Aston University

Module Information for CS2TP, 2022/3 - APPROVED

Module Title/Name: **Team Project**	Module Code: **CS2TP**
School: **Engineering & Physical Sciences**	Module Type: **Standard Module**
New Module? **No**	Module Credits: **30**

Module Management Information

Module Leader Name	Shereen Fouad
Email Address	s.fouad@aston.ac.uk
Telephone Number	Not Specified
Office	Not Specified

Additional Module Tutor(s):
Sherazade Boussag. (s.boussag@aston.ac.uk,)Shereen Fouad. (s.fouad@aston.ac.uk,)

Level Description: Level 5 (Foundation Degree/Dip He)	**Programmes in which available:** Computer Science with Business Computer Science Cybersecurity
Contributing School & Subject Groups (and %): None Specified	**Credit Value & ECTS Credits:** 15
Available to Exchange Students? No	

Module Dependencies

Pre-requisites: (CS1OOP). (CS1FOP). (CS1IAD).	**Co-requisites:** Software Engineering (CS2SE).
Prohibited Combinations: None Specified	

Module Learning Information

Module Aims:
To develop the skills and attitudes necessary to effectively produce a complete, high-quality software system, including team-working, project management, reflective practice, software engineering, delivery, and quality assurance.

Module Learning Outcomes:

On successful completion of the module students will be able to:

- 1. Apply all phases of the software engineering lifecycle in the production of a complete, high-quality software product for an identified need. Delivering the product effectively will require effective teamwork, project management, communication, value generation, and other appropriate teamworking and software development skills.
- 2. Present, explain and justify their work, including design decisions made, in the context of effective software engineering practice and relevant computer science theory.
- 3. Analyse, evaluate, and critically reflect on their team's application of the software engineering lifecycle, as well as the team's culture and behaviour, in order to formulate their own strategies for continuous ongoing learning in the context of future team-based experiences.
- 4. Develop, justify, implement, and evaluate a strategy to ensure software quality, including quality assurance processes, an appropriate test plan, use of appropriate tools for the support of effective team-based software engineering activity (e.g. version control, task and bug trackers, automated testing and deployment).

Indicative Module Content:
During the first term, students are introduced via classroom sessions to collaborative software development concepts (e.g. version control with Git, issue tracking, agile project management using Scrum/Kanban, and software quality assurance including continuous integration approaches), and practice these concepts in pairs. Students are then given a domain for a software product to develop in the remaining weeks of the first term, which they are expected to design and develop according to good software engineering principles in small teams.
&nbsp;
During the second term, students will use what they learned during their first project to work on a second software product in a similar domain, but with a larger team and scope.
&nbsp;
Software engineering activities to be demonstrated throughout the module include requirements capture, prototyping, implementation, testing and troubleshooting detected issues, deployment, maintenance, and extension. Various artifacts are expected to be produced at different times during the module, according to the software engineering methodology employed. Students work largely independently in small teams, and receive support during weekly laboratory sessions.

International Dimensions:
N/A

Corporate Connections:
Through links with the Computer Science Industry Club, students are provided
with the opportunity to present to, and receive feedback from, representatives from
the software engineering industry.

Ethical Approval:
N/A

Links to Research:
N/A

Ethics, Social Responsibility and Sustainability:
N/A

Module Delivery

Methods of Delivery & Learning Hours (by each method):

Method of Delivery	Learning Hours
Large Group Activity:	14 hours
Specialist Session Activity:	5 hours
Independent Activity:	233.5 hours
Structured Activity:	47.5 hours
Total Learning Hours:	**300 hours**

Learning & Teaching Rationale:

Large Group Activity

Lectures will teach the theory surrounding collaborative software development, including version control, task tracking, software quality assurance, and agile project management. Additional support lectures may be delivered based on the domain chosen for the projects.

Specialist Session Activity

During term 2, students will be asked to demonstrate their second project at the MVP stage and at the completed product stage through recorded Collaborate sessions (which will replace their lab for the week).

