

Software Engineering Reviews and Audits

Boyd L. Summers



CRC Press
Taylor & Francis Group

AN AUERBACH BOOK

Software Engineering Reviews and Audits

Boyd L.



CRC Press

Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **Informa** business
AN AUERBACH BOOK

Auerbach Publications
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2011 by Taylor and Francis Group, LLC
Auerbach Publications is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number: 978-1-4398-5145-6 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the Auerbach Web site at
<http://www.auerbach-publications.com>

Software Engineering Reviews and Audits

BOOKS ON SOFTWARE AND SYSTEMS DEVELOPMENT AND ENGINEERING FROM AUERBACH PUBLICATIONS AND CRC PRESS

Design and Safety Assessment of Critical Systems

Marco Bozzano and
Adolfo Villafiorita
978-1-4398-0331-8

Implementing and Developing Cloud Computing Applications

David E. Y. Sarna
978-1-4398-3082-6

Secure Java: For Web Application Development

Abhay Bhargav and B. V. Kumar
978-1-4398-2351-4

Scrum Project Management

Kim H. Pries and Jon M. Quigley
978-1-4398-2515-0

Engineering Mega-Systems:

**The Challenge of Systems
Engineering in the
Information Age**
Renee Stevens
978-1-4200-7666-0

Certified Function Point Specialist Examination Guide

David Garmus, Janet Russac, and
Royce Edwards
978-1-4200-7637-0

Enterprise Systems Engineering: Advances in the Theory and Practice

George Rebovich, Jr.
and Brian E. White
978-1-4200-7329-4

Process-Centric Architecture for Enterprise Software Systems

Parameswaran Seshan
978-1-4398-1628-8

Secure and Resilient Software Development

Mark S. Merkow and
Lakshmikanth Raghavan
978-1-4398-2696-6

Real Life Applications of Soft Computing

Anupam Shukla, Ritu Tiwari,
and Rahul Kala
978-1-4398-2287-6

Product Release Planning: Methods, Tools and Applications

Guenther Ruhe
978-0-84932620-2

Process Improvement and CMMI® for Systems and Software

Ron S. Kenett and Emanuel Baker
978-14200-6050-8

Applied Software Product Line Engineering

Kyo C. Kang, Vijayan Sugumaran,
and Sooyong Park
978-1-42006841-2

CAD and GIS Integration

Hassan A. Karimi and Burcu Akinci
978-1-4200-6805-4

Applied Software Product-Line Engineering

Kyo C. Kang, Vijayan Sugumaran,
and Sooyong Park, eds.
978-1-4200-6841-2

Enterprise-Scale Agile Software Development

James Schiel
978-1-4398-0321-9

Handbook of Enterprise Integration

Mostafa Hashem Sherif, ed.
978-1-4200-7821-3

Architecture and Principles of Systems Engineering

Charles Dickerson, Dimitri N. Mavris,
Paul R. Garvey, and Brian E. White
978-1-4200-7253-2

