COM1001 Introduction to Software Engineering Semester 2 Team Project

Dr Donghwan Shin

Senior Lecturer

Agenda

Team Project

- Scrum
- Project brief
- GitLab live demo

Assessment

- Assessment subjects
- Marking criteria

Ask questions during the lecture

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
HKCZBZ

Enable answers by SMS

University of Sheffield

Team Project

Project Goal

Implement a web-based system based on the user stories your team collected in the Autumn semester.

Work in a team to implement, refactor, and test a software system.

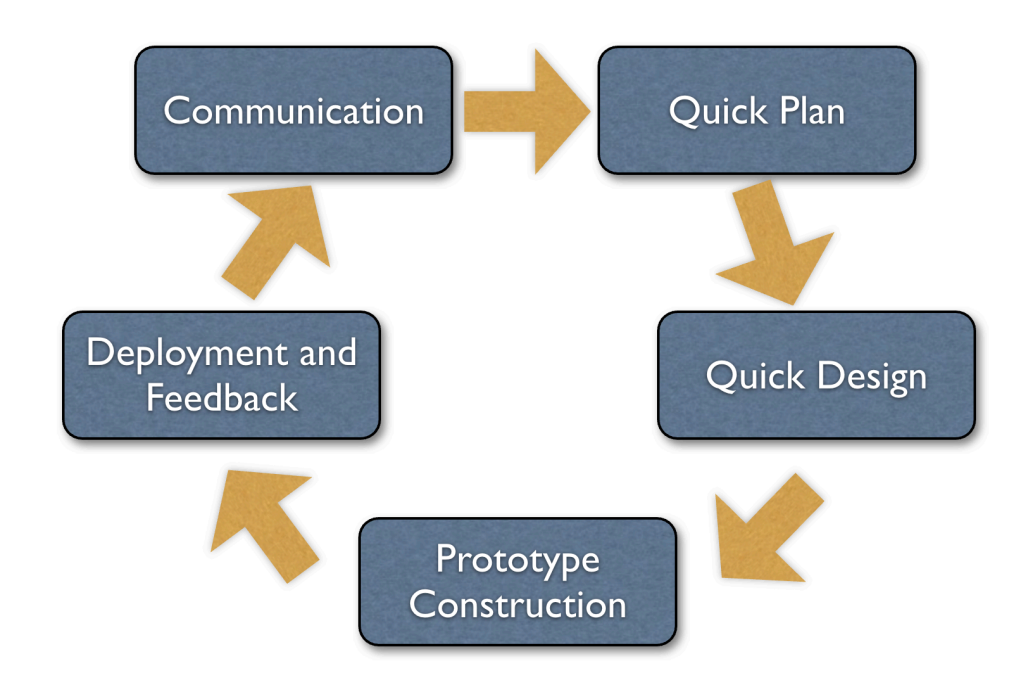
