School of Computer Science and Engineering (CSE)

COMP9900 Information Technology Project
COMP3900 Computer Science Project

2023 Term 3

Week 1

Dr Rachid Hamadi (Lecturer in Charge)
Dr Basem Suleiman (Project Clients Coordinator)

Outline

- Course Introduction
- Project Clients Coordination (by Dr Basem Suleiman)
- Assessment Tasks Overview
- User Stories and Acceptance Criteria
- Jira Software
- Week 1 Lab Tasks
- Q & A

Course Introduction

Course Introduction

- Moodle Course Website
  https://moodle.telt.unsw.edu.au/course/view.php?id=79493

- Lectures: **Wednesday 18:00 - 20:00**
  Location: **Online** using **Blackboard Collaborate**
  Weeks 1-5, 7-10

- Labs : 2 hours per week
  Location: **F2F** or **Online** using **Blackboard Collaborate**
  Weeks 1-5, 7-10 (Week 6 is Flex Week)
  Make Up Class for Monday Week 4 (Labour Day Public Holiday)

Course Summary

- A **software** project **capstone** course
- Students work in teams to **define**, **implement** and **evaluate** a **real-world** software system
- Most of the work in this course is **team-based project work**
- There are some **introductory** lectures on **software project management techniques** and **teamwork strategies**

Course Summary (cont’d)

- Project teams meet **weekly** with project **mentors** to report on the progress of the project

- Assessment is based on a project **proposal**, **progressive demonstrations** and **retrospectives**, a **final project demonstration** and **report**, and on the **quality of the software** system itself

- Students are also required to reflect on their work and to provide **peer assessment** of their **teammates' contributions** to the project

Labs (or Mentoring Sessions)

- **Weekly** project progress meeting with **tutor/mentor**
- **Two progressive demos** to mentor in **Weeks 5 and 8** and **final demo/presentation** in **Week 10**
- **Two retrospectives** right after progressive demos and report due **Saturday Weeks 5 and 8** resp.
- Attendance to labs is then **mandatory**
- Regular group meetings (more than once per week) among team members

Readings

- No textbook
- Slides will be provided in the course website (Moodle)
- Online resources may be provided from time to time (e.g., in lectures or via the tutor/mentor in the lab) and may be uploaded to Moodle
- For Project Management part of the course, the following text is highly recommended:

  Kathy Schwalbe. Information Technology Project Management. 9th Edition. Cengage. 2018

Assumed Knowledge

Before commencing this course, students should be able to:

- produce correct software programs in Python, Java or C/C++, i.e., compilation, running, testing, debugging, etc.
- produce readable code with clear documentation
- have basic knowledge of database programming, Web programming and/or script programming (such as Python, PHP, and JavaScript)

Assumed Knowledge (cont'd)

For **COMP9900**, students must:

- Be in their **final** term of study, and
- Have completed at least **66** UOC towards MIT program **8543**

Assumed Knowledge (cont'd)

For **COMP3900**, students must:

- Have successfully completed **COMP1531** and **COMP2521** (or the old equivalent COMP1927)
- Be enrolled in a **BSc Computer Science** major
- Have completed at least **102** UOC

Learning outcomes

1. work from a set of **requirements**, elaborate them, and produce a **specification**
2. design and build a **correct**, **efficient** and **robust** software system from **specification**
3. use software **development** and software **project management tools**
4. validate the **correctness** and **robustness** of **software**
5. work **effectively** in a project **team**, and **lead** when required
6. **manage** their **time** effectively, and make reasoned **trade-offs** over competing demands
7. **communicate** technical information clearly, both **verbally** and in **writing**

COMP9900 – IT Project
COMP3900 – CS Project

- A brief **description** of **25 real-world projects** with **clients** to choose from this term is provided under **Project Topics** section in Moodle

- As a team, decide on the project that best matches the **team's areas of specialization**, **knowledge** and **skills** by end of Week 1

Project Clients Coordination

(by Dr Basem Suleiman)

Project Clients Coordination

- Roadmap to improve capstone project experience – CSE, Engineering, ACS
- Projects proposed by **clients** (*project owners*) – real-life project experience
- Project description
  - Client, project specialization, background, requirements and scope, required knowledge and skills, expected outcomes
- Projects focus – **development** work and **discipline-specific** if agreed
  - Traditional software/system development and interdisciplinary
  - Web/software/mobile application, AI-based development, cybersecurity, computer vision
- Teams to select project based on knowledge and skills – important!

