Software Assurance Curriculum Master
Bibliography and Course References

Nancy R. Mead, Software Engineering Institute
Julia H. Allen, Software Engineering Institute
Mark Ardis, Stevens Institute of Technology
Thomas B. Hilburn, Embry-Riddle Aeronautical University
Andrew J. Kornecki, Embry-Riddle Aeronautical University
Richard Linger, Software Engineering Institute

September 2010; Revised June 2011
# Table of Contents

Section 1: Software Assurance Curriculum Master Bibliography .................. 2

Section 2: Master of Software Assurance Course Outlines References .......... 32

- **Assurance Assessment (3.1, 3.2, 3.3, 6.4)** .................................................. 32
- **Assurance Management (2.1, 2.2, 2.3, 4.1, 4.2, 4.3)** ............................... 35
- **Assured Software Analytics (6.3, 6.4)** ....................................................... 37
- **Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])** .......................................................... 38
- **Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design])** ............................................................... 41
- **Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation]** ............................................................... 43
- **System Operational Assurance (7.1, 7.2, 7.3)** .............................................. 46
- **System Security Assurance (5.1, 5.2, 5.3)** ................................................. 48
Section 1: Software Assurance Curriculum Master Bibliography

This master bibliography was compiled from *Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum* and *Software Assurance Curriculum Project Volume II: Undergraduate Course Outlines*. It also includes references from the *Master of Software Assurance Course Outlines*,¹ which are included in their entirety as Section 2 of this document. URLs are valid as of August 2010.

This bibliography was expanded in June 2011 to include references from *Software Assurance Curriculum Project Volume III: Master of Software Assurance Course Syllabi* and *Software Assurance Curriculum Project IV: Community College Education*.


[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements]), Software Assurance Curriculum Project Volume III References*]

[Abstract from publisher:
CMMI® (Capability Maturity Model® Integration) is an integrated, extensible framework for improving process capability and quality across an organization. It has become a cornerstone in the implementation of continuous improvement for both industry and governments around the world. Rich in both detail and guidance for a wide set of organizational domains, the CMMI Product Suite continues to evolve and expand.

Updated for CMMI Version 1.2, this third edition of *CMMI® Distilled* again provides a concise and readable introduction to the model, as well as straightforward, no-nonsense information on integrated, continuous process improvement. The book now also includes practical advice on how to use CMMI in tandem with other approaches, including Six Sigma and Lean, as well as new and expanded guidance on preparing for, managing, and using appraisals.

Written so that readers unfamiliar with model-based process improvement will understand how to get started with CMMI, the book offers insights for those more

¹ This document is available for download at http://www.cert.org/mswa.
experienced as well. It can help battle-scarred process improvement veterans, and experienced suppliers and acquirers of both systems and services, perform more effectively. CMMI® Distilled is especially appropriate for executives and managers who need to understand why continuous improvement is valuable, why CMMI is a tool of choice, and how to maximize the return on their efforts and investments. Engineers of all kinds (systems, hardware, software, and quality, as well as acquisition personnel and service providers) will find ideas on how to perform better.

The three authors, all involved with CMMI since its inception, bring a wealth of experience and knowledge to this book. They highlight the pitfalls and shortcuts that are all too often learned by costly experience, and they provide a context for understanding why the use of CMMI continues to grow around the world.


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 Engineering* draws 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


[Source: *Master of Software Assurance Course Outline References – Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design]), Software Assurance Curriculum Project Volume III References*]


[Source: *Master of Software Assurance Course Outline References – System Security Assurance (5.1, 5.2, 5.3), Software Assurance Curriculum Project Volume III References*]

[Abstract from publisher:
The world has changed radically since the first edition was published in 2001. Spammers, virus writers, phishermen, money launderers, and spies now trade busily with each other in a lively online criminal economy—and as they specialize, they get better. New applications, from search to social networks to electronic voting machines, provide new targets. And terrorism has changed the world. In this indispensable, fully updated guide, Ross Anderson reveals how to build systems that stay dependable whether faced with error or malice.