Agile Development Process

Iterative & Incremental

Customer Collaboration

Adaptability

Teamwork & Communication



From Week 3 in Semester 1

Scrum

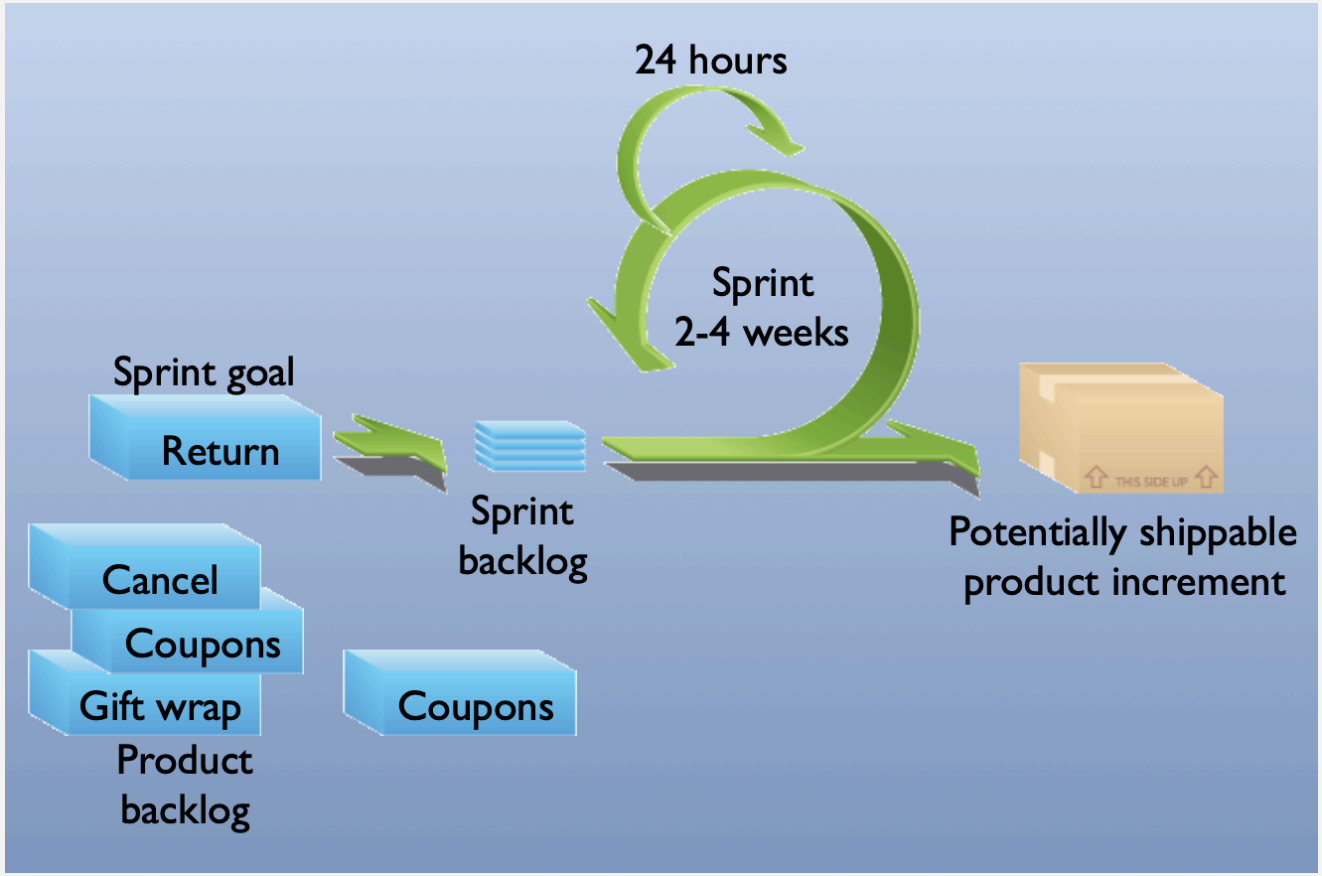
Autumn semester, Week 3

Part of Agile methodologies

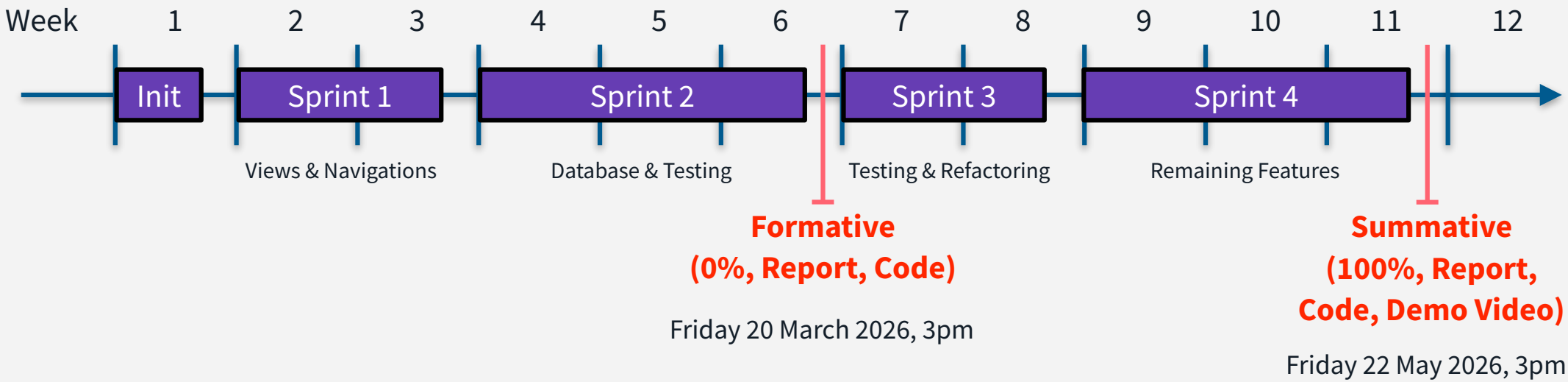
Self-organising team of 5-9

Product owner, scrum master, product backlog, sprints, burndown charts, daily scrum, etc.

Essential in the industry.



Weekly Schedule based on mini-Scrum



After the final submission, students will do a **peer assessment** (BuddyCheck) for individual scailing.

