Agile software development Syllabus

Module description

This module aims to provide insights and practice in software development using contemporary methods to produce software that meets the needs of users and supports an organisation's business function. The module will enable you to gain competence in the conceptualisation of a technology-based solution to a real-world problem, fulfilling the requirements of users and taking constraints imposed by the prevailing and foreseen market conditions and lessons learned from prototypes into account. Topics include:

1. Project management and team working

2. Requirements gathering and specification

3. Market and solutions research

4. User-centred design and prototyping

5. Project proposal

6. Agile development methodologies

7. Software testing

8. Software validation and user testing

9. Professional practice

10. Final project

Course structure and item numbers

Each topic spans over two weeks, e.g. Topic 1 is covered in week 1 and 2, etc. Each week consists of two or three lessons. A number of each lesson starts with the topic number, e.g. Lesson 6.0 in introductory lesson in topic 6.

A number of each activity based on the following format: lesson_number.item_number. For example, '6.01 What is and what is NOT agile?' is the first item in an introductory lesson in topic 6.

Module goals and objectives

The module goals are to introduce you to a variety of topics around the practicalities of software engineering including professional and agile practice and collaborative development. You will be given the opportunity to present a proposal for a technical project including a structured plan for implementing the solution using the agile development methodology and user-centred development practices. During the whole process from concept to solution presentation, you will be required to work in a distributed team using online collaboration, project tracking and version control tools.

Upon successful completion of this module, you will be able to:

1. Understand and apply the agile project management process applied to software development

2. Recognise the purpose of prototypes in answering open technical questions and in designing to meet user requirements

3. Devise technical tests and user tests to validate the functionality and performance of software

4. Work in a distributed team using appropriate online tools

5. Apply appropriate professional practices taking societal, ethical, marketing and technical constraints into account

Textbook and Readings

Specific essential readings for this module will be taken from the following text book and online resource:

Interaction Design: Beyond Human-Computer Interaction. Fifth edition.

Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban (chapter 1.)

The specific pages for the reading activities will be given in the platform, and there is no need to read beyond the recommended pages. In addition to the text book, there are additional activities written by the course author, some of which involve exploration of concepts through discussion or through iterative, collaborative development. There will also be discussion prompts asking you to do some independent research using online sources.

Module outline

The module consists of 10 topics, each of which spans two weeks.

Topic 1. Project management and teamworking

Key concepts:

• Describe progress in a linear way • Define milestones and dependencies in a project • Be able to build contingency into a workflow model • Delegate tasks into meaningful chunks of work • Be able to break a project down into deliverable components

Learning outcomes:

• Describe events and sequences of actions in a coherent manner • Manage risk • Manage assets and resources

Topic 2. Requirements gathering and specification

Key concepts:

• Formally define requirements for a project • Understand the critical path and the implications of divergence • Formally specify the component parts of the system • Consider the scope and limitations of your technology solution • Define and describe stakeholders

Learning outcomes:

• Consider the wider implications of building a system for purpose • Explore formal specifications from both a functional and technical perspective • Identify key stakeholders, challenges, risks and innovations

Topic 3. Market and solutions research

Key concepts:

• Explore the concept of research. • Understand the fundamental differences between qualitative and quantitative research. • Consider the scope of your project, including any limitations and your MVP. • Be able to produce a competitor analysis exploring where your concept fits in a broader context. • Explore techniques for generating your own research data.

Learning outcomes:

• Consider the wider implications of building a system for purpose • Identify interesting and valuable sources of information • Make decisions based around research findings

Topic 4. User-Centred Design and Prototyping

Key concepts:

• Explore the user-centred design methodology • Understand how to apply user-centred design methodologies to different types of projects • Consider prototyping as a process to refine your designs and improve the quality of your final product • Explore a variety of prototyping techniques, evaluating the results of these • Consider a strategy for continual improvement through iterative design cycles

Learning outcomes:

• Understand user-centred design and what it means to employ said methods and strategies • Consider how user-centred design techniques can be used to iteratively improve software quality • Formalise your findings and develop them as a series of prototypes

Topic 5. Project proposal

Key concepts:

• Consider your project proposal in terms of research, design and your prototype • Understand the role that user centred design plays in mitigating research, design and production • Describe your proposal in a coherent way • Explore the wider implications of your designs and proposal • Consider a strategy for continual improvement through iterative design cycles that will allow your intended design to come to fruition

Learning outcomes:

• Present research around your topic • Explore a series of iterative design cycles that culminate in a series of design propositions • Present your proposition to be assessed

Topic 6. Agile development methodologies Key concepts:

• Explore agile as a development methodology • Consider how projects are managed • Identify challenges and risks of project management approaches • Consider iterative development as a strategy for achieving goals • Explore techniques to manage risk, build in contingency and delegate responsibilities fairly

Learning outcomes:

• Appreciate the value in different project management methodologies • Understand the value of working in a collaborative way to converge on shared goals • Explore iterative development strategies

Topic 7. Software Testing Key concepts:

• Consider how your software might be used and misused • Explore strategies for identifying points of failure • Consider strategies for ensuring good software quality • Explore the types of inputs and outputs that your system will expect • Consider how individual components of your software will integrate with one another and challenges present therein

Learning outcomes:

• Understand some basic software testing techniques • Consider the wider implications of functional testing • Explore testing as a construct for reducing bugs

Topic 8. Software Validation and User Testing Key concepts:

• Consider evaluation techniques to identify how well your system works • Explore contexts of use and think about ways in which software can be used and misused • Identify key measures for success or failure in engineering software • Explore the wider implications of your designs and proposal • Consider a strategy for continual improvement through iterative design cycles that will allow your intended design to come to fruition

Learning outcomes:

• Processes for continual improvement of software • Explore a series of iterative design cycles with user feedback in mind • Consider approaches for formative and summative evaluation

Topic 9. Professional Practice Key concepts:

• Consider what it means to be a professional • Identify what it means to be a good team player • Explore risk and identify contingency techniques to overcome points of failure • Consider ethical design and development as desirable outcomes

Learning outcomes:

• Think about risks and potential points of failure in a system design lifecycle • Think about how your system will be used and the real world implications of design • Consider the implications for design of software systems

Topic 10. Final project Key concepts:

• Consider the lifecycle of a software product or service beyond delivery • Think about ways in which a deliverable piece of software can be supported • Critically reflect on development activities • Consider your own development lifecycle • Revise milestones and key deliverables, evaluating the success of your project holistically

Learning outcomes:

• Present your final deliverable • Consider your deliverable beyond delivery • Explore the wider implications of development software for purpose

Activities of this module

The module is comprised of the following elements:

1. Lecture videos will include overviews and discussions of topics.

2. Practice quizzes will be used to reinforce your learning and understanding.

3. Programming activities.

4. Discussions with your peers will help to guide your work and encourage you to explore different types of solutions to problems.

5. Readings will help to reinforce your learning of concepts.

6. Peer reviews to provide feedback to your classmates on a piece of work they produced.

How to pass this module

The module has two major assessments. The Preliminary report (project proposal) is worth 30% of your grade and your final project report and software will be worth 70% of your grade.