

Black Box Software Testing

(Professional Seminar)

Cem Kaner, J.D., Ph.D.

Professor of Computer Sciences
Florida Institute of Technology

Section:30

Testing User Documentation

Summer, 2002

Contact Information:

kaner@kaner.com

www.kaner.com (testing website)

www.badsoftware.com (legal website)

I grant permission to make digital or hard copies of this work for personal or classroom use, with or without fee, provided that (a) copies are not made or distributed for profit or commercial advantage, (b) copies bear this notice and full citation on the first page, and if you distribute the work in portions, the notice and citation must appear on the first page of each portion, (c) each page bear the notice "Copyright (c) Cem Kaner" or if you changed the page, "Adapted from Notes Provided by Cem Kaner". Abstracting with credit is permitted. The proper citation for this work is Cem Kaner, *A Course in Black Box Software Testing (Professional Version)*, Summer-2002, www.testing-education.org. To copy otherwise, to republish or post on servers, or to distribute to lists requires prior specific permission and a fee. Request permission to republish from kaner@kaner.com.

Documentation is an Express Warranty

A warranty is a statement of fact, either articulated or implied by law, respecting the quality or character of the goods to be sold. Under the Uniform Commercial Code an express warranty is:

2-313(a) Any affirmation of fact or promise made by the seller to the buyer which relates to the goods and becomes part of the basis of the bargain . . .

2-313(b) Any description of the goods which is made part of the basis of the bargain . . .

2-313(c) Any sample or model which is made part of the basis of the bargain.

Documentation is an Express Warranty

*You can't disclaim an express warranty --
you are accountable for your claims.*

Uniform Commercial Code 2-316 (1):

Words or conduct relevant to the creation of an express warranty and words or conduct tending to negate or limit warranty shall be construed whenever reasonable as consistent with each other; but . . . negation or limitation is inoperative to the extent that such construction is unreasonable.

Black Box Testing: Testing Documentation

Doc testing is important because:

- Errors in the manual increase risks of legal liability.
- Testing the documentation improves the reliability of the program.
- The documentation may be your mainstream test plan and your most up-to-date specification.
- Confusion in the manual reflects confusion in the program's design.

Refer to Testing Computer Software, Chapter 10

Testing Documentation: What to Test

- Verify every statement of fact and every reasonable implication
- Check the placement and accuracy of figures
- Audit the completeness of the manual (check that every feature is documented)

Track errors in the documentation in a way that is normal for the Doc group. This probably doesn't involve the bug tracking system (but put code/doc mismatches there). If you give back marked up manuscripts, keep photocopies of your markups. Check your corrections against the next circulating draft of the manual.

On average, you will cover 4 pages per hour in a reasonably stable program. Your second pass will go more quickly because the program is in better shape, but it will still take several minutes per page because you will actually test every page.

Testing Documentation: Things to Say

- Your role is *not* editorial: you are not the authority on style and layout.
- Keep your tone non-judgmental.
- Point out upcoming changes (design changes, new error handling, data structures, etc.) that might affect the manual. Mark these in appropriate sections on the manuscript.
- Name experts or references to consult when the writer is confused.
- Suggest examples.
- Point to useful existing data files (for examples).

Testing Documentation: Things to Say

- When appropriate, you might do some writing. The writer might or might not use what you have written. You might write in two ways:
 - » words that you think belong in the manual “as is” (you’re saying, “Here, say it like this and it will be right.”)
 - » explanations that are background material for the writer. Maybe a rough draft of what she’ll write.
- Note: the final review is a meeting just a few days before the book goes to the printer. If you have many small-detail late comments, offer the writer a chance to review them privately with you a few days before the review.

Documentation Testing: On-Line Help

- Contents of the help
- Cross-reference jumps
- Glossary lookups
- Browse sequences
- Graphic hotspots
- Graphic display - color or resolution
- Window size, e.g. compile at 1024x768 and display at 640x480
- Procedure sequences
- Balloon help / tool tips
- Index
- Search
- Context jumps
- Error messages

Publisher Liability for Content-Related Errors

Winter v. G.P. Putnam's Sons, 938 F.2d 1033, (9th Circuit) 1991.

Winter became seriously ill from picking and eating mushrooms after relying on *The Encyclopedia of Mushrooms*, published by Putnam. Putnam did not verify the material in the book and did not intentionally include the error.

- Putnam was not liable for errors in the book.
- Noted that Jeppesen cases consistently held publisher liable for information error when information published was to be used as a tool.
- The court said that a software publisher might be liable for program that does not work as intended.

ALM v. Van Nostrand Reinhold 480 NE.2d 1263, 1985. ***The Making of Tools*. Plaintiff used it, and a tool shattered, causing injury. VNR not liable, but author of the book might be.**

Liability might also attach if the program provides professional services (such as tax preparation) and gives bad information.

Warranties & Misrepresentations: What Must You Test?

- Advertisements
- Published specifications
- Interviews
- Box copy
- Fax-backs
- Manual
- Help system
- Warranty
- Web pages
- Readme
- Advice given to customers on Internet, CompuServe or AOL

