UNSW Course Outline

SENG2021 Requirements and Design Workshop - 2024

Published on the 28 Jan 2024

General Course Information

Course Code : SENG2021
Year : 2024
Term : Term 1
Teaching Period : T1
Is a multi-term course? : No
Faculty : Faculty of Engineering
Academic Unit : School of Computer Science and Engineering
Delivery Mode : In Person
Delivery Format : Standard
Delivery Location : Kensington
Campus : Sydney
Study Level : Undergraduate
Units of Credit : 6

Useful Links

Handbook Class Timetable

Course Details & Outcomes

Course Description

This course is the first of the workshops series that are a key feature of the Software Engineering degree. These workshops are designed to teach students to work in teams (or groups) using agile methods and apply their theoretical knowledge to solve real-life problems. Building on foundational Software Engineering principles of first year core course COMP1531, this workshop focuses on the processes of designing and building components (APIs) of a large software application.

This workshop will offer students the opportunity to get practical experience on the activities of analysing software requirements and producing design artifacts as well as use collaborative techniques and tools to support these processes. In addition, it aims to provide students with teamwork skills and software design techniques that an engineer would use in the development process. The students will also be getting experience on different aspects of designing and developing an API microservice and a web-based application. The requirements for this course have been determined in collaboration with industry partners and will relate to developing a realistic application. The course has a number of industry sponsors that include Nine and Macquarie Bank.

Course Aims

SENG2021 is the first Software Engineering-specific workshop that builds on the foundation of COMP1531, and where we begin to answer the question "What does it mean to be a Software Engineer?". A critical element of the workshop is to provide teams of students with industry-proven collaborative software development tools from Atlassian (Jira and Confluence) to help them apply agile methods in practice. The students will also get experience in project management with the development of project plans, brainstorming, requirement documents, prototyping techniques, issues and tasks management and peer reviews. Throughout a realistic business application in the area of e-invoicing, they will learn about the processes of converting requirements to design artifacts in a realistic context and acquire practical design skills, particularly in architectural design and software component integration using APIs.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Explain the principles and processes involved in the software development life cycle
CLO2 : Synthesise and interpret requirements to create tangible architectural and technical designs in a team-based dynamic development environment
CLO3 : Understand stack-based software architecture and acquire skills in full-stack design and construction
CLO4 : Build software systems using appropriate languages, libraries and frameworks
CLO5 : Understand and apply the principles of project management in a dynamic software development environment
CLO6 : Contribute to system documentation for a software product

Course Learning Outcomes	Assessment Item
CLO1 : Explain the principles and processes involved in the software development life cycle	• Sprint 1: Planning a Service • Sprint 4: Moving a Proof of Concept towards a Product • Final Portfolio Submission
CLO2 : Synthesise and interpret requirements to create tangible architectural and technical designs in a team-based dynamic development environment	• Sprint 2: Architecting a Service • Sprint 1: Planning a Service
CLO3 : Understand stack-based software architecture and acquire skills in full-stack design and construction	• Sprint 3: Building a Proof of Concept • Sprint 2: Architecting a Service
CLO4 : Build software systems using appropriate languages, libraries and frameworks	• Sprint 3: Building a Proof of Concept • Sprint 4: Moving a Proof of Concept towards a Product • Sprint 2: Architecting a Service
CLO5 : Understand and apply the principles of project management in a dynamic software development environment	• Sprint 1: Planning a Service • Sprint 3: Building a Proof of Concept • Sprint 4: Moving a Proof of Concept towards a Product
CLO6 : Contribute to system documentation for a software product	• Final Portfolio Submission • Sprint 4: Moving a Proof of Concept towards a Product

Learning and Teaching Technologies

webCMS, Confluence

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Sprint 1: Planning a Service Assessment Format: Group	10%	Due Date: Week 3 Monday, 1pm
Sprint 2: Architecting a Service Assessment Format: Group	20%	Due Date: Week 5 Monday, 1pm
Sprint 3: Building a Proof of Concept Assessment Format: Group	20%	Due Date: Week 8 Monday, 1pm
Sprint 4: Moving a Proof of Concept towards a Product Assessment Format: Group	35%	Due Date: Week 10 Wednesday 1pm
Final Portfolio Submission Assessment Format: Individual	15%	Due Date: Week 11 Wednesday, 1pm

Assessment Details

Sprint 1: Planning a Service

<u>Assessment Overview</u>

Sprint 1 is all about planning and designing. Teams will be planning work up until the end of Sprint 2 (Week 5). The learning outcomes of this sprint can be broken down into four major pillars:

1. **Requirements Engineering**. To succeed in creating a solution, teams first need to understand the problem they're solving. This means analysing the problem domain and defining requirements for the selected service/API.

