Course Proposal

COMP9820 Software Project Management

Code: COMP9820 (replacing GSOE9280 Engineering Project Management)

Study level: Postgraduate

Unit of Credit: 6

School: Computer Science and Engineering

Multi-term course: no

Proposed start date: T1 2026

Course Description

This course aims to introduce the students to the fundamental principles of modern management processes and practices, as well as tools and technologies for collaborative software development, deployment and release. The course will enable students to develop the knowledge and skills of managing software development in team environments. It will inspire students to be software development managers or team leaders. This course should equip students with the required knowledge and skills for carrying out industry-based capstone projects, managing project stakeholders and ensuring the delivery of software products that meet their desired needs.

Overview and Aims

- This course focuses on managing software projects from technical, social and professional perspectives.

- It emphasizes managing software process models following modern Agile software development practices, tools and technologies across all the software development lifecycle and beyond.

- The course will provide students with core principles of Agile development process to define, plan, build, deploy and maintain software applications collaboratively as a team.

- Students will develop knowledge and skills on how to work collaboratively in a team and manage software development projects with different agile roles.

- Students will also follow Agile social and technical practices, and various modern tools to support the planning, designing, development and management activities of software applications.

- The course will also cover essential techniques for managing the deployment and release of software applications in cloud environments and using modern software architectures.

- Students will apply the methods, practices, tools and technologies to plan, design, build, and deploy a software application as a team.

Pre-requisite

- Recommended: COMP9044 - Software Construction: Techniques and Tools
- Recommended: COMP6080 – Web Front-end Programming

Delivery Mode

IN-PERSON

- In-person and hybrid (lectures)
- In-person individual and group work (Laboratory)

Assumed knowledge

Students are expected to be familiar with the following knowledge and skills:

- O-O Design and Programming
- Web application design and development, including a technical proficiency of one or more programming frameworks (e.g. Python/JavaScript/Java)
- Database management systems;
- System Modelling;

Teaching Strategies and Rational

Lectures

Technical Lectures: will cover core concepts, methods, and practices which will introduce the students to fundamentals of modern software project management and related knowledge in the field.

Professional Lectures: will cover practical topics of managing software projects in practice (enterprise environment) which will exposure students to managing software projects in real-world development environment.

Laboratory Work

The laboratory will cover practical work to apply core concepts, methods and practices introduced in the lectures into real-world project scenarios and problems. This will involve hands-on exercises and tasks which will allow students to learn how to use and apply modern tools and practices that will support their learning. Students will also apply their learning from these exercises to manage an evolving project over multiple phases as a team.

Project Work

Students will work as a team to manage the development of a software project collaboratively through multiple phases including initiation, planning, executing, monitoring and delivery and deployment of a software product to production environment. Teams will need to apply management techniques as well as methodologies and practices such as Agile and DevOPs.

Schedule (Lecture Topics)

Week	Lecture Topics	Lab
W1	Course Introduction, Software Development Process Models & Lifecycle. Agile Software Development Introduction to DevOps, Managing Software Artefacts – Software Version Management with Git (sole development)	
W2	Advanced Software Version Management – Github Fundamentals (collaborative development), Issue Tracking and Management Software Configuration Management, Semantic Versioning, Software Build Automation	
W3	Software Quality Assurance, Software Testing Continuous Integration (CI), Continuous Delivery and Deployment (CD), Containerization	
W4	Cloud Service and Deployment models, Infrastructure as a Code (IaC) Scrum Method, Agile Management – Team Dynamics and Team Structure	
W5	Agile Management – Expressing Requirements (User Stories), UI Designs and Prototyping Agile Management – Planning, Estimation and Monitoring Progress	

W6	Flexi week	
W7	Software Design and Architecture, Microservices Architecture, Software Operations Management (Observability)	
W8	Ethics, Intellectual Property Management, and Open-Source Software, Licensing	
W9	API Development Practices – Managing third-party APIs in Software Development	
W10	Project Reflection and Lessons Learned	

Major Project

- Software requirements will be provided to all teams to design and implement a web-based application. Each team will create, plan, execute and manage the development infrastructure to design and build the web application.

Learning Outcomes

1. **Apply software project management principles** to effectively plan, execute, monitor, and deliver software applications using Agile and DevOps methodologies.

2. **Demonstrate technical proficiency in managing collaborative software development**, including version control (Git/GitHub), issue tracking, and configuration management.

3. **Implement software quality assurance practices**, including automated testing, continuous integration (CI), and continuous deployment (CD), to improve software reliability.

4. **Manage cloud-based software deployment and release processes**, leveraging Infrastructure as Code (IaC) and cloud service models.

5. **Analyze and apply Agile project management techniques**, including Scrum methodologies, user story development, sprint planning, and progress tracking.
6. **Work effectively in software development teams**, assuming Agile roles (e.g., Scrum Master, Product Owner, Developer) and applying best practices in team dynamics and leadership.

7. **Design and manage scalable software architectures**, including microservices and API-driven systems, to support modern software development practices.

8. **Address ethical, legal, and professional responsibilities** in software project management, including intellectual property management, licensing, and open-source contributions.

9. **Critically reflect on project management strategies** and lessons learned from real-world software projects to improve future practices.

Assessments

- **Group project work with multiple phases (3 Sprints/phases)**
    - **Project Work (Sprint 1):**
        - Mode: group work
        - Task: creating and managing software development infrastructure using CI/CD (DevOps)
        - Deliverables: technical group report and demonstration of work during lab sessions
        - Weight: 25%
        - Due: Week 4

    - **Project Work (Sprint 2):**
        - Mode: group work
        - Task: managing the development of a software application using Scrum and the development infrastructure created by the team
        - Deliverables: technical group report and demonstration of work during lab sessions
        - Weight: 25%
        - Due: Week 7

    - **Project Work (Sprint 3):**
        - Mode: group work
        - Task: evolving and managing the software application, delivery, deployment and monitoring
        - Deliverables: technical group report and demonstration of work during lab sessions
        - Weight: 30%
        - Due: Week 10

- **Peer assessment and Individual contributions (20%)**
    - Mode: individual
    - Tasks:
        - Individual learning activities (checkpoints during lab sessions and project demonstration)
        - Evaluation of group members and their contributions to the project work (3 Sprints)
        - Self-reflection of own work (portfolio summary)
    - Deliverables: individual report and presentation
    - Weight: 20%
    - Due: weeks 4, 7, and 10

Strategic alignment:

The course is required to meet the accreditation requirements. It also supports the transformation of capstone projects into industry-based projects.

Distinctive feature

This course will cover the knowledge and skills required for carrying industry-based capstone projects. It will help students to plan, execute and manage technical and social aspects of software development activities to deliver quality software applications for a client.

Contribution to Learning Outcomes of Programs and Specialisations

The course will cover the modern software development management practices, tools and technologies for MIT