1. Introduction

Andrea Polini

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

October 16th, 2014

Course and Teacher

- Software Engineering II – Software Testing
  - Lessons: Tuesday and Thursday from 11am to 1pm
  - web: http://didattica.cs.unicam.it/doku.php?id=softengii:softengii2014
- Andrea Polini
  - e-mail: andrea.polini@unicam.it
  - web: http://www.cs.unicam.it/polini
  - Weekly student reception: Thursday from 2pm to 3pm
- Exam dates:
  - February 4th and 25th, 2015
  - June 10th and July 8th, 2015
  - September 9th and 30th, 2015
  - February 3rd and 24th, 2016

Exam

Course Objectives

- At the end of the course the students should have learned different software testing strategies. Using this knowledge they should be able to apply the different techniques in real case scenarios selecting each time the most suitable one.

Contents

- Introduction and Basic Concepts
- The testing process
  - Test selection
  - Test execution
- Testing strategies
  - Black-box strategies
    - Behavioural Models
    - Data Models
  - White box strategies
    - Structural Testing
    - Data Flow Testing
- Testing tools

SE – Some definition

- IEEE:
  Adoption of a sistematic, 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 sapplicable theories and methods. Engineers also recognize that they must work to organizational and financial constraints

SE – Some definition

- IEEE:
  Adoption of a sistematic, 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 sapplicable 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 involvment of a development team

Focus on

- Software development methodologies for **big size software systems**
- SE recommends **disciplined and systematized**
- 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

Verification and Validation

- Static strategies
- Dynamic strategies

Testing

Software testing concers the run of some “experiments” in a controlled environment in order to acquire enough confidence on the behaviour of a software system. Software Testing typically concerns functional properties but it can also refer to extra-functional properties

Two different objectives and “moods”:

- Try to demonstrate that the system correctly satisfy the needs of the user
- Discover the bugs

> *Testing can never guarantee the absence of fault but just their existence*
>
> *E.W. Dijkstra*

Testing

Software testing concers the run of some “experiments” in a controlled environment in order to acquire enough confidence on the behaviour of a software system. Software Testing typically concerns functional properties but it can also refer to extra-functional properties

Two different objectives and “moods”:

- Try to demonstrate that the system correctly satisfy the needs of the user
- Discover the bugs

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

E.W. Dijkstra

Testing

- Testing Strategies
- Testing techniques

Testing is not debugging

Genesis of failures

Error

Fault

Failure