

Project Management Course  
*Zenhub + Github, when agile become a reality*

Aitor Corchero

# Project Management

Introduction .....	1
Agile Methodology in Github & Zenhub .....	2
Hands on Github and Zenhub .....	3
STEP 1. Initialise Github .....	3
STEP 2. General Backlog .....	6
STEP 3. Sprint Backlog .....	9
STEP 4. Sprint Planning .....	11
STEP 5. Product Release .....	11
Conclusion .....	14
References .....	14

# Introduction

Agile development or simply agile is a software methodology development that promotes the flexibility, interactivity and transparency [Szalvay-2004].

Agile software development projects are based on the nature of evolving (not fixed) requirements, (re-) usable code and the continuous testing. This methodology (see [Table 1](#)) key elements are:

*Table 1. Agile key elements*

Element	Description
Stories/User Stories	Refers to the user requirements that also corresponds with product features.
Sprint Backlog	The collection of work scheduled for a sprint. At the beginning of each sprint, the team decides the work to do during the sprint. These tasks should be organised by priority.
Global backlog (or icebox)	A collection of all user-stories. When a new user-story is defined, it is stored in the global backlog for their selection. Thus, the development team select them and introduce into the sprint. The main intention is to prioritize as well the tasks for making simple their selection.
Sprints	Iteration cycles of the development cycle. Normaly 2-4 weeks. The main objective is to provide a small development (or prototype) to the customers.

The agile development cycle ([Figure 1](#)) starts with the definition of the product backlog (user-stories, featues, etc.). During the spring planning, stories are selected to be implemented. Thus, it pass to the sprint backlog.

During the sprint development the stories are implemented and evaluated during 1-4 weeks. Each day during the sprint, several daily scrum meetings are performed to evaluate the work, the level of accomplishent and the project execution. Then, burndown and velocity charts are revised to see if the sprint is delayed, on time or faster than expected.

At the end of each sprint, a sprint review is performed in order to underestand the mistakes (learn from the experience). Moreover, the product is provided to the customer and a feedback from user is received, an new stories could appear.

