

Overview:

Organic Unit	Escola Superior de Tecnologia e Gestão	Academic Year	2018/2019					
Study Cycle	Master in Computer Engineering - Mobile Computing (D) [M238]	Degree	Mestrado					
Curricular Year	1	Period	S2					
Curricular Unit	Software Quality	ECTS	6					
Scientific Area	Computer Engineering	Type	Opcional					
Work Load	161.6	T 0	TP 22.5	PL 30	TC 0	S 0	E 0	OT 0
<i>T - Theoretical; TP - Theoretical and Practical; PL - Practical and Laboratorial; TC - Field Work; S - Seminar; E - Trining; OT - Tutorial</i>								

Responsible academic staff member:

Ricardo Filipe Gonçalves Martinho (17,50 horas semanais de contacto: PL: 10,00; TP: 7,50;)

Catarina Isabel Ferreira Viveiros Tavares Reis (17,50 horas semanais de contacto: PL: 10,00; TP: 7,50;)

José Carlos Bregjeiro Ribeiro (17,50 horas semanais de contacto: PL: 10,00; TP: 7,50;)

Prerequisites:

None.

Language:

Inglês

Outline:

This course allows students to deepen their knowledge in Software Quality Assurance (SQA), with regard to the various models and features most widely used nowadays by the industry. Standards, norms and certifications are fundamental to ensure quality, but the existing pressure of the "time-to-market" of a software product can jeopardize its quality. Thus, this course provides certification contents and introduces deployment pipelines that support continuous integration and delivery. All the concepts of one of the most recognized certifications in the area of Software Quality Assurance - the initial level of the ISTQB (Foundation Level) are provided, setting the basis for students to take the certification exam. The principles and technical practices that promote a fast and incremental delivery to the market, while ensuring high quality standards are presented. Through the automation of the build, deployment and testing processes, and an improved collaboration within the team, it is possible to establish an entire deployment pipeline from requirements to release. The key issues, state-of-the-art techniques and software tools, as well as best practices provided in the course demonstrate how to mitigate risks, while delivering software products and value to business in a fast and reliable way.

Learning outcomes:

C1: Decide on the main issues and tradeoffs involving continuous delivery;

C2: Define a software testing strategy aligned with the company's business strategy;

C3: Implement a configuration management environment to improve team communication and integrate the testing and continuous delivery strategies;

C4: Acknowledge and apply the main testing techniques throughout the software lifecycle;

C5: Acknowledge main techniques and tools for automated software testing;

C6: Implement a full continuous delivery pipeline using state-of-the-art software tools and techniques for each stage of the pipeline. C7: Design, implement, deploy and monitor a software architecture backed by quality attributes (specifically non-functional requirements).

Syllabus:

Syllabus:

1. Software Testing
 - 1.1. Fundamentals of Testing
 - 1.2. Testing Throughout the Software Lifecycle
 - 1.3. Static Techniques
 - 1.4. Test Design Techniques
 - 1.5. Test Management & Tool Support for Testing
2. The Deployment Pipeline that supports Continuous Integration
 - 2.1. Foundations of Software Delivery
 - 2.2. Defining a Deployment Pipeline
 - 2.3. The Commit Stage
 - 2.4. Automated Acceptance Testing
 - 2.5. Deploying and Releasing Applications
3. Quality Attributes in Software Architecture
 - 3.1. Understanding Quality Attributes
 - 3.2. Architecture and Quality
 - 3.3. Containerisation

Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

1. Software Testing
 - 1.1. Fundamentals of Testing (C1)
 - 1.2. Testing Throughout the Software Lifecycle (C2)
 - 1.3. Static Techniques (C4)
 - 1.4. Test Design Techniques (C4)
 - 1.5. Test Management & Tool Support for Testing (C5)
2. The Deployment Pipeline that supports Continuous Integration
 - 2.1. Foundations of Software Delivery (C1)
 - 2.2. Defining a Deployment Pipeline (C2, C3, C6)
 - 2.3. The Commit Stage (C3, C6)
 - 2.4. Automated Acceptance Testing (C4, C6)
 - 2.5. Deploying and Releasing Applications (C6)
3. Quality Attributes in Software Architecture
 - 3.1. Understanding Quality Attributes (C1, C2)
 - 3.2. Architecture and Quality (C4, C5, C7)
 - 3.3. Containerisation (C4, C6, C7)

