



Module Specification

Software Engineering [TSI]

Version: 2023-24, v2.0, 09 Aug 2023

Contents

| | |
|--|----------|
| Module Specification | 1 |
| Part 1: Information | 2 |
| Part 2: Description | 2 |
| Part 3: Teaching and learning methods | 3 |
| Part 4: Assessment..... | 5 |
| Part 5: Contributes towards | 6 |

Part 1: Information

Module title: Software Engineering [TSI]

Module code: UFCE68-24-2

Level: Level 5

For implementation from: 2023-24

UWE credit rating: 24

ECTS credit rating: 12

College: College of Arts, Technology and Environment

School: CATE School of Computing and Creative Technologies

Partner institutions: Transport and Telecommunication Institute

Field: Computer Science and Creative Technologies

Module type: Module

Pre-requisites: None

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: Not applicable

Features: Not applicable

Educational aims: Learning the principles and techniques used in various stages of complex software development and training in the organisation of software development processes.

Outline syllabus: •Models of software development process.

- Project management and project metrics.
- Estimation during the project management.
- Basics of software design.
- The basics of object-oriented view of software systems.
- Unified Modelling Language (UML).
- Formation of the requirements model for the software product.
- Behaviour diagrams of program system.
- Architectural design basics.
- Detailed design.
- Fundamentals of component object model.
- The development process of the object-oriented software system.
- Assessment of the object-oriented software quality.
- Design concepts.
- Architectural design – recommended approach.
- Component-level design.
- Design for mobility.
- Pattern-based design.
- Cloud architecture and development platforms.
- Basic concepts of testing.
- Methods of structural testing.
- Methods of functional testing.
- The organization of the software testing process.
- Features of object-oriented testing.
- Test automation.
- Software Quality management.

Part 3: Teaching and learning methods

Teaching and learning methods: Learning and teaching will be provided to students in two forms: lectures and labs. During lectures, theoretical aspects of the course will be provided to students by the teaching staff. Lectures will be supported by presentation published and available to the students on e.tsi.lv under the module

section. Also, additional materials, like code examples, text books, publications on the internet, videos etc will be presented in e.tsi.lv.

During labs, each student receives an individual task to perform.

In addition to learning activities during the face-to-face lessons, students must spend time outside the classroom on independent learning activities. These might include completing assignment tasks, independent reading, practising new skills on personal projects, watching informative videos, completing self-assessment test etc.

Module Learning outcomes: On successful completion of this module students will achieve the following learning outcomes.

MO1 Understand and apply principles and methods throughout the software development lifecycle.

MO2 Effectively plan, lead, evaluate quality, and provide support for software projects.

MO3 Understand the software development processes, use established scientific and methodological foundations to create robust and innovative approach to software solutions.

MO4 Understand the capacities and apply modern tools for automated software development.

Hours to be allocated: 240

Contact hours:

Independent study/self-guided study = 144 hours

Face-to-face learning = 102 hours

Total = 246

Reading list: The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://rl.talis.com/3/uwe/lists/116D8907-B765-6A8F-C0D5-EDA1551C2740.html?lang=en-gb&login=1) via the following link <https://rl.talis.com/3/uwe/lists/116D8907-B765-6A8F-C0D5-EDA1551C2740.html?lang=en-gb&login=1>

Part 4: Assessment

Assessment strategy: To assess the learning outcomes of this course, several types of activities are provided, which include:

1) Written assignment - Students are required to produce code for a given scenario, they will evaluate against the code they have produced against code which has been produced using automated tools and discuss the differences.

2) Examination (2h)

Resits will be like for like; however, using a different scenario for the written assignment.

Assessment tasks:

Written Assignment (First Sit)

Description: The project dedicated to examine and discuss two solutions, differences and efficiency of the software prepared by students and one by automated tools.

Weighting: 50 %

Final assessment: No

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4

Examination (First Sit)

Description: Closed booked examination (2 hours)

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO3

Written Assignment (Resit)

Description: The project dedicated to examine and discuss two solutions, differences and efficiency of the software prepared by students and one by automated tools.

Weighting: 50 %

Final assessment: No

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4

Examination (Resit)

Description: Closed booked examination (2 hours)

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO3

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Computer Science and Software Development {Double Degree} [Feb][FT][TSI][4yrs]
BSc (Hons) 2021-22

Computer Science and Software Development {Double Degree} [Oct][FT][TSI][4yrs]
BSc (Hons) 2021-22

Computer Science and Software Development {Double Degree} [Oct][PT][TSI][5yrs]
BSc (Hons) 2020-21

Computer Science and Software Development {Double Degree} [Feb][PT][TSI][5yrs]
BSc (Hons) 2020-21