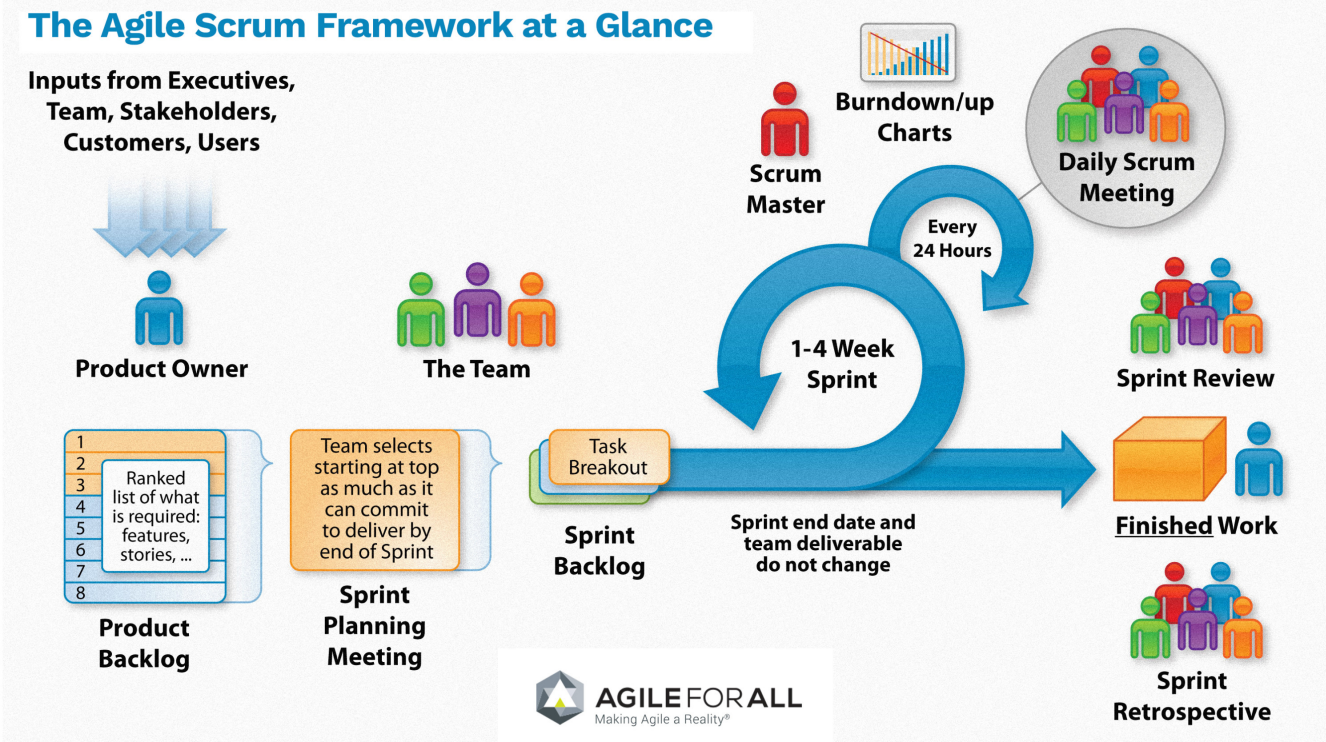


Figure 1. Agile development cycle

# Agile Methodology in Github & Zenhub

The agile methodology is fully compatible with Github and Zenhub. In detail, Github provides a framework for managing documents (code, text, etc), continuous follow-up of the project (using the Wiki, for example), and more important, managing the product releases (output of each sprint).

Complementary, Zenhub provides a kanban model for managing the global backlogs, sprint backlogs, provide burndown reports, managing the issues life-cycle and, define objectives (sprints and product releases).

Considering these overall description of the Github and Zenhub usage, the [Table 2](#) provide the association between Agile and the depicted tools.

Table 2. Agile vs Github & Zenhub

Element	Description
Stories/User Stories	Issues
Sprint Backlog	Backlog list in the board (Zenhub). Milestoned & unassigned issues.
Global backlog (or new issues)	Icebox in the board (Zenhub). Unmilestoned & unassigned issues
Sprints Planning	Issues associated to the milestone
Sprints	Milestones
Sprint Planning Meeting	Wiki (Github)
Daily Scrum Meeting	Wiki (Github) & Burndown/up charts (Zenhub reports)

Element	Description
Sprint Review	Wiki (Github)
Finished Work	Code release (Github)

# Hands on Github and Zenhub



### Customer need

eCORP, a larger company in business development, wants to elaborate the course documentation for training their customers and employees. With this regards, we need to manage the development and resources to do it efficiently in 1 week.

## STEP 1. Initialise Github

To initialise Github, it is necessary to go to <https://github.com>. Once in the github page, you have to log in or sign in into the platform.

**Join GitHub**  
The best way to design, build, and ship software.

**Step 1:** Create personal account

**Step 2:** Choose your plan

**Step 3:** Tailor your experience

**Create your personal account**

**Username**  
  
This will be your username — you can enter your organization's username next.

**Email Address**  
  
You will occasionally receive account related emails. We promise not to share your email with anyone.

**Password**  
  
Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

**Create an account**

**You'll love GitHub**

- Unlimited collaborators
- Unlimited public repositories
- ✓ Great communication
- ✓ Frictionless development
- ✓ Open source community

Figure 2. GitHub Sign In page

Once the Github is created, the next step is to log in the platform and create a new repository.