Key Points in mini-Scrum

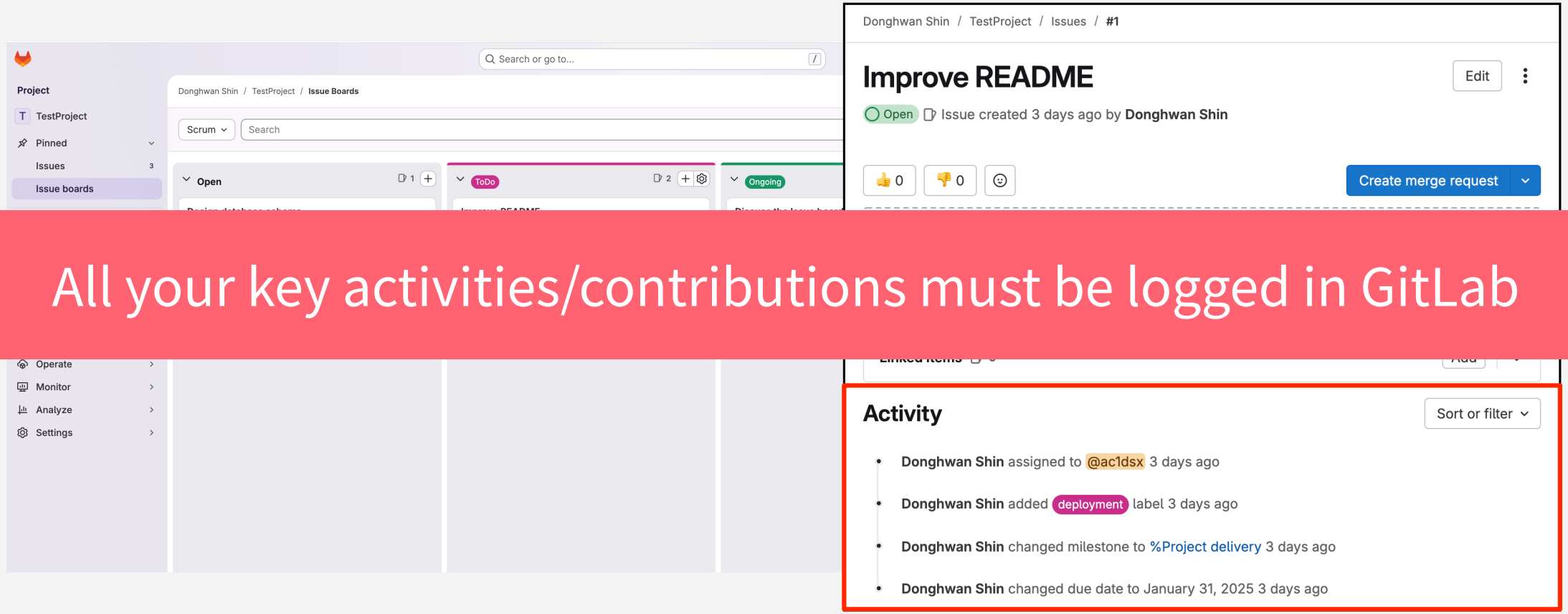
Scrum master (team lead)

- Lead scrum events (sprint planning, individual stand-ups, and sprint review).
- Help clear “blockers” to keep the team productive.
- No need to have the same lead throughout the semester.

Weekly Scrum

- A weekly meeting is timetabled (Tuesdays), but your team may have more meetings regularly.
- Individual **stand-ups** (max 3 min each): Each team member must share what they did, what they will do, and any blockers they have.
- Do ***not*** interrupt the stand-ups; any blockers should be discussed ***after*** completing the stand-ups.

GitLab Issues & Issue boards (subject to assessment)



