STUDENT AND ACADEMIC SERVICES

UWE Bristol University of the West of England

MODULE SPECIFICATION

Part 1: Information			
Module Title	Software Engineering		
Module Code	UFCFK6-30-2	Level	Level 5
For implementation from	2020-21		
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		
Module Type:	Standard		
Pre-requisites	Introduction to OO Systems Development 2020-21		
Excluded Combinations	None		
Co-requisites	None		
Module Entry Requirements	None		
PSRB Requirements	None		

Part 2: Description

Overview: Pre-requisites: students must take one out of UFCFF6-30-1 Programming in C or UFCFC3-30-1 Introduction to OO Systems Development.

Educational Aims: See Learning Outcomes

Outline Syllabus: Introduction to software engineering with emphasis on the engineering discipline to software development, the nature of the software product, and professional issues.

The Software Engineering Process: generic process models such as the waterfall, evolutionary, spiral, and reuse models in addition to further models such as Rational Unified Process, and Agile software development methods. Also, coverage of the triangle of software development: process model, modelling language, and tools (with emphasis on upper case tools).

Software Engineering Activities: Business Process Modelling, Requirements Analysis, Design, Implementation, Testing, and deployment along with supporting workflows to include configuration management, and project management. Testing: Verification and Validation, and testing strategies.

Project Management: developing the initial proposal, costing, planning, scheduling, personnel selection, project monitoring and reviews.

The use of the Unified Modelling Language (UML) for system modelling with emphasis on context, interaction, structural, behavioural models in addition to covering model driven engineering.

Software Cost Estimation: cost estimation methods with emphasis on both algorithmic and non-algorithmic models.

Configuration Management: change, version, and release management, and system building.

Software Quality: definition, the process of software quality management, software quality standards, reviews and inspections, and software metrics.

State-of-the-art and emerging software engineering paradigms including service-oriented software engineering, software engineering and the Clouds, and Dependability.

Teaching and Learning Methods: Contact time: 72 hours
Assimilation and development of knowledge: 148 hours
Exam preparation: 40 hours
Coursework preparation: 40 hours
Total study time: 300 hours

Scheduled learning: these include a course of lectures and practical sessions to provide theoretical understanding of software engineering, and develop the necessary practical skills in the engineering of software development, respectively. Students will be engaged from the beginning of each lecture with some reasonable interaction through the essential parts of the lecture and associated key questions. During practical sessions, students will work in groups formed from the start of the module and carried over till the end of the module run. In addition, about two-thirds of the practical sessions will be utilised for the work as a group on the module's assignment, which includes a portfolio of deliverables that will be informally assessed by the respective tutor with feedback given for each group within the respective tutorial session.

Independent learning: these include hours engaged with essential reading of lecture notes, working on deliverables of the group-project assignment ahead for informal assessment and feedback by the respective tutor in tutorial sessions. Supportive guidance will be provided to students regarding the most appropriate sources of information such as books, research and practical articles, lectures notes, and requirements specifications templates that will be made available, where possible, via the Blackboard VLE. Such independent learning will yield two outcomes: (1) contribute to higher quality deliverables of the group project assignment and hence enhance guiding and enhancing the student learning experience, and (2) reinforcing higher interactivity (with critical appraisal) in the module's key areas initiated by individuals and within groups, and hence improving the quality of the anticipated module's learning outcomes.

Part 3: Assessment

The assessment strategy for this module is based on a written assignment in relation to a software group project with deliverables over three coursework elements CW1, CW2, and CW3 with 15%, 65% and 20% of the module assessment, respectively.

CW1 is a project proposal to be submitted before the end of the first term and is 15% of the total module assessment. This is group-based with allocation of marks dependent on the individual students' contribution that will be controlled by the respective resources allocated to the associated tasks' set in the assignment specifications. Feedback will be given to students in the beginning of the second term, and hence students will utilise such feedback to inform better attainment in the remaining parts of the assignment CW2 and CW3.

CW2 is 65% of the total module assessment and includes detailed group-based project deliverables such as requirements specifications, software design, software prototyping, and software testing specifications, with some specific project individual contributions.

CW3 is 20% of the total module assessment and is an individual reflective report in relation to problems faced, software engineering lessons learned, portfolio of tasks undertaken and suggestions to enhance the run of the group project itself and also in relation functional and non-functional enhancements of the respective software application specified, designed and developed.

For the resit assignment, this is not group-based and it relates to a critical reflective essay (3000 words) on key selected software engineering aspects as per the module's learning outcomes, and includes critical summative view of the current state of the art literature and suggested future directions in relation to the aspects outlined in the resit assignment specifications.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Reflective Piece - Component A		20 %	Individual Reflective Report (1500 words)
Project - Component A		65 %	Group-based and Individual Project Deliverables
Project - Component A		15 %	Group Project Proposal
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Written Assignment - Component A		100 %	Written assignment (3000 words)

Part 4: Teaching and Learning Methods

Learning Outcomes

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

Module Learning Outcomes	**Reference**
Recognise the engineering dimension in software development including professional practice in contrast to non-engineering disciplines	MO1
Show detailed knowledge of the notion of "software development process", and the different generic software development process models along with their key distinguished features	MO2
Understand the concept of software quality, the generic process of software quality management, and software metrics	MO3
Explain the key methods and models used in software cost estimation	MO4
Explain the role of configuration management during the entire software development life cycle	MO5
Apply software engineering techniques and methods in a process centred approach to the development of a software application in a group-based setting covering most stages of the software development life cycle	MO6
Understand and apply specific UML modelling techniques and notations in relation to the key stages of the software development life cycle	MO7
Provide good understanding of state-of-the-art and emerging software engineering paradigms and their potential	MO8

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:* https://uwe.rl.talis.com/modules/ufcfk6-30-2.html	

Part 5: Contributes Towards

This module contributes towards the following programmes of study:

Computing [Sep][FT][Frenchay][3yrs] BSc (Hons) 2019-20

Computing [Sep][SW][Frenchay][4yrs] BSc (Hons) 2019-20

Computing {Dual} [Aug][FT][Taylors][3yrs] BSc (Hons) 2019-20

Computing {Dual} [Aug][SW][Taylors][4yrs] BSc (Hons) 2019-20

Computing {Dual} [Mar][FT][Taylors][3yrs] BSc (Hons) 2019-20

Computing {Dual} [Mar][SW][Taylors][4yrs] BSc (Hons) 2019-20

Software Engineering [Sep][SW][Frenchay][4yrs] BSc (Hons) 2019-20

Software Engineering [Sep][FT][Frenchay][3yrs] BSc (Hons) 2019-20

Software Engineering {Dual} [Aug][FT][Taylors][3yrs] BSc (Hons) 2019-20

Software Engineering {Dual} [Mar][FT][Taylors][3yrs] BSc (Hons) 2019-20

Software Engineering [Jan][FT][Northshore][3yrs] BSc (Hons) 2019-20

Software Engineering {Foundation} [Oct][FT][GCET][4yrs] BEng (Hons) 2018-19

Computing {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19

Computing {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19

Software Engineering {Foundation} [Feb][FT][GCET][4yrs] BEng (Hons) 2018-19

Computer Science [Jan][FT][Villa][3yrs] BSc (Hons) 2019-20

Computer Science [May][FT][Villa][3yrs] BSc (Hons) 2019-20

Computer Science [Sep][FT][Frenchay][3yrs] BSc (Hons) 2019-20

Computer Science [Sep][FT][Villa][3yrs] BSc (Hons) 2019-20

Computer Science [Sep][SW][Frenchay][4yrs] BSc (Hons) 2019-20

Computer Science {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19

Computer Science {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19