Here’s straight talk about
• Technical engineering basics—cryptography, protocols, access controls, and distributed systems
• Types of attack—phishing, Web exploits, card fraud, hardware hacks, and electronic warfare
• Specialized protection mechanism—what biometrics, seals, smartcards, alarms, and DRM do, and how they fail
• Security economics—why companies build insecure systems, why it’s tough to manage security projects, and how to cope
• Security psychology—the privacy dilemma, what makes security too hard to use, and why deception will keep increasing
• Policy—why governments waste money on security, why societies are vulnerable to terrorism, and what to do about it]

[Source: Software Assurance Curriculum Project Volume IV References]

[Source: Software Assurance Curriculum Project Volumes I and II References]

[Source: Software Assurance Curriculum Project Volume IV References]

[Source: Software Assurance Curriculum Project Volume IV References]

[Source: Software Assurance Curriculum Project Volume IV References]

[Source: Software Assurance Curriculum Project Volume IV References]

[Source: Software Assurance Curriculum Project Volume IV References]

[Source: Software Assurance Curriculum Project Volume IV References]

[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Master of Software Assurance Course Outline References – System Security Assurance (5.1, 5.2, 5.3), Software Assurance Curriculum Project Volume I and III References*]


[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Software Assurance Curriculum Project Volumes I and II References*]


[Source: *Software Assurance Curriculum Project Volumes I and II References*]


[Source: *Software Assurance Curriculum Project Volumes I and II References*]

Association for Computing Machinery (ACM) Inc. ACM Code of Ethics and Professional Conduct. [Source: Software Assurance Curriculum Project Volume IV References]


[Source: Software Assurance Curriculum Project Volume IV References]

[Source: Software Assurance Curriculum Project Volume IV References]


[Source: Software Assurance Curriculum Project Volume II References]

[Source: Software Assurance Curriculum Project Volume I References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volumes I, II, and IV References]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements]). Software Assurance Curriculum Project Volume III References]

[Source: Master of Software Assurance Course Outline References – Assurance Management (2.1, 2.2, 2.3, 4.1, 4.2, 4.3)]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: *Master of Software Assurance Course Outline References – System Operational Assurance (7.1, 7.2, 7.3)*]


[Source: *Master of Software Assurance Course Outline References – System Operational Assurance (7.1, 7.2, 7.3)*]

CERT. *The CERT Oracle Secure Coding Standard for Java.*

[Source: *Software Assurance Curriculum Project Volumes I and II References*]

CERT. *CERT® Resilience Management Model* (reports and presentations).

[Source: *Master of Software Assurance Course Outline References – System Operational Assurance (7.1, 7.2, 7.3), Software Assurance Curriculum Project Volume III References*]


[Source: *Master of Software Assurance Course Outline References – System Operational Assurance (7.1, 7.2, 7.3)*]


[Source: *Master of Software Assurance Course Outline References – System Operational Assurance (7.1, 7.2, 7.3)*]

[CERT’s *Information Security for Technical Staff* courseware (introductory, advanced)

- Introductory: This five-day course is designed to provide participants with practical techniques for protecting the security of an organization's information assets and resources, beginning with concepts and proceeding on to technical implementations. The courses focus on understanding and applying the concept of survivability through the effective management of risk, threats, policy, system configuration, availability, and personnel. The course also addresses incident response and provides a technical foundation for working with TCP/IP security and cryptography. The final section of the course helps participants learn to design secure network architecture managing host systems, securing network services, and infrastructure, working with firewalls, and understanding intrusion detection and prevention.

- Advanced: This four-day course is designed to increase the depth of knowledge and skills of technical staff charged with administering and securing information systems and networks.]


Certifying the protection of IT assets is critical to achieving the goal of Defense-in-Depth. Students are introduced to the CERT Defense-in-Depth Framework: eight operationally focused and inter-dependent management components which will be synergistically applied to a fictitious organization's IT enterprise. Through lectures, demonstrations, scenario-based exercises, small group activities, and open discussions, students will learn high-level best practices for effectively integrating each of these eight components into all aspects of IT operations. Further, the course scenario is used extensively to reinforce these best practices with technical information security implementations.


