Lab: Agile Planning Using GitHub

Estimated time: 60 minutes

Welcome to this lab on Agile Planning Using GitHub. In this lab, you will build a sprint plan, or Sprint 0, in preparation for the first sprint for your customer accounts microservice development project. The plan is to create an account service to read, update, and delete accounts. The accounts service database will contain basic information about customer names and addresses.

> Note: For this lab, you will be working in GitHub. You will not be using the lab environment.

Objectives

In this lab, you will create a GitHub repository, set up a GitHub Kanban board, develop a user story template, and add user stories to the board. You will then sort these stories to prepare for backlog refinement, refine the product backlog to make it sprint-ready, and finally build a comprehensive sprint plan.

Deliverables

You will have a two options to submit your deliverable in the final project.

- **Option 1: AI-Graded Submission and Evaluation**
- **Option 2: Peer-Graded Submission and Evaluation**

In this lab, you need to save your responses in the form of images in the .jpeg or .png format and GitHub repository URL on your device for the final project.

To take screenshots, you can use various free screen-capture tools or, depending on your operating system, the following shortcut keys:

- **Mac:** You can use `Shift + Command + 3` (⇧ + ⌘ + 3) on your keyboard to capture your entire screen, or `Shift + Command + 4` (⇧ + ⌘ + 4) to capture a window or area. They will be saved as a file on your Desktop folder.

- **Windows:** You can capture your active window by selecting `Alt + Print Screen` on your keyboard. This command copies an image of your active window to the clipboard. Next, open an image editor, paste the image from your clipboard to the image editor, and save the image.

Project overview

You have been asked by the customer account manager at your company to develop an account microservice to keep track of the customers on your e-commerce website. Since it is a microservice, it is expected to have a well-formed REST API that other microservices can call. This service initially needs to create, read, update, delete, and list customers.

You have also been told that someone else has started on this task. They have already developed the database model and a Python Flask-based REST API with an endpoint to create a customer account. You just need to plan to add the REST APIs to read, update, delete, and list accounts. Since you will be working in an online lab environment, you will need to plan your work to get that environment ready for development.

Exercise 1: Create a GitHub repository

In this exercise, you will create a GitHub repository using a starter template that will be provided for you. You will need your own GitHub repository to be able to push modifications that you make back to GitHub and save them.

> Note: It is important to understand that the lab environment that you will be developing is ephemeral. It is short-lived and may be deleted at any time. For this reason, all of your work must be saved in GitHub so that it can be easily restored.

Steps to be performed

1. Create a new GitHub repository for your project, and name is as `devops-capstone-project`. This is the name that graders will be looking for to grade your work.

   Note: Ensure you select the **Public** option for your repository and then create it.

- Select the URL to open the starter code project: https://github.com/ibm-developer-skills-network/aolwx-devops-capstone-template

- Use the green `[Use this template]` button to clone this repository to your own private GitHub account. (Do not use Fork; use the Template button.)

- Name the created repository as `devops-capstone-project`. This is the name that graders will be looking for to grade your work.

2. Once the repository is created, open the `README.md` file in your new repository. And update it with the project name `devops-capstone-project`. Provide a brief description of the capstone project. Save and commit the changes so that the updated details appear on your repository's main page.

Evidence

- For **Option 1: AI-Graded Submission and Evaluation**, copy and paste the **public GitHub repository URL** of the `README.md` file that contains the project name details.

- For **Option 2: Peer-Graded Submission and Evaluation**, take the **screenshot** and save it as `planning-repository-done.jpeg` or `planning-repository-done.png` that shows the repository name as `devops-capstone-project`, and is Public.