2. **Designing a Microservice**. Teams will need to decide on the **stack** for their microservice - what technologies they'll use at each layer to build and deploy a working prototype, and defining the technical requirements (interface) of the selected API.

3. **Managing a project**. Using Jira and Confluence, industry-grade tools, teams will be tracking and managing tasks and team progress.

4. **Working as a Scrum team**. In adopting Scrum team roles and following Scrum rituals - standups and retrospectives, teams will be working in the same fashion as real-world software teams. They'll have to work through issues as they arise - technical and people related! This is

all a normal part of real Software Engineering.

Sprint 1 is expected to take each team a 2-week effort and will be marked by the mentor.

The course authority may adjust the marks of team members based on the contribution of each individual towards the project. Team members have the opportunity to address issues as they arise and raise them with their tutor. Issues that persist can be escalated by emailing the course account.

Teams are expected to raise issues in teamwork with their tutor as they arise; there is no 'peer review' process at the end of the project, instead the 'peer review' is ongoing throughout the term.

Course Learning Outcomes

- CLO1 : Explain the principles and processes involved in the software development life cycle
- CLO2 : Synthesise and interpret requirements to create tangible architectural and technical designs in a team-based dynamic development environment
- CLO5 : Understand and apply the principles of project management in a dynamic software development environment

Sprint 2: Architecting a Service

Assessment Overview

In Sprint 2, teams will be building an MVP of the selected microservice/API. The learning outcomes of this Sprint can be broken down into three major pillars:

1. **Iterating on the Design**. Teams will have received feedback about their design from Sprint 1 - they will need to iterate and improve. Teams will also need to start going deeper into the detailed design by completing a data model for their application.

2. **Building an MVP**. *Fast, good, cheap. Pick two*. Teams only have two weeks to build the service. They need to start simple and build outwards. It is suggested for teams to first get a basic “hello world” API working and deploy that. Then add in more functionality and keep iterating on the MVP.

3. **Mastering the Craft**. Software Engineering is about so much more than just coding - writing clean code, testing, deploying, documenting and working as an Agile team. Teams will be building your skills in these areas that will help you in future Sprints and beyond.

Sprint 2 is expected to take each team a 2-week effort and will be marked by the mentor.

The course authority may adjust the marks of team members based on the contribution of each individual towards the project. Team members have the opportunity to address issues as they arise and raise them with their tutor. Issues that persist can be escalated by emailing the course account.

Teams are expected to raise issues in teamwork with their tutor as they arise; there is no 'peer review' process at the end of the project, instead the 'peer review' is ongoing throughout the term.

Course Learning Outcomes

- CLO2 : Synthesise and interpret requirements to create tangible architectural and technical designs in a team-based dynamic development environment
- CLO3 : Understand stack-based software architecture and acquire skills in full-stack design and construction
- CLO4 : Build software systems using appropriate languages, libraries and frameworks

Sprint 3: Building a Proof of Concept

Assessment Overview

The learning outcomes of this Sprint can be broken down into four major pillars.

1. **Defining a Product**. In this Sprint teams will need to find a **target market** to **solve a real-world problem**, and define a product to provide value to end users. This will involve Requirements Engineering and planning work with Jira.

2. **Pitching a Proof of Concept**. Major projects begin with an idea, and a proof of concept. This is a working prototype which the team demonstrates to stakeholders to demonstrate the value of the idea. The stakeholders can then decide whether to further invest (budget, resources) in the idea. We're calling this a Shark Tank pitch in this course.

3. **Iterating on your API** - software doesn't stop once it's deployed; that's when it starts. Teams will have the chance to work on leftover work from Sprint 2, and improve their API by iterating on it.

4. **Full Stack & Frontend Development**. For many teams, this'll be their first foray into frontend programming! By the end of this Sprint, teams will have built and deployed a full stack application.

Sprint 3 is expected to take each team a 2-week effort and will be marked by the mentor and an additional marker.

The course authority may adjust the marks of team members based on the contribution of each individual towards the project. Team members have the opportunity to address issues as they arise and raise them with their tutor. Issues that persist can be escalated by emailing the course account.