All your key activities/contributions must be logged in GitLab

https://docs.gitlab.com/ee/user/project/issue_board.html

https://youtu.be/-4SGhGpwZDY?si=kOySFbbhWxykNwY2

GitLab Exercise

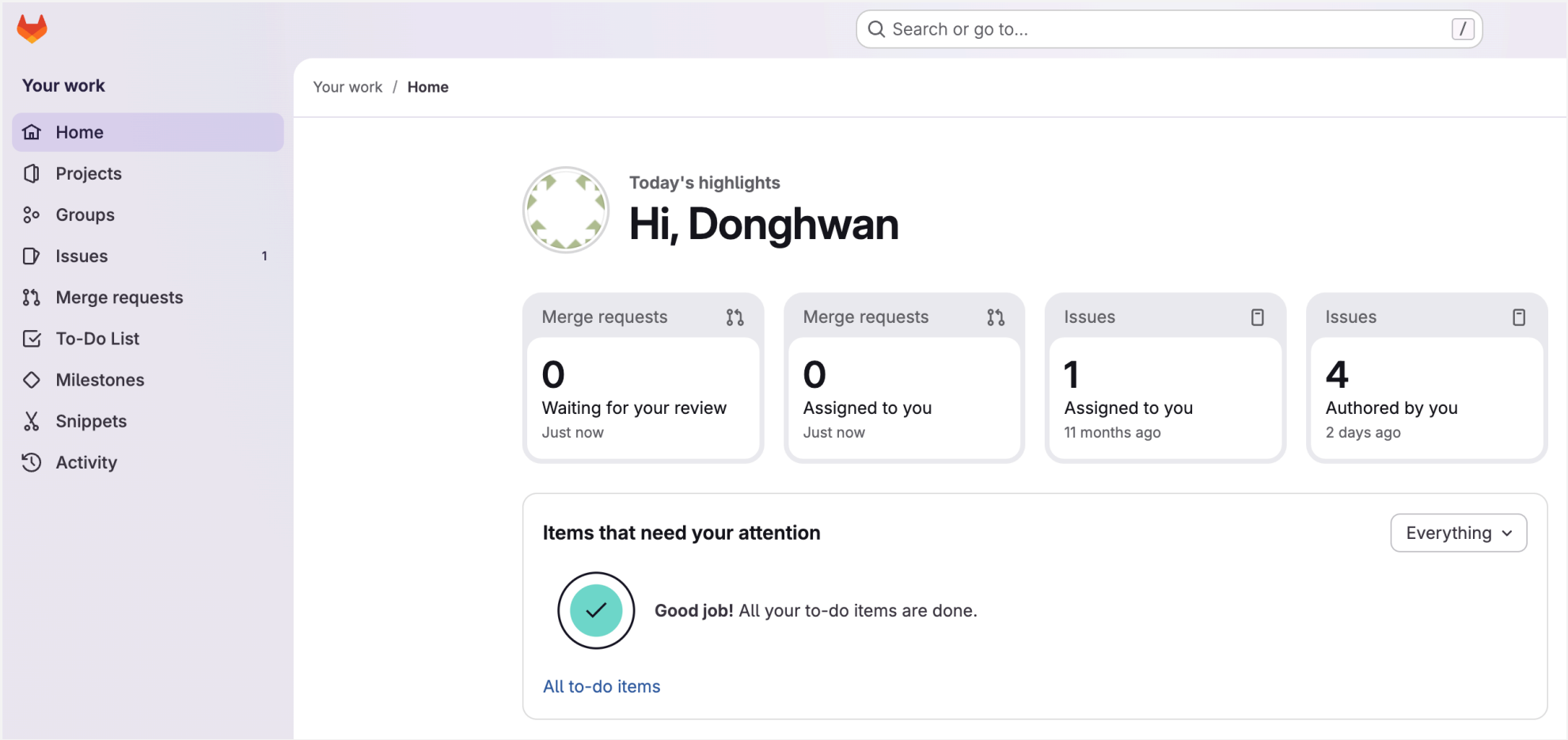
1. Go to your team project repository in GitLab.
2. Go to Manage > Labels, and create “ToDo”, “Ongoing”, and “Meeting Notes”.

   Only one team member needs to do this; no need to repeat.
3. Go to Plan > Issues, and create an Issue (and try Assignee, Labels, Dates).

   You can create as many as you want.
4. Go to Plan > Issue boards, and click “New list” to create Todo and Ongoing lists.

   Only one team member needs to do this; no need to repeat.
5. Try drag & drop issues between lists, and open & edit issues in the board.
6. Remove the test issues just created to make a clean start later.

Live Demo


Search or go to...
/
Your work
Home
Projects
Groups
Issues
1
Merge requests
To-Do List
Milestones
Snippets
Activity
Your work / Home
Today's highlights
Hi, Donghwan
Merge requests
0
Waiting for your review
Just now
Merge requests
0
Assigned to you
Just now
Issues
1
Assigned to you
11 months ago
Issues
4
Authored by you
2 days ago
Items that need your attention
Everything
Good job! All your to-do items are done.
All to-do items

