



2. Software Testing – Generalities I

Andrea Polini

Software Engineering II – Software Testing
MSc in Computer Science
University of Camerino

October 21st, 2014

Quality Dimensions

- **Static**
 - understand-ability
 - maintainability
 - structuredness
- **Dynamic**
 - reliability
 - correctness
 - completeness
 - consistency
 - usability
 - performance

Other dimension **internal** or **external**

Quality dimensions ask for the definition of **metrics**, and requirements should define specific **measures** to satisfy

Measure what is measurable, and make measurable what is not so

Quality Dimensions

- **Static**
 - understand-ability
 - maintainability
 - structuredness
- **Dynamic**
 - reliability
 - correctness
 - completeness
 - consistency
 - usability
 - performance

Other dimension **internal** or **external**

Quality dimensions ask for the definition of **metrics**, and requirements should define specific **measures** to satisfy

Measure what is measurable, and make measurable what is not so



Quality Dimensions

- **Static**
 - understand-ability
 - maintainability
 - structuredness
- **Dynamic**
 - reliability
 - correctness
 - completeness
 - consistency
 - usability
 - performance

Other dimension **internal** or **external**

Quality dimensions ask for the definition of **metrics**, and requirements should define specific **measures** to satisfy

Measure what is measurable, and make measurable what is not so

(Galileo Galilei)

Correctness

Correctness is just an ideal property, therefore it is more important to have a perception of **how likely is** that a software system will fail

Ex: The system should provide arithmetic operations for Natural numbers

```
public class math1{
    public double sum(double x, double y)
    { return x+y; }

    public double subtract(double x, double y)
    { return x-y;}

    public double abs(double x) {
        if (x>0) {return x;}
        else {return x;}
    }
    ....
}
```

```
public class math2{
    public int sum(int x, int y)
    { return x+y; }

    public int subtract(int x, int y)
    { return x-y;}

    public int abs(int x) {
        if (x>0) {return x;}
        else {return -x;}
    }
    ....
}
```

Reliability

ANSI/IEEE STD 729-1983: Reliability

Software reliability is the probability of failure free operation of software over a given time interval and under given conditions

- considers an operational profile

Reliability

Software reliability is the probability of failure free operation of software in its intended environment

Reliability

ANSI/IEEE STD 729-1983: Reliability

Software reliability is the probability of failure free operation of software over a given time interval and under given conditions

- considers an operational profile

Reliability

Software reliability is the probability of failure free operation of software in its intended environment

Operational profile

`sort`

Consider a `sort` program able to order input sequences both of strings and numbers (obviously not mixed).

Operational profile

An OP is a numerical description of how a program is used

- Different operational profile can be defined for `sort`

Requirements and testers

REQ 1

It is required to write a program that inputs two integers and outputs the maximum of these

REQ 2

It is required to write a program that inputs a sequence of integers and outputs the sorted version of this sequence

Definition of tests can help in clarifying requirements. Incompleteness of requirements can lead to ineffective testing activities.

Input domains should be covered to include **valid and invalid inputs**

Requirements and testers

REQ 1

It is required to write a program that inputs two integers and outputs the maximum of these

REQ 2

It is required to write a program that inputs a sequence of integers and outputs the sorted version of this sequence

Definition of tests can help in clarifying requirements. Incompleteness of requirements can lead to ineffective testing activities.

Input domains should be covered to include **valid and invalid inputs**

Some testing taxonomy

Test case

A **test case** is a pair consisting of test data to be input and expected output

Test set

A **test set** is a collection of zero or more test cases, generally homogeneous in terms of the functionality they stress

Test plan

A **test plan** is the definition of test requirements to be satisfied by the selection of a test set

Test selection

The following checks must be carried on to test the `sortAD` program (where the AD stays for Ascending/Descending and the program takes in input A or D to define which behaviour to perform)

- Execute the program on at least two input sequences, one with “A” and the other with “D”
- Execute the program on an empty input sequence
- Test the program for robustness against erroneous inputs such as “R” typed in as the request character
- All failures of the test program should be recorder in a suitable file using an appropriate form

Minimal test set?

Test selection

The following checks must be carried on to test the `sortAD` program (where the AD stays for Ascending/Descending and the program takes in input A or D to define which behaviour to perform)

- Execute the program on at least two input sequences, one with “A” and the other with “D”
- Execute the program on an empty input sequence
- Test the program for robustness against erroneous inputs such as “R” typed in as the request character
- All failures of the test program should be recorder in a suitable file using an appropriate form

Minimal test set?

Test execution

Test Execution

Test execution is the activity of performing the selected test case

At a first glance can seem an easy activity ...

- load tests
- bring the system in the right status for test execution
- record results
- check results

Test execution

Test Execution

Test execution is the activity of performing the selected test case

At a first glance can seem an easy activity ...

- load tests
- bring the system in the right status for test execution
- record results
- check results

The oracle problem



How can we assess the results provided by the system under test (SUT)?

This is the famous oracle problem

The oracle problem



How can we assess the results provided by the system under test (SUT)?

This is the famous **oracle problem**

Soundness vs. Completeness

- A test set is **sound** if all discovered faults are actually faults in the system (**no false positive**)
- A test set is **complete** if it can discover all faults but can generate **false positive**

Which kind of test set would you like to define?

Soundness vs. Completeness

- A test set is **sound** if all discovered faults are actually faults in the system (**no false positive**)
- A test set is **complete** if it can discover all faults but can generate **false positive**

Which kind of test set would you like to define?

Testability

Testability

Degree to which a system or component facilitates the establishment of test criteria and the performance of test to determine whether those criteria have been met

Depending on how you measure testability it can be classified as static or dynamic