Module Specification

Distributed and Enterprise Software Development

Version: 2022-23, v2.0, 31 May 2022

Contents

Part 1: Information

Module title: Distributed and Enterprise Software Development

Module code: UFCFTR-30-3

Level: Level 6

For implementation from: 2022-23

UWE credit rating: 30

ECTS credit rating: 15

Faculty: Faculty of Environment & Technology

Department: FET Dept of Computer Sci & Creative Tech

Partner institutions: None

Delivery locations: Frenchay Campus

Field: Computer Science and Creative Technologies

Module type: Standard

Pre-requisites: Advanced Software Development 2021-22, Principles of Programming 2020-21, Systems Development Group Project 2021-22

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: This module consolidates the previously attained programming and systems development knowledge in years 1 & 2 and prepares students for the challenges of solving and implementing solutions for complex organisations like enterprises. These are decentralised, networked multilevel systems that produce and process very large volumes of data. Being able to assess the suitability, select the

right tools and techniques and implement enterprise wide systems is an important set of knowledge and valuable skill set for a systems developer.

Features: Not applicable

Educational aims: This module aims to introduce advanced and enterprise level software development using contemporary distributed technologies.
It also aims at developing students understanding of enterprise system development and to contribute to their understanding of legal, ethical, social and professional aspects.

Outline syllabus: Initially the module teaches distributed and parallel computing concepts including file system, concurrency, synchronisation, messaging, persistence, replication, scalability, integrity, latency, fault-tolerance and security to implement small to large-scale software applications. Here, students will learn various architectural patterns for distributed systems and study n-tier architecture, service oriented and microservices architectures in detail. Theory and concepts of components, interfaces and services will be taught by using a suitable modelling notation. Students will also learn to design trusted distributed transaction management and apply Extract, Transform and Load (ETL) concepts in implementing secure software.

Subsequently, students will learn various programming models and algorithms to deal with large-scale data e.g., MapReduce. Focus will be on parallel and distributed algorithms in batch processing, shared-memory, message passing and models for stream processing.

Finally, students will learn enterprise system development and associated legal, ethical, social and professional aspects. This will cover understanding about enterprises of planetary scale and associated complexity. Software systems by acquisition, integration, configuration and customisation, and subsequent interoperability challenges will be covered. The economics of various software acquisition strategies, from DIY to components-of-the-shelf (COTS), to software packages, to service-based capabilities in the Cloud and Edge computing will be covered. Use of virtualisation and containerisation will be taught to understand various scalability, performance, storage, process and deployment related issues

associated with enterprise scale software development in a cloud/edge environment. The significance, capabilities and examples of contemporary technologies such as blockchain or distributed ledgers and cloud and edge computing will be explored in the context of enterprise system development. Students will test these technologies and find out how these technologies impact enterprise software development.

They will learn and apply an agile approach for developing enterprise software by using an enterprise scale framework and gain clear understanding of complexity and challenges associated to stakeholder and requirements analysis, design, implementation, testing and maintenance processes.

Part 3: Teaching and learning methods

Teaching and learning methods: The module will be delivered via a combination of lectorial/workshop and lab sessions, with face-to-face and online help provided by tutors. Where relevant, employers will be invited to give a guest lecture on their company's development practices.

Lectures will focus on providing basic concepts and introduction to lab sessions and independent learning. Lab sessions will focus on allowing the students to apply the concepts learned in the lectures to various problems and contexts. Online resources such as UWE e-library and LinkedIn learning are also available.

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

MO1 Evaluate distributed and parallel computing concepts and paradigms with their legal, ethical, social and professional implications in developing large scale software systems.

MO2 Describe and evaluate the concepts and paradigms of modern enterprise systems with particular focus on components, interfaces and services.

MO3 Apply distributed computing concepts, algorithms and models to develop enterprise level software systems.

MO4 Apply a current industrial project management approach when undertaking a software development project.

MO5 Identify security issues in distributed or enterprise level software systems in order to implement preventive measures.

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 300

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link [from https://rl.talis.com/3/uwe/lists/874C62FC-4042-00DE-7C1C-CD496104F65F.html](https://rl.talis.com/3/uwe/lists/874C62FC-4042-00DE-7C1C-CD496104F65F.html)

Part 4: Assessment

Assessment strategy: The assessment components are designed to ensure that students' understanding and skills are developed incrementally and the assessment strategy provides continual formative verbal feedback opportunities and allows students to develop their skills with the materials being presented in the lectures and laboratory sessions. The group-based working also provides numerous peer-learning opportunities.

The assessment will be carried out over component A with two elements; (1) an individually developed report and (2) a group project developed and demonstrated.

While the Report Component A tests students understanding of theoretical concepts of distributed and enterprise system development through a reflective and an evaluative report written on how a small-scale project is done following professional skills developed through lecture and practical sessions. Each student will have opportunities to demonstrate the practical knowledge and transferable skills

developed so far. Students are expected to reflect on legal, ethical, social and professional aspects of the small-scale individual project, which is expected to let students prepare for the group project.

The project element (Project Component A) will allow students to practice team work via a group project to prepare for professional life. It will let students implement the practical knowledge and transferrable skills following the good practices to contribute to group activities. There will be a group mark and an individual contribution weight calculated from peer assessment data as well as tutor observations. The individual mark will be derived from these two components (group mark and individual weight) and will be supported with individual performance in Q/As during the demonstration. The tutors observation will help adjustment the marks if an evidence of significant unbalanced contributions from the group members is sensed. Each group will submit project code, and be expected to demo their finished project illustrating both group and individual programming skills. Students will get verbal and written feedback.

For resit, both element of Component A will be replicated noting that if some members of any group happen to fail while other members of the same group pass, all leftover members will be allocated to a group.

Assessment components:

Report - Component A (First Sit)
Description: Individual reflection on small project and professional practices.
Weighting: 40 %
Final assessment: No
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Project - Component A (First Sit)
Description: Group project with demonstration
Weighting: 60 %
Final assessment: No

Group work: Yes
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Report - Component A (Resit)
Description: Individual reflection on small project and professional practices.
Weighting: 40 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Project - Component A (Resit)
Description: Group project with demonstration.
In case, few members of a group fails in the main sit, they will be merged with other groups / individual to carry on the group work for the referral.
Weighting: 60 %
Final assessment: No
Group work: Yes
Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Computer Science [Sep][FT][Frenchay][3yrs] BSc (Hons) 2020-21