

Agile Planning and Management using GitHub -

by **RAJ KIRAN**

Scenario

Remember this module's scenario in which you're working for a software development company in the retail industry that is planning to migrate its online store to a new app but is experiencing difficulties planning the project due to little collaboration and communication between the development and operations teams. Since you have decided to use GitHub for Agile planning and management, this lab gives you the opportunity to create a GitHub repo, associated milestones and issues, a project, and project board. Additionally, you'll be able to add a draft item to the project board and an item based on an issue and review the automation settings.

Objectives

In this lab, you will:

- Create a GitHub repo, project, and project board
- Create and manage project board items

Note: For this and subsequent labs, you will need a GitHub account. We strongly recommend creating a separate account to use in the labs. To create a GitHub account, follow instructions in the article [Creating an account on GitHub](#).

Exercise 1: Create a GitHub repo, project, and project board

In this exercise, you'll create a GitHub repo, project, and project board.

Note: As per GitHub documentation, Projects offer *an adaptable, flexible tool for planning and tracking work on GitHub*. Projects provide access to *boards*, which serve the role of *Kanban* boards. Kanban is a common and widely used framework in an Agile environment to represent the state of the project's work.

The exercise consists of the following tasks:

- Task 1: Create a GitHub repo
- Task 2: Create GitHub milestones and issues
- Task 3: Create a GitHub project
- Task 4: Create a GitHub project board

Task 1: Create a GitHub repo

1. Start a web browser and navigate to the [GitHub](#) home page.
2. When prompted to authenticate, sign in by using your GitHub user account.
3. On the GitHub main page, select the Repositories tab and then select New.
4. On the Create a new repository page, do the following actions:
 - In the Owner drop-down list, select your GitHub user account name.
 - In the Repository name text box, enter `DevOpsCoreIntroRepo`.
 - Change the visibility of the repo to Private.
 - Enable the Add a README file checkbox.
 - In the Add .gitignore drop-down list, select Visual Studio.
5. **Note:** To learn more about .gitignore, refer to [Ignoring files](#).
 - In the Choose a license drop-down list, select MIT.
6. **Note:** To learn more about license selections, refer to [Licensing a repository](#).
7. Select Create repository.

Task 2: Create GitHub milestones and issues

1. On the DevOpsCoreIntroRepo page, select the Issues tab.
2. **Note:** You will create a milestone and an issue associated with the newly created repo that you will subsequently use when working with the project board.
3. On the Issues page, to the left of the New issue button, select Milestones.
4. On the Milestones page, select New milestone.
5. On the New milestone page, do the following actions:
 - In the Title text box, enter `alpha release`.
 - In the Due date (optional) text box, enter the date one week ahead from the current date.
 - In the Description text box, enter `Completion of the alpha release`.
6. Select Create milestone.
7. Repeat the last three steps to create an beta release milestone with the due date two weeks ahead of the current date. In the Description text box, enter `Completion of the beta release`.

8. Navigate back to the Issues page and select New issue.
9. In the Add a title text box, enter Repo README page is empty.
10. In the Add a description text box, enter Brevity might be a virtue, but this README page can really use some text.
11. Select the gear icon next to the Milestone entry and, in the drop-down list, select alpha release.
12. Select the gear icon next to the Labels entry and, in the drop-down list, select bug.
13. Select Submit new issue. Note that the issue has been automatically assigned #1.

Task 3: Create a GitHub project

1. On the GitHub main page, select the avatar icon at the right side of the toolbar and, in the drop-down menu, select Your projects.
2. Select New project.
3. In the Select a template pane, select Team planning, and then select Create project.
4. Note: Alternatively, you could start from scratch and display the project in the table, board, or roadmap format.
5. On the new project page, select the autogenerated project name. This will automatically display the Project settings page.
6. In the Project name text box, enter DevOps Core Intro Project.
7. In the Short description text box, enter Introduction to GitHub Projects and select Save.
8. In the README section, enter the following text
9. Note: The README section includes a simplified Markdown editor, assisting you with creating visually appealing README page for the project. You could use the toolbar icons to format the text and use the Preview tab to review the resulting changes. Copy and paste the following text into the README editor section:
10. codeCopy

```
### Welcome to DevOps Core Intro Project ###
```

```
**Projects are a customizable, flexible tool for planning and tracking your work.**
```

To find out more, refer to GitHub documentation [about Projects] (<https://docs.github.com/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>) .

- 11.
12. Select Preview to review the resulting page.
13. Select Save.
14. Scroll down to the Danger zone and note the option allowing you to switch the visibility of the project between Private and Public, close the project, and delete it.
15. Note: Do not close or delete the project at this point. Only note the options available to you.
16. Scroll up to the top of the page and note that you have the options to Manage access to the project and create custom fields in the project interface.
17. Select the left facing arrow in the <- Settings label to exit the Settings page.

