Module Specification

Collaborative Software Development Project

Version: 2023-24, v2.0, 12 Jul 2023

Contents

Part 1: Information

Module title: Collaborative Software Development Project

Module code: UFCFCR-30-3

Level: Level 6

For implementation from: 2023-24

UWE credit rating: 30

ECTS credit rating: 15

Faculty: Faculty of Environment & Technology

Department: FET Dept of Computer Sci & Creative Tech

Partner institutions: None

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: The primary role of a software engineer is to be able to design, build and test high-quality software solutions following best practices and industry standards. They will typically be working as part of a larger collaborative team, in which they will have responsibility for significant elements of the overall project. The developer will need to be able to interpret requirements, specification documentation and designs in order to develop and test software that meets its requirements, even when these

requirements may change. Using a collaborative environment allows developers to bring together independent modules together to create a complete solution

Features: Not applicable

Educational aims: Undertake the roles and responsibilities of a software engineer

Collaborate as a team

Interpret and implement a software design

Develop a software solution collaboratively

Create effective and secure software solutions

Participate in code reviews, debugging and refactoring processes

Thoroughly test a solution

Outline syllabus: The roles and responsibilities that are required from a software engineer at every stage of the development lifecycle

Collaboration as a team to apply systems analysis and design to a project specification creating artefacts
(e.g. use case)

Collaboratively creating a program based on user requirements, embracing an industry based methodology

Industry standard build processes, and tools for configuration management, version control and software build (e.g. GitHub, Bitbucket, Tortoise SVN), release and deployment

Contemporary software development languages

Quality metrics

Code reviews, debugging and refactoring processes to improve code quality and efficiency

Testing to ensure resilience of code and that it meets the functional and non-functional requirements (e.g. black box, white box, unit testing)

Part 3: Teaching and learning methods

Teaching and learning methods: Introductory lectures covering the fundamentals and technical underpinning of the module for the first assessment before progressing onto practical delivery through a series of lessons, workshops and practical tasks in the classroom to develop the tools and techniques required to complete the practical assessment for this module. Students are also provided with access to a suitable collaborative software development tools to aid the completion of this module.

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

MO1 Identify, justify and review all stages of the software development lifecycle and teamwork for a collaborative software development project.

MO2 Analyse, interpret and implement a design compliant with functional and nonfunctional requirements

MO3 Perform code reviews, debugging and refactoring to improve code quality and efficiency

MO4 Select appropriate testing techniques to ensure that functional and non-functional requirements are met.

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 192 hours

Face-to-face learning = 108 hours

Total = 300

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://rl.talis.com/3/uwe/lists/AD653781-0CB7-33AB-46FD-7EE27F7B2D76.html

Part 4: Assessment

Assessment strategy: At both first sit and resit, this module is assessed by a combination of techniques: a practical portfolio and a presentation.

The practical portfolio is completed in small groups and used to evidence the student's software project, assessing their collaborative software developments, planning, testing, and team working.

The final group presentation requires students to identify, justify and communicate all stages of their software development lifecycle critically review the stages their software solution and evaluation of team performance.

Grading and feedback will be individual assessed based upon contribution, collaboration and input towards the group objective and predefined tasks.

Tutor-lead formative feedback will be available throughout the module.

The resit opportunity should follow the same format as the first sit. Due to the size and complexity of the project a re-working of the practical component is recommended.

Assessment tasks:

Portfolio (First Sit)
Description: Group Portfolio - Design, Build and Test a Software Project built using a Collaborative Approach
Weighting: 70 %
Final assessment: No

Group work: Yes
Learning outcomes tested: MO2, MO3, MO4

Presentation (First Sit)
Description: Group Presentation (20 mins)
Weighting: 30 %
Final assessment: Yes
Group work: Yes
Learning outcomes tested: MO1

Portfolio (Resit)
Description: Portfolio - Design, Build and Test a Software Project built using a Collaborative Approach.
Weighting: 70 %
Final assessment: No
Group work: Yes
Learning outcomes tested: MO2, MO3, MO4

Presentation (Resit)
Description: Group Presentation (20 mins)
Weighting: 30 %
Final assessment: Yes
Group work: Yes
Learning outcomes tested: MO1

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Applied Computing[Sep][FT][UCW][3yrs] BSc (Hons) 2021-22