WM278-30 Software Development Life Cycle

24/25

Department
> WMG

Level
> Undergraduate Level 2

Module leader
> Avleen Malhi

Credit value
> 30

Module duration
> 12 weeks

Assessment
> 100% coursework

Study locations
> University of Warwick main campus, Coventry Primary
> Distance or Online Delivery

Description

Introductory description

With the increasing trend across the industry, software engineering skills are becoming a vital part of any product development. It goes hand-in-hand with the integration of the mechatronics hardware devices, which is becoming crucial for many sectors, not just the software industry.

This module contributes to AHEP 4 learning outcomes for Partial CEng and BCS accreditation criteria for Full CITP as referenced in the learning outcomes.

[Module web page](#)

Module aims

The aim of this module is to cover the main phases of the software development life cycle including requirements analysis, design, development, testing, and deployment. This involves analysing user requirements, defining functional and non-functional requirements, and subsequently designing a system architecture that aligns with these gathered requirements. In particular, this module introduces the use of various UML diagrams such as use case diagrams to

capture user interactions with contexts, class diagrams for illustrating system structure, sequence diagrams for depicting the flow of interactions between system components.

This module also explores key software process models including the Agile Manifesto, Scrum, XP, Lean, and Kanban and aims to gain a deeper understanding of different software design models to effectively manage projects in software development. More importantly, by engaging in a series of carefully designed work-based projects and case studies, the module aims to deliver to students the software development mindset, which allows them to apply to real-world problems using tools and techniques for managing changes in software, code, and documentation.

Learners will be taught to manage their own professional development as engineers / IT professionals (as individuals, team members or leaders), by introducing them to reflective practice and the UK Standard for Professional Engineering Competence (UK SPEC) / Chartered IT Professional (CITP) standard.

Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

Importance of Software Engineering
Software development methodologies
Stages of software development life cycle
Gathering and analysing requirements
Functional and non-functional requirements
Software Design using UML
Designing the Software using UML
Use Case diagram
Activity diagram
Sequence diagram
State diagram
Deployment diagram
Class diagram
Overview of software quality assurance
Defining quality: requirements and specifications
Security as a fundamental aspect of quality
Quality by design: building in quality
Program verification technologies and methods
Inspections and code reviews
Software testing
Software configuration control
Requesting, evaluating, and approving software changes
Implementing software changes
Software documentation
Generate documentation from the code
Writing user manual
Software process modelling
The Agile Manifesto

Sprint management
XP values and principles
Pair programming
Test-driven development
Continuous integration
Coding standards
Lean Software Development
Kanban
Measure and manage workflow

Learning outcomes

By the end of the module, students should be able to:

- Identify user requirements systematically to meet needs from stakeholders in software development. [CITP: 2.1.1, 2.2.5].
- Analyse and define project requirements to recognise key software processes used in software development [CITP: 2.1.2, 2.1.10, 2.3.1].
- Employ software design process by effectively producing artefacts and contexts to optimise process workflow as part of a team [CITP: 2.1.8].
- Apply the range of software techniques, tools, and methods to improve configuration management, version control and software build [CITP: 2.1.12, 2.1.4].
- Distinguish between key different design patterns and software testing strategies comprehensively [AHEP:4 - C6] [CITP: 2.1.7].
- Effectively demonstrate software development on a project based on user requirements and software design [CITP: 2.2.1, 2.2.4].

