

Black Box Software Testing

(Professional Seminar)

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

Professor of Computer Sciences
Florida Institute of Technology

Section:9

Overview of Black Box Testing Paradigms

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.

Paradigms of Black Box Testing

ACKNOWLEDGEMENT

This section is based on work done jointly with James Bach, and uses slides that James and I prepared jointly for presentation at the USPDI and STAR conferences. We also thank Bob Stahl, Brian Marick, Hans Schaefer, and Hans Buwalda for several insights.

ABOUT JAMES BACH

- James Bach writes, speaks, and consults on software quality assurance and testing. In his sixteen years in Silicon Valley-style software companies, including nine years at Apple Computer, Borland International, and Aurigin Systems, he's been a programmer, tester, QA manager, and roving problem-solver. For three years, James was Chief Scientist at ST Labs, an independent software testing company in Seattle that performs substantial testing and tester training for Microsoft.
- James is a frequent speaker and writer on the subject of software quality assurance and development processes. James writes and edits the "Software Realities" column in IEEE Computer magazine, is a former section chair of the American Society for Quality and the Silicon Valley Software Quality Association, and was part of the ASQ committee that created the Certified Software Quality Engineer program.

The Puzzle

Black box testing groups vary widely in their approach to testing.

Tests that seem essential to one group seem uninteresting or irrelevant to another.

Big differences can appear even when both groups are composed of intelligent, experienced, dedicated people.

Why?

Paradigms

A paradigm provides a structure of thinking about an area.

- Typically, the description of a paradigm includes one or a few paradigmatic cases -- key example. Much of the reasoning within the paradigm is based on analogies to these key cases.
- The paradigm creates a mainstream of thinking--it provides insights and a direction for further research or work. But it implicitly also defines limits on what is relevant, interesting, or possible. Things outside of the paradigm are uninteresting. People who solve the types of puzzles that are characteristic of the paradigm are respected, whereas people who solve other types of puzzles instead are outsiders, and not well respected.
- *A testing paradigm would define the types of tests that are (to the person operating under this paradigm) relevant and interesting.*

Paradigms (Kuhn)

See Thomas Kuhn, *The Structure of Scientific Revolutions*. He describes paradigms as follows:

- **A paradigm is a model based on a shared experiment that includes law, theory, application and instrumentation together and from which springs particular coherent traditions of scientific research.**
- **A paradigm includes a concrete scientific achievement as a locus of professional commitment, prior to the various concepts, laws, theories and points abstracted from it.**
- *The pre-paradigm period . . . is regularly marked by frequent and deep debate over legitimate methods, problems, and standards of solution, though these serve rather to define schools than to produce agreement.*

Black Box Testing Paradigms

- There are strikingly different paradigms (or sub-paradigms) within black box testing.
- This list reflects our (Kaner's/Bach's) observations in the field and is not exhaustive.
- We put one on the list if we've seen credible testers drive their thinking about black box testing in the way we describe. A paradigm for one person might merely be a technique for another.
- *We recommend that you try to master the “right” combination of two or three approaches. They are not mutually exclusive. The right combination will depend on your situation.*

A List of Paradigms

- Function testing
- Domain testing
- Specification-based testing
- Risk-based testing
- Stress testing
- Regression testing
- Exploratory testing
- User testing
- Scenario testing
- Stochastic or Random testing

Sample Exam Questions

Consider the tools in TI interactive. For each tool, identify the two testing strategies that you would consider most useful, and explain why. (Across the different tools, please describe a variety of *different*

TI InterActive! toolbar

The buttons on the TI InterActive! toolbar give you quick access to the program's main features.



Performs calculations and defines variables and functions.



Performs typical spreadsheet operations.



Specifies the mode settings for each object.



Performs statistics regression calculations on lists of data.



Graphs functions and plots statistical data.



Transfers data to/from a connected calculator.



Generates a table of values for defined functions.



Captures the screen of a connected calculator (TI-83 or TI-83 Plus).



Enters and/or edits lists of data.



Browses the Web and extracts data directly from Web pages.



Enters and/or edits matrices.



Sends e-mail attachments of your current document.

Paradigm Exercise

- **Do any of the paradigms listed reflect a dominant approach in your company? Which one(s)?**
- **Looking at the paradigms as styles of testing, which styles are in use in your company? (List them from most common to least.)**
- **Of the ones that are not common or not in use in your company, is there one that looks useful, that you think you could add to your company's repertoire?**

Black Box Software Testing

Paradigms:

Function Testing

Function Testing

Tag line

- “Black box unit testing.”

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

- Thorough analysis of each item tested

Blind spots

- Misses interactions, misses exploration of the benefits offered by the program.

Some Function Testing Tasks

Identify the program's features / commands

- 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

Identify variables used by the functions and test their boundaries.

Identify environmental variables that may constrain the function under test.

Use each function in a mainstream way (positive testing) and push it in as many ways as possible, as hard as possible.

