

# Black Box Software Testing

Fall 2005

## Overview—Part 2 (Mission of Testing)

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

Professor of Software Engineering

Florida Institute of Technology

and

James Bach

Principal, Satisfice Inc.

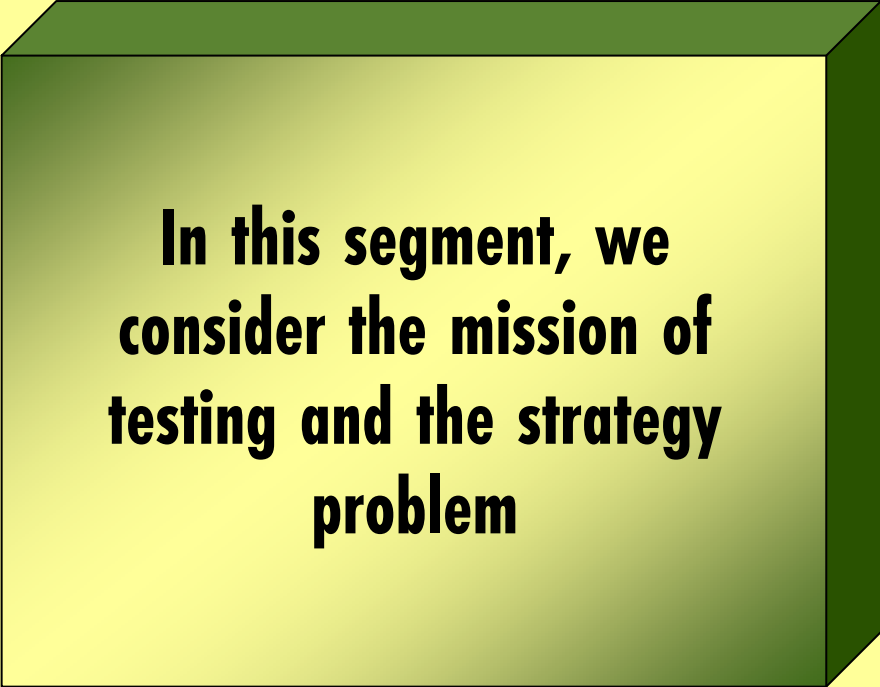
### **Copyright (c) Cem Kaner & James Bach, 2000-2005**

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.

## Some fundamental questions in software testing

- Why are you testing? What are you trying to learn? [*What is the mission of your testing?*]
- How should you organize your work to achieve your mission? [*The strategy problem*]
- How will you know whether the program passed or failed the test? [*The oracle problem*]
- What would it take to do a complete testing job? [*The impossibility of complete testing.*]
- How much testing is enough? [*The measurement problem.*]



**In this segment, we consider the mission of testing and the strategy problem**

## What is testing?

*A technical investigation of the product under test conducted to provide stakeholders with quality-related information.*

# Defining Testing

## **A technical**

- We use technical means, including experimentation, logic, mathematics, models, tools (testing-support programs), and tools (measuring instruments, event generators, etc.)

## **investigation**

- An organized and thorough search for information.
- This is an active process of inquiry. We ask hard questions (aka run hard test cases) and look carefully at the results.

# Defining Testing

**A technical  
Investigation ...**

**of the product under test**

- The product includes the data, the documentation, the hardware, whatever the customer gets. If it doesn't all work together, it doesn't work.

**conducted to provide  
stakeholders**

- Someone who has a vested interest in the success of the testing effort
- Someone who has a vested interest in the success of the product

# Defining Testing

**A technical  
investigation  
of the product under test  
conducted to provide  
stakeholders ...**

**with quality-related information**

- The information of interest is **often** about the presence (or absence) of bugs.
- Other types of information are sometimes more vital to the stakeholders

# Information Objectives

- Find important bugs, to get them fixed
- Assess the quality of the product
- Help managers make release decisions
- Block premature product releases
- Help predict and control costs of product support
- Check interoperability with other products
- Find safe scenarios for use of the product
- Assess conformance to specifications
- Certify the product meets a particular standard
- Ensure the testing process meets accountability standards
- Minimize the risk of safety-related lawsuits
- Help clients improve product quality & testability
- Help clients improve their processes
- Evaluate the product for a third party

**Different objectives  
require different  
testing strategies and  
will yield different  
tests, different test  
documentation and  
different test results.**

# Testing strategy: A brief introduction

Your testing strategy is your plan for:

- Meeting your information objectives
- In the context of your project



**Let's use an example to illustrate  
the idea of a testing strategy**



# A thought experiment

Suppose you were testing a program that does calculations, like a spreadsheet.

Consider 4 development contexts:

- a) Computer game
- b) Early development of a commercial product, at the request of the project manager, to help her identify product risks and help her programmers understand the reliability implications of their work
- c) Late development of a commercial product, to help the project manager decide whether the product is finished
- d) Control medical equipment

## A thought experiment (2)

For each context:

- What is your mission?
- How should you organize your testing to help you achieve the mission?
  - How aggressively should you hunt for bugs? Why?
  - Which bugs are less important than others? Why?
  - How extensively will you document your work? Why?
- Suppose the program has a numeric input field. The spec says it must accept single digits, but not how it should respond to letters. Should you test with letters? What if you're time pressed?

**Next stop:**

**Test oracles**