

[illegible]

Goal

- To make a graduate with a STEM background to be able to:
 - Understand and apply quality assurance in an incremental/agile development process
 - Analyse the requirements of a problem presented by a client and, as a team, specify a quality assurance plan
 - Collaborate with the development team in the design of the solution using appropriate architectural and design patterns that promote a testable architecture
 - Implement the tests in JavaScript/Typescript using Behaviour Driven Development (BDD) and Continuous Deployment (CD)
 - Analyse the results and apply quality control and improvement techniques

The QACT approach

- The programme aims to fulfil this objective
 - A careful selection of the candidates
 - 1 school year (32 weeks of classes)
 - An innovative pedagogical approach based on project based learning
 - A barebones approach, focusing on the minimum content necessary for the students to be successful in the typical professional environment
 - A semi-professional work environment based on teamwork, peer-learning and Scrum
 - Focused and intensive: 25 hours/week evening course



Pedagogical approach

- We call it CDIO-IL and it is a combination of
 - Disciplinary teaching (35%)
 - Lectures and regular lab classes
 - Project Based Learning (65%)
 - Project development in a team of 8-10 elements
 - Scrum software development process (usually 3-weeks sprints)
 - Project drives the learning process
 - Global backlog of user stories and sprints backlogs adapted to the learning process' needs
 - Involvement of real software companies as product owners



Pedagogical approach

- How can it be so fast and effective?
 - By the careful application of selected active learning pedagogical patterns and project based learning
 - Learn by example
 - Providing solutions upfront help the student scaffolding knowledge
 - Learn by doing
 - All work is to be produced in the context of the project, including in regular classes
 - The immediate application of new knowledge in a real application helps the students cement the knowledge and learning more effectively

Pedagogical approach

- How can it be so fast and effective?
 - Teamwork and Peer-learning
 - Provides a first support network and helps the student consolidate learning
 - Feedback
 - The teacher's main role is to provide quality feedback, promoting rework and improvement
 - Focus on quality of work done
 - Grades reflect quality (below standards, met the standards, noteworthy)
 - Deliveries that don't meet the requirements and minimum quality standards are not accepted



Programme structure

- Two 16-week semesters
- Up to 32 students, 8-10 elements/group
- 25 hours/week workload
 - 12 hours/week of regular classes
 - 13+ hours/week of autonomous work in a scrum team
- 3 simultaneous courses each semester
 - 1 software engineering and testing course
 - 1 technical course covering key technologies and competencies
 - 1 project course to apply testing in a enterprise-like context

