UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Sprint 1 Instructions and Rubric

1. Instructions

At this stage, you should know who your group members are, what project you have been assigned to, and who your client is. Your first priority is to contact your client as soon as possible to organise a Sprint Planning meeting. In your sprint planning meeting, your client will tell you what they want out of the project, and what to prioritise working on for this sprint (it is your responsibility to get all the necessary information from your client).

Once you have had your sprint planning meeting, you must begin working together on your project, following the scrum methodology. You will take the requirements your client gave you and turn them into user stories, which you will then implement. At the same time, you should be following Continuous Integration and Test-Driven Development principles.

At the end of the sprint you will have a version of the product whose features should work correctly and which can form a foundation for further work in the next sprints. At the end of every sprint, you will be assessed on your progress.

The following criteria will be used to assess your work.

1.1 Scrum Methodology

You need to follow the scrum methodology throughout the project. This means you need to engage in all the types of scrum meetings. You will be required to show proof of your engagement with these meetings (e.g. minutes). For sprint 1, you will not need to have had a sprint review meeting.

Make sure that your daily scrum meetings are frequent and that you engage in your backlog refinement and sprint retrospective meetings prior to being marked.

1.2 Requirements Engineering

You will need to maintain a backlog of requirements for the entire project scope, ideally using software such as Taiga, Trello or Notion. The requirements being addressed in sprint 1 should be pulled into a separate track and should be recorded in the form of user stories. All user stories should be valid (i.e. focusing on user interactions with the software) and in the correct format (Who-What-Why), and should be divided into tasks, with each task assigned to one or more team members. Every task should also have a status (e.g. In Progress, Testing, Complete, etc.).

1.3 Software Implementation

By the end of the sprint you should have completed at least four user stories. A user story is complete if it passes its associated User Acceptance Test (see below).

1.4 CI/CD

You should create a GitHub repository for your project, on which all team members are making commits. You should also be using an issue tracker for any issues you encounter during development. CI/CD pipelines, via tools like Github Actions, should be set up correctly on your repository. Your software should be contained in a publicly available standalone build, rather than running from a development environment on someone's laptop.

1.5 TDD

Every user story should have one or more associated user acceptance tests in the correct format (Given-When-Then). These acceptance tests must be used to determine whether or not your user stories have been completed.

You will not have to write unit tests this sprint, although you should be investigating which frameworks to use. You will have to set up a code coverage tool, such as Jest, on your GitHub repository.

1.6 Project Polish

A portion of your mark will be determined by your project's polish. To maximise this mark, make sure your software is easy to use, bug-free, aesthetically pleasing and intuitive. Your code should be well-structured and legible, with comments explaining the code. Good documentation is desirable.

Rubric

	Poor	Acceptable	Good	Excellent	Weight
Scrum Methodology	Proof of 1 meeting	Proof of 2 meetings	Proof of 3 meetings	Proof of 4 meetings	20
Requirements Engineering	Documentation created, user stories present for sprint 1	Correct format for user stories, product backlog maintained	User stories divided into tasks	All tasks assigned and statuses tracked	25
Software Implementation	1 user story complete	2 user stories complete	3 user stories complete	4 user stories complete	15
Continuous Integration	GitHub repository set up and in use	Everybody making commits, issue tracker being used	CI configured on GitHub and able to auto build/run	Full use of GitHub including releases for this sprint & site deployed to azure	20
Test Driven Development	Some acceptance tests present	UATs for every user story	UATs in correct format	Code coverage tool integrated on GitHub	10
Project Polish	Poor attempt at project so far given elapsed time.	Project is where we would expect it to be given the elapsed time.	Team has done work beyond what is expected of them given elapsed time.	Team on track for being in the top projects this year.	10