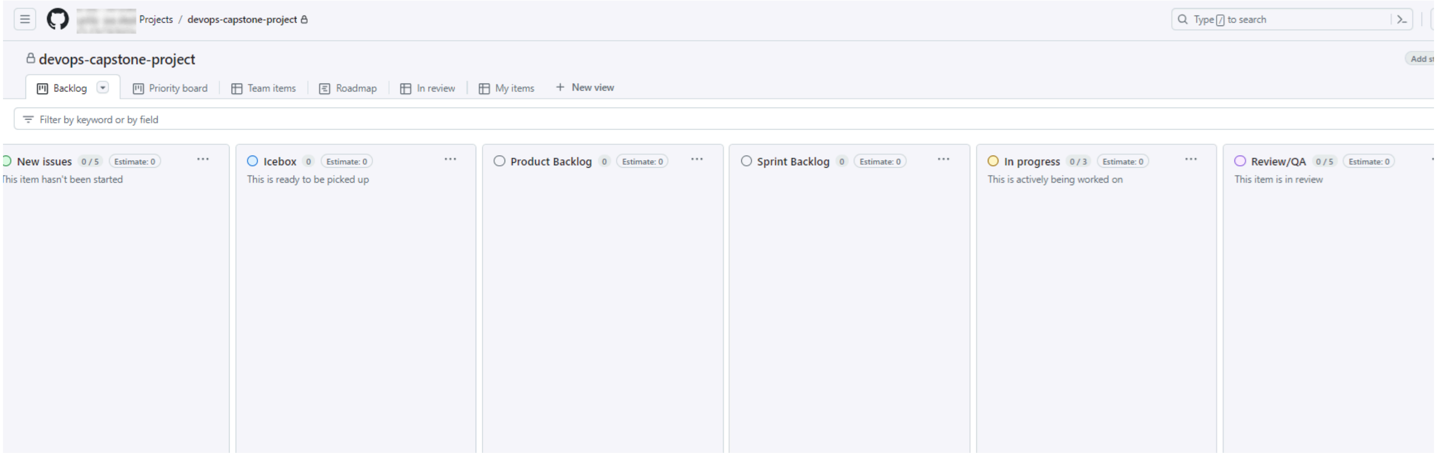
Exercise 2: Create a GitHub Kanban board in your repository

In this exercise, you will set up a kanban board using GitHub Projects for the created repository.

1. Set up your GitHub kanban board by incorporating the seven columns give below:

- **New issues**
- **Icebox**
- **Product backlog**
- **Sprint backlog**
- **In progress**
- **Review/QA**
- **Done**

Refer the screenshot given below:


[redacted]
Projects / devops-capstone-project
Type / to search
devops-capstone-project
Add st
Backlog
Priority board
Team items
Roadmap
In review
My items
+ New view
Filter by keyword or by field
New issues 0 / 5 Estimate: 0
This item hasn't been started
Icebox 0 Estimate: 0
This is ready to be picked up
Product Backlog 0 Estimate: 0
Sprint Backlog 0 Estimate: 0
In progress 0 / 3 Estimate: 0
This is actively being worked on
Review/QA 0 / 5 Estimate: 0
This item is in review

2. If you face any challenges while setting up the GitHub kanban board as per the lab requirements, you can refer to the lab Get Set Up In Git Hub from the **Introduction to Agile Development and Scrum** course.

Exercise 3: Create a user story template

In this exercise, you will create a user story template that will help you write well-formatted user stories for your GitHub kanban board.

Steps to be performed

1. Create a template for your project's GitHub repository. Ensure the template includes the components listed below. You may want to copy, paste, and then edit this text because it also contains the correct markdown syntax you will need for the template.

```
**As a** [role]
**I need** [function]
**So that** [benefit]

### Details and Assumptions
    * [document what you know]
### Acceptance Criteria
```

```
gherkin
Given [some context]
When [certain action is taken]
Then [the outcome of action is observed]
```

2. Ensure that you have a new folder in your repository with name `.github/ISSUE_TEMPLATE`. This folder will contain your new user story template named **user-story.md**.

Evidence

- For **Option 1: AI-Graded Submission and Evaluation**, copy and paste the **public GitHub repository URL** of the `user-story.md` file that contains the user story template and save it in a text file.
- For **Option 2: Peer-Graded Submission and Evaluation**, take the **screenshot** and save it as `planning-storytemplate-done.jpeg` or `planning-storytemplate-done.png` that shows the new folder .github/ISSUE_TEMPLATE created and the user story template.

Exercise 4: Assemble your product backlog

In this exercise, you will create user stories based on the customer accounts microservice for the e-commerce capstone project. The accounts service will have basic customer information like names and addresses.