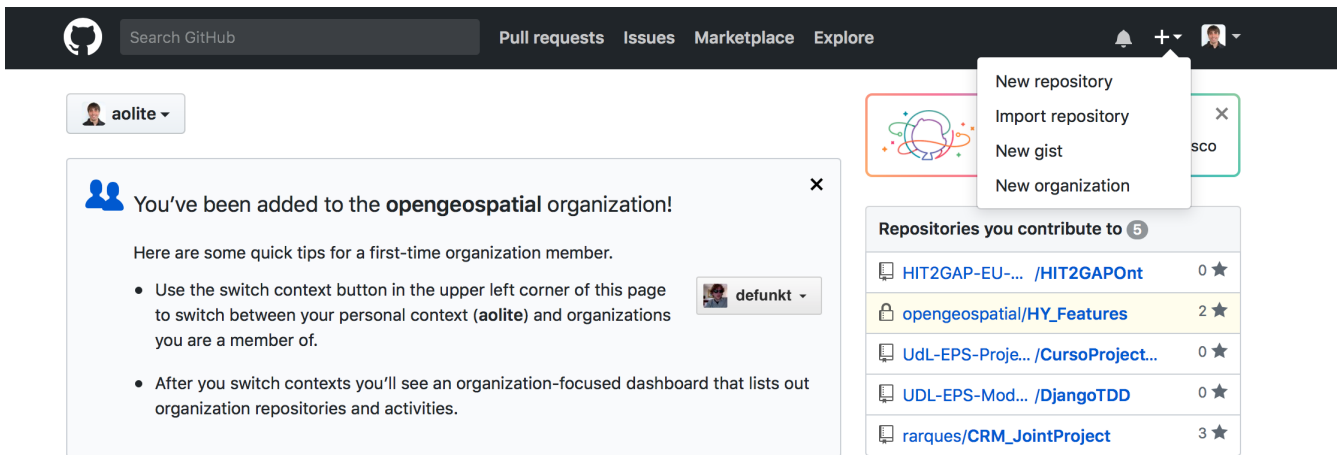


Figure 3. GitHub main user page

In the page of creating a new repository, you have to fill the form including a repository name, description, nature of the repository, and some files for the init (readme and licence files). Once specified this parameters, press the button create the repository to start working.

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

Great repository names are short and memorable. Need inspiration? How about **urban-octo-spork**.

Description (optional)

Public

Anyone can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Figure 4. GitHub Github init repository

Yo have your repository created!!!! Now, browse to the repository to start working with Github.