Ask questions during the lecture

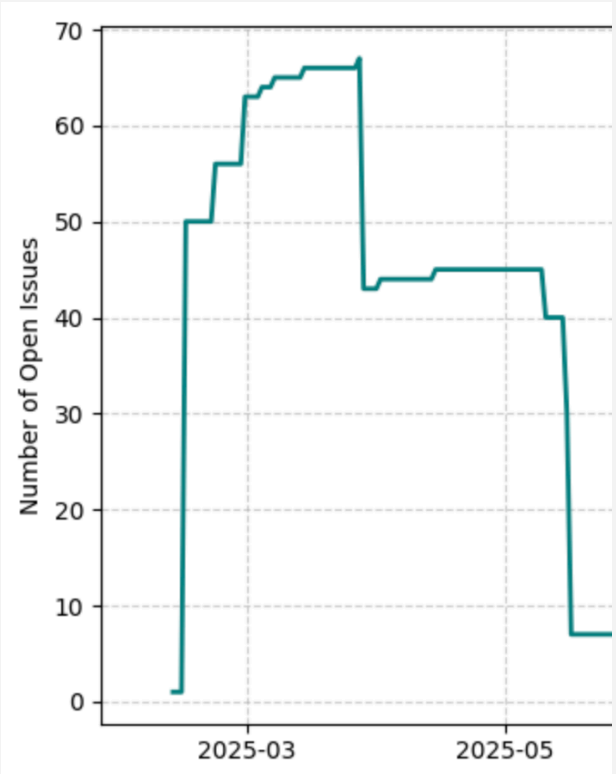
1 Go to wooclap.com

2 Enter the event code in the top banner

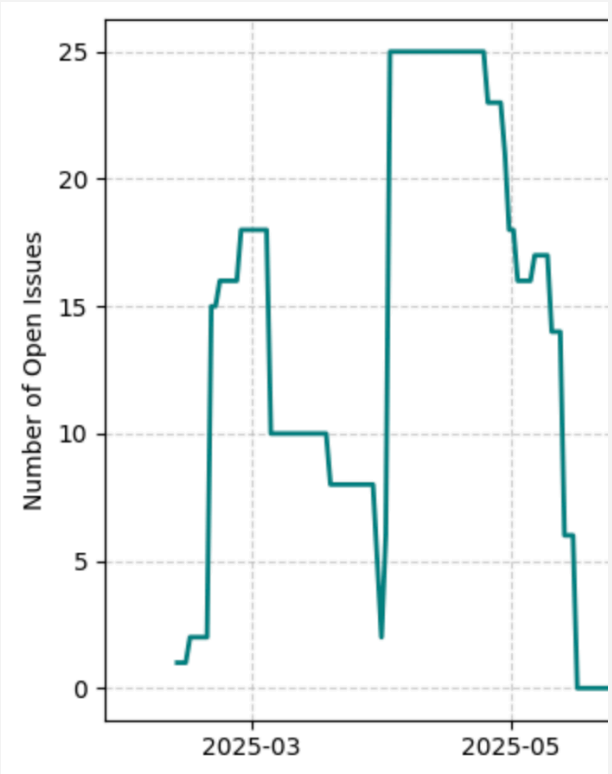
Event code
HKCZBZ

Enable answers by SMS

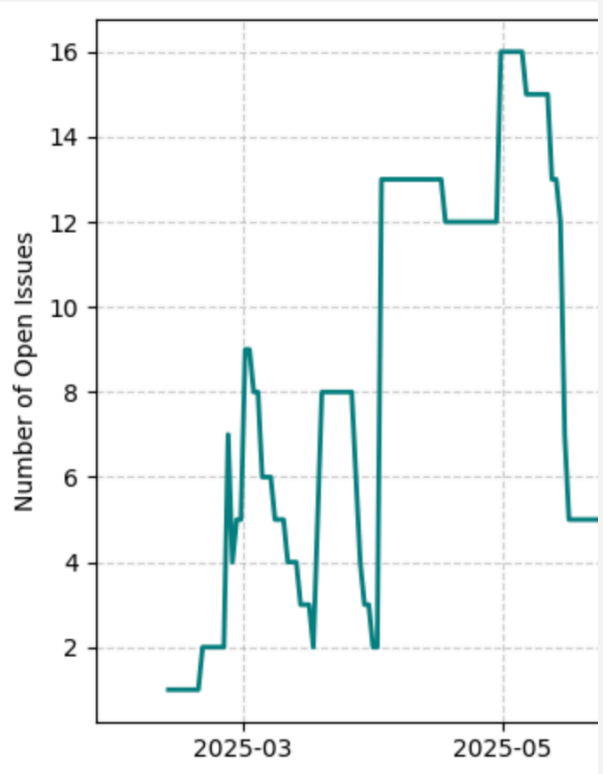
Burndown Chart Examples



Good



Fine



Bad

Team Facilitators

Each team will be allocated a ***facilitator*** (a member of the school's staff).