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volumes I and II References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Software Assurance Curriculum Project Volumes I and II References*]

CyberWatch. *College Curriculum.*


Department of Homeland Security (DHS) Software Assurance (SwA). *Build Security In.*


[This textbook covers methods for reverse engineering of software.]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]

http://www.sei.cmu.edu/library/abstracts/reports/10tn016.cfm
[Source: Software Assurance Curriculum Project Volumes I and III References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volumes I and II References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volumes I and II References]

[Source: Software Assurance Curriculum Project Volume III References]

http://www.sei.cmu.edu/library/abstracts/reports/91tr002.cfm
[Source: Software Assurance Curriculum Project Volume I References]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements]), Software Assurance Curriculum Project Volume III References]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Software Assurance Curriculum Project Volume III References]
[A nice introduction to buffer overflows and stack-smashing using animation and student exercises.]


[Source: Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]

Goertzel, Karen Mercedes; Winograd, Theodore; McKinley, Holly Lynne; Oh, Lyndon; Colon, Michael; McGibbon, Thomas; Fedchak, Elaine; & Viennuea. Software Security Assurance: Stateof-the-Art Report (SOAR). Information Assurance Technology Analysis Center (IATAC) & Data and Analysis Center for Software (DACS), 2007.
[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volumes I and III References]

[Source: Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]

https://www.vte.cert.org/vteweb/go/2699.aspx
[Source: Master of Software Assurance Course Outline References – Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Master of Software Assurance Course Outline References – Assured Software Analytics (6.3, 6.4), Software Assurance Curriculum Project Volume III References*]

This textbook includes wireless network security analysis and methods. It could be used as a reference that provides a picture of assurance issues in the pervasive wireless networks that support software and service operations across organizations and that are themselves software enabled.


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Assurance Management (2.1, 2.2, 2.3, 4.1, 4.2, 4.3), Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements], Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design]), Software Assurance Curriculum Project Volume II and III References*]

[Abstract from publisher:

Your in-depth, expert guide to the proven process that helps reduce security bugs. Your customers demand and deserve better security and privacy in their software. This book is the first to detail a rigorous, proven methodology that measurably minimizes security bugs—the Security Development Lifecycle (SDL). In this long-awaited book, security experts Michael Howard and Steve Lipner from the Microsoft Security Engineering Team guide you through each stage of the SDL—from education and design to testing and post-release. You get their first-hand insights, best practices, a practical history of the SDL, and lessons to help you implement the SDL in any development organization.

Discover how to:

- Use a streamlined risk-analysis process to find security design issues before code is committed
- Apply secure-coding best practices and a proven testing process
- Conduct a final security review before a product ships
- Arm customers with prescriptive guidance to configure and deploy your product more securely
- Establish a plan to respond to new security vulnerabilities
- Integrate security discipline into agile methods and processes, such as Extreme Programming and Scrum]
Includes a CD featuring:

- A six-part security class video conducted by the authors and other Microsoft security experts

Sample SDL documents and fuzz testing tool][Hughes, Katherine. “CTE Dual Enrollment: Preparing Students for College and Careers.” Presented at the CA Community College Association for Occupational Education Conference, 2011.


[Source: Software Assurance Curriculum Project Volume IV References]


[Source: Software Assurance Curriculum Project Volumes I and II References]

IBM. IBM Point of View: Security and Cloud Computing.


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volumes I and II References]


[Source: Software Assurance Curriculum Project Volume I References]


[Source: Software Assurance Curriculum Project Volume I References]


[Source: Software Assurance Curriculum Project Volume I References]


[Source: Software Assurance Curriculum Project Volumes I and II References]


[Source: Software Assurance Curriculum Project Volume IV References]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Master of Software Assurance Course Outline References – Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements]), Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume I References]


[Source: Master of Software Assurance Course Outline References – System Operational Assurance (7.1, 7.2, 7.3)]


[Source: Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Software Assurance Curriculum Project Volume III References]


[Source: Master of Software Assurance Course Outline References – Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design]), Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume I References]


[Source: Master of Software Assurance Course Outline References – Assured Software Analytics (6.3, 6.4), Software Assurance Curriculum Project Volume III References]

[This textbook is out of print but contains material on rigorous methods for structuring and reverse engineering of software to verify functionality. It is intended for faculty use only.]