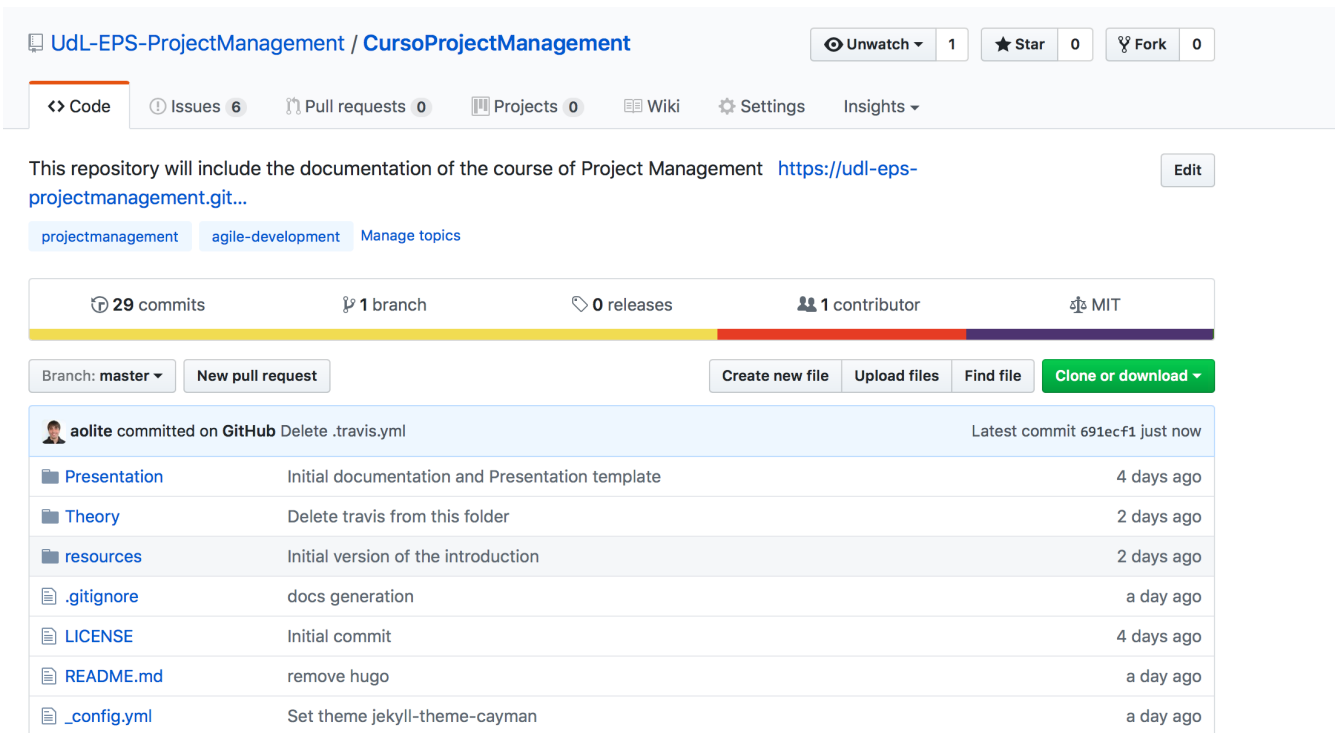


Figure 5. GitHub repository main view

Considering the repository, the main part is the code repository. But before starting with it, we will start with the Wiki (Figure 6). The Wiki will store the sprint planning meeting, daily scrum meeting and sprint review. Moreover, it will serve also to store some ideas or additional information relevant for the project.

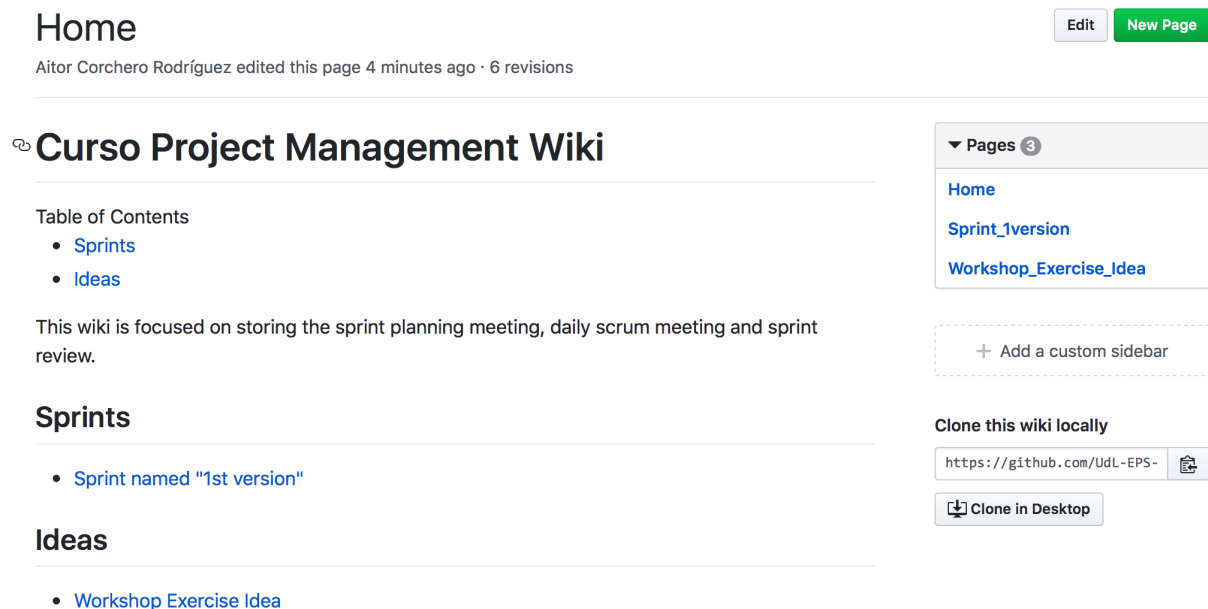


Figure 6. Wiki for storing the meetings minutes


## Sprint Planning Meeting

The sprint planning meeting will cover the agenda for passing the main issues from the backlog to the sprint backlog according to the customer requirements and feedbacks.

## Sprint Planning Meeting

---

<https://github.com/UdL-EPS-> 

 Clone in Desktop

### Agenda

1. Introduction (5-10min)
2. Selection of main user stories (7h max)
3. Conclusion (10 min)

Figure 7. Sprint Planning Minutes and Agenda

## Daily scrum meeting

The daily scrum meeting wiki section will cover the agenda and the main situation and actions to make successful the sprint.

