EDITH COWAN UNIVERSITY

CSI3105: Software Testing

Module 1a: Introduction to Software Testing

Creative thinkers made here.

Unit Summary

EDITH COWAN UNIVERSITY

- Computing is ubiquitous
- We test to avoid failures
  - Bad publicity
  - Loss of users
  - Sued for damages
- Best course of action is to follow a systematic process

Module Summary

- To find out what is software testing and software testing activities
- To learn why software testing is important and why software fail?
- To examine some (in)famous failures in the discipline
- To learn software testing terminology
- To learn classification of software testing techniques

Module Summary

EDITH COWAN UNIVERSITY

- Video 1b
  - Define what is software testing
- Video 1c
  - Outline why it is important
- Video 1d
  - Define some common terminology
- Video 1e
  - How we can classify software testing techniques

What is Software Testing?

EDITH COWAN UNIVERSITY

- Definition: Testing is the process of executing a program with the intention of finding errors (Myers,1976).
- Testing - an unnatural process as its aim is to make the program fail.
  - Errors, faults, failure
- To be successful in his/her testing activities, the tester must construct his/her test case such that; if faults are present in the software, these faults will be exposed.
- It is not possible to guarantee the absence of errors by testing the software using a large set of test cases.
- The optimal result of testing - the maximum exposure of errors present.
  - Testing can demonstrate the presence of errors but not their absence.
- Effective testing is a result of adequate preparations being made before hand:
  - to arrange the system such that it easy to test and
  - by preparing a plan of testing so that the sequence of testing is well organised.

Software Testing Activities

EDITH COWAN UNIVERSITY

- Steps in carrying out testing:
  - Establish test objectives.
  - Design test cases and writing test cases.
  - Evaluate the test cases.
  - Execute the tests using the developed program.
  - Examine test results (output from the previous step).

Software Testing: Why

EDITH COWAN UNIVERSITY

Testing is an important step in the development of programs

- Untested programs are likely to contain bugs, security flaws, logic errors or other undesirable behaviour
  - Leading to downtime, lower sales, bad reputation, potential litigation, etc…

Software failures are costly.

- $60 billion each year [NIST 2002]
- On average, 1 hour downtime (financial company) costs >$6M [Gartner'98].
- System defects account for up to 40% of system failures [Marcus 2000]

Causes of Software Failure

EDITH COWAN UNIVERSITY

- Incorrect memory Usage (e.g. segmentation fault)
- Deadlocks
- Memory leaks
- Wrong Implementation
- Regression Bugs

Famous Software Failures

ECU
EDITH COWAN UNIVERSITY

Examples:

- Therac-25 (1985-1987)
- Morris Worm (1988)
- Kerberos Random Number Generator (1988-1996)
- Denver Baggage Handling System
- Ariane 5
- Apple's "goto fail" bug (CVE-2014-1266) (2014)
- OpenSSL's "Heartbleed" bug (CVE-2014-0160) (2014)