Teaching/Learning methodologies:

In Class:

PT = Presential Teaching

PT1. Theoretical classes - presenting and discussing the course syllabus items;

PT2. Laboratory classes - implementing a continuous integration and delivery environment, dealing with automatic testing tools and monitoring containerised and orchestrated environments;

PT3. Tutor sessions, done in small workgroups to drive the learning process and problem solving.

Autonomous:

AL1. Study
AL1.1 Bibliography reading;
AL1.2 Problem solving regarding exercises recommended;
AL1.3 Workgroup project development;
AL2. E-learning - Reading of the materials made available.

Specific Resources:

N/A.

Evaluation:

Description:

PA = Periodic assessment:

PA1. 70% - workgroup assignment project, with minimum of 9,5 out of 20; Project discussion;

PA2. 30% - written test, with minimum of 9,5 out of 20;

$PA = 70\% * PA1 + 30\% * PA2$

FANE = Final Assessment Normal Exam:

FANE1. 70% - workgroup assignment project, with minimum of 9,5 out of 20; Project discussion;

FANE2. 30% - written test, with minimum of 9,5 out of 20;

$FANE = 70\% * FANE1 + 30\% * FANE2$

FA = Final Assessment (other exams):

FA1. 70% - computer practical test, with minimum of 9,5 out of 20.

FA2. 30% - written test, with minimum of 9,5 out of 20;

$FA = 70\% * FA1 + 30\% * FA.2$

NOTE:

- classifications greater or equal than the minimums required for each theoretical or practical components can be used in any assessment period within the same school year.

- improvement of grades can only be made by realising both theoretic and practical exams of the same assessment period.

Number of elements in the final evaluation:

2

Number of elements for continuous evaluation:

2

Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

PT = Presential Teaching

PT1. Theorics - presenting and discussing the course syllabus items; (C1 to C7)

PT2. Laboratory classes - implementing a continuous integration and delivery environment, dealing with automatic testing tools and monitoring containerised and orchestrated environments; (C2, C3, C5, C6 e C7)

PT3. Tutor sessions, done in small workgroups to drive the learning process and problem solving. (C1 to C7)

AL = Autonomous Learning

AL1. Study

AL1.1 Bibliography reading; (C1, C2, C4, C6)

AL1.2 Problem solving regarding exercises recommended; (C1 to C7)

AL1.3 Workgroup project development; (C1, C6 and C7)

AL2. E-learning - Reading of the materials made available (C1 to C7)

Bibliography:

Main Bibliography:

- Course slides;
- The Certified Tester Foundation Level in Software Testing - Foundation Level Syllabus (2018) - <http://www.istqb.org/downloads/syllabi/foundation-level-syllabus.html>
- Continuous Delivery: Reliable Software Delivery (2010), Jez Humble, David Farley, Addison-Wesley Professional;
- Continuous integration: improving software quality and reducing risk (2007), Duvall, P. M., Matyas, S., & Glover, A., Addison-Wesley Professional;
- Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice (Third Edit., p. 624).
- Martin, R. C. (2017). Clean architecture: a craftsman's guide to software structure and design. Prentice Hall Press.

Additional Bibliography:

- Guide to the Software Engineering Body of Knowledge - SWEBOK v3.0 (2014), edited by Pierre Bourque and Richard E. Fairley.
- Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley Professional.
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook:: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution.
- Newman, S. (2015). Building microservices: designing fine-grained systems. " O'Reilly Media, Inc.".
- Poulton, N. (2018). The Kubernetes Book (3rd Edition).