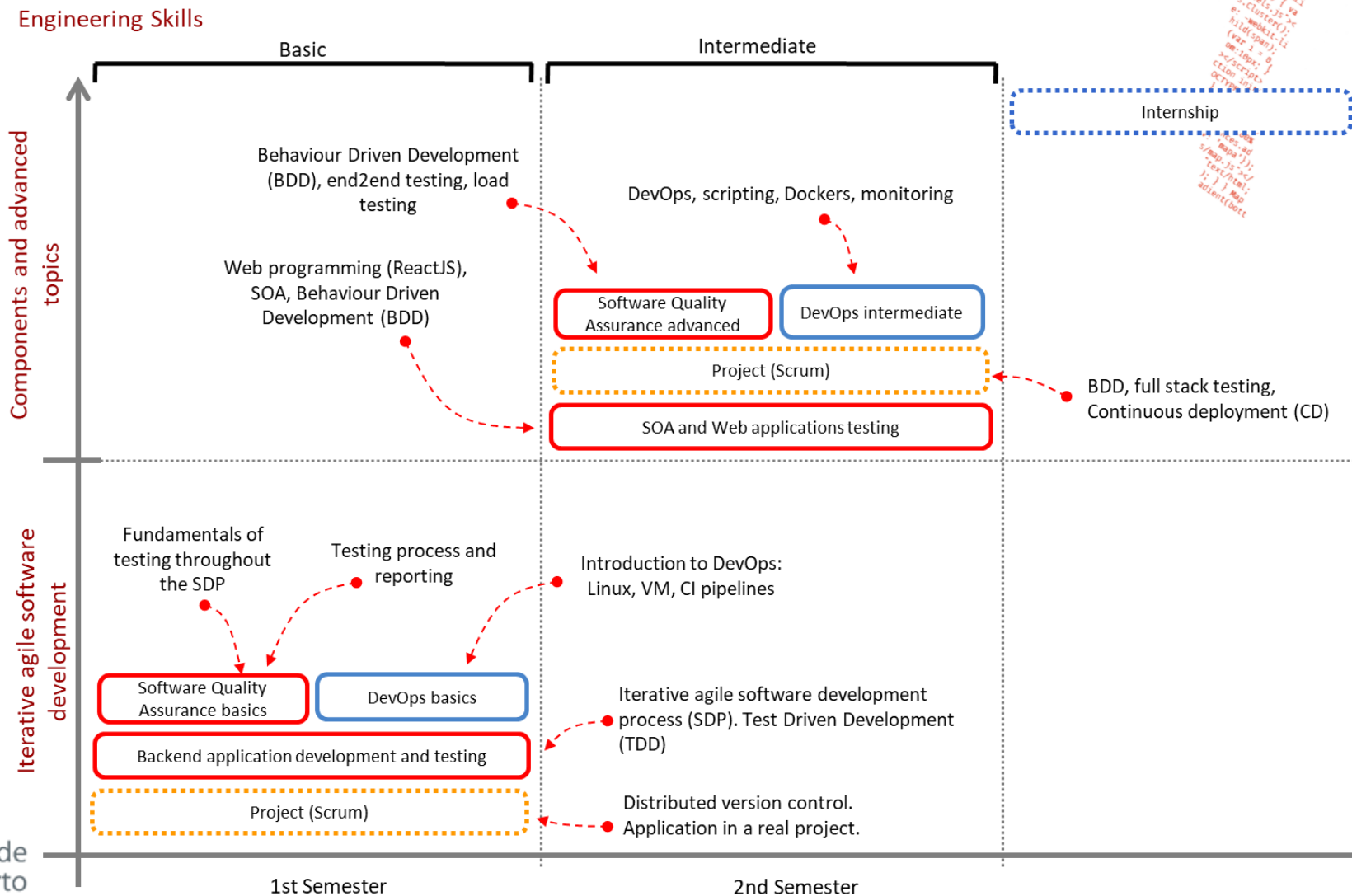


[illegible]

- 2nd semester

- Software Development and Testing II
- Software Quality Assurance II
- DevOps II
- Project II

Programme structure



Course Information



Software Quality Assurance I

■ Main goals

- Understand the concepts of quality assurance and testing
- Understand the testing process and know the tests to apply in every phase of the software development process
- Understand the test-driven development approach
- Design a test plan for system according to requirements
- To compute test results and elaborate a test report

• References

- <https://www.istqb.org/certifications/certified-tester-foundation-level>
- <https://www.istqb.org/certifications/test-manager>
- <https://www.istqb.org/certifications/test-analyst>

Software Development and Testing I

■ Main goals

- Understand the need for the existence of a software development process (SDP) and the dimensions and stakeholders of the SDP
- Apply an iterative and incremental agile SDP, using appropriate artefacts and notations for describing the problem, the analysis, the design and the solution
- Apply methodologies and principles of OO analysis and design, including simple OO design principles, patterns, and secure by design principles
- Apply a Test Driven Development approach throughout the software development life cycle
- Apply data persistence

DevOps I

■ Main goals

- Understand the fundamentals of computer systems administration and security
- Be able to install and manage Linux systems
- Apply distributed version control management
- Use virtual machines locally and in virtualization infrastructures
- Implement continuous integration servers and pipelines

Project I

■ Main goals

- Work in a team, in the context of a software development project, applying Scrum
- Use the software development and teamwork tools and approaches that are appropriate for each moment and context
- Apply methodologies of OO analysis and design, including simple OO design principles and patterns
- Apply the appropriate methodologies and tools to implement and test the solution
- Understand essential documentation and presentation best practices and apply them to communicate the results of the software project

Technologies 1st semester

■ Software Development Process

- UML, Software design patterns
- Scrum
- DVCS

■ Implementation

- JavaScript/Typescript
- NodeJS
- DVCS (e.g. Git)
- Databases (e.g. MongoDB)
- Unit testing frameworks (e.g. Mocha, Jest)
- Virtualization (e.g. Vagrant, VirtualBox)
- Continuous Integration Server (e.g. Jenkins)

Software Quality Assurance II

■ Main goals

- Understand the Behaviour-Driven Development approach as a combination of TDD, DDD and object-oriented analysis and design
- Understand end-to-end testing
- Understand the fundamentals of load testing
- Design a test plan for system according to requirements by applying BDD
- References
 - <https://www.istqb.org/certifications/test-analyst>
 - <https://www.istqb.org/certifications/technical-test-analyst>

Software Development and Testing II

- Main goals
 - Analyse the requirements and concepts of a software system, especially Distributed and Decentralized Software Systems (DDSS)
 - Understand the design of a DDSS based on industry standard design and architecture styles and patterns
 - Design a test plan and the tests for a DDSS using appropriate test design patterns and best practices
 - Apply a Behaviour Driven Development approach throughout the software development life cycle

DevOps II

■ Main goals

- Understand the fundamentals of DevOps
- Use appropriate DevOps/scripting tools to automate the deployment of applications and the infrastructure provisioning
- Use the different components of a continuous delivery pipeline
- Use containers to package and deploy applications
- Apply specific monitoring best practices and tools



Project II

■ Main goals

- Work with a team, in the context of a software development project, applying Scrum and continuous deployment (CD)
- Apply industry standard methodologies and architectural patterns for DDSS analysis and design
- Apply DevOps methodologies and tools to implement, test and deploy a full stack web application
- Understand essential documentation and QA reporting best practices and apply them to communicate the results of the software project

Technologies 2nd semester

- Software Development Process
 - UML, DDSS software design and architectural patterns
 - Scrum
 - DVCS
- Implementation
 - JavaScript/Typescript
 - NodeJS, ReactJS
 - DVCS (e.g. Git)
 - Databases (e.g. MongoDB)
 - Testing frameworks and tools (e.g. Cypress, Cucumber, JMeter)
 - Virtualization (e.g. Vagrant, VirtualBox, Docker)
 - Automation and Provisioning (e.g. Ansible)
 - Monitoring (e.g. Prometheus)





Join us in shaping the future.

April-2022
geral@portotechhub.com