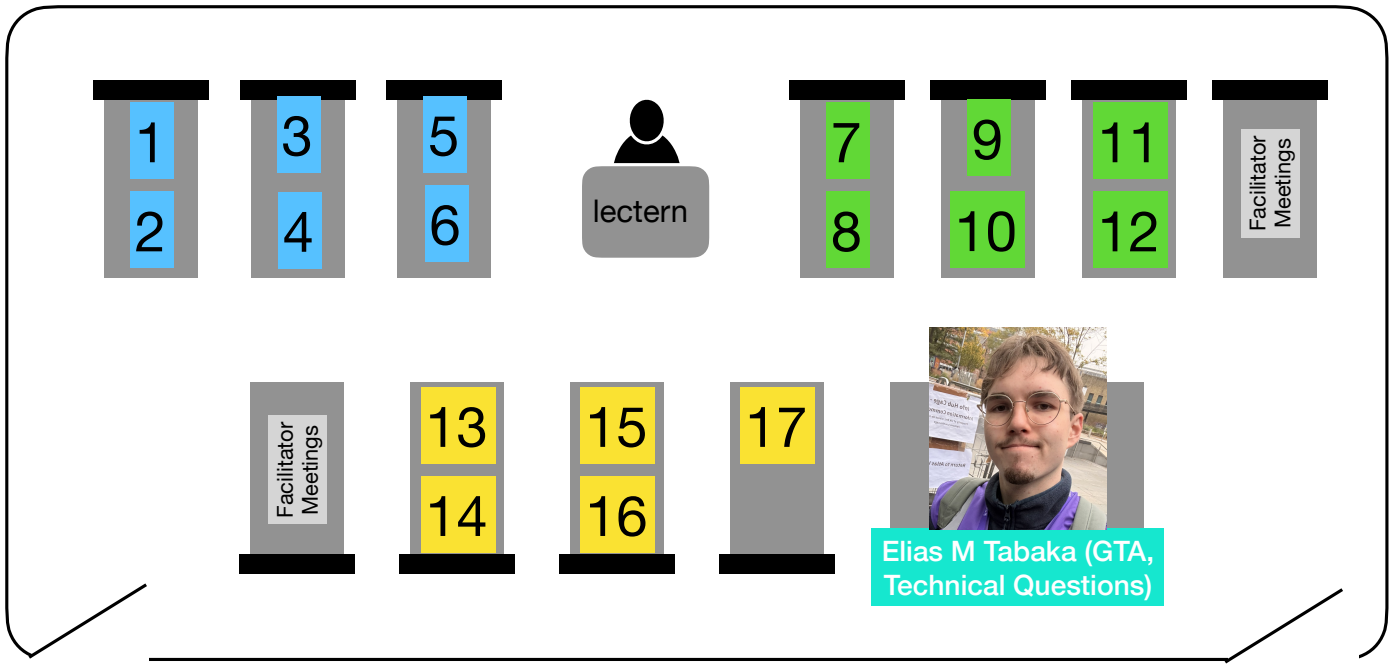
During the weekly scrum meetings, your team facilitator will conduct a quick check-in on progress.

> Note: the facilitator will *not* provide technical feedback.

The facilitator may hold a ***one-on-one meeting*** with a member who has not made active contributions, as indicated by GitLab activity.

Team ID	Room	Team Supervisor
Team 1-6	Diamond Computer Room 3 (CR3)	Islam Elgendy
Team 7-12	Diamond Computer Room 3 (CR3)	Charles Grellois
Team 13-17	Diamond Computer Room 3 (CR3)	Nafise Sadat Moosavi
Team 18-22	Diamond Computer Room 5 (CR5)	Nikos Aletras
Team 23-27	Diamond Computer Room 5 (CR5)	Robert Loftin
Team 28-32	Diamond Computer Room 5 (CR5)	Yoshi Gotoh
Team 33-38	Diamond Computer Room 5 (CR5)	Andrew Stratton