Software Testing Terminology

Error, Faults, Failures

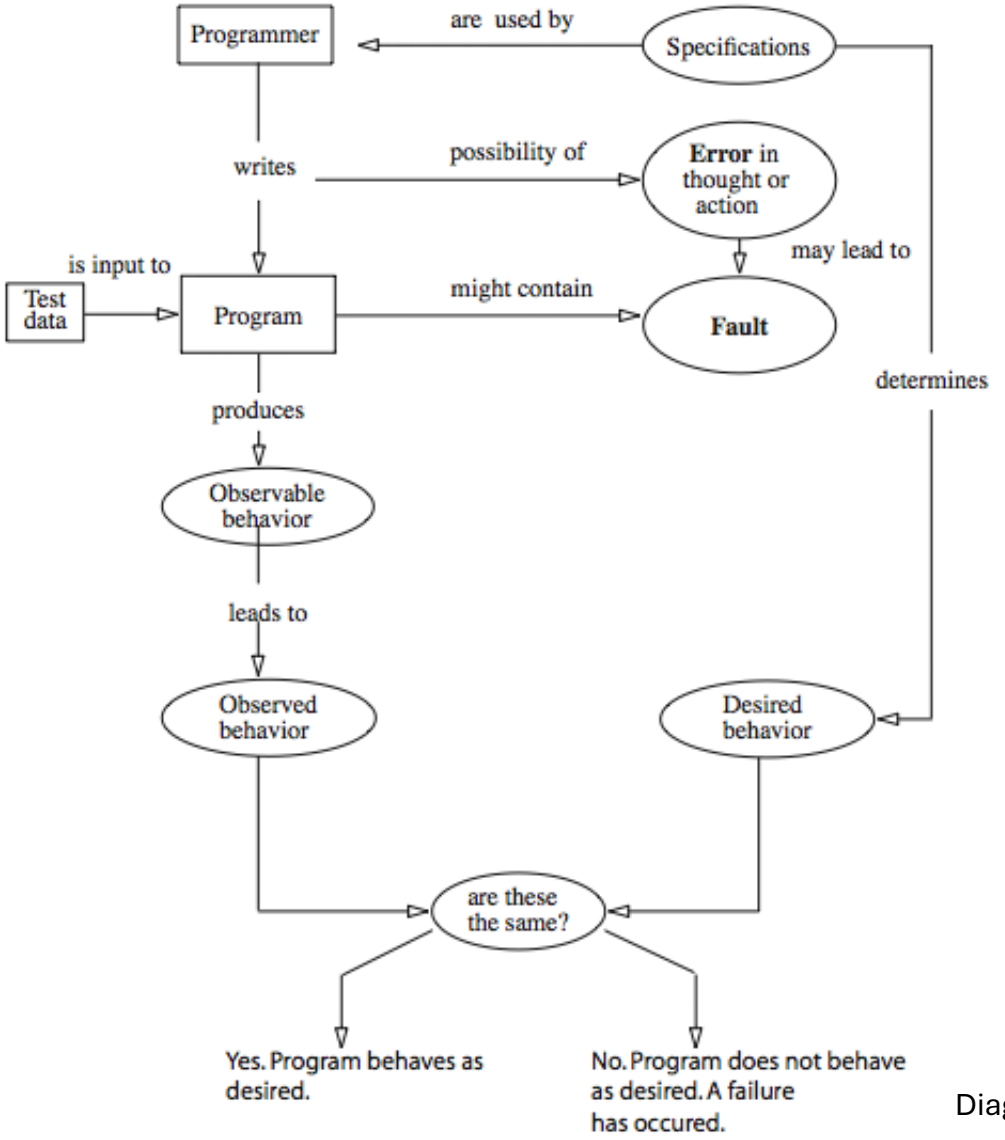

Diagram from Mathur(2014)

Software Testing Terminology: Software Quality

EDITH COWAN UNIVERSITY

Static quality attributes: structured, maintainable, testable code as well as the availability of correct and complete documentation.

Dynamic quality attributes: software reliability, correctness, completeness, consistency, usability, and performance

- **Software reliability** is the probability of failure free operation of software in its intended environment.
- **Correctness:**
  - Establish correctness via testing would imply testing a program on all elements in the input domain. In most cases that are encountered in practice, this is impossible to accomplish.
  - Established via mathematical proofs of programs
- **Testability:**
  - *The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met* (IEEE)

Software Testing Terminology

EDITH COWAN UNIVERSITY

Program Behaviour

Program states

- state of a program is the set of current values of all its variables (state vector) and an indication of the next statement to be executed.
- Program execution of its statements – causes the program to move from one state to the next.
- Program Behaviour = a sequence of different program states

To specify:

- Simplest way - Natural Language (multiple interpretations)
- State Diagram, Formal Mathematical Specifications

Software Testing Terminology:

EDITH COWAN UNIVERSITY

Operational Profiles:

- An operational profile is a numerical description of how a program is used.

Examples:

Operational profile #1

Sequence	Probability
Numbers only	0.9
Alphanumeric strings	0.1

Operational profile #2

Sequence	Probability
Numbers only	0.1
Alphanumeric strings	0.9

Software Testing Terminology

Test/debug cycle

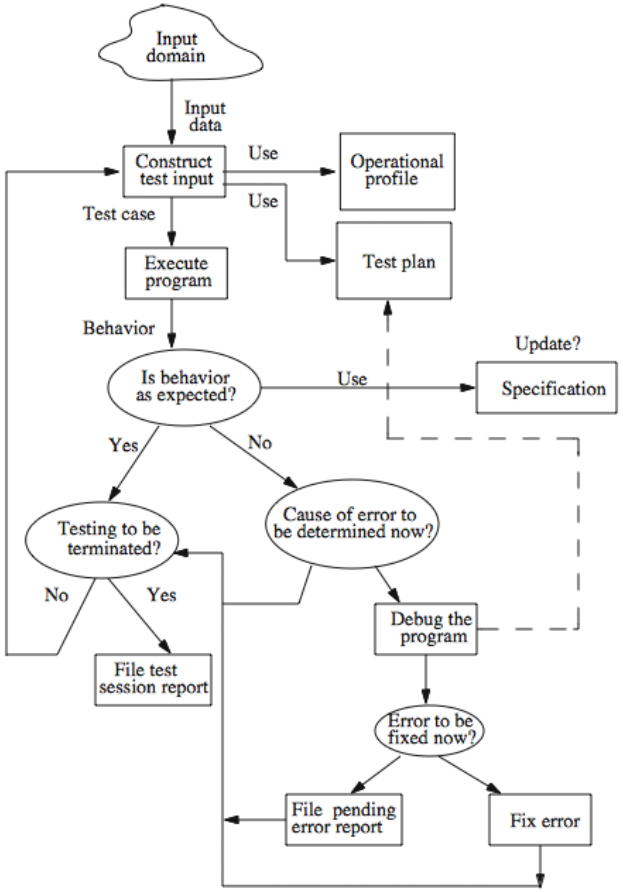

Diagram from Mathur(2014)