Theory of Science and Technology Transfer and Applications

Sifeng Liu, Zhigeng Fang,
Hongxing Shi, and Benhai Guo
978-1-4200-8741-3

The SIM Guide to Enterprise Architecture

Leon Kappelman
978-1-4398-1113-9

Getting Design Right: A Systems Approach

Peter L. Jackson
978-1-4398-1115-3

Software Testing as a Service

Ashtaque Ahmed
978-1-4200-9956-0

Grey Game Theory and Its Applications in Economic Decision-Making

Zhigeng Fang, Sifeng Liu,
Hongxing Shi, and Yi LinYi Lin
978-1-4200-8739-0

Quality Assurance of Agent-Based and Self-Managed Systems

Reiner Dumke, Steffen Mencke,
and Cornelius Wille
978-1-4398-1266-2

Modeling Software Behavior: A Craftsman's Approach

Paul C. Jorgensen
978-1-4200-8075-9

Design and Implementation of Data Mining Tools

Bhavani Thuraisingham, Latifur Khan,
Mamoun Awad, and Lei Wang
978-1-4200-4590-1

Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems

Duane W. Hybertson
978-1-4200-7251-8

Requirements Engineering for Software and Systems

Phillip A. Laplante
978-1-4200-6467-4

Preface

I have worked in systems engineering, software development, software configuration management, software testing, and software quality assurance organizations for more than 30 years. My experience in these selected fields has been somewhat different from that of most people. I always wanted to experience the software disciplines required in each area of expertise. I know that many individuals or groups reading this book will be surprised to know that successful software engineering reviews and audits are beneficial to the success of software industries and military and aerospace programs. The commercial software world can benefit from this book by helping companies realize that they can succeed in this crazy and sometimes confusing software market and by being aware that effective reviews and audits for software will help them to be more successful.

I attended college and majored in business management with an emphasis in information systems, production and operations management, quantitative analysis methods, statistical analysis, computer science, and application programming. During this time, I worked for an aerospace company and began my journey into software development as a senior systems analyst. I dove into software requirements, software design, code and unit testing, configuration control, software builds, software deliveries, and providing software documentation supporting each area of expertise. I have been involved in performing numerous software engineering reviews and audits to ensure compliance with contractual requirements.

In leading multiple software engineering development teams, I continually tackled complex technical challenges to ensure that system/software engineering problems were addressed and resolved. The main objective of the technical leadership role for software design teams, through the use of common software tools for both UNIX® and Windows® platforms, was to ensure that common software tools were institutionalized. In order to establish and implement common software tools, the necessary capabilities must be provided for teams to learn from other software groups using similar software tools and processes. Adapting these new and proven tools will decrease the software build flow times from days to hours, so teams can be more responsive to customer needs, issues, and concerns. Software metrics should be provided weekly and monthly to senior management

for problems and software build time evaluations to ensure that issues are addressed and resolved. The software tasks for engineering reviews and audits performed during the development lifecycle prior to deliveries and formal audits will improve quality and execution.

Aerospace programs present challenges for personnel working in key technical roles. An example is a replacement of key multiplexer computer systems for the United States Air Force (USAF). Technical software issues are addressed and coordinated daily for the customer. Software teams are directed to capture software baselines and ensure that engineering reviews and audits for software have been performed for installations of new software baselines during test and integration activities.

My career culminated with moving to numerous military and aerospace programs and being involved in the Capability Maturity Model Integration® (CMMI®) activities for systems engineering and software development. I supported and participated in CMMI audit teams and performed CMMI appraisals to certify military and aerospace software programs for accomplishing Level 2 and Level 5 ratings for capability certifications.

SUMMARY

Before software, military, and aerospace programs implement software engineering reviews and audits, it is important to understand the software development lifecycle. Chapters are included that define methods for systems engineering, software design, software quality assurance, software configuration management, and software suppliers/subcontractors. The intent of this book is to ensure that software engineering reviews and audits are conducted and performed incrementally in software development schedules. The implementation of review and audit disciplines will benefit software companies and military and aerospace programs and ensure that formal audits (i.e., FAI, FCA, and PCA) are successful the first time.

Acknowledgments

In the last 30 years of working and gaining experience in software industries and military and aerospace programs, I have been motivated to write a book related to the understanding the importance of performing effective software engineering reviews and audits. My career as a software engineer has included designing, controlling, building, installing, and validating software, which is my passion. Before a company for which I work delivers software to its customers, both in the United States and internationally, I feel a sense of accomplishment that my software has successfully passed critical software reviews and major formal audits. The experience I gained working in multiple data centers for commercial banking, Hercules Aerospace, Mini-System software contractors, and the Boeing Company for 30 years has given me the knowledge of numerous software process improvement activities. It is outstanding to work with the software teams with which I am associated. Senior management and software managers have always been a support to me, and I greatly appreciate their guidance. This book has been on my mind for 30 years, due to seeing the good and the bad of performing effective software engineering reviews and audits.