### Daily Scrum Meeting (2017-09-01)

---

#### Agenda

1. Introduction (5 min)
2. Sprint Status (5 min)
3. Conclusion (5 min)

#### Situation & Actions

- Action 1: Issue #3 needs to be redefined.
- Situation 1: As the sprint is delayed, we need to adjust it to the current dates.
  - Corrective Action 1: Make faster features firstly and in parallel perform more heavy ones.

Figure 8. Daily Scrum Meeting Minutes and Agenda

## Sprint Review

This part will comprise the sprint review and customer feedback main points.

### Sprint Review

---

- Documentation OK.
- Colors do not fit well, maybe UdL color palette could be defined.

Figure 9. Sprint review main aspects

Once initialising Github and understanding main documents, lets we will start defining the general backlog (issues) and the corresponding sprints.

## STEP 2. General Backlog

For defining and managing the issues, we will use Zenhub (<https://www.zenhub.com/>). For that, we have two options:



- Chrome users: We could install the Zenhub extension (<https://chrome.google.com/webstore/detail/zenhub-for-github/ogcgkffhplmphkaahpmffcafajaocjbd>)
- Rest of web users, we have the web app (<https://www.zenhub.com/>)

Once signed up with the Github account into zenhub, we can start a board based on the desired repository. With that aim, the chrome users could click on the "board" button in the repository page. The others we have to select the desired project from the project list in the Zenhub webpage.

