Prifysgol Wrecsam
Wrexham University

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	COM758
Module Title	Advanced Development Methodologies
Level	7
Credit value	20
Faculty	FACE
HECoS Code	100956
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
MSc Software Engineering	Core
MSc Software Engineering with Advanced Practice	Core

Pre-requisites

N/A

Breakdown of module hours

Learning and teaching hours	11 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	10 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	**21** hrs
Placement / work based learning	0 hrs
Guided independent study	179 hrs
Module duration (total hours)	**200** hrs

For office use only	
Initial approval date	08/11/2023
With effect from date	Sept 2024

For office use only	
Date and details of revision	
Version number	1

Module aims

The module aims to develop a comprehensive understanding of advanced software development methodologies and their theoretical foundations.

- To explore and critically evaluate the strengths and weaknesses of various development methodologies in different project contexts.
- To enhance the ability to select and apply appropriate development methodologies based on project requirements, team dynamics and organizational constraints.
- To examine the role of software development processes, tools, and techniques in improving productivity, efficiency, and quality of software development projects.
- To explore the impact of advanced development methodologies on software quality, scalability, maintainability, and overall project success.

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

1	Critically evaluate different software development methodologies.
2	Demonstrate proficiency in selecting and applying appropriate development methodologies based on project requirements.
3	Analyse and compare various software development processes, including requirements gathering, design, implementation, testing, deployment and maintenance.
4	Apply advanced techniques for managing software development projects.

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 indicative assessment would consist of multiple elements, one element possibly being; students are given a real or hypothetical software development project and are required to analyse and evaluate the suitability of different development methodologies for the given task.

They should consider project requirements, team composition, organizational constraints, and other relevant factors. The assessment could involve a written report or a presentation summarizing their analysis and recommendations. Another element might be students are tasked with conducting independent research on a specific aspect of advanced development methodologies. They could explore emerging trends, evaluate the effectiveness of certain methodologies in specific contexts, or analyse the impact of development methodologies on software quality or team collaboration. Students are expected to present their findings in a well-structured research paper following academic conventions. Practical application, students are given a set of project requirements and asked to apply a specific development methodology to complete the task.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1, 2, 3, 4	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 advanced techniques for managing software development projects.

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:

- Agile methodologies, such as Scrum, Kanban and Extreme Programming (XP)
- Lean Development
- Continuous Integration/Continuous Delivery (CI/CD)
- Feature-Driven Development (FDD)
- Rational Unified Process (RUP)
- Spiral Model
- Dynamic Systems Development Method (DSDM)
- Feature-Driven Agile (FDD Agile)
- Crystal Methodologies

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update.

Essential Reads

N/A

Other indicative reading

R.C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship,* Prentice Hall, 2008.

G. Kim, *The Unicorn Project: A Novel about Digital Disruption, Redshirts, and Overthrowing the Ancient Powerful Order: A Novel about Developers, Digital Disruption, and Thriving in the Age of Data,* IT Revolution Press, 2019.

T. Winters, H. Wright, T. Manshrek, *Software Engineering at Google: Lessons Learned from Programming Over Time*, Oreilly, 2020.