Teams are expected to raise issues in teamwork with their tutor as they arise; there is no 'peer review' process at the end of the project, instead the 'peer review' is ongoing throughout the term.

Course Learning Outcomes

- CLO3 : Understand stack-based software architecture and acquire skills in full-stack design and construction
- CLO4 : Build software systems using appropriate languages, libraries and frameworks
- CLO5 : Understand and apply the principles of project management in a dynamic software development environment

Sprint 4: Moving a Proof of Concept towards a Product

Assessment Overview

In this sprint, teams will be expected to:

1. Finalise all elements of their design for the selected Service API and Application, and compile a written report;

2. Continue developing and maintaining the Service API and make it available to other teams;

3. Continue developing and maintaining their Application, moving it from a proof of concept towards a usable product; and

4. Prepare a follow-up “Shark Tank” pitch of their product for stakeholders

Sprint 4 is expected to take each team a 3-week effort and the demo will be marked by the mentor and additional marker. All reports will be independently marked by the LIC.

The course authority may adjust the marks of team members based on the contribution of each individual towards the project. Team members have the opportunity to address issues as they arise and raise them with their tutor. Issues that persist can be escalated by emailing the course account.

Teams are expected to raise issues in teamwork with their tutor as they arise; there is no 'peer review' process at the end of the project, instead the 'peer review' is ongoing throughout the term.

Course Learning Outcomes

- CLO1 : Explain the principles and processes involved in the software development life cycle
- CLO4 : Build software systems using appropriate languages, libraries and frameworks
- CLO5 : Understand and apply the principles of project management in a dynamic software development environment
- CLO6 : Contribute to system documentation for a software product

Final Portfolio Submission

Assessment Overview

The portfolio submission will need to contain for each student three on-line pages (using Confluence platform), one on each of:

1. Time & Project Management
2. Technical Maturity
3. Leadership & Teamwork

These pages are continuously maintained by students throughout the course by adding personal reflections on how the course is going. Students can include links to relevant forum / blog posts related to the portfolio content. Students are expected to spend around 2 hours a week maintaining their portfolios.

Students are encouraged to provide feedback on each other's portfolios occasionally. All portfolios are marked by the mentors at the end of the course.

Course Learning Outcomes

- CLO1 : Explain the principles and processes involved in the software development life cycle
- CLO6 : Contribute to system documentation for a software product

General Assessment Information

Grading Basis

Standard

Course Schedule

Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

General Schedule Information

- Week1:
  - Overview of Workshop
  - Intro to e-invoicing
  - Announcements (Team Formation)
  - Software Stacks (Guest Lecture)
  - Application Programming Interfaces (Guest Lecture)
- Week2:
  - Software Security
  - Cloud Services (Guest Lecture)
- Week3:
  - [TBA]
- Week5:
  - [TBA]
- Week7:
  - Detailed Design
  - [TBA]

The schedule above may be subject to change depending on the availability of our industry partners. For the most current version of the course schedule, please refer to the schedule on our webCMS page.

Course Resources

Prescribed Resources

Relevant resources will be provided throughout the term in the "Self Learning Resources" tab on webCMS.

Course Evaluation and Development

The course is evaluated each session using the myExperience system. However, students are encouraged to provide feedback throughout the term, including during lectures, project check-ins, by contacting course staff via email and completing the feedback form linked in the sidebar.

The following changes have been implemented from the previous offering in 23T1:

- Students are given the flexibility in choosing their method of API deployment, provided it is well-documented and accessible.
- Guest lectures are tailored to meet the specific needs of our students.
- The use of Generative AI tools, particularly ChatGPT, is now permitted. Students are urged to explore different tools and share their experiences with the class. However, it is essential for students to confirm their use of these modern tools early on with their mentors to prevent unintentional plagiarism.
- Sprint specs will be released earlier in the term to facilitate better planning.
- The marking criteria for Sprint4 has been revised to enhance clarity and simplicity for both students and mentors.
- At the beginning of each week, the Lecturer in Charge will address any feedback and requests submitted through the feedback form.
- The ed forum platform will be accessible from the start of the term to facilitate early engagement.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Armin Chitizadeh					Yes	Yes

Other Useful Information

Academic Information

I. Special consideration and supplementary assessment

If you have experienced an illness or misadventure beyond your control that will interfere with your assessment performance, you are eligible to apply for Special Consideration prior to, or within 3 working days of, submitting an assessment or sitting an exam.