Project Clients Coordination

- **Teams to work with project client**
  - Project stakeholders – (e.g., industry partners, research assistants)
  - Requirements elicitation and scope – by Week 3 Friday
  - Sprint demos, deliverables/outcomes, handover/documentation
  - Bi-weekly/milestone meetings (e.g., demos) and communication – efficient and effective
  - Early feedback and changes (Sprint review)
  - Team lead (product owner) – act on the client's behalf
  - Be creative – suggest/lead novel ideas
  - Projects coordinator (Basem Suleiman) to facilitate with clients

- **Project's structure**
  - One big project and sub-projects (big team and sub-teams)
  - Same project carried out by multiple teams
  - Coordination and communication among teams and the client

Project Clients Coordination

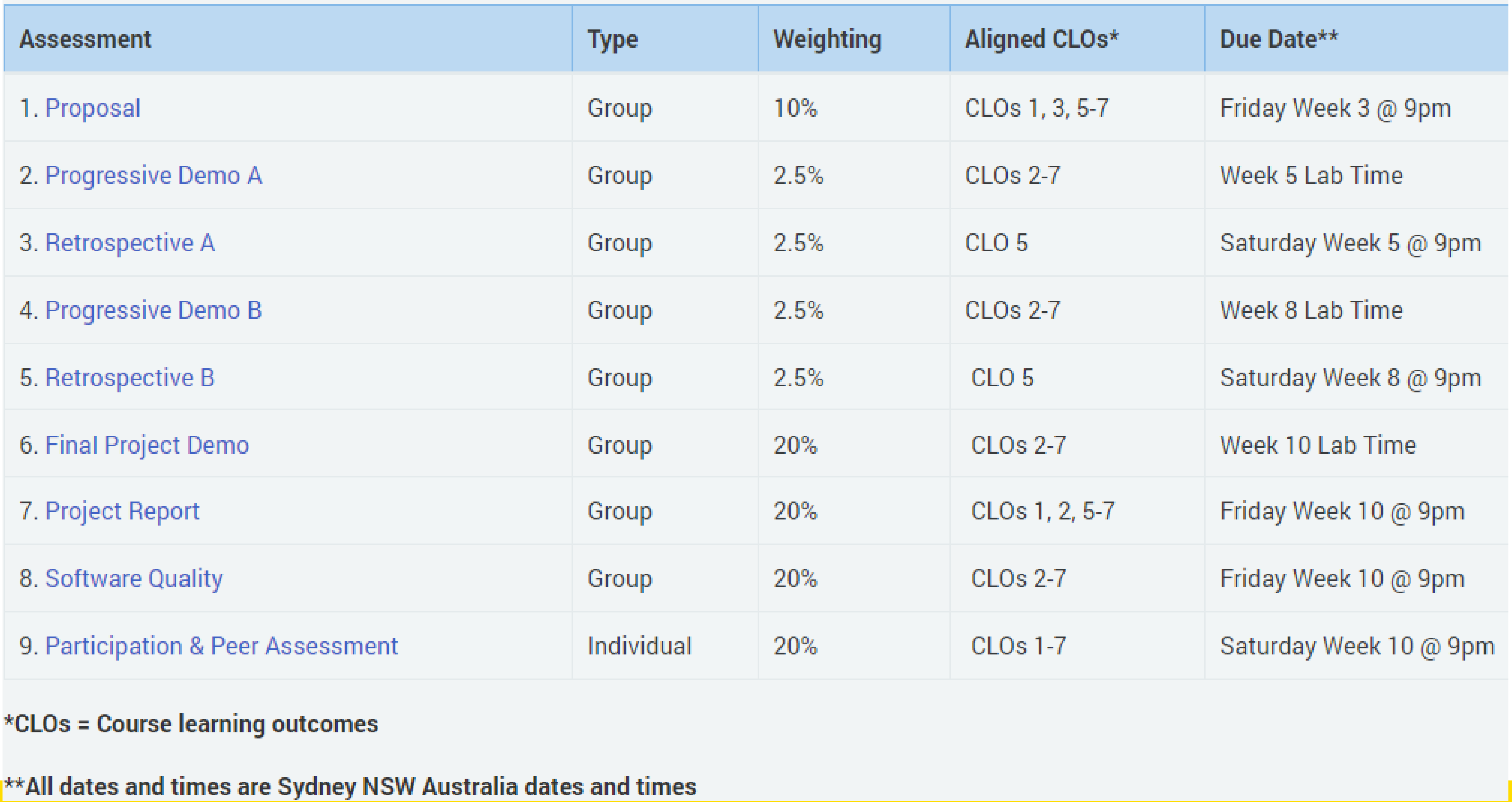
- Project’s timeline
  - **Week 2-3**: meet/communicate with your client (lead/product owner)
  - **Week 3**: proposal due (clear requirements and scope, project plan)
  - **Week 4-10**: development work and major milestones/demos
    communicate with client (meeting, communication channels)
  - **Week 5:** Sprint 1 demo
  - **Week 8:** Sprint 2 demo
  - **Week 10:** Sprint 3 (final) demo, project document, and handover

