



Module Specification

1. Factual information			
Module title	TM354: Software Engineering	Level	3
Module tutor	TBA	Credit value	30
Module type	Taught	Notional learning hours	8

2. Rationale for the module and its links with other modules

Software engineering (TM354) – the intellectual tools needed to design, build, and test software systems. This module aims to provide you with an understanding of software engineering concepts and a view of practical software development. It follows a disciplined approach to the development of software systems to meet specified requirements. You will become familiar with a wide range of techniques to support the dialogue between software engineers and an organisation's stakeholders, and the work of the developers. You will also develop a good understanding of the different approaches to, and practices of, software development, including those followed by agile methods.

3. Aims of the module

1. To understand the business domain for a problem requiring a software solution or a change to an existing solution
2. To acquire the tools and knowledge to analyse and design such a solution or change
3. To understand how any chosen software architecture will impact on the satisfaction of all users requirements and expectations
4. To apply and reuse design expertise from a set of design patterns
5. To develop the skills for testing outputs of all activities throughout the development process.

4. Pre-requisite modules or specified entry requirements

M251

5. Intended learning outcomes	
A. Knowledge and understanding	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>A1. Understand concepts of software development and maintenance, specialising in such topics as Web and Internet design and programming, advanced database techniques or human computer interaction</p> <p>A2. Acquire the methods and tools used to develop a range of software systems</p> <p>A3. Identify a range of situations in which computer systems are used, the ways in which people interact with them, and the ethical, social and legal problems that computer software can create.</p>	<ul style="list-style-type: none"> • 25% face-to-face tutorial sessions • TMA work • Module learning booklets and support material
B. Cognitive skills	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>B1. Explain advanced software development concepts and apply them to practical problems, including in an extended piece of work</p> <p>B2. Analyse problems, and design and evaluate realistic solutions to them</p> <p>B3. Compare and contrast a variety of computing methods and tools, identifying the best choices to apply to specific problems</p> <p>B4. Explain the various roles, functions and interactions of members of a software development team.</p>	<ul style="list-style-type: none"> • 25% face-to-face tutorial sessions • TMA work • Module learning booklets and support material

C. Practical and professional skills	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>C1. Work independently, planning, monitoring, reflecting on and improving your own learning and working practices</p> <p>C2. Work in a group, communicating computing ideas effectively in speech and in writing</p> <p>C3. Find, assess and apply information from a variety of sources, using information technology where necessary, in a number of assignments, including at least one significant piece of work</p> <p>C4. Use numerical and analytical techniques confidently to solve complex problems.</p>	<ul style="list-style-type: none"> • 25% face-to-face tutorial sessions • TMA work • Module learning booklets and support material

D Key transferable skills	Learning and teaching strategy
<p>Upon completing this module, students will be able to:</p> <p>D1. Design, program, test and evaluate software systems</p> <p>D2. Use modern software tools, both within and outside your workplace</p> <p>D3. Identify and handle the ethical, social and legal issues that may arise during software development and use.</p>	<ul style="list-style-type: none"> • 25% face-to-face tutorial sessions • TMA work • Module learning booklets and support material

6. Indicative content.
<p><u>Block 1: From domain to requirements</u></p> <p><u>Unit1</u> Approaches to Software Development</p> <p><u>Unit2</u> Requirements Concepts</p> <p><u>Unit3</u> From Domain Modelling to Requirements Analysis</p> <p><u>Unit4</u> The Case Study: Part 1</p>

6. Indicative content.

Block 2 – From Analysis to Design

Unit5

Classes and Associations (Structural modelling of domain versus the solution)

Unit6

Interactions (Dynamic modelling)

Unit7

State and Activities (More dynamic modelling)

Unit8

The Case Study: Part 2

Block 3 – From Architecture to Product

Unit9

From Model to Implementation (architecture, patterns and reuse)

Unit10

Components, Patterns and Architecture (building blocks and enterprise architectures)

Unit11

Product Quality: Metrics, Verification, Validation, Testing

Unit12

The Case Study: Part 3

Unit13

Process Quality Management, Human Resources, Quality Assurance

Unit14

Human Factors and Professional Issues

7. Assessment strategy, assessment methods and their relative weightings

TMA Work: 20%

MTA: 30%

Exam: 50%

8. Mapping of assessment tasks to learning outcomes														
Assessment tasks	Learning Outcomes													
	A1	A2	A3	B1	B2	B3	B4	C1	C2	C3	C4	D1	D2	D3
TMA	x	x	x	x	x	x	x	x	x	x	x	x	x	x
MTA	x	x	x	x	x						x	x		x
Final	x	x	x	x	x	x	x				x	x		x

9. Teaching staff associated with the module	
Tutor's name and contact details	Contact hours
TBA	

10. Key reading list				
Author	Year	Title	Publisher	Location
Module adopted from OU, UK.				

11. Other indicative text (e.g. websites)
https://lms.arabou.edu.kw/