1. Create seven user stories in your GitHub kanban board, one for each of the following steps of your project:

- Setup the development environment
- Read an account from the service
- Update an account in the service
- Delete an account from the service
- List all accounts in the service
- Containerize your microservice using Docker
- Deploy your Docker image to Kubernetes

2. Ensure that all your stories are listed in the **New Issues** pipeline.

Evidence

For both **Option 1: AI-Graded Submission and Evaluation** and **Option 2: Peer-Graded Submission and Evaluation**.
Take a **screenshot** of your **Kanban board** showing all user stories listed under the **'New Issues'** column and save as `planning-userstories-done.jpeg` or `planning-userstories-done.png`.

Exercise 5: Triage new issues

In this exercise, you will begin to conduct **Backlog Refinement** by inspecting the issues in the `New Issues` pipeline and moving them to the `Product Backlog` or `Icebox` depending on when you plan to work on them. Containerizing with Docker and deploying to Kubernetes is something you will do a few sprints from now, so it is not immediately important.

Steps to be performed

1. Determine which user stories you will work on immediately and move them from the `New Issues` pipeline to the `Product Backlog` pipeline.

2. Move the remaining stories from `New Issues` into the `Ice box` as you will work on them later.

▼

Click here for a hint.

Five stories will be implemented now, and two will be implemented later.

Evidence

For both **Option 1: AI-Graded Submission and Evaluation** and **Option 2: Peer-Graded Submission and Evaluation**.
Take a **screenshot** of your **Kanban board** and save it as `planning-productbacklog-done.jpeg` or `planning-productbacklog-done.png`.

Exercise 6: Refine your product backlog

In this exercise, you will follow the steps of conducting a backlog refinement meeting. You will be the product owner, getting the product backlog ready for your next sprint planning meeting. The goal of this preparation is to make all your stories sprint ready.

Steps to be performed

1. Make sure that all the stories in the `Product Backlog` have sufficient details to be considered "sprint ready." Pay special attention to the **Acceptance Criteria** to be sure that you have defined the definition of "done."

2. Create a label called `technical debt` with a `yellow` color code and add it to your repository.

3. Assign labels to your stories. Remember that anything that brings value to the customer is an `enhancement`, and `technical debt` can be things that developers need but provide no visible customer value.

4. Rank the stories in the `Product Backlog` from highest to lowest priority by dragging them higher or lower in the pipeline column, respectively. Think about the order in which you should implement them.

