Module specification

When printed this becomes an uncontrolled document. Please access the Module Directory for the most up to date version by clicking on the following link: Module directory

Module Code	COM478
Module Title	Software Development Methodologies
Level	4
Credit value	20
Faculty	FACE
HECoS Code	100753
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
BSc (Hons) Computer Science	Core
BSc (Hons) Computer Science with Industrial Placement	Core
BSc (Hons) Software Engineering	Core
BSc (Hons) Software Engineering with Industrial Placement	Core

Pre-requisites

N/A

Breakdown of module hours

Learning and teaching hours	24 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	12 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	**36** hrs
Placement / work based learning	0 hrs
Guided independent study	164 hrs
Module duration (total hours)	**200** hrs

For office use only	
Initial approval date	08/11/2023
With effect from date	Sept 2024
Date and details of revision	
Version number	1

Module aims

This module will introduce students to the fundamental concepts and principles of software development methodologies.

- Provide an overview of software development life cycle (SDLC) and its phases.
- Familiarize students with different software development methodologies, including traditional and agile approaches.
- Develop an understanding of the significance of methodologies in software development projects.
- Examine the characteristics, advantages and limitations of different methodologies.
- Evaluate the suitability of different methodologies for specific project types and organizational environments.

Module Learning Outcomes - at the end of this module, students will be able to:

1	Identify the key principles and concepts of software development methodologies.
2	Analyse and compare different software development methodologies.
3	Develop skills in project planning, estimation and scheduling.

Assessment

Indicative Assessment Tasks:
This section outlines the type of assessment task the student will be expected to complete as part of the module. More details will be made available in the relevant academic year module handbook.

The assessments would be made of multiple tasks, which may include, write a report comparing and contrasting two software development methodologies; such as Waterfall vs. Agile, in terms of their key principles, advantages and limitations. Analyse case studies or real-world examples to demonstrate the application and outcomes of each methodology. Evaluate the suitability of each methodology for different project types and scenario

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1,2,3	Portfolio	100%

Derogations

None

Learning and Teaching Strategies

In line with the Active Learning Framework, this module will be blended digitally with both a VLE and online community. Content will be available for students to access synchronously and asynchronously and may indicatively include first and third-party tutorials and videos, supporting files, online activities any additional content that supports their learning.

As this module progresses, the strategies will change to best support a diverse learning environment. Initially, the module will start with a heavier reliance on engaging tutor-led lectures, demonstrations, and workshops to ensure that the students get the relevant threshold concepts. As the module continues experiential and peer learning strategies will be encouraged as the students' progress with their portfolio work.

Assessment will occur throughout the module to build student confidence and self-efficacy in relation to applying appropriate software development methodologies in relation to the chosen scenario.

Indicative Syllabus Outline

Yearly content will be updated to represent the most appropriate content for current industry technologies, but a list of indicative topics could include:

- Overview of software development life cycle (SDLC) and its phases.
- Understanding the importance of methodologies in software development.
- Comparison between traditional and modern methodologies.
- Project planning, estimation and scheduling.
- Managing risks and uncertainties in software projects.
- Examining success stories and lessons learned from various methodologies.
- Evaluating the applicability and effectiveness of methodologies in different contexts.

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update. Please *ensure correct referencing format is being followed as per University Harvard Referencing Guidance.*

Essential Reads

B. Hughes, B, Project Management for IT-Related Projects: 3rd edition, BCS, The Chartered Institute for IT, 2019.

Other indicative reading

J.L. Brewer, Methods of IT Project Management, Purdue University Press, 2022.