2. Software Testing – General Concepts

Andrea Polini

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

ToC

1. Software Qualities
2. Test Activities and Taxonomy
3. Types of Testing

ToC

1 Software Qualities

2 Test Activities and Taxonomy

3 Types of Testing

SE and Software Qualities

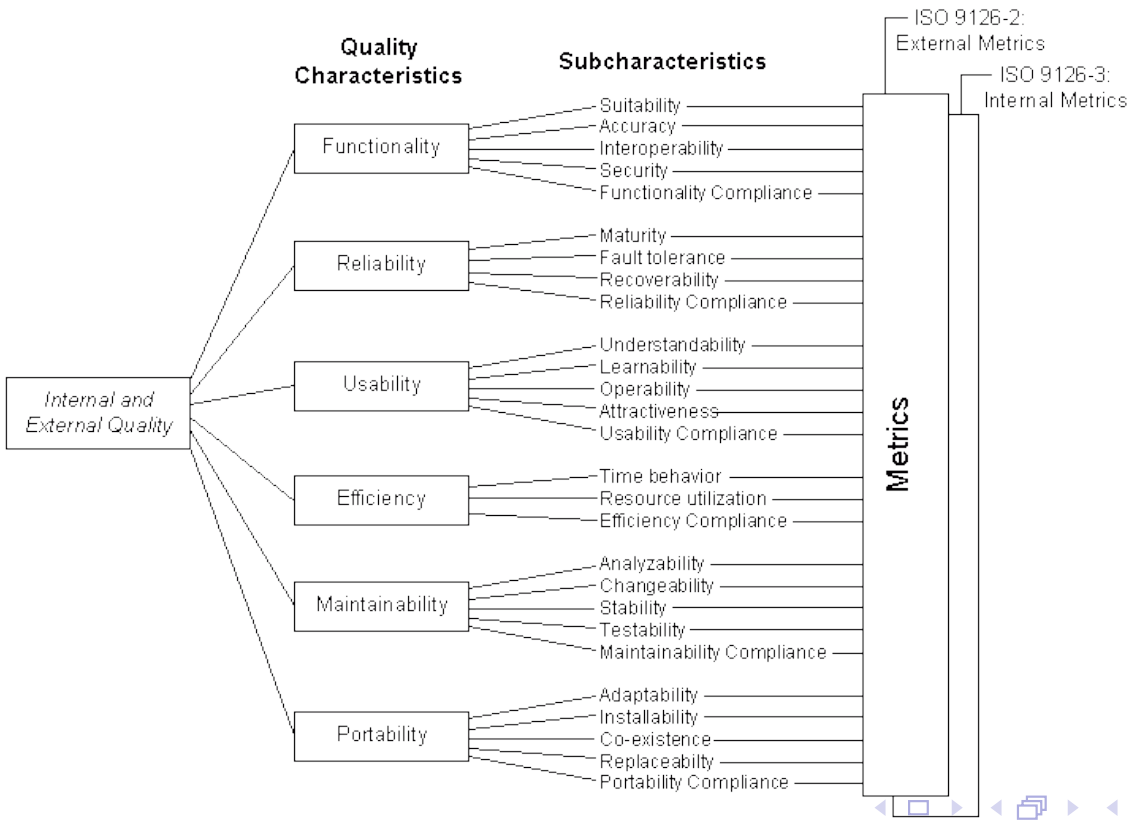
Software Engineering provides you with methodologies, techniques, approaches, and tools to build GOOD software

What does good stand for?

SE and Software Qualities

Software Engineering provides you with methodologies, techniques, approaches, and tools to build GOOD software

What does good stand for?



Software Quality

Software Quality

Degree to which a software . . .

1. conforms to specified requirements (functional and non-functional)
2. meets the needs and expectations of customers, users and stakeholders in general
3. is designed and developed according to sound engineering practices and standards

Metrics and Measures

To assess the degree to which a software conforms to a quality we need to define metrics and measurements procedures. A metric define a correspondence between entity and attributes of the real world with mathematical models and sets in order to better understand the real world itself. In order to make comparisons we need to consider sets with ordering relations.

