TVVS - Teste, Verificação e Validação de Software (*Software Testing, Verification and Validation*)

Ana Paiva

apaiva@fe.up.pt

//web.fe.up.pt/~apaiva

José Campos

jose.carlos.campos@fe.up.pt

//jose.github.io

Objectives

The objective of this curricular unit is:

- familiarization with the terminology used in software testing; raising awareness of issues related to software quality; and exploring and practicing different Verification and Validation (V&V) techniques necessary for building quality software systems.

Learning Outcomes and Competences

- At the end of the course, students should be able to design and execute a Verification and Validation (V&V) plan. More specifically, students are expected to be able to:
  - Plan a Verification and Validation strategy.
  - Select the best software testing techniques and tools for a given context.
  - Design and develop tests at different levels (e.g., unit, integration, system and acceptance).
  - Test exceptional situations (e.g., boundary value analysis).
  - Reflect on the limitations and quality of the tests developed.
  - Use metrics to assess test quality (e.g., instruction coverage).
  - Write maintainable test code, avoiding known issues (e.g., flakiness, unreadable, dependent, fat testing, etc.).

Program

- Introduction to Software Verification and Software Validation
- Static Testing
- Equivalence Class Partitioning / Category Partition
- Boundary Value Analysis
- Model-based Testing
- Structural Testing (Line and Decision coverage)
- Structural Testing (Path coverage) and Logical Coverage (Condition coverage)
- Logical Coverage (Modified Condition/Decision Coverage (MC/DC))
  Mutation Testing
- Dataflow Testing
- Integration Testing, System Testing, Acceptance Testing, and Regression Testing
- Test management and documentation

Literature

- Mandatory
  - Aditya P. Mathur; Foundations of Software Testing, 2013. ISBN: 9789332517660
  - Maurício Aniche; Effective software testing, 2022. ISBN: 978-1-633-43993-1
  - Paul Ammann; Introduction to software testing. ISBN: 978-1-107-17201-2
  - Mark Utting; Practical Model-Based Testing, 2007. ISBN: 978-0-12-372501-1
  - Paul C. Jorgensen; Software Testing A Craftsman's Approach, 2013. ISBN: 978-1-466-56069-7
- Complementary
  - Dorothy Graham, Rex Black, Erik van Veenendaal; Foundations of Software Testing: ISTQB Certification, 2020. ISBN: 978-1-473-76479-8

Teaching methods and learning activities

- **Theoretical classes** are used to formally explain the concepts and fundamental aspects of V&V. Whenever possible, theoretical concepts are accompanied by presenting and discussing real practical cases.

  [Friday]

- **Theoretical-practical classes** are used to carry out small exercises, apply V&V techniques, use/experiment with various testing tools and support implementing the practical project.

  [Wednesday]

Grades

Eligibility for exams

- To complete this curricular unit, students must obtain:
  - minimum grade of 47.5% in the **practical exercises (C)**
  - minimum grade of 47.5% in the **project (P)**
  - minimum grade of 47.5% in the **final exam (E)**

- Calculation formula of the final grade

C x 10% + P x 40% + E x 50%

Grades

Practical exercises (C) (10%)

- Small exercises that should be performed in class and delivered at the end of the class

Project (P) (40%)

- An open-source project to be tested by applying different techniques/tools
- You will get more details around the middle of October

* [Both (C and P) to be performed individually]

Grades

- Special assessment (TE, DA, …)
  - The practical work is mandatory for all students
  - If not attending the practical class, must submit the work by the end of the class day
- Improve grades
  - The grades obtained in practical work/projects can be improved in the next edition of the course.
  - The classification of the final exam can be improved in the appeal exam.