



# 1. Introduction

Andrea Polini

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

## WARNING

Slides are distributed to help students in their preparation to the exam. In **no way** they intend to substitute text books. Instead a **thorough study of the text books** constitutes the **most wise strategy** to maximize the chances to pass the final exam.

- 1 General Information
- 2 Introduction to Software Testing

# ToC

- 1 General Information
- 2 Introduction to Software Testing

# Course and Teacher

- Software Engineering II – Software Testing
  - Lessons:
    - Tuesday from 3pm to 5pm
    - Wednesday from 9am to 11am
  - web: [http://didattica.cs.unicam.it/...](http://didattica.cs.unicam.it/)
- Andrea Polini
  - e-mail: [andrea.polini@unicam.it](mailto:andrea.polini@unicam.it)
  - web: <http://www.cs.unicam.it/polini>
  - weekly office hours: after Wednesday lesson
- Exam dates:
  - June 15th and July 6th, 2016
  - September 7th and 28th, 2016
  - February 8th and 22nd, 2017

# Course Objectives

- The course permits to the student to acquire the knowledge needed to understand software testing issues and solutions. The course then aims at permitting the development of competences needed to operate in real scenarios in order to test complex software systems.

# Study material

- **Reference book:**



Aditya P. Mathur

*Foundations of Software Testing, 2<sup>nd</sup> Ed.*

Pearson, 2014.

- **Further references provided by the teacher**

# Exam

- 1 **Short survey presentation:** One or two scientific papers will be assigned to each student. A short survey on the topic will be performed, and a presentation will be given by the student in one of the last lessons. A precise plan of the presentation will be provided.
- 2 **Small Software Project** – students (it is possible to work in groups of 2) will have to choose an open source software system (possibly one made by them), and provide a complete test plan with artifacts. A short report have to be delivered before the oral paper. Selection has to be submitted to my evaluation. In case a group is not able to select a software system will have to notify the teacher by mid of April.
- 3 **Oral paper** – date fixed for the exam

# ToC

- 1 General Information
- 2 Introduction to Software Testing**



# Before starting

## SE – Some definition

- IEEE:  
Adoption of a **systematic, disciplined and quantifiable** approach to the the development, support and maintenance of software systems
- Sommerville:  
Software Engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification to maintaining the system after it has gone to use. Software Engineers apply theories, methods and tools where these are appropriate, but they use them selectively and always try to discover solutions to problems even when there are no applicable theories and methods. Engineers also recognize that they must work to organizational and financial constraints

# Before starting

## SE – Some definition

- IEEE:  
Adoption of a **systematic, disciplined and quantifiable** approach to the the development, support and maintenance of software systems
- Sommerville:  
Software Engineering is an engineering discipline that is concerned with **all aspects of software production** from the early stages of system specification to maintaining the system after it has gone to use. Software Engineers apply **theories, methods and tools** where these are **appropriate**, but they use them selectively and always try to discover solutions to problems even when there are no applicable theories and methods. Engineers also recognize that they must work to **organizational and financial constraints**

# SE – Some definition

... continued

- Ghezzi, Jazayeri, Mandrioli:  
Software Engineering concerns the development of software systems which given **their size** require the involvement of a **development team**

# Focus on

- Software development methodologies for **big size software systems**
- SE recommends **disciplined and systematic** methods
- SE recommends the introduction of **quantifiable methods in order to have the possibility to compare different solutions**
- **Development team – Communication** is one of the most complex aspect to manage.

# Software Processes – How

- Activity perspective
- Workflow perspective
- Data-flow perspective
- Role/action perspective

# Software Engineering Activities

- Requirements Elicitation and Analysis
- Design
- Implementation
- **Verification and Validation**
  - Static strategies
  - Dynamic strategies

# Software Engineering Activities

- Requirements Elicitation and Analysis
- Design
- Implementation
- Verification and Validation
  - Static strategies
  - Dynamic strategies

# Testing intuition

Software testing concerns the execution of **some** “experiments” in a **controlled environment** in order to acquire **enough confidence** on the behaviour of a software system when deployed in the real environment. Software Testing can equally aim at assessing **functional properties** and **extra-functional properties** (some of them at least)

Two different objectives and “moods”:

- Try to demonstrate that the system correctly satisfy the specifications, and the needs of users and customers
- Try to discover bugs in the code

*Testing can never guarantee the absence of fault but just their existence*

*E.W. Dijkstra*



# Testing intuition

Software testing concerns the execution of **some** “experiments” in a **controlled environment** in order to acquire **enough confidence** on the behaviour of a software system when deployed in the real environment. Software Testing can equally aim at assessing **functional properties** and **extra-functional properties** (some of them at least)

Two different objectives and “moods”:

- Try to demonstrate that the system correctly satisfy the specifications, and the needs of users and customers
- Try to discover bugs in the code

*Testing can never guarantee the absence of fault but just their existence*

*E.W. Dijkstra*

# Testing vs. Debugging

Clearly strictly related but different objectives!

- **Testing**: find bugs and show possible divergences between what is observed and what it is expected
- **Debugging**: remove bugs and align the characteristics of the system to what is expected

# Testing vs. Debugging

Clearly strictly related but different objectives!

- **Testing**: find bugs and show possible divergences between what is observed and what it is expected
- **Debugging**: remove bugs and align the characteristics of the system to what is expected

# Genesis of failures

**Error:** *an error occurs in the process of writing a program (or document)*

**Fault** – aka bug, defect: *a fault is the manifestation of one or more errors and is constituted by a piece of code that do not correspond to what is needed*

**Failure:** *a failure is the observation of a behaviour that does not correspond to the desired one*

# Genesis of failures

**Error:** *an error occurs in the process of writing a program (or document)*

**Fault** – aka bug, defect: *a fault is the manifestation of one or more errors and is constituted by a piece of code that do not correspond to what is needed*

**Failure:** *a failure is the observation of a behaviour that does not correspond to the desired one*

# Genesis of failures

**Error:** *an error occurs in the process of writing a program (or document)*

**Fault** – aka bug, defect: *a fault is the manifestation of one or more errors and is constituted by a piece of code that do not correspond to what is needed*

**Failure:** *a failure is the observation of a behaviour that does not correspond to the desired one*

# How and when errors can occur?