Indicative reading list

- I. Sommerville, "Software Engineering", Pearson education, 2016, ISBN: 9781292096131
- R. Stephens, "Beginning Software Engineering", Indiana: Wrox 2015, ISBN: 9781118969168.
- Andrew Stellman, Jennifer Greene: "Learning Agile: Understanding Scrum, XP, Lean, and Kanban", 2014, ISBN: 9781449331924.
- Jonathan Rasmusson: "The agile samurai: how agile masters deliver great software", 2010, ISBN: 9781934356586

Subject specific skills

Contributes to the occupational standard for Digital and Technology Solutions Professional (ST0119):

- Apply relevant standard processes, methods, techniques and tools. For example, ISO Standards, Waterfall, Agile in a digital and technology solution project (S5).
- Manage digital and technology solutions projects. For example, identifying and resolving deviations from specification, applying appropriate Project Management methodologies (S6).
- Work effectively within teams, leading on appropriate digital technology solution activities

(S7).
- Initiate, design, implement and debug a data product for a digital and technology solution (S10).

Transferable skills

- Teamwork (Operate within, and contribute to, a respectful, supportive and cooperative group climate. Sensitive to the impact of actions on others.)
- Digital Literacy (Has the capabilities that enable living, learning and working in a digital society. Comfortable with using digital media to communicate, solve problems, manage information, collaborate, create and share content.)
- Problem Solving (Use rational and logical reasoning to deduce appropriate and well-reasoned conclusions. Retain an open mind, optimistic of finding solutions, thinking laterally and creatively to look beyond the obvious. Knows how to learn from failure.)
- Critical Thinking (Recognise patterns, themes and key messages from sometimes confused and incomplete data. Make informed decisions on the value of a range of sources allowing an evidence based conclusion based on this analysis.)
- Professionalism (Prepared to operate autonomously. Aware of how to be efficient and resilient. Manages priorities and time. Self-motivated, setting and achieving goals, prioritising tasks)
- Information Literacy (Critical awareness of how information is gathered, used, managed and synthesised. Understanding of the relative value of different sources and the importance of provenance. The systematic collection, analysis and evaluation of information in the investigation of a topic.)

Study

Study time

Type	Required
Lectures	20 sessions of 1 hour (7%)
Seminars	5 sessions of 1 hour (2%)
Tutorials	5 sessions of 1 hour (2%)
Work-based learning	30 sessions of 1 hour (10%)
Online learning (scheduled sessions)	30 sessions of 1 hour (10%)
Online learning (independent)	10 sessions of 1 hour (3%)
Other activity	10 hours (3%)
Private study	70 hours (23%)
Assessment	120 hours (40%)
Total	300 hours

Private study description

Private study and independent learning possibly include:

- Pre-block reading list given on Moodle to encourage flipped learning approach
- Post-block activities released on Moodle.
- Post-block online Quizzes for revision
- Online forum for discussing queries with course peers and tutor.
- Online consulting session for providing one to one support to help struggling students.

Other activity description

Online support and consultancy in support of assessments and group project work.

Costs

No further costs have been identified for this module.

Assessment

You must pass all assessment components to pass the module.

Assessment group A1

	Weighting	Study time	Eligible for self-certification
Software Design Specification (Group)	40%	48 hours	No

This assessment is based on a group work to produce a group report on the design of a software system.

	Weighting	Study time	Eligible for self-certification
Software Documentation (Individual)	60%	72 hours	Yes (extension)

This assessment includes a written report and plus system development based on the report (a half of the assessment time required).

- Design of system architecture to effectively apply the requirement described on the report.
- Implementation of basic functionalities for the system.
- Unit testing for the basic functionalities

Feedback on assessment

Please use the following as guidance and update as necessary:

- verbal feedback given during seminar/tutorial sessions
- formative feedback on the individual contributions
- written feedback on the final group reports

Availability

Courses

This module is Core for:

- Year 2 of DWMS-H655 Undergraduate Digital and Technology Solutions (Cyber) (Degree Apprenticeship)
- Year 2 of DWMS-H652 Undergraduate Digital and Technology Solutions (Data Analytics) (Degree Apprenticeship)
- Year 2 of DWMS-H653 Undergraduate Digital and Technology Solutions (Network Engineering) (Degree Apprenticeship)
- Year 2 of DWMS-H654 Undergraduate Digital and Technology Solutions (Software Engineering) (Degree Apprenticeship)