Assessment Tasks Overview

Assessment Tasks Overview

- No final exam
- A team-based project
- Each team has **ideally five (5) members**

Assessment Tasks Overview (cont’d)



Assessment	Type	Weighting	Aligned CLOs*	Due Date**
1. Proposal	Group	10%	CLOs 1, 3, 5-7	Friday Week 3 @ 9pm
2. Progressive Demo A	Group	2.5%	CLOs 2-7	Week 5 Lab Time
3. Retrospective A	Group	2.5%	CLO 5	Saturday Week 5 @ 9pm
4. Progressive Demo B	Group	2.5%	CLOs 2-7	Week 8 Lab Time
5. Retrospective B	Group	2.5%	CLO 5	Saturday Week 8 @ 9pm
6. Final Project Demo	Group	20%	CLOs 2-7	Week 10 Lab Time
7. Project Report	Group	20%	CLOs 1, 2, 5-7	Friday Week 10 @ 9pm
8. Software Quality	Group	20%	CLOs 2-7	Friday Week 10 @ 9pm
9. Participation & Peer Assessment	Individual	20%	CLOs 1-7	Saturday Week 10 @ 9pm

***CLOs = Course learning outcomes**

****All dates and times are Sydney NSW Australia dates and times**

User Stories and Acceptance Criteria

User Stories

- A **user story** helps **agile software development teams** capture a **simplified** and **high-level** description of a **requirement** from an **end user** perspective

- A user story often follows the following **Connextra** format/template:

  As a [who] I want to [what] so that [why]

- Example:

  As an **online shopper**, I want to **add an item to my cart**, so that **I can purchase it**

User Stories (cont’d)

- As a **<type of user>** – this is the **WHO**
  - Who are we building this for? Who is the user?
- I want **<some feature>** – this is the **WHAT**
  - What are we building? What is the intention?
- So that **<some reason>** – this is the **WHY**
  - Why are we building this? What is the value for the customer?

User Stories (cont’d)

User Stories Checklist

- Keep them short
- Keep them simple
- Write them from the user perspective
- Make the reason/value/benefit of the story clear
- Describe only one piece of functionality
- Write stories as a team
- Use **acceptance criteria** to show a **Minimum Viable Product** (**MVP**), that is, is a **working** and **usable** product

Acceptance Criteria

- **Acceptance criteria**, also called ***satisfaction conditions***, provide a detailed scope of end users requirements

- Help the development team understand the **value of the user story** and **set expectations** as to when a team should consider something **done**

- Acceptance Criteria Goals
  - clarify what the team should build before they start
  - ensure everyone has a common understanding of the problem
  - help the team members know when the story is complete
  - help verify the story via automated tests

Acceptance Criteria (cont’d)

Example:

- As an online banking customer, I want a strong password, so that my credit card information is secure

Acceptance Criteria:

- The password must be at least eight (8) characters
- The password must contain at least one character from each of the following groups:
  - lower case alphabet
  - upper case alphabet
  - digit
  - special characters (!, @, #, $, %, ^, &, *)

Acceptance Criteria (cont’d)

Acceptance criteria should include:

- **Negative** scenarios of the functionality
- **Functional** and **non-functional** use cases
- **Performance** concerns and guidelines
- What the system/feature intends to do
- The **impact** of a user story to other features
- **User experience** concerns