Please note that UNSW has a Fit to Sit rule, which means that if you sit an exam, you are declaring yourself fit enough to do so and cannot later apply for Special Consideration.

For details of applying for Special Consideration and conditions for the award of supplementary assessment, please see the information on UNSW's Special Consideration page.

II. Administrative matters and links

All students are expected to read and be familiar with UNSW guidelines and polices. In particular, students should be familiar with the following:

- Attendance
- UNSW Email Address
- Special Consideration
- Exams
- Approved Calculators
- Academic Honesty and Plagiarism
- Equitable Learning Services

III. Equity and diversity

Those students who have a disability that requires some adjustment in their teaching or learning environment are encouraged to discuss their study needs with the course convener prior to, or at the commencement of, their course, or with the Equity Officer (Disability) in the Equitable Learning Services. Issues to be discussed may include access to materials, signers or note-takers, the provision of services and additional exam and assessment arrangements. Early notification is essential to enable any necessary adjustments to be made.

IV. Professional Outcomes and Program Design

Students are able to review the relevant professional outcomes and program designs for their streams by going to the following link: https://www.unsw.edu.au/engineering/student-life/student-resources/program-design.

Note: This course outline sets out the description of classes at the date the Course Outline is published. The nature of classes may change during the Term after the Course Outline is published. Moodle or your primary learning management system (LMS) should be consulted for the up-to-date class descriptions. If there is any inconsistency in the description of activities between the University timetable and the Course Outline/Moodle/LMS, the description in the Course Outline/Moodle/LMS applies.

Academic Honesty and Plagarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. *Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.*

Plagiarism is a type of intellectual theft. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement. UNSW has produced a website with a wealth of resources to support students to understand and avoid plagiarism, visit: student.unsw.edu.au/plagiarism. The Learning Centre assists students with understanding academic integrity and how not to plagiarise. They also hold workshops and can help students one-on-one.

You are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and the proper referencing of sources in preparing all assessment tasks.

Repeated plagiarism (even in first year), plagiarism after first year, or serious instances, may also be investigated under the Student Misconduct Procedures. The penalties under the procedures can include a reduction in marks, failing a course or for the most serious matters (like plagiarism in an honours thesis or contract cheating) even suspension from the university. The Student Misconduct Procedures are available here:

www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf

Submission of Assessment Tasks

Work submitted late without an approved extension by the course coordinator or delegated authority is subject to a late penalty of five percent (5%) of the maximum mark possible for that assessment item, per calendar day.

The late penalty is applied per calendar day (including weekends and public holidays) that the assessment is overdue. There is no pro-rata of the late penalty for submissions made part way through a day. This is for all assessments where a penalty applies.

Work submitted after five days (120 hours) will not be accepted and a mark of zero will be awarded for that assessment item.

For some assessment items, a late penalty may not be appropriate. These will be clearly indicated in the course outline, and such assessments will receive a mark of zero if not completed by the specified date. Examples include:

- Weekly online tests or laboratory work worth a small proportion of the subject mark;
- Exams, peer feedback and team evaluation surveys;

- Online quizzes where answers are released to students on completion;
- Professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date; and,
- Pass/Fail assessment tasks.

Faculty-specific Information

Engineering Student Support Services – The Nucleus - enrolment, progression checks, clash requests, course issues or program-related queries

Engineering Industrial Training – Industrial training questions

UNSW Study Abroad – study abroad student enquiries (for inbound students)

UNSW Exchange – student exchange enquiries (for inbound students)

UNSW Future Students – potential student enquiries e.g. admissions, fees, programs, credit transfer

Phone

(+61 2) 9385 8500 – Nucleus Student Hub

(+61 2) 9385 7661 – Engineering Industrial Training

(+61 2) 9385 3179 – UNSW Study Abroad and UNSW Exchange (for inbound students)

School Contact Information

CSE Help! - on the Ground Floor of K17

- For assistance with coursework assessments.

The Nucleus Student Hub - https://nucleus.unsw.edu.au/en/contact-us

- Course enrolment queries.

Grievance Officer - grievance-officer@cse.unsw.edu.au

- If the course convenor gives an inadequate response to a query or when the course convenor does not respond to a query about assessment.

Student Reps - stureps@cse.unsw.edu.au

- If some aspect of a course needs urgent improvement. (e.g. Nobody responding to forum queries, cannot understand the lecturer)