

Black Box Software Testing

(Academic Course - Fall 2001)

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

Florida Institute of Technology

Section: 7 :

The Black Box Testing Organization

Contact Information:

kaner@kaner.com

www.kaner.com (testing practitioners)

www.badsoftware.com (software law)

www.testingeducation.org (education research)

Copyright (c) Cem Kaner 2001.

I grant permission to make digital or hard copies of this work for personal or classroom use, 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. Abstracting with credit is permitted. The proper citation for this work is Cem Kaner, *A Course in Black Box Software Testing (Academic Version)*, Fall 2001, 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.

Black Box Software Testing

The Black Box Testing Organization

Assigned Reading:

Brian Marick, *Classic Testing Mistakes*,
<http://www.testing.com/writings/classic/mistakes.pdf>

Kaner, Hendrickson & Smith-Brock: *Managing the Proportion of Testers to Other Developers*, PNSQC, 2001.

Role of the Testing Group

- Many testing groups call themselves “Quality Assurance.” This gives everyone the wrong idea.
- A QA group has the resources and authority to set standards and solve problems. A QA group owns the project’s quality and thus the satisfaction of the company’s customers. The role includes prevention as well as late-stage damage control. For example, the *real* head of QA can send programmers to courses in defensive programming. Normally, the *real* head of QA in a company is the President or a Senior Vice-President.
- A Testing Group provides important technical services:
 - We find and report bugs. *This is high skill work, not to be under-rated.*
 - We identify weak areas of the program;
 - We identify high risk areas of a project;
 - We explain our findings in ways that help customer service staff to help customers;
 - We explain our findings in ways that help management make reasoned business decisions about each bug.

Role of the Testing Group

- **Within a testing group, the typical responsibilities of a tester are:**
 - Find problems
 - find bugs
 - find design issues
 - find more efficient ways to find bugs
 - Communicate problems
 - report the bugs and design issues
 - report on testing progress
 - evaluate and report the program's stability
 - Manage / supervise testing projects
 - Prepare test plans and schedules
 - Estimate testing tasks, resources, time and budget
 - Measure and report testing progress against milestones
 - Teach other testers to find bugs

It is Not Our Job to Verify That a Program Works Correctly

- If you can't test the program completely, you can't verify that it works correctly.
- Besides, the program *doesn't* work correctly, so no one can verify that it does.
- If your goal is to verify that the program works correctly, then as a tester, you fail every time you find an error.
- *If you set your mind to showing that the program works correctly, you'll be more likely to miss problems than if you want and expect the program to fail.*
- This is a critical psychological problem that affects researchers in all areas of science and that shows up even in basic human sensation and perception.
- Read Rosenthal's *Experimenter Effects in Behavioral Research*, for example.

Our Mission: Which Group is Better?

Testing Group 1

	Found pre-release
Function A	100
Function B	0
Function C	0
Function D	0
Function E	0
Total	100
Function A	50
Function B	6
Function C	6
Function D	6
Function E	6
Total	74

Testing Group 2

**From Marick,
*Classic Testing
Mistakes***

Two groups test the same program.

- The functions are equally important
- The bugs are equally significant

This is artificial, but it sets up a simple context for a discussion of tradeoffs.

So? Purpose of Testing?

- **A testing group has two key priorities, bug finding and assessment. Sometimes, these conflict:**
 1. Find the bugs (preferably in priority order).
 2. Help the project manager make tough decisions.
- *Assessment is the underlying reason for testing, from management's viewpoint. But if you aren't hammering hard on the program, you aren't doing anyone any favors.*

Varying Missions of Test Groups

(Some of these might be impossible)

- Find defects
- Maximize bug count
- Block premature product releases
- Help managers make ship / no-ship decisions
- Assess quality
- Minimize technical support costs
- Conform to regulations
- Minimize safety-related lawsuit risk
- Assess conformance to specification
- Find safe scenarios for use of the product (find ways to get it to work, in spite of the bugs)
- Verify correctness of the product
- Assure quality

A Different Take: Public vs. Private Bugs

- A programmer's **public bug rate** includes all bugs left in the code when she gives it to someone else (such as a tester.) Rates of one bug per hundred statements are not unusual, and several programmers' rates are higher (such as three bugs per hundred).
- A programmer's **private bug rate** includes all the bugs that she makes, including the ones she fixes before passing the program to testing.
- Estimates of private bug rates have ranged from 15 to 150 bugs per 100 statements. Therefore, programmers must be finding and fixing between 80% and 99.3% of their own bugs before their code goes into test. (Even the sloppy ones find and fix a lot of their own bugs.)
- What does this tell us about our task?
- It says that we're looking into the programmer's (and her tools') blind spots. Merely repeating the types of tests that the programmers did won't yield more bugs. That's one of the reasons that an alternative approach is so valuable.

On Hitting The Wall

- What do you do when you've been testing the product for a while and you finally hit the wall / lose your edge?
- A few suggestions:
 - test against the manual for fresh perspective
 - rotate program areas
 - bring in users for observation
 - Pull out top bugs, see where the weak areas of the program are and re-attack
 - be systematic (if uncreative)
 - create complex documents (use competitors' demos)
 - switch focus to automation or some other useful, technical task.

Our Mission: Which Group is Better?

	Found pre- release	Found later	Total
Function A	100	0	100
Function B	0	12	12
Function C	0	12	12
Function D	0	12	12
Function E	0	12	12
Total	100	48	148
Function A	50	50	100
Function B	6	6	12
Function C	6	6	12
Function D	6	6	12
Function E	6	6	12
Total	74	74	148