Jira Software

Jira Software

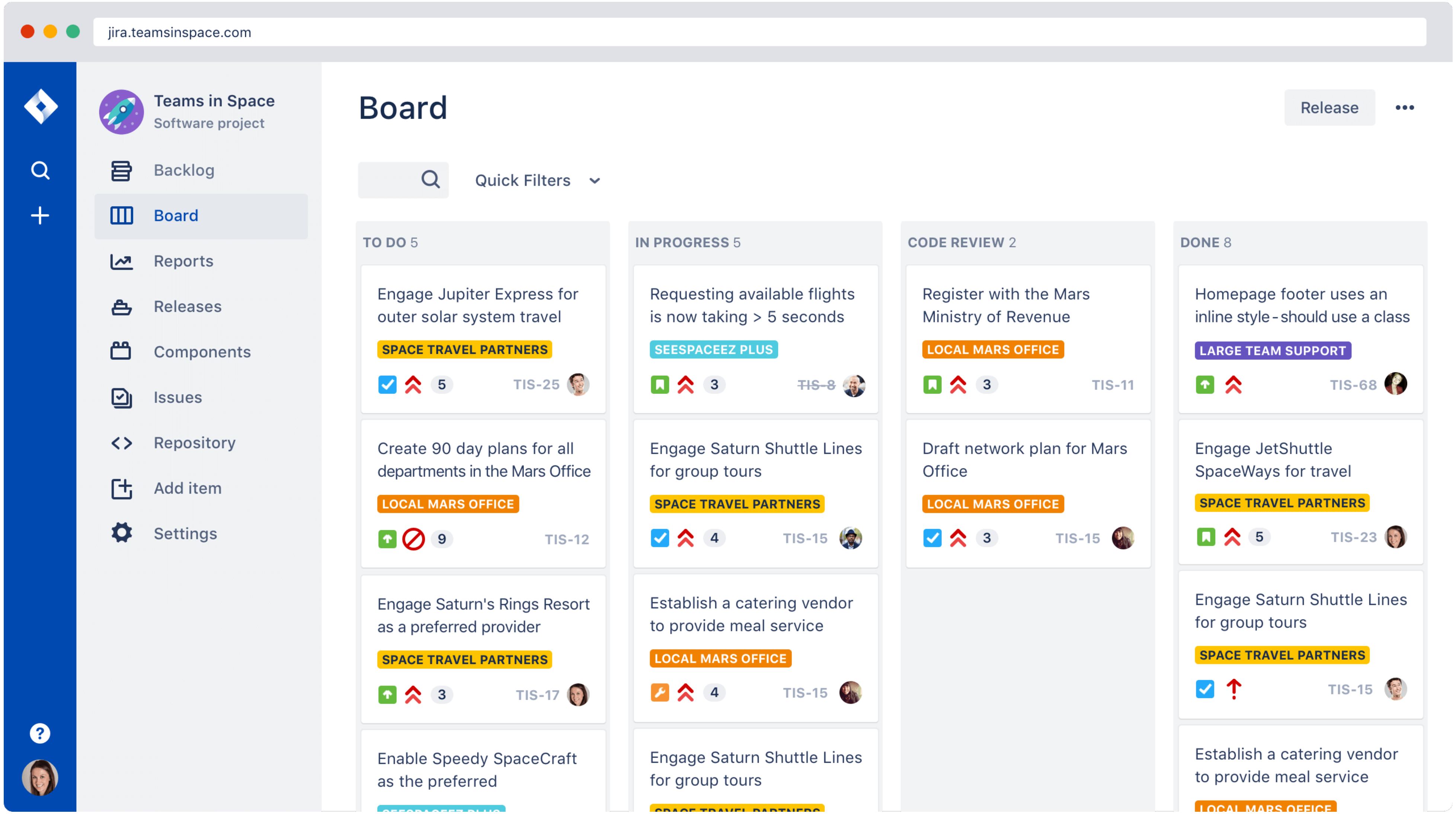
- Jira Software is an **agile** project management tool

- Jira Software supports any agile project management methodology such as **Scrum** and **Kanban**

- From agile **boards** to **reports**, you can **plan**, **track**, and **manage** all your agile software development projects from a single tool

- See https://www.atlassian.com/software/jira/agile for more details

Jira Software (cont’d)