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

(Galileo Galilei)

Quality Dimensions

Quality attributes can be classified according to other several dimensions:

- Static
  - understandability
  - maintainability
  - structuredness
- Dynamic
  - reliability
  - correctness
  - completeness
  - consistency
  - usability
  - performance

Once metrics have been defined for a given quality requirements will have to declare measures to satisfy

Organizational Metrics

The results of measurments on multiple projects can be considered as measurements of the organization on such an attribute

Correctness

Correctness

A program is considered correct if it behaves as expected on each element of its input domain

Correctness is just and ideal property, it asks for exhaustive testing, therefore it is more important to have a perception of how likely is that a software system will fail

Ex: The system should provide aritmetic 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 takes in input two integers and provides in output the maximum of the of the two

REQ 2

It is required to write a program that takes in input a sequence of integers and provide in output the sorted version of this sequence

Definition of tests can help in clarifying requirements. Incompletness of requirements can lead to ineffective testing activites.

Robustness

Input domains should be covered to include valid and invalid inputs

Requirements and testers

REQ 1

It is required to write a program that takes in input two integers and provides in output the maximum of the of the two

REQ 2

It is required to write a program that takes in input a sequence of integers and provide in output the sorted version of this sequence

Definition of tests can help in clarifying requirements. Incompletness of requirements can lead to ineffective testing activites.

Robustness

Input domains should be covered to include valid and invalid inputs

ToC

1 Software Qualities

2 Test Activities and Taxonomy

3 Types of Testing

Some testing taxonomy

Test case

A test case is a pair consisting of test data to be provided in input and exepected as output

Test set (aka test suite)

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 A/D 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 A/D 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 Harness

A test harness is a tool that helps the tester in performing one or more testing execution activities

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 Harness

A test harness is a tool that helps the tester in performing one or more testing execution activities

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 Harness

A test harness is a tool that helps the tester in performing one or more testing execution activities

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

The Oracle

Manually defined oracles:

- \- Costly
- \- Error Prone
- \+ More precise conditions

Automatically derived oracles:

- \- Difficult to implement
- \- Necessary conditions are generally checked (more false negative)
- \+ More reliable
- \+ Cheap

Logs of previous system usage can help in deriving oracles

The Oracle

Manually defined oracles:

- \- Costly
- \- Error Prone
- \+ More precise conditions

Automatically derived oracles:

- \- Difficult to implement
- \- Necessary conditions are generally checked (more false negative)
- \+ More reliable
- \+ Cheap

Logs of previous system usage can help in deriving oracles

Soundness vs. Completness

- 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. Completness

- 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 tests, to determine whether those criteria have been met
Related aspects are controllability and observability

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

ToC

1 Software Qualities

2 Test Activities and Taxonomy

3 Types of Testing

Test generation

Test generation

Test generation deals with the definition of strategies for the selection of appropriate data input and invocation sequences in order to form test sets satisfying given properties

Strategies can be defined for:

- Requirements
- FSM
- Statecharts
- PN
- Timed I/O Automata
- Algebraic and logic specifications
- Code (generally using monitored run-time data)

Type of testing

Testing can be classified in many different dimensions in some case orthogonal with respect to each other. A classification framework helps in clarifying concepts:

- Source of test generation
- Lyfe cycle phase in which testing takes place
- Goal of a specific testing activity
- Characteristics of the artefact under test
- Test process

Source of test generation

- Requirements > Black-box testing
- Code > White box testing
- Formal model > Model based testing (BB special case)
- Component interface > Interface testing (BB special case)

Lyfe cycle phase in which testing takes place

In the software production life-cycle different test are carried on with different objectives:

- Coding > Unit Testing
- Integration > Integration Testing
- System integration > System Testing
- Maintenance > Regression testing
- Pre-release > Beta testing

Goal of a specific testing activity

Goal oriented testing aims at showing specific properties for the system and then intends to show specific failures of the system

- Robustness
- Vulnerability
- Security
- GUI
- Stress
- Performance
- Acceptance
- Compatibility

Characteristics of the artefact under test

The focus here is on the characteristics of the artefact that is under test:

- OO testing
- Real-time testing
- Software testing
- Web service testing
- ...

Test process

In this case the focus is on the development process model and its relation to testing activities:

- Testing in the waterfall model
- Testing in the V-Model
- Spiral testing
- Agile testing
- Test driven development
- ...

The saturation effect

Confidence vs. Reliability

- Confidence is a subjective assessment of the quality of the software with respect to its “correctness”
- Reliability should be an objective assessment of the quality of the software with respect to its “correctness”

The saturation effect warns testers onf the efficacy of test generation strategies and suggest to apply more than one strategy