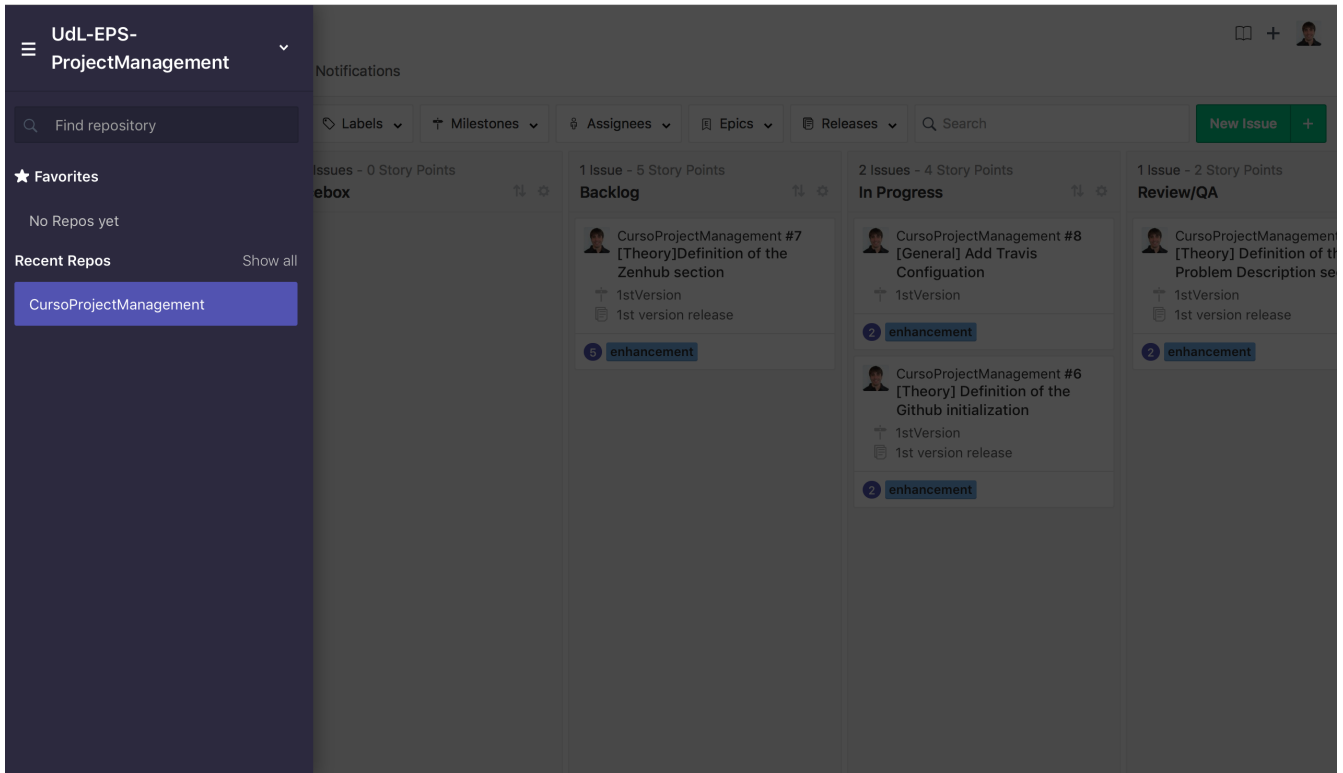


Figure 10. Project selection in Zenhub

The product backlog or "New Issues" appears in the board, so, we need to fill it with the corresponding issues. For that, we have to create new issues indicating the title and the description. The more, we have to specify the label (action of the activity) and the estimation for doing the task. The importance of the task could be determined by the order of the tasks inside the list.

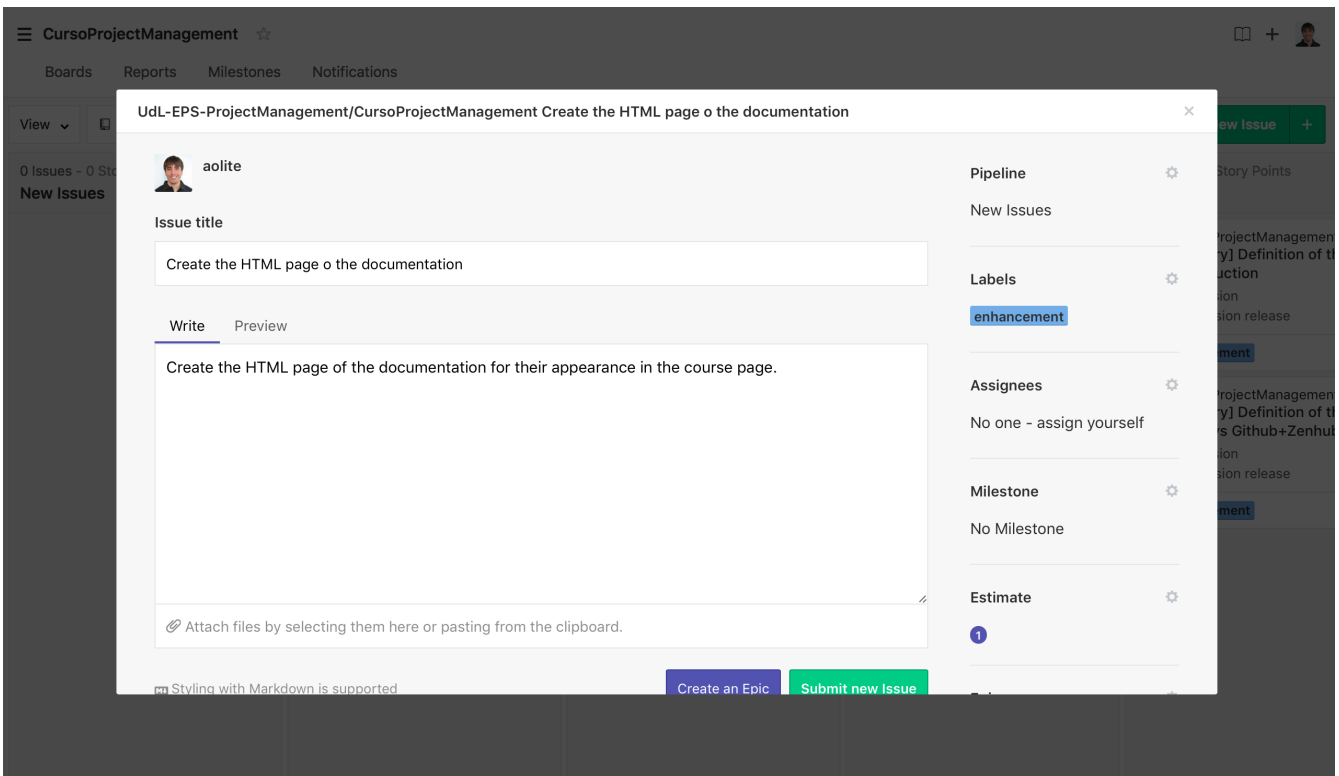


Figure 11. Create new issue for the project management

After this, the board "new issue" contains the desired task to be performed.