Board

TO DO 5

- Engage Jupiter Express for outer solar system travel — SPACE TRAVEL PARTNERS — 5 — TIS-25
- Create 90 day plans for all departments in the Mars Office — LOCAL MARS OFFICE — 9 — TIS-12
- Engage Saturn's Rings Resort as a preferred provider — SPACE TRAVEL PARTNERS — 3 — TIS-17
- Enable Speedy SpaceCraft as the preferred

IN PROGRESS 5

- Requesting available flights is now taking > 5 seconds — SEESPACEEZ PLUS — 3 — ~~TIS-8~~
- Engage Saturn Shuttle Lines for group tours — SPACE TRAVEL PARTNERS — 4 — TIS-15
- Establish a catering vendor to provide meal service — LOCAL MARS OFFICE — 4 — TIS-15
- Engage Saturn Shuttle Lines for group tours

CODE REVIEW 2

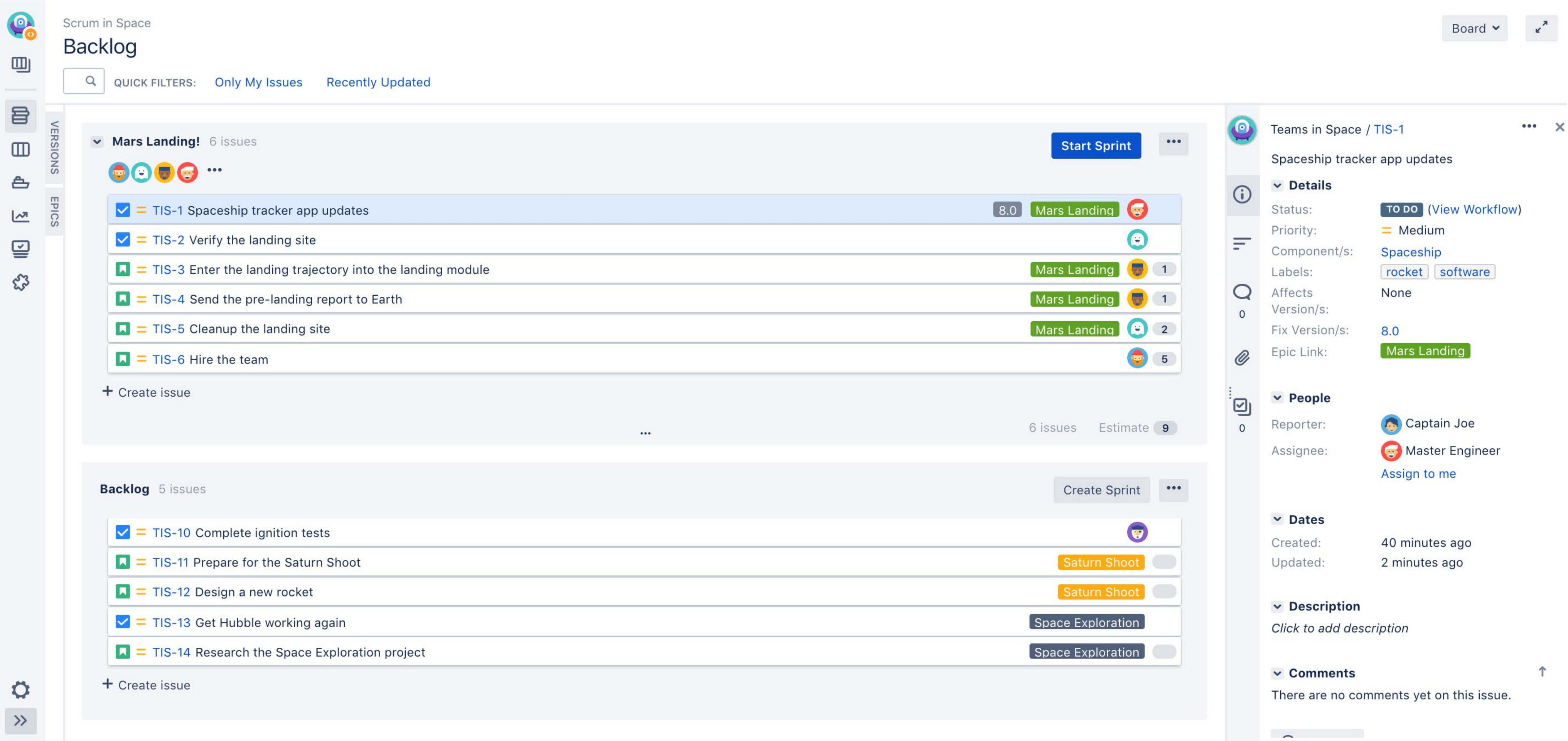
- Register with the Mars Ministry of Revenue — LOCAL MARS OFFICE — 3 — TIS-11
- Draft network plan for Mars Office — LOCAL MARS OFFICE — 3 — TIS-15

