

Black Box Software Testing

Spring 2005

PART 7 -- FUNCTION TESTING

by

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

Professor of Software Engineering

Florida Institute of Technology

and

James Bach

Principal, Satisfice Inc.

Copyright (c) Cem Kaner & James Bach, 2000-2004

This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

These notes are partially based on research that was supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Function Testing

- Tag lines
 - “Black box unit testing.”
 - "Test what it can do"
- Fundamental question or goal
 - Test each function thoroughly, one at a time.
- Paradigmatic case(s)
 - Spreadsheet, test each item in isolation.
 - Database, test each report in isolation
- Strengths
 - Best for assessing capability rather than reliability
 - If used as a dominant technique, thorough analysis of each item tested
- Blind spots
 - Misses interactions, misses exploration of the benefits offered by the program.

Some Function Testing Tasks

- A function is something the product can do
 - Functions may be called features / commands or they may be identified only by what they do (usually, some of each)
- Identify each function and sub-function
 - From specifications or the draft user manual
 - From walking through the user interface
 - From trying commands at the command line
 - From searching the program or resource files for command names
- From this information, build the function list

Primary Artefact: The Function List

- **Category of functions** (*e.g. Editing*)
 - **Individual functions** (*e.g. Cut, Paste, Delete*)
 - **Inputs to the function**
 - Variable
 - » Maximum value
 - » Minimum value
 - » Other special cases
 - **Outputs of the function**
 - **Possible scope of the function** (*e.g. Delete word, Delete paragraph*)
 - **Options of the function** (*e.g. configure the program to Delete the contents of a row of a table, leaving a blank row versus Delete the row along with its contents*)
 - **Circumstances under which the function behaves differently** (*e.g. deleting from a word processor configured to track and display changes or not to track changes*)

Concept Mapping Tools

- Overviews:
 - <http://www.graphic.org/>
 - <http://www.mindtools.com/mindmaps.html>
 - http://users.edte.utwente.nl/lanzing/cm_bibli.htm
 - <http://www.innovationtools.com/resources/mindmapping.asp>
- Cmap (freeware, supports collaborative mapping among several users), <http://cmap.ihmc.us/>
- FreeMind (another free mind mapping program)
http://freemind.sourceforge.net/wiki/index.php/Main_Page

Some Function Testing Tasks

- Identify each function and sub-function
- Build the function list
 - From specifications or the draft user manual
 - From walking through the user interface
 - From trying commands at the command line
 - From searching the program or resource files for command names
- **Key tasks when building or using the function list:**
 - **Determine how you would know if the function worked.**
 - **Identify variables used by the functions and test their boundaries.**
 - **Identify environmental variables that may constrain the function under test.**
 - **Check that each function does what it's supposed to do and does not do what it's not supposed to do.**

Primary Uses of Function Testing

- Orients you to coverage issues
 - you might check your test plan or set of test cases against the function list, for coverage of every feature, and every option and subfunction of every feature.
- Useful for initial testing of the product
 - Sympathetic introduction to the capabilities of the product
 - Fast scan for serious problems that should be addressed early

Risks of Function Testing

- Excellent starting point, but some people rely on this as the primary testing method.
- ***As a primary testing method***, it is a problem that it emphasizes unit testing, isolating tests of individual items:
 - Misses feature interactions
 - Misses load-related issues, interaction with background tasks, effects of interrupts
 - Often focuses on the features without considering boundaries or other tests of special data
 - Doesn't address user tasks—whether the customer can actually achieve the benefits promised by the program