[Source: Master of Software Assurance Course Outline References – Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design])]}


[Source: Master of Software Assurance Course Outline References – Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design])]}


[Source: Software Assurance Curriculum Project Volume III References]

McGarry, John; Card, David; Jones, Cheryl; Layman, Beth; Clark, Elizabeth; Dean, Joseph; & Hall, Fred. Practical Software Measurement: Objectives for Decision Makers. Addison-Wesley Professional, 2001. [Source: Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4)]


[Abstract from publisher: The software security best practices, or touchpoints, described in this book have their basis in good software engineering and involve explicitly pondering security throughout the software development lifecycle. This means knowing and understanding common risks (including implementation bugs and architectural flaws), designing for security, and subjecting all software artifacts to thorough, objective risk analyses and testing. Software Security is about putting the touchpoints to work for you. Because you can apply these touchpoints to the software artifacts you already produce as you develop software, you can adopt this book’s methods without radically changing the way you work. Inside you’ll find detailed explanations of
• Risk management frameworks and processes
• Code review using static analysis tools
• Architectural risk analysis
• Penetration testing
• Security testing
• Abuse case development

In addition to the touchpoints, Software Security covers knowledge management, training and awareness, and enterprise-level software security programs.]


[Source: *Master of Software Assurance Course Outline References – Assured Software Development I - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]*

[Source: *Software Assurance Curriculum Project Volume III References]*

[Source: *Master of Software Assurance Course Outline References – Assured Software Development I - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]*

[Source: *Master of Software Assurance Course Outline References – Assurance Management (2.1, 2.2, 2.3, 4.1, 4.2, 4.3), Software Assurance Curriculum Project Volume I, II, and III References]*

[Source: *Software Assurance Curriculum Project Volume II, and III References]*

[Not directly cited but part of the Software Assurance Curriculum Project report series.]

[Source: *Software Assurance Curriculum Project Volume IV References]*

[Source: *Master of Software Assurance Course Outline References – System Security Assurance (5.1, 5.2, 5.3)]*

[Source: *Software Assurance Curriculum Project Volume III References]*

[Source: Software Assurance Curriculum Project Volume III References]


[Source: Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Software Assurance Curriculum Project Volume III References]

[Abstract from publisher:
Although many software books highlight open problems in secure software development, few provide easily actionable, ground-level solutions. Breaking the mold, Secure and Resilient Software Development teaches you how to apply best practices and standards for consistent and secure software development. It details specific quality software development strategies and practices that stress resilience requirements with precise, actionable, and ground-level inputs.

Providing comprehensive coverage, the book illustrates all phases of the secure software development life cycle. It shows developers how to master non-functional requirements including reliability, security, and resilience. The authors provide expert-level guidance through all phases of the process and supply many best practices, principles, testing practices, and design methodologies.]


[Source: Software Assurance Curriculum Project Volume I References]

[The SDL addresses, in separate sections, practices for product software vs. practices for internal line of business (LOB) software. Only those for product software were reviewed.]


[Source: Master of Software Assurance Course Outline References – Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Software Assurance Curriculum Project Volume III References]


[Source: Master of Software Assurance Course Outline References – Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design]), Software Assurance Curriculum Project Volume III References]


Moss, Michele & Nadworny, Margaret. “Update on the Assurance for CMMI Practices.”


- [Source: Master of Software Assurance Course Outline References – System Operational Assurance (7.1, 7.2, 7.3)]


- [Source: Master of Software Assurance Course Outline References – Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design]), Software Assurance Curriculum Project Volume III References]


- [Source: Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements]), Software Assurance Curriculum Project Volume I and III References]


- [Source: Software Assurance Curriculum Project Volume I References]


- [Source: Software Assurance Curriculum Project Volume I References]


- [Source: Software Assurance Curriculum Project Volume III References]


- [Source: Software Assurance Curriculum Project Volume III References]


- [Source: Software Assurance Curriculum Project Volume I References]


- [Source: Software Assurance Curriculum Project Volume IV References]


- [Source: Software Assurance Curriculum Project Volume III References]
[Source: Software Assurance Curriculum Project Volumes I and II References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])]

http://www.smartgridinformation.info/pdf/2283_doc_1.pdf
[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volume III References]

[Source: Software Assurance Curriculum Project Volumes I, II, and III References]

[Source: Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Software Assurance Curriculum Project Volume II References]

