Module Specification

Enterprise and Systems Architectures Modelling and Development

Version: 2021-22, v2.0, 27 Jul 2021

Contents

Part 1: Information

Module title: Enterprise and Systems Architectures Modelling and Development

Module code: UFCFDD-15-M

Level: Level 7

For implementation from: 2021-22

UWE credit rating: 15

ECTS credit rating: 7.5

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: Object-Oriented Analysis, Design and Programming 2021-22

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: See Learning Outcomes

Outline syllabus: Enterprise-scale software systems development:

Enterprises of planetary scale and complexity, virtualisation.

Examples of enterprise scale systems, the types of organisation and kinds of complex software systems that each uses.
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.

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).

Architectures and patterns:
Notion of software architecture and its significance to enterprise system developments.
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:
Comparison of Microsoft .NET versus Java Enterprise Edition.
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. EJBs, servlets and JSPs, web services.
Also persistence via design, implementation and use of Java Database Connectivity (JDBC) and relational databases, including transaction management.

Cloud Computing:

Significance and (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 humankind in the future.

Part 3: Teaching and learning methods

Teaching and learning methods: Scheduled contact hours comprise a 1 hour lecture and 1 hour of practical activities or tutorials, making 2 hours per week in total.

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 M 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 their findings to their tutor and peers. Activities relate to realistic case studies of level M complexity, and include the production of implementation artefacts such a 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 M.

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 development artefacts, promoting virtualisation of learning location. The learning achieved from self-study is 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.

Module Learning outcomes:

MO1 Describe the essential characteristics of enterprise-scale software systems and their development in detail

MO2 Explain the theory and concepts of components, interfaces and services and their various modelling notations

MO3 Discuss in detail the application of the notion of software architectures and software patterns in relation to analysis and design techniques for enterprise-scale software systems development

MO4 Understand and apply a state-of-the-art development framework e.g. Java Enterprise Edition (Java EE) to the design and development of web-based applications

MO5 Provide an overview understanding of Cloud Computing and its potential

MO6 Locate and critique relevant research in the field via journal and conference papers, and other appropriate sources

Hours to be allocated: 150

Contact hours:

Independent study/self-guided study = 126 hours

Face-to-face learning = 24 hours

Total = 150

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://uwe.rl.talis.com/modules/ufcfdd-15-m.html

Part 4: Assessment

Assessment strategy: The assessment strategy for this module comprises a written examination and software demonstration.

The written examination is of two hours duration and comprises six questions mapping to each of the module's learning outcomes. 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.

A demonstration of software developed to a requirements specification is highly appropriate to a module concerned with software development. The ability of the student to effectively communicate the enterprise system design rationale and trade-offs to others is valued highly on this module and rewarded as such.

A combination of written examination and software demonstration is appropriate for this module as the culmination of the journey experienced by students learning Enterprise Systems Development. This journey is a highly interactive one, wherein regular and continual formative feedback, much enabled by individual and group evaluation and reflection and critical appraisal, is essential to enrich and deepen the on-going learning of the student.

Resit opportunities are provided as follows: The resit examination is similar in format and approach to the first sit, while the resit coursework assignment is an individual essay covering the same learning outcomes as in the first sit, which allows students to critically reflect on the feedback provided with previous assignments.

Assessment components:

Examination (Online) - Component A (First Sit)
Description: Online Final written examination (2 hours)
24 hour window
Weighting: 50 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO6

Presentation - Component B (First Sit)
Description: Demonstration of software (20 minutes)
Weighting: 50 %
Final assessment: No
Group work: No
Learning outcomes tested: MO3, MO4, MO5, MO6

Examination (Online) - Component A (Resit)
Description: Online Final written examination (2 hours)
24 hour window
Weighting: 50 %
Final assessment: Yes
Group work: No
Learning outcomes tested:

Written Assignment - Component B (Resit)

Description: Written assignment

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested:

Part 5: Contributes towards

This module contributes towards the following programmes of study: