

Title: The Certified Software Quality Engineer Handbook
Author: Linda Westfall
Publisher: ASQ Quality Press, Milwaukee, WI
ISBN: 978-0-87389-730-3
No. Pages: 644 plus a supplementary CD-ROM
BOK: all

There are those books you expect will be wonderful reading and you look forward to the time doing so. These are usually fiction or non-fiction. Then there are the technical books that you expect will be boring and tedious and you force yourself to begin reading. This was a book I expected to be boring and plodding. In fact, it has turned out to be anything but that.

Linda Westfall has done a masterful job of taking material that could have been dull and boring and presenting it in such a way that it is actually interesting and extremely useful. More than being a rag-tag list of items and their accompanying examples or illustrations, the book is actually a well thought through compendium of user needs and quality-based techniques to fulfill those needs.

Yet, along with the kudos, this reviewer has taken some issues with the book. The primary issue concerns the failure to synch the contents of this book with the referenced IEEE Standard. This, it turns out, is not the author's fault as much as it is the failure of the CSQE BOK to conform to the applicable standards. What is the author's fault is the occasional reference to a recalled IEEE Standard.

Linda Westfall methodically follows the CSQEBOK from beginning to end as she divides this book into seven (7) parts, each part corresponding to a section of the BOK. Within each section there may be as few as two or as many as six chapters with each chapter providing information about a single topic. Following the last chapter is an Appendix containing the CSQE BOK, a glossary of terms, and an immense list of references to enhance the reader's understanding.

The seven parts of the book include:

- Part I General Knowledge
- Part II Software Quality Management
- Part III Systems and Software Engineering Processes
- Part IV Project Management
- Part V Software Metrics and Analysis
- Part VI Software Verification and Validation

Part VII Software Configuration Management

Appendix A Software Quality Engineer Certification Body of Knowledge

Assessment of Parts:

Each chapter defines or describes the topic to be discussed. Each section of a chapter is introduced with the applicable text taken from the BOK. The chapter may contain definitions, examples, methodologies, and techniques that taken together form a framework for the reader. Each chapter also contains graphics that illustrate the points made in the text. Each new topic is introduced by the applicable citation from the BOK.

Part I General Knowledge

The topics in this section encompass: quality principles; ethical and legal compliance; standards and models; leadership skills; and, team skills.

I thought the first two chapters provided a good overview of the covered topics. However I was surprised that the chapter describing standards and models did not mention the relationship of standards to the identified models. For example, the CMMI model recommends that there be a configuration management plan but the reader may not be aware that IEEE Standard 828-2005 should be used to generate a configuration management plan.

Part II Software Quality Management

The topics in this section include: quality management system; methodologies for quality management; and, audits.

While reading the chapter describing a quality management system, I was chagrined to discover that it addressed not a quality management system but rather described a testing overview. I thought the examples would show the totality of a quality management system rather than focus, exclusively on the test process. One of the diagrams shows the hierarchical relationships of standards, policies, process, procedures and instructions. I understand that these are components of the SQA Process; however I was looking for additional information that showed how these items link together.

Audits were covered comprehensively. However, missing was an identification of those things that might trigger an audit as well as what types of audits might be done on a routine basis and included on the project schedule. I recall one situation when I noticed that a document I was given to evaluate as part of the verification process contained a lower level identification number than the previously reviewed document of the same name. This led to a full review of the configuration management process. This review identified numerous inherent weakness of the existing process and led to numerous changes being implemented.

This section would be enhanced by including a template (outline) of an audit document and an audit report such as might be found in IEEE Std. 1028-2008

Part III Systems and Software Engineering Processes

The topics in this section include: life cycles and process models; system architecture; requirements engineering; requirements management; software analysis; design and development; and, maintenance management.

The first thing this reviewer found confusing was that the illustrations of the various development models did not immediately follow the textual description. For example, the text described the V-model and the illustration was that of the W-model. Following the textual description of the spiral-model was the illustration of the W-model.

A helpful graphic for this section would have been a grid that showed the distinguishing characteristics of each to the development models. For example, which ones require all the documentation to be completed in advance and which ones require the system requirements to be completed in advance. Which ones require continuous system testing and which do not.

The description of techniques for requirements elicitation was excellent, especially the detail needed to develop use cases. Also useful were the use of diagrams normally associated with object oriented programming but which are of great use for requirements engineering. Mentioning that use cases and object oriented diagrams are exceedingly useful in developing test cases for user acceptance testing as well as for other levels of testing would have been a nice inclusion.

