MODULE SPECIFICATION

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	Field	Computer Science and Creative Technologies
Department	FET Dept of Computer Sci & Creative Tech		
Contributes towards	Computer Science BSc (Hons) 2020-21 Computer Science {Foundation} BSc (Hons) 2020-21		
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		
Module Entry requirements	None		

Part 2: Description

This module consolidates the previously attained programming and systems development knowledge in years 2 & 3 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.

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.

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.

Part 3: Assessment

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.

While the component A tests students understanding of theoretical concepts of distributed and enterprise system development through a written exam, component B tests students practical knowledge and transferable skills. In the individual project demonstration element of Component B, there will be a small project for each student. Each student is expected to submit project code, a brief video of the project demo and reflection on legal, ethical, social and professional aspects of the project. Students will get formative feedback for this element and this individual project demonstration will allow students to prepare for the group project.

In group project demonstration element of Component B, outputs are from group as well as individual tasks. There will be a group mark and a mark for an individual's contribution to group dynamics e.g., applying a System of Systems concept. Marks adjustment may take place where there is an evidence of significant unbalanced contributions from the group members. Each group will submit project code, a brief video of their project demo, reflecting on the group dynamics, project testing and evaluation of the project. Also, each group will be expected to

demo their finished project illustrating both group and individual programming skills. Students will get verbal and written feedback.

For resit, component A has 2 hours exam to test the theoretical knowledge of the students. For component B, each student will need to work on a small individual project. The project specifications will be scaled down to balance the work required from each student. Students will be expected prepare a brief video of the project demo. They will also required to write a report and reflect on how will this project be impacted if the same project is undertaken by a group of students.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Project - Component B		24 %	Individual small project with demonstration - at the end of semester 1
Group work - Component B		36 %	Group project with demonstration - at the end of semester 2
Examination - Component A	✓	40 %	Exam (2 hours)
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Written Assignment - Component B	✓	60 %	Individual project code and report
Examination - Component A		40 %	Exam (2 hours)

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will be able to:

	Module 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 (A, B).
MO2	Describe and evaluate the concepts and paradigms of modern enterprise systems with particular focus on components, interfaces and services (A).
MO3	Apply distributed computing concepts, algorithms and models to develop enterprise level software systems (A, B).
MO4	Apply a current industrial project management approach when undertaking a software development project (B).
MO5	Identify security issues in distributed or enterprise level software systems in order to implement preventive measures (A, B).

Contact Hours

Contact Hours	
Independent Study Hours:	
Independent study/self-guided study	228

	Total Independent Study Hours:	228
	Scheduled Learning and Teaching Hours:	
	Face-to-face learning	72
	Total Scheduled Learning and Teaching Hours:	72
	Hours to be allocated	300
	Allocated Hours	300
Reading List	*The reading list for this module can be accessed via the following link:* from https://rl.talis.com/3/uwe/lists/874C62FC-4042-00DE-7C1C-CD496104F65F.html	