Structured Activity

During the 2.5-hour weekly labs, students will initially gain practice with the tools and processes enabling collaborative software development theory, producing a training portfolio to be done in pairs. After this training period, students will use the labs to advance their projects in teams while having access to the module staff for support on their project management, teamwork, and technical work.

Independent Activity

The majority of the module will be conducted through independent experiential learning, as team-based project work.

Module Assessment

Methods of Assessment & associated weighting (including approaches to formative assessment as well as summative):

Assessment Type	**Category**	**Duration/ Submission Date**	**Common Modules/ Exempt from Anonymous Marking**	**Assessment Weight**
Project	**Group Assignment**	**TBC**	**No**	**80%**
Details	Project: A structured portfolio of submissions, including project plans, demonstrations, source code and a final report of approximately 7,500 words. These are delivered at intervals through the module by the team as whole. The portfolio is designed as an opportunity for students to provide evidence against the module's learning outcomes. Students will be provided worksheets to complete in pairs for their initial training. For the team-based projects, students will follow detailed submission specifications provided at the beginning of the module, which will set out the evidence that students are expected to provide. The evidence in the portfolio is used to identify achievement in terms of four criteria: 1. Software Quality & Scope, which encompasses everything about the software artifact itself, and is assessed primarily through the demonstration of the software. 2. Team Working, which includes aspects relating to team culture, leadership, team roles, peer-learning and team development. 3. Project Management & Process, which includes the software engineering process, quality assurance, project management, and use of task and bug tracking tools, and version control. 4. Exposition & Impact, of both written reports and demonstrations, and relationship with the stakeholders. In addition, the portfolio and other sources of evidence are used to identify individual students' contributions to the team's work. A student's mark for this element of assessment, therefore, reflects i) the overall performance of the team throughout the project, and ii) the student's contribution to that overall performance.			
Reflective Learning	**Individual Assignment**	**TBC**	**No**	**20%**
Details	Reflective learning: 3 short critical reflections, including a final reflection on the project as a whole. Delivered at intervals through the project, for regular feedback, as well as a final reflection submitted at the end of the module. This element assesses a student's ability to engage in regular effective reflective practice, based on the evidence in these submissions. Students are expected to articulate their learning in terms of the learning outcomes, and aspects captured in the work leading to their portfolio, in a structured way. Reflections should, therefore, articulate the student's learning in terms of teamwork, project management, software development, and the impact their work creates.			

Total:	**100%**

Method of Submission:
Electronic Copy Only

Assessment Rationale:
The assessment is as follows:
1. Team Element, assessed through a portfolio of project work produced by the pairs and teams (80%). Most learning outcomes will be assessed by this component of assessment.
2. Individual Element, assessed through a series of pieces of reflective writing (20%). Here we assess their ability to analyse, evaluate, and critically reflect on their team's application of the software engineering lifecycle, as well as the team's culture and behaviour. Both of these will be continuously assessed throughout the module. Element 1 is designed to embrace the range of work that effective software project teams typically engage in. Students are expected to be active members of the team and engage in project work in order to access learning opportunities. Students are then expected to reflect on their experiences both formally and informally, to identify and articulate what they have learned, and to feed this back into their work to enhance its effectiveness. This form of reflective practice is a key skill in professional individual and team-based work and is captured in Element 2.

Feedback Rationale:
During term 1, students will receive feedback throughout the year at various stages of their project, including their project proposal, their team retrospectives, and their progress on the development of the project. The feedback on their project proposal and team retrospectives will be given through discussion at the weekly laboratories.

At the beginning of term 2, students will receive written feedback on the outcome of their first project, with suggestions on what to improve on for their second project. The laboratory sessions will be used again for informal formative feedback on their second project. Written feedback will be given for their MVP demos in the middle of the term, to help them steer their projects.
Throughout the module, students will also receive cohort-level formative feedback about the submitted reflections, addressing common issues and generally observed themes in their reflections.