

London South Bank
University

Module Guide

Systems and Software Engineering

ENG_6_539

<http://vle.lsbu.ac.uk>

School of Engineering

2015-16

Level 6

Table of Contents

1.	Module Details	3
2.	Short Description	3
3.	Aims of the Module	3
4.	Learning Outcomes	3
5.	Assessment of the Module	4
6.	Feedback.....	4
7.	Introduction to Studying the Module	4
7.1	Overview of the Main Content	4
7.2	Overview of Types of Classes	4
7.3	Importance of Student Self-Managed Learning Time	4
7.4	Employability	4
8.	The Programme of Teaching, Learning and Assessment	5
9.	Student Evaluation	5
10.	Learning Resources	5
10.1	Core Materials	6
10.2	Optional Materials.....	6
	NOTES	6

1. MODULE DETAILS

Module Title:	System and Software Engineering
Module Level:	6
Module Reference Number:	ENG_6_539
Credit Value:	15
Student Study Hours:	156
Contact Hours:	52
Private Study Hours:	104
Pre-requisite Learning (If applicable):	None
Co-requisite Modules (If applicable):	None
Course(s):	BEng Computer systems and networks Full-Time (2388) Part-time (2419)
Year and Semester	Year 4, Semester 2
Module Coordinator:	Gabor Nyerges
MC Contact Details (Tel, Email, Room)	ext:7535, nyergeg@lsbu.ac.uk, room T401
Teaching Team & Contact Details (If applicable):	Gabor Nyerges (details above) Clive Hayton (ext: 7511, Clive.Hayton@lsbu.ac.uk)
Subject Area:	Computer Systems
Summary of Assessment Method:	End of module exam + coursework

2. SHORT DESCRIPTION

Software is assuming ever increasing importance as a central part of systems engineering within the technology as well as commerce and financial sectors of industry. Almost every technology product has some software components. Furthermore, most financial institutions rely heavily on software to provide their respective services.

The formal study of software engineering and, of course, its application to problems in the field of engineering is now essential for all engineers. In addition, Software Engineering is a microcosm of engineering as such and many aspects, particularly, those related to management are immediately transferable to other aspects of engineering.

This guide tells you how the teaching will be structured, and how you will be assessed. Its main purpose is to enable you to organise your own time and efforts to get the most out of the module. Preparation and reading are essential to making progress and there will be little time for teacher led revision, until the end of the module.

3. AIMS OF THE MODULE

The aim of this module is to teach students the discipline of software engineering. Thus, content is geared towards the practice of software engineering rather than software development. This means that the material will focus on aspects that a software engineer should know in order to be able to manage and deliver software engineering projects

4. LEARNING OUTCOMES

On successful completion of this unit, students will be able to;

- Discuss different lifecycle models and make informed decisions as to which should be used for a particular problem.
- Appreciate the role of requirements analysis in the lifecycle and write requirements documents for simple systems.
- Describe the functional and object oriented approaches to software, identify situations in which they are appropriate, and use appropriate notations for the presentation of system models and designs, take into account reliability and reusability considerations.
- Deal with problems associated with real-time system design.

- Distinguish between validation and verification, use methods appropriate to the stage in the lifecycle and the objective of the testing, and describe ways in which testing and development can be integrated.
- Describe and use metrics for measurement and estimation with a view to costing software development exercises.
- Describe different approaches to the organisation and management of software projects.

5. ASSESSMENT OF THE MODULE

Part I: End-of-module examination (70%)

Part II: Assignment (30%)

6. FEEDBACK

The Coursework feedback will be given in two weeks after the deadline.

7. INTRODUCTION TO STUDYING THE MODULE

7.1 Overview of the Main Content

Part I: Concepts of software engineering, design, modelling and specification. Software reliability, validation and verification. Documentation and costing

Part II: Finite State Machine Modelling of systems

7.2 Overview of Types of Classes

Part I

Teaching will consist of lectures and tutorial sessions. All the main points of the subject matter will be covered in the lectures. Relevant tutorials will be carried out in at various points during the timetabled lecture sessions to enforce the practical implications of lecture material.

Part II:

Teaching will consist of two lectures followed by four weeks of independent study for the assignment, accompanied by weekly consultations.