My acknowledgments go to all the second-level software managers I've worked with for allowing me to excel in the commercial software world and military and aerospace programs. In my early years, and with the support of my wife and family, I received the Outstanding Young Man of America Award and achieved a Bachelor of Science (BS) in business management with an emphasis on computer science while attending Weber State University. Thank you to my lovely wife, for her support and patience with me throughout these past 34 years.

After college the Hercules Aerospace Division employed me as a senior software designer. The software team I worked with gave me insight to the software development environment. I will never forget when my first software manager told me to go work for Boeing and sharpen my software skills. I was soon employed by the Boeing Company in the Software Configuration Management (SCM) organization supporting the B2 and F-22 Raptor programs. The software manager and leads were instrumental in my areas of expertise for advancement in the Boeing Company. Special recognition goes to the Boeing software managers in Denver, Colorado,

for allowing me the opportunity to expand my skills in CMMI, software design, software configuration management, and software quality assurance methods. The F-22 Raptor and Airborne Early Warning & Control (AEW&C) software managers are an inspiration and have given me the opportunity to organize and establish software configuration management teams. Currently I am a software engineer for supporting quality assurance activities and provide software expertise to the F-22 Raptor Air Vehicle System (AVS) and AEW&C programs.

Examples or outlines are made available in the appendices to show how software plans could be defined and documented. Just think about this! Important software lifecycle disciplines are defined and discussed all through this book.

Let's get started!

Boyd Lambert Summers
Maple Valley, Washington

Contents

List of Figures	xiii
List of Tables	xv
Preface	xvii
Acknowledgments	xix

Chapter 1	Introduction	1
1.1	Senior Management	2
1.2	Process and Quality	3
1.3	Software Lifecycle	4
1.3.1	Requirements Phase	4
1.3.2	Design Phase	4
1.3.3	Implementation Phase	5
1.3.4	Test Phase	5
1.3.5	Installation and Checkout Phase	5
1.3.6	Operation and Maintenance Phase	6
1.3.7	Conclusion	6
	Further Reading	6

Chapter 2	Process Model	7
2.1	Process Model	7
2.2	CMMI® Understanding	8
2.3	Process Areas	9
2.4	Staged Maturity Levels	12
2.5	Common Features	12
2.5.1	Software Engineering Process Group	13
2.5.2	Process Change Management	14
	Further Reading	15

Chapter 3	Quality Management Systems	17
3.1	Quality Management Systems	17
3.2	Quality Management Systems Understanding	18
3.3	AS9100 Program Audit	19
3.4	Future Quality Management Systems Objectives	19

- 3.5 Quantitative Management Process 20
- Further Reading..... 20

- Chapter 4 Policy..... 21**
 - 4.1 Policy Understanding.....21
 - 4.1.1 Organization-Level Policy.....22
 - 4.2 Planning Process.....23
 - 4.3 Planning Procedures.....23
 - 4.4 Control Procedures24
 - Further Reading.....24

- Chapter 5 Systems Engineering 25**
 - 5.1 Systems Engineering Method25
 - 5.2 Systems Analysis of Requirements.....26
 - 5.2.1 Requirements Analysis26
 - 5.2.2 Functional Analysis of Requirements26
 - 5.2.3 Systems Integration of Requirements.....27
 - 5.2.4 Systems Verification and Validation of Requirements27
 - 5.3 Systems Engineering Management Plan27
 - Further Reading..... 28

- Chapter 6 Software Design 29**
 - 6.1 Software Design Method29
 - 6.2 Software Development Process..... 30
 - 6.3 Software Design Decisions.....31
 - 6.3.1 Evaluate Requirements and Architecture31
 - 6.3.2 Analyze Requirements and Architecture.....31
 - 6.4 Computer Application Methods.....33
 - 6.4.1 Waterfall Software Method.....33
 - 6.4.2 Spiral Software Method33
 - 6.5 Definition of Agile.....35
 - 6.5.1 Agile Software Development Method35
 - 6.6 Definition of Lean.....36
 - 6.7 Definition of Firmware36
 - 6.7.1 Software Components of Firmware.....37
 - 6.7.2 Hardware Components of Firmware.....37