This reviewer was disappointed to see that requirements analysis and bi-directional traceability were included in the chapter concerning requirements management rather than the chapter on verification, where they more correctly belong. Again this appears to be a problem associated with the BOK rather than with the author's judgment. Although identifying that the change control board was responsible for handling changes to the requirements, there was no mention of the need for all requirements to be baselined and controlled. Nor was there any mention of the need not to re-use requirement numbers or to create a new requirement number whenever the wording in a requirement is changed or modified. These would be the minimum tasks required to manage the requirements.

To counter the random discussions that software maintenance activities are a separate and distinct part of the SDLC, it would be helpful for the author to have mentioned that software maintenance activities take place concurrently with operations activities. While this may be considered a "truism" by experience software quality assurance individuals, it may not be considered a "truism" by those who are less experienced.

Part IV Project Management

This section includes: planning, scheduling, and deployment; tracking and controlling; and, risk management.

This reviewer takes a strong exception to the author's statement that "...many tasks cannot be planned to any level of detail until predecessor activities are completed." It would be more accurate for the author to state that some tasks cannot be executed until predecessor activities have been completed.

However, the author then continues to describe situations in which lower level plans may have to be modified based on new information and previously unforeseen conditions. Yet, even the modification of a plan is very different than the absence of a plan or the absence of a level of detail.

If the identification of project-level standards is not accomplished during the project planning process, *when* will they be identified? The answer to this question is not contained within this section, yet it can be crucial to the ultimate implementation of the software development project.

Thanks go to the author for including the artifacts required for each of the Phase Gate Reviews. This reviewer has been asked to participate in numerous Milestone Reviews that lacked the required artifacts.

Part V Software Metrics and Analysis

This section includes: metrics and measurements theory; process and product measurement; and, analytic techniques.

Kudos to the author for making readily understandable what easily might have been not so understandable. The author has captured the heart of any metrics discussion by stating, and I paraphrase -- that any metric that does not feed a decision is a metric not worth collecting. However, having collected the metric the next logical questions might be – what does it mean and how will it best be represented.

Part VI Software Verification and Validation

The topics included in this section are: theory of verification and validation; test planning and design; reviews and inspections; test execution documentation; and, customer deliverables.

Rather than cover the broad aspects of verification and validation, this section makes it appear that testing covers both of these concepts. Creating an illusion that testing covers both, does a great disservice to the reader of this handbook. Unfortunately this is a problem of the BOK rather than ignorance on the part of the handbook's author.

For a good understanding of the distinction between verification vs. validation and the tasks that go into each, the reader is urged to read IEEE Standard 1012-2004. These tasks encompass the full development life cycle, are appropriate to all types of life cycles, and recommended tasks are determined by the integrity level of the project. It further states that an integrity level may be determined by risk or by any other criterion deemed important by the organization.

Again, the author (or the BOK) considers testing to be the same as V&V by stating that each test plan “can be incorporated as part of a single V&V plan.” Neither the IEEE Standard for Verification and Validation nor the IEEE Standard for Test Documentation mention this possibility. Both standards, and my own 25 years of experience, indicate that V&V will create documentation and execute tests for only software with a high integrity level and that for all other software V&V will verify the software documentation produced by others and may witness test execution. In each of these possible instances,

there will be a V&V Plan and one or multiple test plans, test designs test case, and test procedure documents. There may be a Master Test Plan that describes the management and strategy of the test effort, especially if test encompasses multiple levels of test, multiple iterations of test, and multiple systems.

The chapter covering review and inspections may do a disservice to the reader only because it may lead the reader to believe that having a peer review ameliorates that need for verification and validation to evaluate the contents of the product against one or more agreed upon standards.

Part VII Software Configuration Management

The topics included in these last chapters include: configuration infrastructure; configuration identification; configuration control and status accounting; configuration audits; and product release and distribution.

I was pleased to note that the description of configuration management tools and their uses included those that allowed related documents to be baselined along with the source code. Documents that might be baselined include requirements, design, and test. The ability to roll-back both code and related documentation is extremely important for organizations that may have multiple versions of code in use.

Conclusion:

The writing of a guide for the CSQEBOOK started many years ago when the ASQ Software Division decided it would be helpful for the members to have such a guide. The project was begun and abandoned numerous times until Linda Westfall accepted the challenge. In the preface she states that this book was written not only to be a guide in preparing for the CSQE exam but also as a resource for software quality engineers and others who need to understand how software quality impacts their work. In both of those objectives, Linda Westfall has succeeded in a bountiful fashion.

This book now occupies a prominent place on this reviewer's bookshelf.