7.3 Importance of Student Self-Managed Learning Time

Self-managed study and practice is important for completing the module successfully. It is essential for Part II, which - after a couple of introductory lectures - will consist of independent work on an assignment, accompanied by weekly consultations.

The breadth of material covered in Part I and the complexity of some of the topics also requires the students to plan for a substantial period of self-study.

7.4 Employability

Students who successfully complete this module should be able to work in the area of systems design and software engineering projects. The breadth of the subject means that there is a large catchment area for graduates in this discipline, ranging from commercial applications to real-time control systems.

8. THE PROGRAMME OF TEACHING, LEARNING AND ASSESSMENT

SEMESTER 2		
WEEK	TOPIC	READING (CORE TEXT)
1	<ul style="list-style-type: none"> • GENERAL CONCEPTS IN SOFTWARE ENGINEERING • PARADIGMS IN SOFTWARE ENGINEERING • PROCEDURAL PROGRAMMING • OBJECT-ORIENTED PROGRAMMING 	Bezanov, 2010, c1 Sommerville, 2016, cc1,2,7
2	<ul style="list-style-type: none"> • BASIC ASPECTS OF SOFTWARE MODELLING • DOCUMENTING THE ARCHITECTURE • Introduction to UNIFIED MODELLING LANGUAGE • REQUIREMENTS ENGINEERING -WRITING A REQUIREMENTS SPECIFICATION 	Bezanov , 2010, cc2-3 Sommerville, 2016, cc4-6
3	<ul style="list-style-type: none"> • SYSTEMS DESIGN: SYSTEMS ENGINEERING • REAL TIME SYSTEMS (RTS) and SCHEDULING • SOFTWARE PROTOTYPING 	Bezanov, 2010, 5-6 Sommerville, 2016, cc3,7,9,21
4	<ul style="list-style-type: none"> • SOFTWARE DEPENDABILITY AND RELIABILITY • SAFETY AND RELIABILITY • DESIGN RECOMMENDATIONS FOR DEPENDABILITY • HAZARD AND RISK ANALYSIS IN SAFETY-CRITICAL SYSTEMS • FAILURE MODE AND EFFECTS ANALYSIS • RISK EVALUATION METHODS 	Bezanov, 2010, cc7-8 Sommerville, 2016, cc10-12
5	<ul style="list-style-type: none"> • VALIDATION AND VERIFICATION • TESTING TO VALIDATE THE SYSTEM, WHITE BOX and BLACK BOX • ASSESSING THE EFFECTIVENESS OF TESTING • SOFTWARE QUALITY • CURRENT METRICS AND MODELS TECHNOLOGY • CORPORATE ATTITUDE TO SOFTWARE QUALITY 	Bezanov, 2010, cc9-11 Sommerville,2016, c8-9
6	<ul style="list-style-type: none"> • SOFTWARE COSTING • COCOMO COST ESTIMATION MODEL • DEVELOPMENT MODES FOR SOFTWARE COSTING 	Bezanov, 2010, c12 Sommerville,2016, c23
7	<ul style="list-style-type: none"> • INTRODUCTION TO THE FINITE STATE MACHINE MODELLING OF SYSTEMS • SIMPLE EXAMPLES OF FINITE STATE MACHINE MODELS 	Nyerges, <i>Finite State Machines Lecture Notes</i>
8	<ul style="list-style-type: none"> • FINITE STATE MACHINE MODELS OF SYSTEMS RELEVANT FOR THE ASSIGNMENT 	Nyerges, <i>Finite State Machines Lecture Notes</i>
9 - 12	WEEKLY CONSULTATION FOR THE ASSIGNMENT	
13	REVISION	

9. STUDENT EVALUATION

N/A

10. LEARNING RESOURCES

10.1 Core Materials

Part I

Bezanov, G (2010) *Software Engineering*. MIG Consulting Ltd, London
Sommerville, I., (2016) *Software Engineering (9th Edition)*, Harlow UK: Pearson

Part II

Nyerges, G *Finite State Machines Lecture Notes*, Online, link to website is on the module's VLE site

10.2 Optional Materials

Part I

Pressman , R. *Software Engineering: A Practitioner's Approach (International Edition)*. (8th Ed.)
McGraw-Hill

Part II

Sommerville, I., (2016) *Software Engineering (9th Edition)*, Harlow UK: Pearson

NOTES

N/A