Team Seating Plan – Diamond CR3



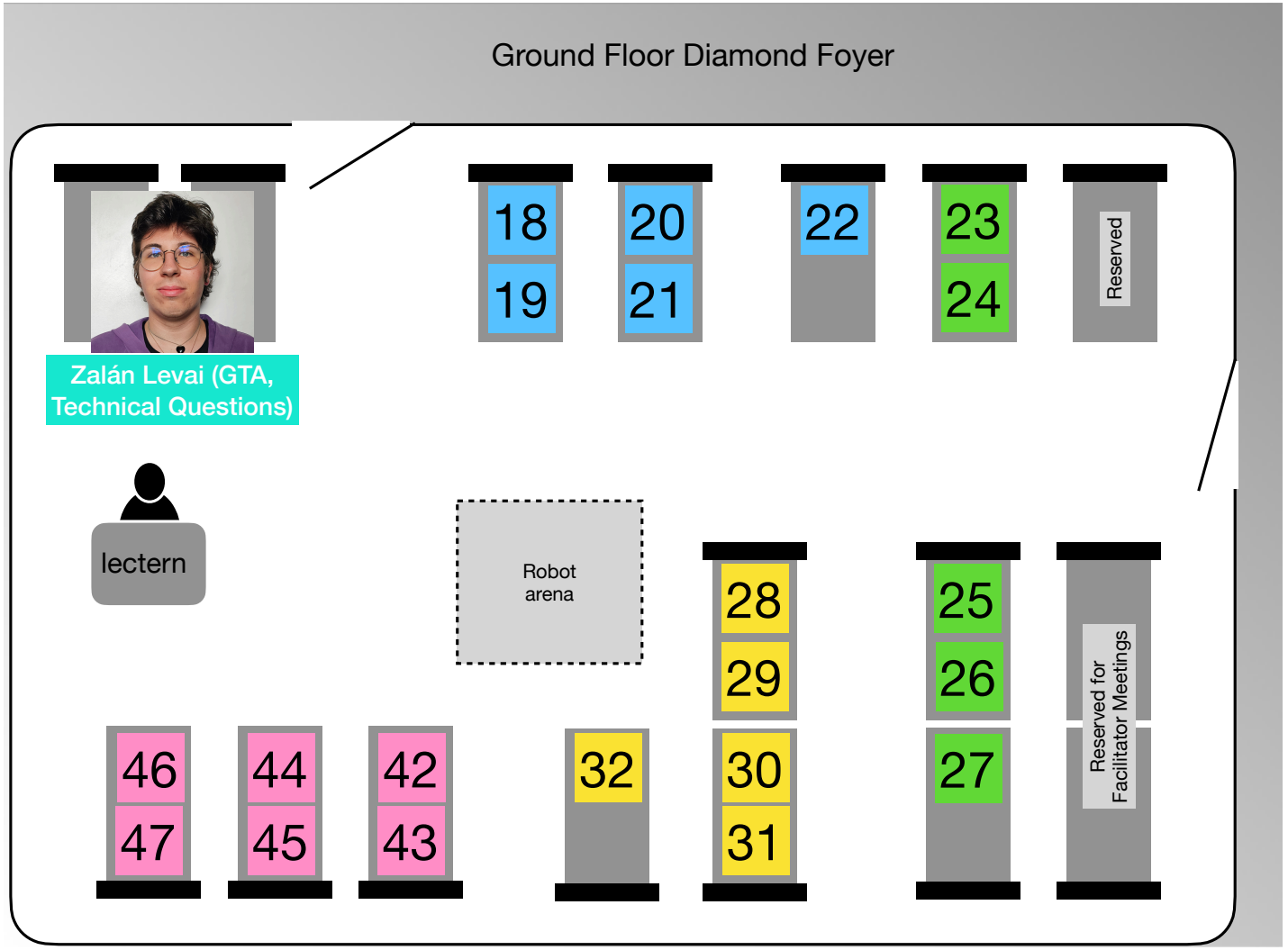
Level 2 Diamond Foyer

Islam Elgendy

Charles Grellois

Nafise Sadat Moosavi

Seating Plan – CR5



Nikos Aletras

Robert Loftin

Yoshi Gotoh

Andrew Stratton

University of Sheffield

Assessment

Assessment Subjects

1. **Formative Assessment** (0% of the semester mark)
    1. Deadline: Friday 20 March 2026, 3pm
    2. Deliverables: Report (interim)
2. **Summative Assessment** (100% of the semester mark)
    1. Deadline: Friday 15 May 2026, 3pm
    2. Deliverables: Report, Demo video
3. **Peer Assessment** (BuddyCheck for individual scaling)
    1. Deadline: Friday 22 May 2026, 3pm
    2. Deliverable: BuddyCheck scores (link will appear on Blackboard)

Important notes

- All deliverables must be submitted via Blackboard.
- While the GitLab repository (https://git.shefcompsci.org.uk) itself does not need to be submitted, all activities within the repository—including commit history, issues, and other contributions—will be reviewed and assessed.