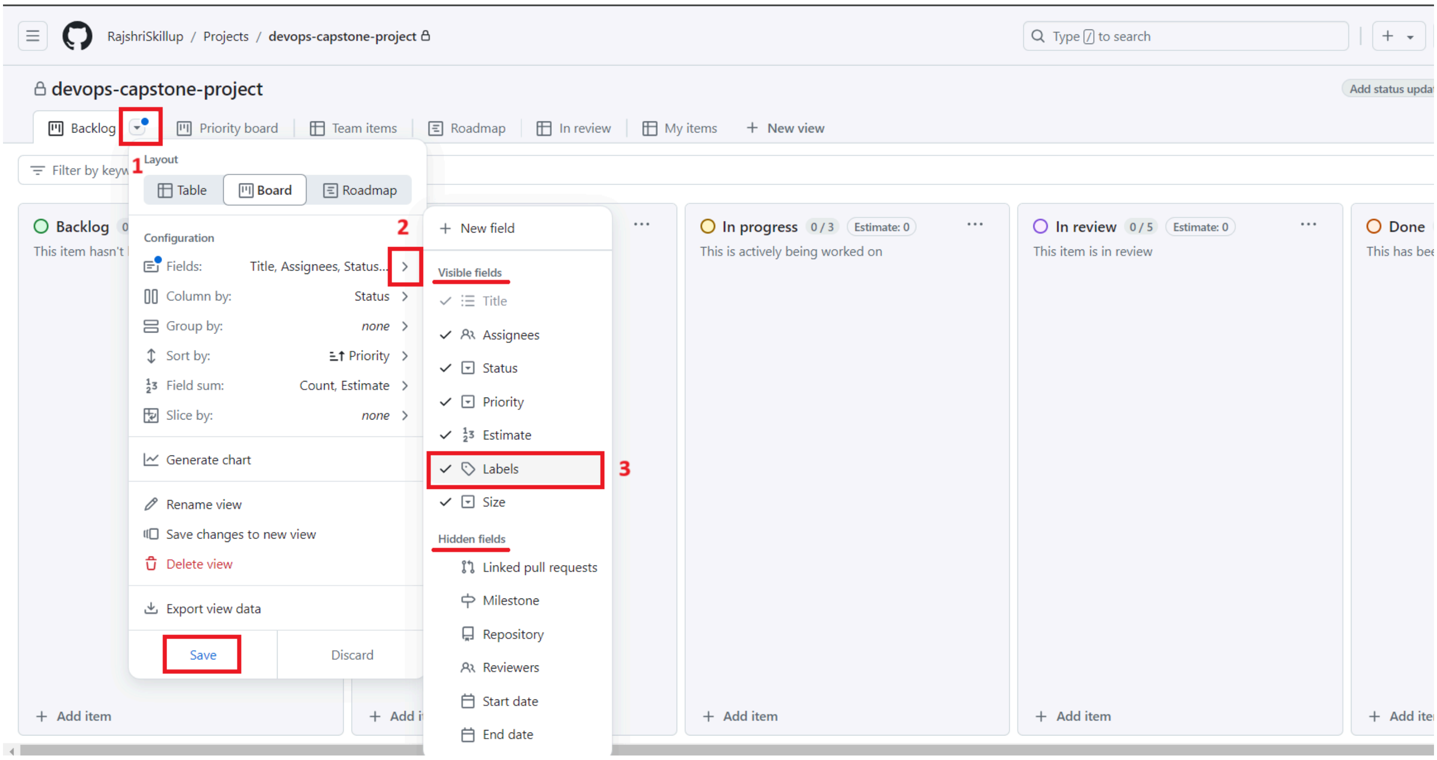
▼

Click here for a hint.

At least one story should be labeled "technical debt."

> **Note:** If you have added the labels but cannot see them on the kanban board, please follow these steps:

- Open the dropdown menu next to the `Backlog` (highlighted as 1 in the screenshot below).
- Select the arrow next to the `Fields` option (highlighted as 2).
- Drag and drop Labels from the `Hidden fields` list to the `Visible fields` list (highlighted as 3).
- Select `Save` to save the changes.



Evidence

For both **Option 1: AI-Graded Submission and Evaluation** and **Option 2: Peer-Graded Submission and Evaluation**
Take a **screenshot** of your **Kanban board** and save it as `planning-labels-done.jpeg` or `planning-labels-done.png`.

Exercise 7: Build your first sprint from your product backlog

In this exercise, you will populate your first Sprint Backlog from your Product Backlog. Typically, you create a sprint plan with your entire team during the sprint planning meeting. But since you are completing this project by yourself, you will have to simulate that meeting.

Steps to be performed

1. Set up three sprints (**Sprint 1**, **Sprint 2**, and **Sprint 3**) in GitHub and make the sprints one week in duration.
2. Open the first story from the top of the `Product Backlog` and assign estimated story points to it. For now, use the scale **3, 5, 8, 13 = S, M, L, XL**.
3. Assign the current story to **Sprint 1**.
4. Close the current story and move it from the `Product Backlog` to the `Sprint Backlog`, being careful to preserve its ranked order. For example, the story at the top of the `Product Backlog` would remain the top story in `Sprint Backlog`.
5. Repeat steps 3–5 for the remaining four stories in the `Product Backlog`, preserving their ranked order.

▼

Click here for a checklist.

Ensure each of these stories is "sprint ready" by checking the following:

- Each story has a label
- Each story has an estimate
- Each story is assigned to Sprint 1

Evidence

For both **Option 1: AI-Graded Submission and Evaluation** and **Option 2: Peer-Graded Submission and Evaluation**
Take a **screenshot** of your **Kanban board** and save it as `planning-kanban-done.jpeg` or `planning-kanban-done.png`.

Summary

Congratulations on completing **Lab: Agile Planning Using GitHub!** In this lab, you have created your first sprint plan for this capstone project. You are now ready to start implementing your account microservice.

Author(s)

Tapas Mandal
John J. Rofrano