Software Testing Terminology

Test Oracle

- The entity that performs the task of checking the correctness of the observed behavior of a program

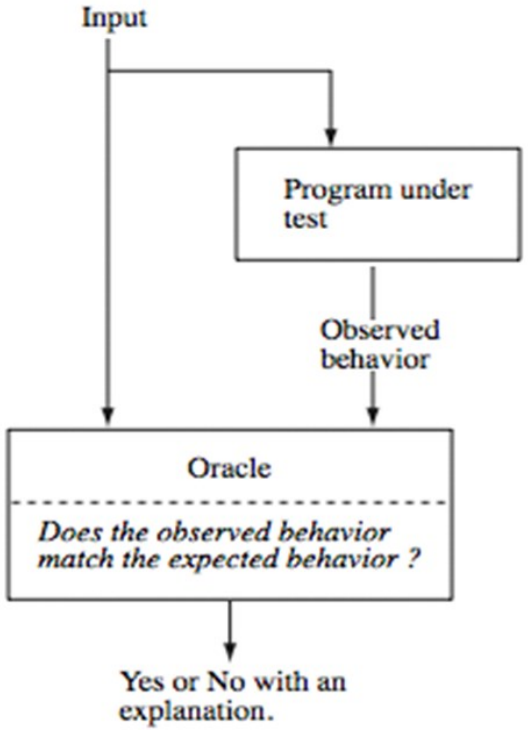

Diagram from Mathur(2014)

Classification of Software Testing Techniques

EDITH COWAN UNIVERSITY

One possible classification:

C1: Source of test generation.

C2: Lifecycle phase in which testing takes place

C3: Goal of a specific testing activity

C4: Characteristics of the artifact under test

Classification of Software Testing Techniques: C1

Artifact	Technique	Example
Requirements (informal)	Black-box	Ad-hoc testing Boundary value analysis Category partition Classification trees Cause-effect graphs Equivalence partitioning Partition testing Predicate testing Random testing
Code	White-box	Adequacy assessment Coverage testing Data-flow testing Domain testing Mutation testing Path testing Structural testing Test minimization using coverage
Requirements and code	Black-box and White-box	
Formal model: Graphical or mathematical specification	Model-based Specification	Statechart testing FSM testing Pairwise testing Syntax testing
Component interface	Interface testing	Interface mutation Pairwise testing

Taken from Mathur (2014)

Classification of Software Testing Techniques: C2

Phase	Technique
Coding	Unit testing
Integration	Integration testing
System integration	System testing
Maintenance	Regression testing
Post system, pre-release	Beta-testing

Taken from Mathur (2014)

Classification of Software Testing Techniques: C3

Goal	Technique	Example
Advertised features	Functional testing	
Security	Security testing	
Invalid inputs	Robustness testing	
Vulnerabilities	Vulnerability testing	
Errors in GUI	GUI testing	Capture/plaback
		Event sequence graphs
		Complete Interaction Sequence
Operational correctness	Operational testing	Transactional-flow
Reliability assessment	Reliability testing	
Resistance to penetration	Penetration testing	
System performance	Performance testing	Stress testing
Customer acceptability	Acceptance testing	
Business compatibility	Compatibility testing	Interface testing
		Installation testing
Peripherals compatibility	Configuration testing	

Taken from Mathur (2014)

Classification of Software Testing Techniques: C4

Characteristics	Technique
Application component	Component testing
Client and server	Client-server testing
Compiler	Compiler testing
Design	Design testing
Code	Code testing
Database system	Transaction-flow testing
OO software	OO testing
Operating system	Operating system testing
Real-time software	Real-time testing
Requirements	Requirement testing
Software	Software testing
Web service	Web service testing

Taken from
Mathur (2014)

Software Testing Tools: Examples

EDITH COWAN UNIVERSITY

There are many software testing tools. Some examples given below:

Python Testing Tools

- Unittest, Pytest
- Check out the tools at: https://pythonhosted.org/testing/

Java Testing Tools

- Junit (http://junit.org)
- TestNG (http://testng.org/doc/index.html)

Defects Tracking

- Bugzilla (https://www.bugzilla.org/)