Assessment Criteria

Category (Mapping to UG Generic Marking Criteria)	1st (70-100%)	2:1 (60-69%)	2:2 (50-59%)	3rd / Pass (40-49%)	Fail (30-39%)	Fail (0-29%)
Implementation (Practical Skills)	Exceptional mastery; all core stories complete + extra value. Polished, autonomous performance.	Highly developed; all core stories complete.Autonomous and proficient.	Most stories completed. Tasks completed accurately.	Demonstrated developing skills; reasonable stories completed. Some functions are broken.	Limited understanding; several stories completed. Frequent errors.	Lacks evidence of skill development. Frequent errors.
Testing (Creative Thinking)	Sophisticated testing; 95%+ coverage. Critically analyse findings and improve test effectiveness.	Consistent and accurate; 70-89% coverage.Evaluates different test levels.	Good attempts; 50-69% coverage. Describes findings with evaluation.	Basic attempt; <50% coverage. Lists results without evaluation.	Some relevant info, but minimal attempt.	Little or no test; limited creative thinking evidence.
Code Quality (Critical Thinking)	Original insight; well-written code with no issues.Innovative solutions.	Substantiated; issues in ~25% of code.Synthesises ideas for conclusions.	Straightforward; issues in ~50% of code. Connects ideas logically.	Basic arguments; issues in ~75% of code. Simple, taught solutions.	Simple arguments; limited coherence. Impractical or descriptive solutions.	No problem-solving or creativity.
Refactoring (Reflection)	Critically analyses; several refactoring techniques applied. Justifies future growth actions.	Evaluates options in various places. Connects experiences to knowledge.	Describes options; basic attempts. Links to prior learning.	Superficial links; basic attempts. Acknowledges some effort.	Describes experiences without analysis. Identifies basic, unspecific ideas.	No analysis or learning is evident. No questioning or future planning.
Deployment (Practical Skills)	Comprehensive instructions; innovation in precision. Clear example commands included.	Complete but not comprehensive. Performed effectively.	Straightforward; includes instructions for installation and use.	Basic instructions or minimal info. Needs guidance to deploy.	Requires substantial guidance. Instructions are missing or inaccurate.	Fails to provide a deployable system. No or empty README.
Report (Communication)	Precise structure; wide range of strategies. Seamless source integration.	Logical and clear; accurate technical language. Good summarisation.	Basic, mostly clear; uses accessibility. Some language errors.	Inconsistent; unpolished style. Limited use of accessibility.	Unclear or fragmented structure. Minimal use of technical language.	No clear structure or logic. Does not use accessibility features.
Demonstration (Communication)	Fluent; highly adapted for audience. Precise technical argumentation.	High-value stories demonstrated, logical. Accurate use of language.	Several stories demonstrated; mostly clear. Attempts to summarise.	Fragmented; some value-adding stories demonstrated. Limited technical language.	Limited awareness of the audience. Relies heavily on direct quotation.	Inappropriate style for the audience. Source material miscommunicated.

Peer Assessment (BuddyCheck)

Rate yourself and your team members on the following criteria:

- **Attendance** and **punctuality** (to team meetings, etc.).
- Ability to **work effectively with other team members**.
- **Quality** of contributions.
- **Timeliness** of contributions.

$$\textit{Your Individual Mark} = \textit{Your Team Mark} \times \textit{Your Scaling Factor}$$

- For example, if your team mark is 60 and your scaling factor is 0.9, your mark will be 54.
- The scaling factor is your average score over the team's average score (see more details).
- It can be further adjusted upon clear evidence (e.g., zero activities in GitLab).

Team Operating Agreement (TOA)

Each team must have a ***signed*** team operating agreement (TOA).

- Use this template (TUOS login required) if your team has not created one.
- It is a ***living document*** and should be updated as needed throughout the project.
- It will set out ground rules for working as a team and serve as an essential baseline in resolving disagreements and conflicts, if any.
- It must be signed by all team members and submitted as an appendix to the ***report***.

Conflict Management Scheme

Step 1: Within team

Use the signed TOA to resolve any conflicts as much as possible.

Step 2: With your facilitator

Discuss outstanding conflicts with the supervisor and ask for their guidance.

Step 3: With the project academic lead (Donghwan)

Discuss outstanding conflicts with the project lead if the facilitator advises.

Week 1: Initialisation (3 pm Today)

1. Complete your “Team Operating Agreement (TOA)” and push it to your team repository.
2. Carefully review the Spring Project Brief and discuss the following:
   - What should we do for each sprint?
   - What should we do for each Scrum meeting?
3. Create issues together (user stories based on the model solution, other tasks, etc.)
   - Make each issue self-contained but concise.
   - Always write the issue you would want to pick up if assigned.
4. Plan the first sprint.

Wrap-up quiz (not assessment)

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
HKCZBZ

Enable answers by SMS

COM1001 Introduction to Software Engineering Semester 2 Team Project

Dr Donghwan Shin

Senior Lecturer