Testing Principles
SFWR ENG 3S03 Software Testing

Alicia Marinache

Department of Computing and Software, McMaster University
Canada L8S 4L7, Hamilton, Ontario

Acknowledgments: Material based on [Mye04, Chapter 2]

Objectives

Testing Principles SFWR ENG 3S03 Software Testing

A.Marinache

Preliminaries

Principles

Summary

- Identify a set of testing principles or guidelines
- Discuss the important principles found in the literature

Software defect: when does it occur?

1. The software does not do something that the specification says it should do.
2. The software does something that the specification says it should not do.
3. The software does something that the specification does not mention.
4. The software does not do something that the specification does not mention but should.
5. The software is dificult to understand, hard to use, slow, or just plain not right.

Acknowledgments: [Pat06, Chapter 1]

Testing Principles
SFWR ENG 3S03
Software Testing
A.Marinache
Preliminaries
Principles
Summary

What is the goal of software testing?

- "To demonstrate that errors are not present"
- "To show that a program performs its intended functions correctly"

What is the goal of software testing?

- "To demonstrate that errors are not present"
- "To show that a program performs its intended functions correctly"

Goal of Software Testing

To find defects, find them as early as possible, and make sure they get fixed.

Testing Principles
SFWR ENG 3S03
Software Testing
A.Marinache
Preliminaries
Principles
Summary

Principle 1

A necessary part of a test case is a definition of the expected output or result.

- A test case must consist of two components:
  - A description of the input data to the program
  - A precise description of the correct output of the program for that set of input data

Principle 2

A programmer should avoid attempting to test his or her own program.

- Does not apply to debugging
- Reason 1: For the person who constructed the program it is difficult to look at it with a destructive eye
- Reason 2: It is likely that the "builder" will carry the same misunderstanding into the tests of the program

Principle 3

A programming organization should not test its own programs.

- Similar to the previous argument
- In general, a programming organization is measured on the ability to produce a program by a given date and for a certain cost
- Testing is more economical if performed by an objective independent party

Principle 4

Thoroughly inspect the results of each test.

- The most obvious principle, yet often overlooked
- Errors that are found on later tests are often missed in the results from earlier tests

Principle 5

Test cases must be written for input conditions that are invalid and/or unexpected, as well as for those that are valid and expected.

- Testing concentrates on the latter, neglecting the former
- Examining a program to see if it does not do what it is supposed to do is only half the battle; the other half is seeing whether the program does what it is not supposed to do

Principle 6

Avoid throwaway test cases unless the program is truly a throwaway program.

- Often observed with interactive systems
- Problem: Test cases represent a valuable investment lost once the testing has been completed

Principle 7

Do not plan a testing effort under the tacit assumption that no errors will be found.

- Mistake due to the use of the incorrect perception of testing: ~~testing is the process of showing that the system functions correctly~~
- Testing is the process of executing a system with the intent of finding errors

Principle 8

The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section.

- A program consists of two modules A and B and
  - five errors found in module A and one error found in module B
  - modul A has not been purposely subjected to a more rigorous test

  then the likelihood of more errors existing in module A is greater than the likelihood of more errors existing in module B

Principle 9

Testing is an extremely creative and intellectually challenging task.

- Creativity required in testing a large system could exceed the creativity required in designing that system
- It is impossible to test a program sufficiently to guarantee the absence of all errors
- Techniques let you develop a reasonable set of test cases, but still require a significant amount of creativity
- Testing Is Not For Dummies!

What makes a good software tester?

- They are explorers
- They are troubleshooters
- They are relentless
- They are creative
- They are perfectionists
- They exercise good judgment
- They are tactful and diplomatic
- They are persuasive

Testing Principles
SFWR ENG 3S03
Software Testing
A.Marinache
Preliminaries
Principles
Summary

- Testing is the process of executing a program with the intent of finding errors
- A good test case is one that has a high probability of detecting an as yet undiscovered error
- A successful test case is one that detects an as yet undiscovered error

Testing Principles
SFWR ENG 3S03
Software Testing
A.Marinache
Preliminaries
Principles
Summary

References I

 Glenford J. Myers, *The art of software testing*, second ed., John Wiley & Sons, Inc., 2004.

 Ron Patton, *Software testing*, second ed., SAMS, 2006.