DONE 8

- Homepage footer uses an inline style - should use a class — LARGE TEAM SUPPORT — TIS-68
- Engage JetShuttle SpaceWays for travel — SPACE TRAVEL PARTNERS — 5 — TIS-23
- Engage Saturn Shuttle Lines for group tours — SPACE TRAVEL PARTNERS — TIS-15
- Establish a catering vendor to provide meal service — LOCAL MARS OFFICE

Jira Software (cont’d)

Backlog



Scrum in Space

Backlog

QUICK FILTERS: Only My Issues Recently Updated

Board

VERSIONS EPICS

Mars Landing! 6 issues — Start Sprint

- ☑ TIS-1 Spaceship tracker app updates — 8.0 — Mars Landing
- ☑ TIS-2 Verify the landing site
- TIS-3 Enter the landing trajectory into the landing module — Mars Landing — 1
- TIS-4 Send the pre-landing report to Earth — Mars Landing — 1
- TIS-5 Cleanup the landing site — Mars Landing — 2
- TIS-6 Hire the team — 5

+ Create issue

6 issues Estimate 9

Backlog 5 issues — Create Sprint

- ☑ TIS-10 Complete ignition tests
- TIS-11 Prepare for the Saturn Shoot — Saturn Shoot
- TIS-12 Design a new rocket — Saturn Shoot
- ☑ TIS-13 Get Hubble working again — Space Exploration
- TIS-14 Research the Space Exploration project — Space Exploration

+ Create issue

Teams in Space / TIS-1

Spaceship tracker app updates

Details

Status:	TO DO (View Workflow)
Priority:	Medium
Component/s:	Spaceship
Labels:	rocket software
Affects Version/s:	None
Fix Version/s:	8.0
Epic Link:	Mars Landing

