

Milestone: Requirements

OVERVIEW

During this milestone, you will:

- Use an agile approach to the gathering of requirements for a software system.
- Collaborate with other teams to develop the final set of requirements.

SPRINTS

The milestone will be completed in two 1-week sprints. Sprint activities are completed as a team.

Sprint 1

Create a backlog of items describing the functionality of the software system.

1. Read the rules for the game.
 - a. There is an issue in the master project repository called “Read the rules.” This issue contains a link to the rules for the game that all teams will use. There may be additional comments related to scope (e.g. No betting for Blackjack) or possible variations (e.g. 8-card Golf instead of 6-card).
 - b. The same links are found on Moodle.
2. Create a use case diagram for the game.
 - a. Identify the actors.
 - b. Identify the system functions.
3. Commit your use case diagram as a PDF or image to your team’s repository with the name `Use Cases.<type>` (e.g. `Use Cases.pdf`, `Use Cases.jpg`)
 - a. You can use Dia on the lab computers or one of a number of online drawing tools like LucidChart or Creately.
4. Create user stories in GitLab for the game that use the format

As a <persona>, I <want to>, [<so that> || <reasoning>].

Although the motivation or reasoning is optional, try to include it even if it seems obvious.

5. For each user story:
 - a. Assign the following labels (This gets you in the practice of labelling your issues so they are easier to sort or find later.)
 - i. `user story`
 - ii. `game::<game name>`
 - b. Set the milestone as `Requirements`.
 - c. Set the iteration to `Backlog Creation`.
6. Play Planning Poker to assign story points to each user story.
 - a. Use the Fibonacci numbers (1, 2, 3, 5, 8, 13, 21)
 - b. Enter the story points into the `Weight` field of the issue.
7. Examine stories with a high number of points to see if they can be broken down into smaller stories.

- a. You may keep the larger story in GitLab and link to the smaller stories.

Grading

The sprint will be graded on the following criteria:

- a. Use Case Diagram [Good/Fair/Poor/Not Done]
- b. User stories provide an appropriate coverage of game requirements [Satisfactory/Unsatisfactory]
- c. User stories are properly written using the prescribed format [Good/Fair/Poor/Not Done]

Sprint 2

1. Examine the backlogs created by the other two teams for the same game as your team.
2. Select a set of user stories that fully describe the game's requirements.
 - a. Record the Team Number (e.g. Team 1) and the issue number (e.g. #9) for the user story.
 - b. The list can be a mixture of different teams' stories, including yours.
 - c. Order the list by Team number and issue number. For example:

```
Team 1 - #4
Team 1 - #7
Team 2 - #1
Team 2 - #5
Team 3 - #2
```

3. Commit to the `requirements` branch a file with the file name `Team <#> Voting.txt` that contains your list of issues.
4. Select one team member to be a Scrum Master to meet with the other Scrum Masters for the same game.
5. The Scrum Masters create a `User Stories.txt` file in the `Master` repository on the `requirements` branch for the game.
 - a. Other team members may attend the meeting but should be "Chickens," not "Pigs."
 - b. Teams may agree to edit a particular team's issue to create an improved version.
6. At the end of the sprint, the instructor will use this file to transfer the listed issues into the game's `Master` repository for the Design milestone.

Sprint Grading

The sprint will be graded on the following criteria:

- a. Team provided a list of user stories for consideration by the Scrum Masters. [Completed/Not Done]
 - a. Teams will receive individual grades for this item.
- b. Teams have created a master list of user stories for the next milestone [Completed/Not Completed]
 - a. All teams will receive the same grade for this item.

MILESTONE DELIVERABLES

Sprint 1	A use case diagram
	Set of user stories
Sprint 2	List of user stories (each team)
	Agreed list of user stories (all teams)
	Contribution report (each team member)