[Abstract from publisher: Commonly exploited software vulnerabilities are usually caused by avoidable software defects. Having analyzed nearly 18,000 vulnerability reports over the past ten years, the CERT/Coordination Center (CERT/CC) has determined that a relatively small number of root causes account for most of them. This book identifies and explains these causes and shows the steps that can be taken to prevent exploitation. Moreover, this book]
encourages programmers to adopt security best practices and develop a security mindset that can help protect software from tomorrow's attacks, not just today's. Drawing on the CERT/CC's reports and conclusions, Robert Seacord systematically identifies the program errors most likely to lead to security breaches, shows how they can be exploited, reviews the potential consequences, and presents secure alternatives.

Coverage includes technical detail on how to

- Improve the overall security of any C/C++ application
- Thwart buffer overflows and stack-smashing attacks that exploit insecure string manipulation logic
- Avoid vulnerabilities and security flaws resulting from the incorrect use of dynamic memory management functions
- Eliminate integer-related problems: integer overflows, sing errors, and truncation errors
- Correctly use formatted output functions without introducing format-string vulnerabilities
- Avoid I/O vulnerabilities, including race conditions

Secure Coding in C and C++ presents hundreds of examples of secure code, insecure code, and exploits, implemented for Windows and Linux. If you're responsible for creating secure C or C++ software---or for keeping it safe---no other book offers you this much detailed, expert assistance.


[Source: Software Assurance Curriculum Project Volumes I and II References, Software Assurance Curriculum Project Volume I and III References]


[Source: Software Assurance Curriculum Project Volume III References]


[Source: Software Assurance Curriculum Project Volume I References]


[Source: Software Assurance Curriculum Project Volumes I and II References]


[Source: Software Assurance Curriculum Project Volume IV References]

[Source: *Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume IV References*]


[Source: *Master of Software Assurance Course Outline References – Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation]*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume I References*]


[Source: *Software Assurance Curriculum Project Volume III References*]


[Source: *Software Assurance Curriculum Project Volume II References*]


[Source: *Software Assurance Curriculum Project Volume I References*]


[Source: *Software Assurance Curriculum Project Volume III References*]
[Source: *Software Assurance Curriculum Project Volume III References*]

[Source: *Master of Software Assurance Course Outline References – Assurance Assessment (3.1, 3.2, 3.3, 6.4), Software Assurance Curriculum Project Volume II References*]

[Source: *Master of Software Assurance Course Outline References – Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation], Software Assurance Curriculum Project Volume III References*]

[Source: *Software Assurance Curriculum Project Volumes I and II References*]

[Source: *Software Assurance Curriculum Project Volume III References*]

[Source: *Software Assurance Curriculum Project Volume III References*]
Section 2: Master of Software Assurance Course Outlines References

This section includes the annotated references from the Master of Software Assurance Course Outline References. All references and annotations also appear in Section 1, Software Assurance Curriculum Master Bibliography.

Assurance Assessment (3.1, 3.2, 3.3, 6.4)

Primary sources:

[Abstract from publisher:
The software security best practices, or touchpoints, described in this book have their basis in good software engineering and involve explicitly pondering security throughout the software development lifecycle. This means knowing and understanding common risks (including implementation bugs and architectural flaws), designing for security, and subjecting all software artifacts to thorough, objective risk analyses and testing. Software Security is about putting the touchpoints to work for you. Because you can apply these touchpoints to the software artifacts you already produce as you develop software, you can adopt this book’s methods without radically changing the way you work. Inside you’ll find detailed explanations of
• Risk management frameworks and processes
• Code review using static analysis tools
• Architectural risk analysis
• Penetration testing
• Security testing
• Abuse case development

In addition to the touchpoints, Software Security covers knowledge management, training and awareness, and enterprise-level software security programs.]