Task 4: Create a GitHub project board

1. On the DevOps Core Intro Project page, select the down-facing caret next to the Backlog tab and, in the Layout section, select Board.
2. Note: You can use this option to easily switch between the table, board, and roadmap views.
3. Review the resulting page that consists of three predefined columns labeled:
 - Todo - including items which haven't been started
 - In Progress - including items being actively worked on
 - Done - including items that have been completed
4. Note: This layout represents a very basic Kanban board. Within each column, you can add individual items. You can also add extra columns.
5. To add an extra column, select the + icon to the right of the Done column and then select + New Column.
6. In the New option window, in the Label text text box, enter Review In Progress and select a color you want to assign to the column. In the Description text box, enter This item is being reviewed, and then select Save.
7. Select the small circle next to the Review in Progress label of the newly added column and use it to drag it between the In Progress and Done column.

Exercise 2: Create and manage project board items

In this exercise, you'll create and manage project board items

Note: There are two basic ways of adding items to a project board. You can create an item draft or add an item representing an existing issue in a GitHub repository.

The exercise consists of the following tasks:

- Task 1: Add a draft item
- Task 2: Add an item based on an issue
- Task 3: Review automation settings

Task 1: Add a draft item

1. On the DevOps Core Intro Project page, in the Todo column, select + Add item.
2. **Note:** In the automatically displayed text box, you can either start typing to create a draft or type # to reference an existing issue in any of your GitHub repositories. We will start with the first of these two techniques.
3. In the text box, enter `Missing Wiki` and then press Enter on the keyboard. This will add a new draft item to the Todo column.
4. In the newly added draft item, select the ellipsis symbol and, in the drop-down menu, select Convert to issue.
5. In the Select an item drop-down list, select DevOpsCoreIntroRepo to add the item to the repo you created in the previous exercise. Note that the issue has been automatically labeled with #2.
6. Select the Missing Wiki issue.
7. In the Missing Wiki #2 pane, note that you have another configuration settings available at this point, including labels and milestones.
8. Select Add labels and, in the Select items drop-down list, select enhancement.
9. Select Add milestone and, in the Select an item drop-down list, select alpha release.
10. Close the Missing Wiki #2 pane.
11. **Note:** Now you will add another draft item and convert it into an issue.
12. On the DevOps Core Intro Project page, in the Todo column, select + Add item.
13. In the text box, enter `Additional collaborators needed` and then press Enter on the keyboard. This will add a new draft item to the Todo column.
14. In the newly added draft item, select the ellipsis symbol and, in the drop-down menu, select Convert to issue and select DevOpsCoreIntroRepo to add the item to the repo.

Task 2: Add an item based on an issue

1. On the DevOps Core Intro Project page, in the Todo column, select + Add item.
2. In the text box, enter # to display a list of existing repositories. In the list, select DevOpsCoreIntroRepo.
3. Next, in the list of existing issues, select Repo README page is empty #1. This will automatically add that issue to the Todo column.
4. Select Repo README page is empty to display the corresponding pane.
5. In the Repo README page is empty pane, in the Assignees section, select Add assignees..., in the Select items drop down list, select your GitHub user name, and close the Repo README page is empty pane.
6. Drag the Repo README page is empty item from the Todo column and drop it in the In Progress column.

Task 3: Review automation settings

1. In the DevOps Core Intro Project page, directly under the avatar icon, select the ellipsis icon.
2. In the drop-down list, select Workflows.
3. On the Workflows page, in the menu on the left side, review the list of default workflows.
4. Note: By default, the Item closed and Pull request merged workflows are enabled.
5. Select Item closed and review the workflow. The workflow automatically sets the status of an issue to Done, whenever that item is closed.
6. Select the avatar icon at the upper right corner of the page and, in the drop-down menu, select Your repositories.
7. In the list of repositories, select DevOpsCoreIntroRepo.
8. In the README section, select the pencil icon to open the file in the edit mode.
9. Replace the existing content with the following text:
10. `codeCopy`

```
### Welcome to DevOps Core Intro Project Repository ###
```

```
**Projects are a customizable, flexible tool for planning and tracking your work.**
```

```
To find out more, refer to GitHub documentation [about Projects] (https://docs.github.com/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects).
```

- 11.
12. Select Commit changes.
13. In the Commit changes window, accept the default settings and select Commit changes again.
14. Select the Issues tab.
15. Select Repo README page is empty.
16. On the Repo README page is empty #1 page, select Close issue.
17. Note that closing the item resulted in the following actions:
 - The status of the item was automatically changed to Done, as indicated by an extra comment stating that github-project-automation bot moved the item from In Progress to Done in DevOps Core Intro Project.
 - alpha release milestone has been marked as 50% complete, as indicated by the green horizontal bar in the Milestone section of the page.
18. Note: In case you don't see the changes, refresh the page.
19. To verify this, navigate back to the board view of DevOps Core Intro Project and note that the Repo README page is an empty item appearing in the Done column.