People

Reporter:	Captain Joe
Assignee:	Master Engineer Assign to me

Dates

Created:	40 minutes ago
Updated:	2 minutes ago

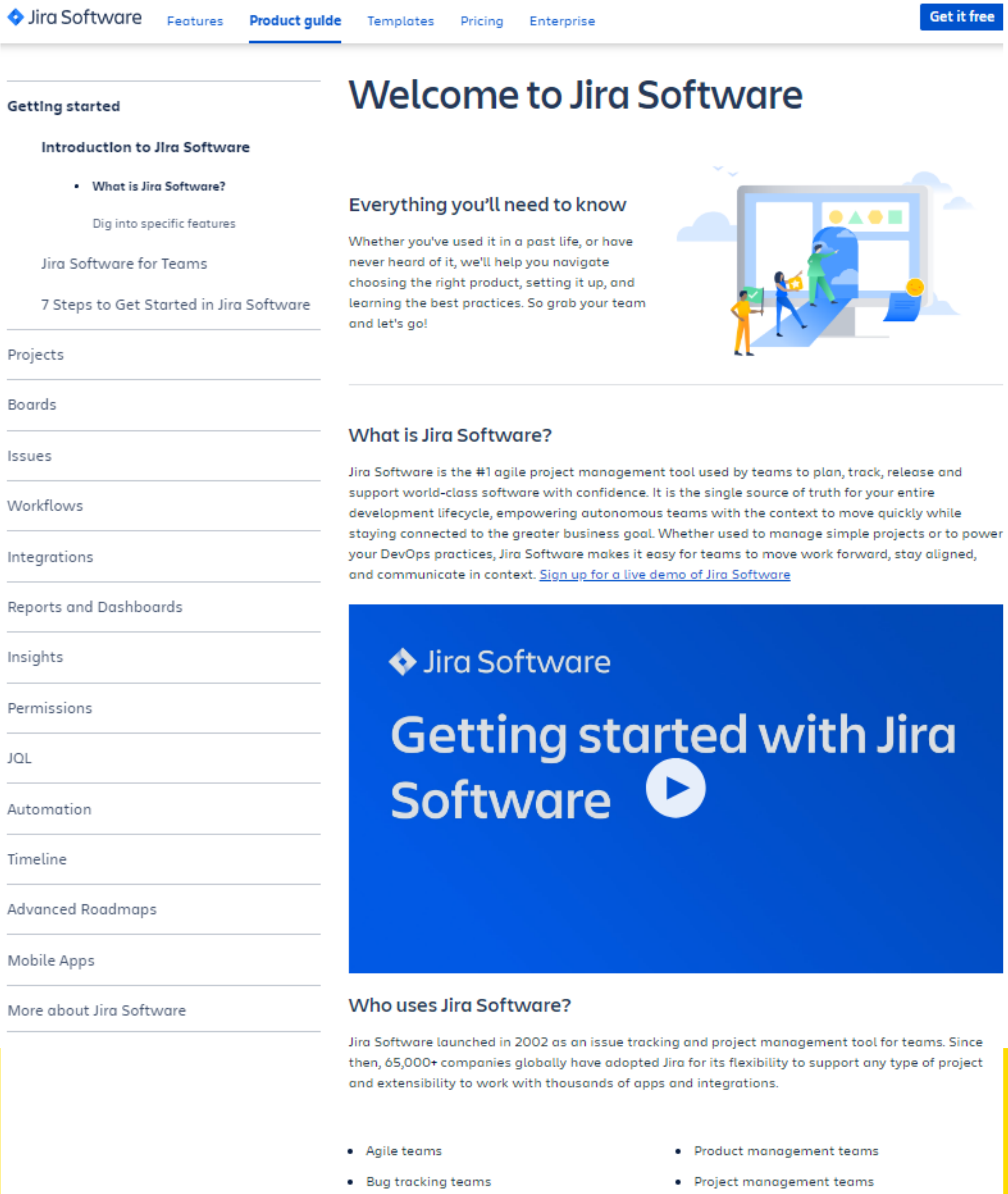
Description

Click to add description

Comments

There are no comments yet on this issue.

Jira Software (cont'd)



Getting started

Introduction to Jira Software

- What is Jira Software?

Dig into specific features

Jira Software for Teams

7 Steps to Get Started in Jira Software

Projects

Boards

Issues

Workflows

Integrations

Reports and Dashboards

Insights

Permissions

JQL

Automation

Timeline

Advanced Roadmaps

Mobile Apps

More about Jira Software

Welcome to Jira Software

Everything you'll need to know

Whether you've used it in a past life, or have never heard of it, we'll help you navigate choosing the right product, setting it up, and learning the best practices. So grab your team and let's go!

What is Jira Software?

Jira Software is the #1 agile project management tool used by teams to plan, track, release and support world-class software with confidence. It is the single source of truth for your entire development lifecycle, empowering autonomous teams with the context to move quickly while staying connected to the greater business goal. Whether used to manage simple projects or to power your DevOps practices, Jira Software makes it easy for teams to move work forward, stay aligned, and communicate in context. Sign up for a live demo of Jira Software

Who uses Jira Software?

Jira Software launched in 2002 as an issue tracking and project management tool for teams. Since then, 65,000+ companies globally have adopted Jira for its flexibility to support any type of project and extensibility to work with thousands of apps and integrations.

- Agile teams
- Bug tracking teams
- Product management teams
- Project management teams

How to use Jira Software

- Jira Software Guides and Tutorials

  https://www.atlassian.com/software/jira/guides

Week 1 Lab Tasks

Week 1 Lab Tasks

- Join or form a team of ideally **5 members** within your lab
- Decide on who will be the **Scrum Master**
- Decide on a **team name**
- Ensure the team's name is prefixed with the last four digits of the course code and lab code, for instance, **9900H14BMagicode** for a team called **Magicode** enrolled in lab **H14B** for **COMP9900**
- Register your team in **Moodle** using **group self-selection** activity under "**Teams Formation**" section

Week 1 Lab Tasks (cont’d)

- Register for **Jira**. Use the exact team’s name for the **Jira site** name. Add all team members and mentor as **site-admin** (see **Jira guide** in **Moodle** under **Jira** section)

- Decide on the **project** you will work on by end of Week 1 to your mentor for approval

- Start working on the **Proposal** assessment due **Week 3 Friday 29 September 2023 @ 9pm**

Q & A