[Abstract from publisher:
Your in-depth, expert guide to the proven process that helps reduce security bugs. Your customers demand and deserve better security and privacy in their software. This book is the first to detail a rigorous, proven methodology that measurably minimizes security bugs—the Security Development Lifecycle (SDL). In this long-awaited book, security experts Michael Howard and Steve Lipner from the Microsoft Security Engineering Team guide you through each stage of the SDL—from education and design to testing and post-release. You get their first-hand insights, best practices, a practical history of the SDL, and lessons to help you implement the SDL in any development organization.
Discover how to:

- Use a streamlined risk-analysis process to find security design issues before code is committed
- Apply secure-coding best practices and a proven testing process
- Conduct a final security review before a product ships
- Arm customers with prescriptive guidance to configure and deploy your product more securely
- Establish a plan to respond to new security vulnerabilities
- Integrate security discipline into agile methods and processes, such as Extreme Programming and Scrum

Includes a CD featuring:

- A six-part security class video conducted by the authors and other Microsoft security experts
- Sample SDL documents and fuzz testing tool

**Secondary sources:**


McGarry, John; Card, David; Jones, Cheryl; Layman, Beth; Clark, Elizabeth; Dean, Joseph; & Hall, Fred. *Practical Software Measurement: Objectives for Decision Makers*. Addison-Wesley Professional, 2001.


**Assurance Management (2.1, 2.2, 2.3, 4.1, 4.2, 4.3)**

**Primary source:**


[Abstract from 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 Engineering* draws 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]

**Secondary sources:**


**Assured Software Analytics (6.3, 6.4)**

**Primary sources:**

[This textbook covers methods for reverse engineering of software.]

[This textbook is out of print but contains material on rigorous methods for structuring and reverse engineering of software to verify functionality. It is intended for faculty use only.]

**Secondary sources:**

[This textbook describes a widely used tool that supports automated analysis of software executables. It could be used as a reference textbook for an example software analysis tool.]

[This textbook includes wireless network security analysis and methods. It could be used as a reference that provides a picture of assurance issues in the pervasive wireless networks that support software and service operations across organizations and that are themselves software enabled.]
Assured Software Development 1 - Process and Requirements (1.1, 1.2, 6.1, 6.2 [requirements])

Primary sources:


[Abstract from publisher:
CMMI® (Capability Maturity Model® Integration) is an integrated, extensible framework for improving process capability and quality across an organization. It has become a cornerstone in the implementation of continuous improvement for both industry and governments around the world. Rich in both detail and guidance for a wide set of organizational domains, the CMMI Product Suite continues to evolve and expand.

Updated for CMMI Version 1.2, this third edition of CMMI® Distilled again provides a concise and readable introduction to the model, as well as straightforward, no-nonsense information on integrated, continuous process improvement. The book now also includes practical advice on how to use CMMI in tandem with other approaches, including Six Sigma and Lean, as well as new and expanded guidance on preparing for, managing, and using appraisals.

Written so that readers unfamiliar with model-based process improvement will understand how to get started with CMMI, the book offers insights for those more experienced as well. It can help battle-scarred process improvement veterans, and experienced suppliers and acquirers of both systems and services, perform more effectively. CMMI® Distilled is especially appropriate for executives and managers who need to understand why continuous improvement is valuable, why CMMI is a tool of choice, and how to maximize the return on their efforts and investments. Engineers of all kinds (systems, hardware, software, and quality, as well as acquisition personnel and service providers) will find ideas on how to perform better.

The three authors, all involved with CMMI since its inception, bring a wealth of experience and knowledge to this book. They highlight the pitfalls and shortcuts that are all too often learned by costly experience, and they provide a context for understanding why the use of CMMI continues to grow around the world.]


[Abstract from 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 Engineering draws 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]

Secondary sources:


[Misuse & abuse cases]


OpenSAMM Project. *Software Assurance Maturity Model (SAMM) v1.0*. 

Assured Software Development 2 - Architecture and Design (6.1, 6.2 [specification, design])

Primary sources:


[Abstract from 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 Engineering draws 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]


[A comprehensive architecture framework and methodology which enables the design, evaluation and implementation of the right architecture for an enterprise.]

