

# *Black Box Software Testing*

## *(Professional Seminar)*

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

Professor of Computer Sciences  
Florida Institute of Technology

### **Section:19**

### **Scenario Testing**

Summer, 2002

Contact Information:

kaner@kaner.com

[www.kaner.com](http://www.kaner.com) (testing website)

[www.badsoftware.com](http://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](http://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](mailto:kaner@kaner.com).

# *Scenario Testing*

## **Tag lines**

- “Do something useful and interesting”
- “Do one thing after another.”

## **Fundamental question or goal**

- Challenging cases that reflect real use.

## **Paradigmatic case(s)**

- Appraise product against business rules, customer data, competitors’ output
- Life history testing (Hans Buwalda’s “soap opera testing.”)
- Use cases are a simpler form, often derived from product capabilities and user model rather than from naturalistic observation of systems of this kind.

# *Scenario Testing*

## **The ideal scenario has several characteristics:**

- It is realistic (e.g. it comes from actual customer or competitor situations).
- There is no ambiguity about whether a test passed or failed.
- The test is complex, that is, it uses several features and functions.
- There is a stakeholder who has influence and will protest if the program doesn't pass this scenario.

## **Strengths**

- Complex, realistic events. Can handle (help with) situations that are too complex to model.
- Exposes failures that occur (develop) over time

## **Blind spots**

- Single function failures can make this test inefficient.
- Must think carefully to achieve good coverage.

# *Scenarios*

## **Some ways to trigger thinking about scenarios:**

- **Benefits-driven:** People want to achieve X. How will they do it, for the following X's?
- **Sequence-driven:** People (or the system) typically does task X in an order. What are the most common orders (sequences) of subtasks in achieving X?
- **Transaction-driven:** We are trying to complete a specific transaction, such as opening a bank account or sending a message. What are all the steps, data items, outputs and displays, etc.?
- **Get use ideas from competing product:** Their docs, advertisements, help, etc., all suggest best or most interesting uses of their products. How would our product do these things?

# *Scenarios*

## **Some ways to trigger thinking about scenarios:**

- **Competitor's output driven:** Hey, look at these cool documents they can make. Look (think of Netscape's superb handling of often screwy HTML code) at how well they display things. How do we do with these?
- **Customer's forms driven:** Here are the forms the customer produces in her business. How can we work with (read, fill out, display, verify, whatever) them?

# *Soap Operas*

- Build a scenario based on real-life experience. This means client/customer experience.
- Exaggerate each aspect of it:
  - » example, for each variable, substitute a more extreme value
  - » example, if a scenario can include a repeating element, repeat it lots of times
  - » make the environment less hospitable to the case (increase or decrease memory, printer resolution, video resolution, etc.)
- Create a real-life story that combines all of the elements into a test case narrative.

**(Thanks to Hans Buwalda for developing this approach and patiently explaining it to me.)**

# *Soap Operas*

(As these have evolved, Hans distinguishes between *normal soap operas*, which combine many issues based on user requirements—typically derived from meetings with the user community and probably don't exaggerate beyond normal use—and *killer soap operas*, which combine *and exaggerate to produce extreme cases.*)

# *Scenario Testing: Interesting Papers*

---

**Hans Buwalda on Soap Operas (in the conference proceedings of STAR East 2000)**

**Kaner, A pattern for scenario testing, at [www.testing.com](http://www.testing.com)**

**Lots of literature on use cases**

# *Sample Exam Questions*

---

**Describe the characteristics of a good scenario test.**

# *Sample Exam Questions*

**Define a scenario test and describe the characteristics of a good scenario test. Imagine developing a set of scenario tests for the TI interactive product, that involved matrix handling.**

- What research would you do in order to develop a series of scenario tests?
- Describe two scenario tests that you would use and
- Explain why each is a good example of a scenario test.

# *Sample Exam Questions*

**Define a scenario test and describe the characteristics of a good scenario test. Imagine developing a set of scenario tests for the TI interactive product, that were focused on configuration-related issues (compatibility of hardware/software with the product).**

- What research would you do in order to develop a series of scenario tests?
- Describe two scenario tests that you would use and
- Explain why each is a good example of a scenario test.

# *Sample Exam Questions*

**Define a scenario test and describe the characteristics of a good scenario test. Imagine developing a set of scenario tests for a C compiler. What research would you do in order to develop a series of scenario tests? (*NOTE: I am not asking for tests of the user interface to the compiler. I'm asking for tests of what we typically think of as compiler functionality.*) Describe two scenario tests that you would use and explain why each is a good test.**

# *Sample Exam Questions*

---

**Imagine that you were testing the feature, “Stat calculation tool.”**

- Describe a scenario test that you would use to test this feature.
- Explain why this is a particularly good scenario test.

# *Sample Exam Questions*

---

**Imagine that you were testing the feature, “Stat calculation tool.”**

- Explain how you would develop a set of soap operas to test this feature.
- Describe one test that might qualify as a soap opera.
- Explain why this is a good soap opera test.

