemaker, Dan; Ingalsbe, Jeffrey A. “Ensuring Cost- Efficient and Secure Software through Student Case Studies in Risk and Requirements Prioritization,” 1-8. *Proceedings of the 42nd Hawaii International Conference on Systems Science,* Big Island, Hawaii, January 5-8, 2009. New York: IEEE Computer Society Press, 2009.

[Mead 2009b]

Mead, Nancy R.; Shoemaker, Dan; & Ingalsbe, Jeffrey A. “Software Assurance Practice at Ford: A Case Study.”*CrossTalk: The Journal of Defense Software Engineering*. (March/April 2009), pp. 4-7. <http://www.crosstalkonline.org/storage/issue-archives/2009/200903/200903-Mead.pdf>

[Mead 2005]

Mead, Nancy R.; Hough, Eric; & Stehney II, Ted. *Security Quality Requirements Engineering* (CMU/SEI-2005-TR-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. [www.sei.cmu.edu/pub/documents/05.reports/pdf/05tr009.pdf](http://www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm)

[Microsoft 2009]

[Microsoft 2009] *IT Infrastructure Threat Modeling Guide* Release 1.0.

[Montalbano 2011]

Montalbano, Elizabeth. *Air Force Says Drone Virus is No Threat: an attack on the network that controls U.S. military unmanned aerial vehicles was only a ‘nuisance’ military arm claims*. [http://www.informationweek.com/government/security/air-force-says- drone-virus-is-no-threat/231900741](http://www.informationweek.com/government/security/air-force-says-%20drone-virus-is-no-threat/231900741) (2011).

[MSCE]

Software Engineering Institute (SEI) Mission Success in Complex Environments (MCSE) Special Project. <http://www.sei.cmu.edu/risk>

[OWASPa]

Open Web Application Security Project[. https://www.owasp.org/](file:///C:/Users/cas/Desktop/June%202009%20NEED%20TO%20FILE/140505%20xfer%20to%20laptop/.%20https:/www.owasp.org/)

[OWASPb]

OWASP Material on Threat Modeling. <https://www.owasp.org/index.php/Threat_Risk_Modeling>

[Saltzer 74]

Saltzer, Jerome H. & Schroeder, Michael D. **“**The Protection of Information in Computer Systems."*Communications of the ACM 17*, 7 (July 1974): 388-402.

[Woody 2012]

Woody, Carol; Mead, Nancy; & Shoemaker, Dan. “Foundations for Software Assurance,” 5368-5374. *Proceedings of the 45th Hawaii International Conference on Systems Science,* Maui, Hawaii, January 4-7, 2012. New York: IEEE Computer Society Press, 2012.