

Software testing

Experimental studies

Marco Aurélio Graciotto Silva¹, Ellen Francine Barbosa¹,
José Carlos Maldonado¹

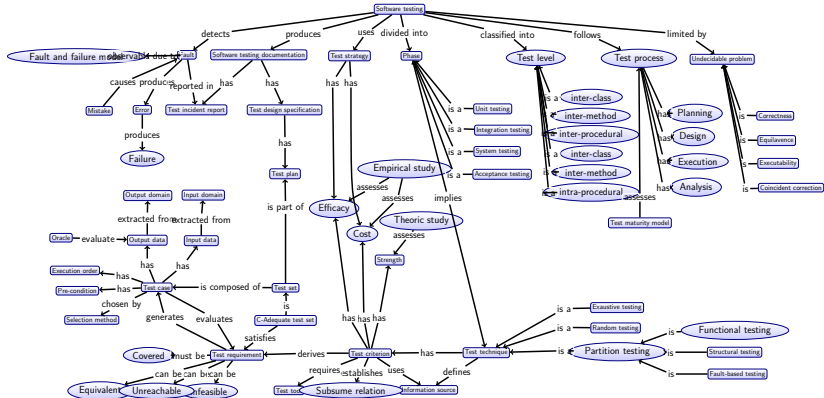
¹University of São Paulo (USP)
São Carlos, SP, Brazil

²Federal University of Goiás (UFG)
Goiânia, GO, Brazil

February 2011

Software testing

Software testing



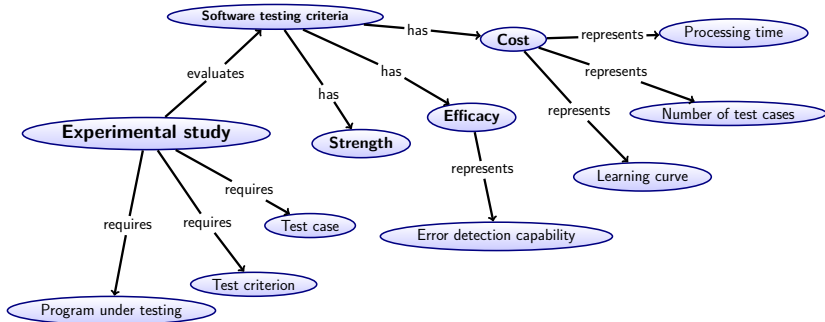
Experimental studies for software testing

Software
testing

Experimental
studies

Evaluated properties

Activities



Experimental studies and software testing

- In the empirical approach, statistics are collected which record the frequency with which different testing strategies reveal the errors in a collection of programs [1].
- Experimental studies allow the establishment of properties and characteristics of the test criteria:
 - cost,
 - efficacy,
 - strength.

Motivation

- Given two test criteria, which is harder to satisfy?
- How to reduce the cost of a test criterion without reducing its capability of error detection?

Cost

Cost is the effort required to use a given test criterion against a program.

Cost metrics

Cost can be measured using the following attributes:

- Quantity of test cases required to satisfy a test criterion.
- Time required to execute all the test cases.
- Time required to identify equivalent and infeasible test requirements.
- Effort required to create test cases.
- Effort required to learn how to use a test tool.

Efficacy

Efficacy is the capability a test criterion has of detecting a greater number of errors than another test criterion.

Strength

Strength is the probability that a test criterion will be satisfied by another test criterion.

Activities

1. Select and prepare the programs to be tested (the population of the experiment).
2. Select the software testing tools.
3. Create test cases adequate for a set of test criteria (the test criteria are the intervention of the experiment).
4. Run the program under test using the test cases.
5. Identify infeasible requirements.
6. Collect and analyze the results.

Program selection

- Programs should be selected for testing.
- Open source software are an interesting choice (as their source code is readily available).
- Simple programs, such as `cal` and `identifier`, can also be used (but with reduced confidence in the evaluation of the properties of the test criteria).

Test tools selection

- It is a requirement that the test be automated as much as possible.
- A common requirement is that the test tools provide a script execution mode (so that the tester intervention is not required to run every aspect of the tool).
- JaBUTi can be run from the command line, using scripts.

Test cases generation

- Test cases are usually generated randomly.
 - It is easy to generate input data randomly.
 - It eliminates the influence of the tester on the generated data (and any threat of validity due to the tester influence).
- If the required test coverage is not achieved, test cases should be manually and added to the test set.

Program execution

- Using the test tool, the program is run using the input data from the test cases.
- The output data of the program is compared with the expected output data (as of the test cases).
- Test requirements satisfaction is analyzed using the program results and execution trace.

Infeasible requirements identification

- Infeasible requirements (non-executable paths, equivalent mutants) should be identified.
- Test set should be augmented until a C – *adequate* test set is defined.



Analysis

- Test results – cost, efficacy and strength – are calculated and compared.
- Relations between test criteria are established.

Example: subsume relation



HOWDEN, W. E. Theoretical and empirical studies of program testing. *IEEE Transactions on Software Engineering*, v. 4, n. 4, p. 293–298, jul. 1978.

Credits

Software
testing

Acknowledgeme

