Agile Planning

Estimated time needed: 60 minutes

Welcome to the hands-on Agile Planning lab. 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 and Zenhub. You will not be using the lab environment.

Objectives

In this lab, you will:

- Create a GitHub repository
- Add a GitHub repository to a Zenhub kanban board
- Develop a user story template
- Add user stories to a kanban board
- Sort user stories to prepare for backlog refinement
- Refine a product backlog to make it sprint ready
- Build a sprint plan

Note on Screenshots

Throughout this lab, you will be prompted to take screenshots and save them on your device. You will need these screenshots either to answer graded quiz questions or to upload and submit for peer review at the end of this course. Your screenshot must have either the .jpg or .png extension.

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.

- **Windows:** You can capture your active window by pressing `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.

You will also connect your GitHub repository to your Zenhub account and, optionally, install the Zenhub plugin in your browser.

Steps to Complete

1. Create a new GitHub repository for your project, called `devops-capstone-project`, from the template provided by the URL below:
    - Click this 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.)
    - Give your repository the name `devops-capstone-project`. This is the name that graders will be looking for to grade your work.
    - Ensure you select the **Public** option for your repository and then create it.
2. Connect your Zenhub account with your GitHub account if it is not already connected.
3. In Zenhub, create a kanban board workspace for your new repository. Call the workspace `capstone-project` and make sure that your repository is added to your workspace.
4. (Optional) Install the [Zenhub browser extension]() to the Chrome or Firefox browser. This extension will add a Zenhub tab so that you do not have to go to zenhub.com to view your kanban board while you are using GitHub.

Evidence

1. Take a screenshot of the repository and save the screenshot as `planning-repository-done.jpg` (or `planning-repository-done.png`).
2. Copy and save the URL for the repository that you just created.

Exercise 2: Create a User Story Template

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

Steps to Complete

1. Create an issue template in GitHub for your project's GitHub repository. Ensure the template includes the components listed below. You may want to copy, paste, 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]
```

Copied!

2. Ensure that you have a new folder in your repository called `.github/ISSUE_TEMPLATES`. This folder will contain your new user story template.

Evidence

1. Take a screenshot of the repository template and save the screenshot as `planning-storytemplate-done.jpg` (or `planning-storytemplate-done.png`).

Exercise 3: 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 Zenhub kanban board, one for each of the following steps of your project:

    - Set up 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.

3. Take a screenshot of your kanban board and save the screenshot as `planning-userstories-done.jpg` (or `planning-userstories-done.png`).

Exercise 4: 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 Complete

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.

Evidence

1. Take a screenshot of your kanban board and save the screenshot as `planning-productbacklog-done.jpg` (or `planning-productbacklog-done.png`).

Exercise 5: 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 Complete

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. In GitHub, 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.

Evidence

1. Take a screenshot of your kanban board and save the screenshot as `planning-labels-done.jpg` (or `planning-labels-done.png`).

Exercise 6: 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 Complete

1. Set up sprints in Zenhub and make the sprints one week in duration.

2. Rename the three sprints that Zenhub creates as: **Sprint 1**, **Sprint 2**, and **Sprint 3**, respectively. This will make it easier for you to track and others to grade.

3. 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**.

4. Assign the current story to **Sprint 1**.

5. 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 the `Sprint Backlog`.

6. Repeat steps 3 through 5 for the remaining four stories in the `Product Backlog`, preserving their ranked order.

▶ Click here for a checklist.

Evidence

1. Take a screenshot of your kanban board and save the screenshot as `planning-kanban-done.jpg` (or `planning-kanban-done.png`).

Conclusion

Congratulations! You have created your first sprint plan for this Capstone project. You are now ready to start implementing your account microservice.

Next Steps

Your next step is to start Module 2: Develop a RESTful Service Using .

Author(s)

Tapas Mandal
John J. Rofrano

Other Contributor(s)

Change Log

Date	Version	Changed by	Change Description
2022-10-03	0.1	Tapas Mandal	Initial version created
2022-10-05	0.2	John Rofrano	Added more details and reformatted
2022-10-11	0.3	Amy Norton	ID Review
2022-10-12	0.4	Beth Larsen	QA review
2022-10-14	0.5	John Rofrano	Updated screenshot image names