



## MODULE SPECIFICATION

Part 1: Information			
Module Title	Enterprise Systems Development		
Module Code	UFCF85-30-3	Level	Level 6
For implementation from	2018-19		
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			
Module type:	Standard		
Pre-requisites	Object-Oriented Systems Development 2018-19		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
<p><b>Educational Aims:</b> See Learning Outcomes.</p> <p><b>Outline Syllabus:</b> The syllabus includes:</p> <p>Enterprise-scale software systems development: Enterprises of planetary scale and complexity, virtualisation. Software systems by acquisition, integration, configuration and customisation, and subsequent interoperability. 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.</p> <p>Theory and concepts of components, interfaces and services: Specification, test specification, implementation and deployment of components and services, and their various modelling notations, including the Unified Modelling Language (UML).</p> <p>Developmental processes: Agile versus plan-based, impact of scale and complexity on the process. Challenges to agile,</p>

## STUDENT AND ACADEMIC SERVICES

test-driven and incremental approaches for contemporary software development. Issues associated with planetary scale stakeholder analysis, requirements capture, virtualisation and testing. Processes for systems development by software reuse – software discovery and evaluation, acquisition, integration, customisation and configuration.

### Architectures and patterns:

The notion of software architecture and its significance to ESD, separation of concerns ('divide and conquer') in global scale software systems via architectures and patterns. Logical versus physical focus i.e. business-driven, logical separation of concerns via cohesive groupings of components and services minimising coupling dependencies, versus the technology capabilities provided by

large scale technical platform infrastructures. The application of architectures and patterns in relation to analysis and design techniques for enterprise-scale software systems development.

### Enterprise scale development frameworks:

Review of the state-of-the-art frameworks for ESD (e.g. comparison of Microsoft .NET, Java Enterprise Edition, Spring etc). Analysis, design and development of multi-tier, distributed web-based applications by the reuse, configuration, customisation and deployment of framework components and services e.g. servlets and JSPs. Also persistence via design, implementation and use of Java Database Connectivity (JDBC) and relational databases, including transaction management.

### Computing paradigms and models for ESD with emphasis on Cloud Computing:

Its significance (competing) definitions, available capabilities and contemporary technologies. Current example applications. Benefits and drawbacks, especially in relation to the economics and risks of utility computing. Drift of applications and data from localised processing to virtual environments, and likely consequences. State-of-the-art research findings on the potential of Cloud Computing, e.g. via multi-channel, asynchronous and adaptive 'systems of systems', to serve mankind in the future.

**Teaching and Learning Methods:** Scheduled learning includes interactive lectures and tutorials, wherein the state-of-the-art of enterprise systems development is demonstrated, discussed and critically evaluated. At lectures, questions from students are proactively encouraged and freely discussed. Questions from the lecturer are prominent at the start of lectures to clearly establish the learning context and obtain the undivided focus of the student cohort. Further questions and answers are initiated during the lecture. As appropriate for level 3 students, discussions at the end of lectures promote formative feedback, evaluation and deep reflection upon the learning outcomes of the lecture. Audio recordings of the lecture and interactive discussions are taken and made available via the Blackboard Virtual Learning Environment (VLE). Indeed, all lecture slides, recommended articles (both research and trade), videos, URLs and tutorial notes are available on the Blackboard VLE.

At tutorials, students are encouraged to attempt scoped activities (e.g. problem analysis and solving, appraisal, design, implementation, and validation) and then articulate and present their findings to their peers. Activities relate to realistic case studies of level 3 complexity, and may also include the production of implementation artefacts such as deployment descriptors, execution log trails, source code implementation etc. Interactive peer-review (directed by the tutor) is essential for student reflection to achieve the deep learning appropriate for level 3. From time to time, students also participate in "pub-quiz"-like sessions in tutorials. (For example, MCQs are provided to small groups of students, enabling immediate and rich formative feedback on the level and extent of student knowledge and understanding at that point in time.)

Independent learning includes hours engaged with essential reading, case study preparation, assignment preparation and completion etc. Explicit guidance is given to students with respect to the sources of information used in self-study. Library resources such as books, research and trade articles are essential to supplement lectures, and are made available via the Blackboard VLE. Use of library search engines is encouraged – please see Reading Strategy section below. In addition, high quality, robust java-based Open Source modelling and development tools (e.g. ArgoUML and NetBeans) have been selected to enable maximum portability and so ease of installation on a variety of students' own laptop platforms for self-study, and are available free of charge. Having the same tools consistently available of faculty workstations and student laptops, when taken together with the Blackboard VLE, enables great interoperability with respect to

## STUDENT AND ACADEMIC SERVICES

development artefacts, promoting virtualisation of learning location. The learning achieved from self-study (is) are then brought forward by students to be reinforced at the interactive tutorials wherein their knowledge and understanding are deepened by directed articulation, presentation and critical appraisal with their peers and tutor.

### Part 3: Assessment

The assessment strategy for this module comprises a written examination and a coursework assignment. Examination questions examine cognate and practical skills via a range of essay, multi-choice questions (MCQs), and appropriate analysis and design technique exercises. Where appropriate, architectural diagrams, UML diagrams, source code fragments or partial text cases may be provided as the basis for the examination question.

The practical nature of the software development activities on this module provides the rationale for the coursework assignments, of which there are two. The first is a groupbased demonstration while the second is an individual written piece of work. The group-based demonstration consists of an appropriately scoped software application developed by the student team using enterprise development frameworks, architectures and patterns. The individual piece of work consists of a critical appraisal of the architectural and pattern-based modelling activities and framework tools conducted in the second semester.

First Sit Components	Final Assessment	Element weighting	Description
Written Assignment - Component B		28 %	Individual written assignment (1000 words)
Presentation - Component B		42 %	Group-based demonstration of software development
Examination - Component A	✓	30 %	Examination (2 hours)
Resit Components	Final Assessment	Element weighting	Description
Written Assignment - Component B		70 %	Individual written assignment (2000 words)
Examination - Component A	✓	30 %	Examination (2 hours)

STUDENT AND ACADEMIC SERVICES

<b>Part 4: Teaching and Learning Methods</b>		
Learning Outcomes	On successful completion of this module students will be able to:	
	<b>Module Learning Outcomes</b>	
	MO1	Describe the essential characteristics of enterprise-scale software systems and their development
	MO2	Show a detailed knowledge of software development process models for enterprise-scale software systems development, including agile
	MO3	Understand the need for developmental frameworks for developing enterprise systems
	MO4	Apply a state-of-the-art developmental framework e.g. Java Enterprise Edition to the team-based design and development of web-based applications
	MO5	Explain the theory and concepts of components, interfaces and services and their various modelling notations
	MO6	. Discuss in detail the application of the notion of software architecture and software patterns in relation to analysis and design techniques for enterprisescale software systems development
	MO7	Understand computing paradigms and associated models for enterprise systems e.g. Cloud Computing and its potential
Contact Hours	<b>Contact Hours</b>	
	<b>Independent Study Hours:</b>	
	Independent study/self-guided study	228
	<b>Total Independent Study Hours:</b>	228
	<b>Scheduled Learning and Teaching Hours:</b>	
	Face-to-face learning	72
	<b>Total Scheduled Learning and Teaching Hours:</b>	72
	<b>Hours to be allocated</b>	300
	<b>Allocated Hours</b>	300
Reading List	<p>The reading list for this module can be accessed via the following link:</p> <p><a href="https://uwe.rl.talis.com/modules/ufcf85-30-3.html">https://uwe.rl.talis.com/modules/ufcf85-30-3.html</a></p>	