We must choose a method:

Waterfall, Spiral or Agile?

I think we should use the agile

We must do these steps in a scrum:

* Product Backlog.
* Task Board (or Scrum Board, or Kanban Board).
* User Stories. When defining a task it should be part of a User Story. This story has a number of tasks associated with it. We need to upload a file. That file needs to be associated with an assessment. An assessment needs to associated with a module. We need to login to the system. And so on.
* Sprints. A Sprint is a period of time when a team does work. The typical length of time is a week. At the end of the Sprint the product must have moved forward and be usable by the customer. That is, the customer has to see new features added and completed so they can judge them. If the product is not usable at the end of the Sprint you have a problem. If a task is not fully finished at the end of the Sprint then it is not done - the effort was wasted in this Sprint and something else could have been developed and shown to the customer instead.
* Sprint Planning. To plan a Sprint, the team selects tasks they believe they can complete during the Sprint. These tasks are selected from the prioritised task list.
* Story Points. Each task is scored based on an estimated cost. We don't talk about hours, days, or any other concept of time. We are allocating points. And points are a relative score. For example, we could rate our tasks using animal sizes:
* Planning Poker.
* Daily Standup.
* Sprint Review and Retrospective.

We must define a product owner and a scrum master:

How do we run a Scrum team:

1. Produce an initial product backlog.
2. Prioritise the backlog.
3. Do an initial estimate of the work.
4. Decide on what work to do in the next Sprint - *Sprint Planning*.
5. Every day have a daily stand-up - *Daily Scrum*.
6. End of the Sprint, review - *Sprint Review*.
7. Analyse how the team has worked and modify based on outcome of review: update estimates, add new tasks, learn lessons, etc. - *Sprint Retrospective*.
8. Repeat 4-7 until product is delivered to the customer.

Scrum uses a **Plan, Do, Check, Act (PDCA) Cycle**

Your Task:

1. Pick a **team**. You need four people for your team.
2. Define a **Code of Conduct**. See [Unit 10a](https://github.com/edinburgh-napier/SET08103/blob/main/units/unit10/unit10a.md) for assistance.

Be present

All voices are equal

Listen actively

Respect everybody

Reliable

On time

Polite

Communicate open and honest

Quality

Focused

Cohesion

Help the team members if necessary

1. Pick a **Product Owner**.
2. Pick a **Scrum Master**.
3. Create and prioritize a **Product Backlog.**
4. Refine and estimate the **Product Backlog.**
5. Schedule **Sprint planning session**.
6. Schedule **daily stand-up**.

Every 2 days online/present for 15 minutes

1. Schedule **Sprint review**.
2. Schedule **Sprint retrospective**.

FAQs - Scrum Usage in the Module:

1. **Do we really need to meet once a day?**
2. **We cannot find a time to all meet. What do we do?**
3. **How will Scrum be used by the teaching team?**
4. **Does my team have to work on the lab material together?**
5. **We have a problem with a team member. What should we do?**
6. **Can we sack a team member?**

Backlog:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Backlog | In Progress | Peer Review | In Test | Done | Blocked |
|  | Create the GitHub |  |  |  |  |
| Project builds to self-contained JAR with Maven |  |  |  |  |  |
|  | Set-up Dockerfile for the Project |  |  |  |  |
| GitHub Actions for project set-up and build is working using JAR, and Docker on GitHub Actions |  |  |  |  |  |
| Create branches for the GitFlow |  |  |  |  |  |
| First release created on GitHub |  |  |  |  |  |
|  | Define the code of conduct |  |  |  |  |
| Issues being used on GitHub |  |  |  |  |  |
| Tasks defined as user stories |  |  |  |  |  |
| Full use cases defined |  |  |  |  |  |
| Use case diagram created |  |  |  |  |  |
| Project integrated with Zube.io |  |  |  |  |  |
| Kanban/Project Board being used |  |  |  |  |  |
| Sprint Boards being used |  |  |  |  |  |
| Suitable integration tests defined |  |  |  |  |  |
| Tests running on GitHub Actions |  |  |  |  |  |
| Deployment working |  |  |  |  |  |
| Bug reporting system set-up |  |  |  |  |  |
| Fast track/defect |  |  |  |  |  |

Therefore, the majority of your coursework grade will be based on your team's ability to work together using the methods defined in the module.

Groups must maintain a spreadsheet detailing individual team members contribution at each of the 5 assessment points

The master branch of your GitHub repository should also be submitted to Moodle at each assessment point along with a spreadsheet in Excel format (csv, xls or xlsx) detailing Individual team members' contributions.

**Code Review 1**

**REVIEW MEETING: Lab of Week 5**

The aim of this code review meeting is to check that the project workflow is set-up for the team. You may choose to meet some of the feature requirements during this review point, but it is not as necessary.

**Checklist Submission 1 (23% of CW mark)**

The following must be in place:

* GitHub project for coursework set-up.
* Product Backlog created.
* Project builds to self-contained JAR with Maven.
* Dockerfile for project set-up and works.
* GitHub Actions for project set-up and build is working using JAR, and Docker on GitHub Actions.
* Correct branches for GitFlow workflow created - includes master, develop, and release branches.
* First release created on GitHub.
* Code of Conduct defined.
* Issues being used on GitHub.
* Tasks defined as user stories.
* Full use cases defined.
* Use case diagram created.

**Graded Criteria Submission 1 (7% of CW mark)**

The following criteria will be assessed for overall quality:

* Metrics from GitHub. Also used to assess individual contribution.
* Use cases well defined.
* Code quality including comments.