6.8	Identify Software Components.....	37
6.8.1	Define Relationships	38
6.8.2	Interfaces and Data	38
6.9	Detailed Design.....	39
6.9.1	Components into Software Units.....	39
6.9.2	Configuration Item Details.....	39
6.9.3	Interface and Data Design.....	39
6.10	Evaluate the Software Design.....	40
6.11	Software Coding and Unit Test	41
6.12	Peer Review Method.....	41
6.13	Peer Review Participants.....	41
6.13.1	Peer Review Steps	43
6.13.2	Establish Peer Review Plan	44
6.14	Prepare the Products for the Peer Review.....	45
6.14.1	Prepare the Peer Review Package.....	45
6.14.2	Distribute the Peer Review Package	46
6.14.3	Conduct the Peer Review	46
6.14.4	Categorize the Comments and Disposition	47
6.14.5	Complete the Peer Review.....	48
6.14.6	Complete the Peer Review Records.....	48
6.15	Design Traceability.....	49
6.16	Design Definition	49
6.17	Measurement Data	49
6.18	Configuration Control	50
6.19	Software Inspections	50
6.20	Software Integration and Testing.....	50
6.21	Software Formal Testing.....	52
6.22	Software User Documentation	52
6.23	Maintain Software Development.....	52
	Further Reading.....	52

Chapter 7 Software Quality 55

7.1	SQA and SQE/EQA Terminology.....	55
7.2	Software Quality Methods	56
7.3	Software Quality Organization	57
7.4	Functional Software Quality Tasks.....	57
7.5	Software Quality Program Planning	58

7.6	Process Audit.....	58
7.7	Support Verification and Validation.....	59
7.8	Support Software Configuration Management.....	60
7.9	Support Milestone Reviews.....	60
7.10	Monitor Supplier Quality	60
7.11	Software Noncompliance Handling.....	61
7.12	Perform Software Product Evaluations	61
7.13	SQA Role in Agile Software Development	62
7.14	Software Quality Assurance Plan.....	63
7.15	Summary.....	64
	Further Reading.....	65
Chapter 8	Software Configuration Management	67
8.1	Software Configuration Management Methods	67
8.2	Organization	68
8.2.1	Software Build Engineer.....	69
8.2.2	Software Change Request.....	69
8.2.3	Software Release	69
8.2.4	Engineering Review Board.....	70
8.2.5	Configuration Management Plan.....	70
8.2.6	Software Development Plan.....	70
8.2.7	Operating Procedures.....	71
8.2.8	Computer Program Library	71
8.2.9	Computer Program Library Contents.....	72
8.2.10	Configuration Control.....	72
8.2.11	Developmental Configuration	72
8.3	Configuration Identification	73
8.3.1	Computer Software Component Identification	74
8.3.2	Computer Software Unit Identification.....	74
8.3.3	Nondeliverable Software Identification.....	74
8.3.4	Nondeliverable Software Control.....	74
8.3.5	Part Number Identification.....	74
8.4	Change Control.....	75
8.4.1	Lower-Level Changes.....	75
8.4.2	Middle-Level Changes	75
8.4.3	High-Level Changes.....	75

8.5	Configuration Status Accounting	76
8.5.1	Software Records	76
8.5.2	Archive Software Records	77
8.5.3	Reports	77
8.5.4	Test Configuration	77
8.6	Software Configuration Audit Checklist.....	77
	Further Reading.....	79

Chapter 9 Software Supplier Audit..... 81

9.1	Starting Point for Software Supplier Audits	81
9.2	Key Requirements for Software Supplier Audits.....	82
9.3	Gate Review for Software Supplier Audits	82
9.4	Software Technical Subcontract Management	83
9.5	Subcontractor Software Configuration Management Audits	84
9.5.1	Subcontractor Configuration Identification	84
9.5.2	Subcontractor Computer Program Library.....	85
9.6	Monitor Supplier Software Quality.....	85
9.6.1	Monitor Supplier Quality Method	85
9.6.2	Supplier Selection Activities	86
9.6.3	Provide Quality Plan.....	86
9.6.4	Quality Management Survey	86
9.6.5	Project-Specific Supplier Audits	87
9.6.6	Review Supplier-Delivered Products	87
9.6.7	Participate in Formal Milestone Reviews	87
9.6.8	Support Supplier Qualification Test.....	87
9.6.9	Acceptance of Software Supplier Reviews	87
	Further Reading.....	90