Secondary sources:


Assured Software Development 3 (6.2) [low-level design, code, test, verification, validation]

Primary sources:

http://www.sei.cmu.edu/library/abstracts/books/0321335724.cfm

[Abstract from publisher:  
Commonly exploited software vulnerabilities are usually caused by avoidable software defects. Having analyzed nearly 18,000 vulnerability reports over the past ten years, the CERT/Coordination Center (CERT/CC) has determined that a relatively small number of root causes account for most of them. This book identifies and explains these causes and shows the steps that can be taken to prevent exploitation. Moreover, this book encourages programmers to adopt security best practices and develop a security mindset that can help protect software from tomorrow's attacks, not just today's. Drawing on the CERT/CC's reports and conclusions, Robert Seacord systematically identifies the program errors most likely to lead to security breaches, shows how they can be exploited, reviews the potential consequences, and presents secure alternatives. Coverage includes technical detail on how to

- Improve the overall security of any C/C++ application
- Thwart buffer overflows and stack-smashing attacks that exploit insecure string manipulation logic
- Avoid vulnerabilities and security flaws resulting from the incorrect use of dynamic memory management functions
- Eliminate integer-related problems: integer overflows, sing errors, and truncation errors
- Correctly use formatted output functions without introducing format-string vulnerabilities
- Avoid I/O vulnerabilities, including race conditions

Secure Coding in C and C++ presents hundreds of examples of secure code, insecure code, and exploits, implemented for Windows and Linux. If you're responsible for creating secure C or C++ software---or for keeping it safe---no other book offers you this much detailed, expert assistance.]

[Abstract from publisher:  
Although many software books highlight open problems in secure software development, few provide easily actionable, ground-level solutions. Breaking the mold, Secure and Resilient Software Development teaches you how to apply best practices and standards for consistent and secure software development. It details specific quality software development strategies and practices that stress resilience requirements with precise, actionable, and ground-level inputs.

Providing comprehensive coverage, the book illustrates all phases of the secure software development life cycle. It shows developers how to master non-functional...
requirements including reliability, security, and resilience. The authors provide expert-level guidance through all phases of the process and supply many best practices, principles, testing practices, and design methodologies.]

**Secondary sources:**


**System Operational Assurance (7.1, 7.2, 7.3)**

**Primary sources:**


[CERT’s *Information Security for Technical Staff* courseware (introductory, advanced)

- Introductory: This five-day course is designed to provide participants with practical techniques for protecting the security of an organization's information assets and resources, beginning with concepts and proceeding on to technical implementations. The courses focus on understanding and applying the concept of survivability through the effective management of risk, threats, policy, system configuration, availability, and personnel. The course also addresses incident response and provides a technical foundation for working with TCP/IP security and cryptography. The final section of the course helps participants learn to design secure network architecture managing host systems, securing network services, and infrastructure, working with firewalls, and understanding intrusion detection and prevention.

- Advanced: This four-day course is designed to increase the depth of knowledge and skills of technical staff charged with administering and securing information systems and networks.]


[This three-day course begins with a brief review of the conceptual foundations of information security. Next, students will be introduced to the CERT Defense-in-Depth Framework: eight operationally focused and inter-dependent management components which will be synergistically applied to a fictitious organization's Information Technology (IT) enterprise. Through lectures, demonstrations, scenario-based exercises, small group activities, and open discussions, students will learn high-level best practices for effectively integrating each of these eight components into all aspects of IT operations. Further, the course scenario is used extensively to reinforce these best practices with technical information security implementations.]

**Secondary sources:**


System Security Assurance (5.1, 5.2, 5.3)

Primary source:


[Abstract from publisher:
The world has changed radically since the first edition was published in 2001. Spammers, virus writers, phishermen, money launderers, and spies now trade busily with each other in a lively online criminal economy -- and as they specialize, they get better. New applications, from search to social networks to electronic voting machines, provide new targets. And terrorism has changed the world. In this indispensable, fully updated guide, Ross Anderson reveals how to build systems that stay dependable whether faced with error or malice.

Here’s straight talk about
- Technical engineering basics—cryptography, protocols, access controls, and distributed systems
- Types of attack—phishing, Web exploits, card fraud, hardware hacks, and electronic warfare
- Specialized protection mechanism—what biometrics, seals, smartcards, alarms, and DRM do, and how they fail
- Security economics—why companies build insecure systems, why it’s tough to manage security projects, and how to cope
- Security psychology—the privacy dilemma, what makes security too hard to use, and why deception will keep increasing
- Policy—why governments waste money on security, why societies are vulnerable to terrorism, and what to do about it]

Secondary sources:


