Black Box Software Testing

Fall 2005

Overview - Part 1 of 3. Fundamental issues.

Cem Kaner, J.D., Ph.D.
Professor of Software Engineering
Florida Institute of Technology
and
James Bach
Principal, Satisfice Inc.

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.

Overview

In a typical testing project, someone gives you a program and asks you to test it.

- The program may (or may not) come with a specification and documentation.
- Documents vary in accuracy and completeness.

You create tests, run them, and report the progress and results of your work.

New versions of the program come to you. You might reuse some old tests, check whether bugs have been fixed, and (or) try new tests.

Eventually, the product is either cancelled or put into production.

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.]***

The issues these questions raise are very difficult, so difficult that they fundamentally influence how testing is done.

Next segment:

The missions
and
strategies of testing