Chapter 10 Software Engineering Reviews..... 91

10.1	Resources	91
10.2	Software Engineering Review Steps.....	92
10.3	Purpose	92
10.3.1	Software Quality Assurance	93
10.3.2	Integrated Product Teams.....	93

10.4	Planning.....	93
10.4.1	Software Development Plan.....	94
10.4.2	Review Plans.....	95
10.5	Performing.....	96
10.6	Document Control Provisions.....	96
10.7	Recording.....	97
10.8	Reporting.....	100
10.8.1	Reporting and Coordination.....	100
10.8.2	Corrective Action/Process Improvement.....	100
10.9	Summary.....	100
	Further Reading.....	101
Chapter 11	Software Engineering Audits	103
11.1	Audit Overview.....	103
11.2	Audit Scope.....	104
11.3	Audit Preparation	104
11.4	Audit Schedule	105
11.5	Audit Plan.....	105
11.6	Audit Communication.....	106
11.7	Audit Entry Criteria	106
11.7.1	Collect Data Package.....	106
11.8	Conduct and Perform Audit	107
11.8.1	Audit Action Items.....	109
11.9	Audit Report.....	109
11.10	Audit Exit Criteria	110
11.11	Audit Record Retention	110
11.12	Summary.....	111
	Further Reading.....	113
Chapter 12	Formal Reviews and Audits.....	115
12.1	System Requirements Review	116
12.2	System Design Review	116
12.3	Software Specification Review	116
12.4	Preliminary Design Review	116
12.5	Critical Design Review.....	117
12.6	Joint Technical Reviews.....	117

12.7	Interim and Final Design Reviews.....	118
12.7.1	Final Design Review Objectives	118
12.9	Required Software Documentation	119
12.9.1	Software Documentation Status.....	119
12.9.2	Software Requirements Specification	119
12.9.3	Interface Design Description.....	120
12.9.4	System/Subsystem Design Description	120
12.9.5	Database Design Description.....	120
12.9.6	Software Design Description	121
12.10	Test Readiness Reviews.....	121
12.10.1	Software Test Plan	123
12.10.2	Software Test Report	123
12.10.3	Version Description Document.....	123
12.11	Software Compliance Status	123
12.11.1	Software Development Files.....	123
12.11.2	Software Verification and Validation	124
12.12	First Article Inspection	125
12.13	Process Steps.....	125
12.13.1	Determine Levels.....	125
12.13.2	Create Records	126
12.13.3	Plan and Schedule	126
12.13.4	Release Information	126
12.13.5	Provide Details.....	126
12.13.6	Perform Quality Inspection and Record Results	126
12.13.7	Software Documentation	127
12.13.8	Audit Exit Criteria	127
12.14	FCA/PCA Audits	127
12.15	Functional Configuration Audit	127
12.16	Physical Configuration Audit	129
12.17	Preliminary Meetings	130
12.18	Audit Preparation	130
12.18.1	Collect Engineering Data Package.....	131
12.19	Perform Audit Pre-Brief	132
12.20	Audit Plan	132
12.21	Audit Metrics	133
12.22	Audit Execution	134
12.23	Audit Final Meeting	134

12.24 Audit Method for Software135

12.25 Final Summary136

Further Reading.....136

Appendix A: Acronyms and Glossary..... 137

Appendix B: Software Development Plan..... 145

Appendix C: Software Quality Assurance Plan 181

Appendix D: Software Configuration Management Plan 201

Appendix E: FCA Checklist..... 219

Appendix F: PCA Checklist..... 223

Acronyms 229

Index..... 235