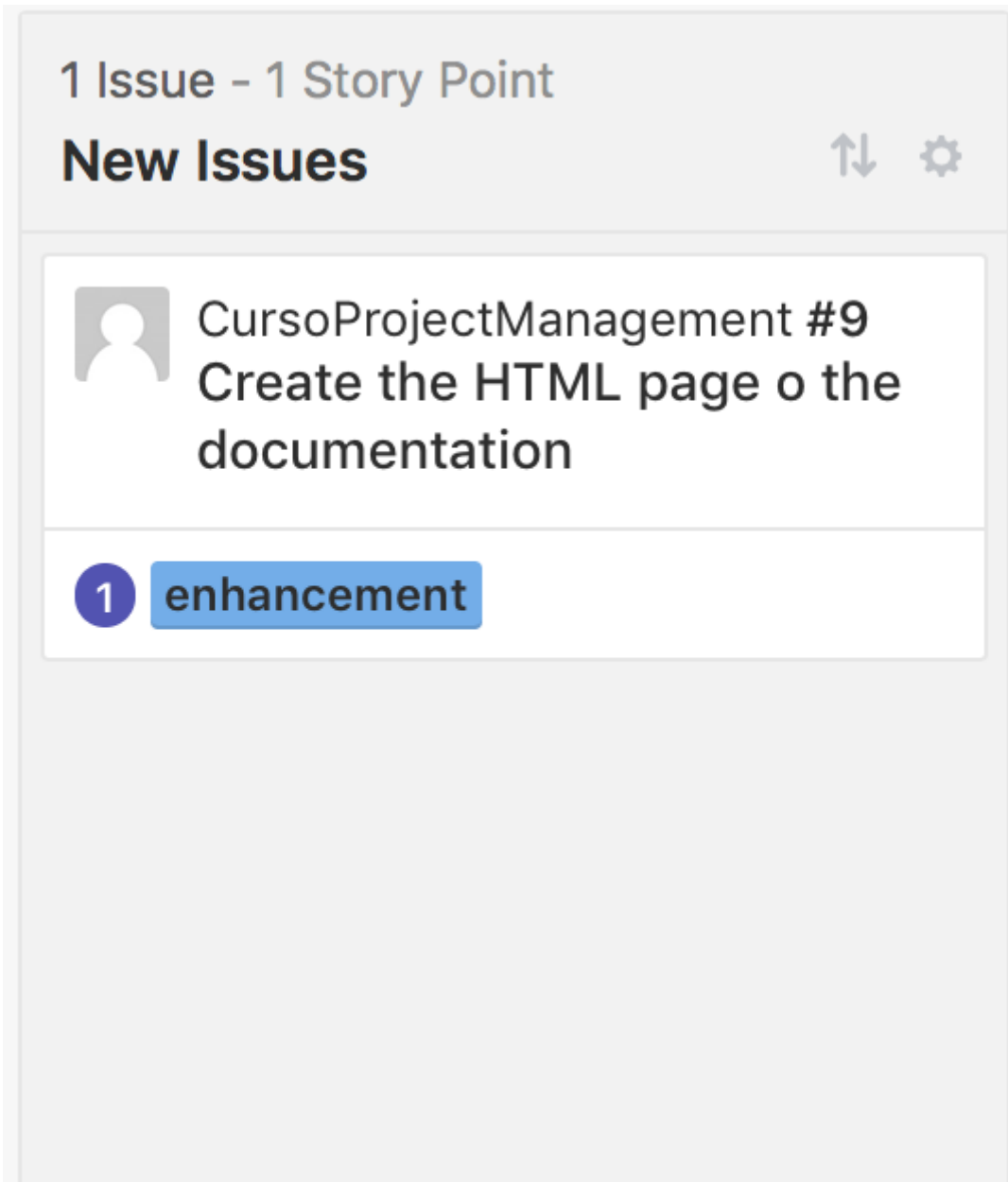


Figure 12. Product Backlog list

### STEP 3. Sprint Backlog

Similar as the product backlog, the sprint backlog (or simply backlog) contains the issues with assignees and milestones.

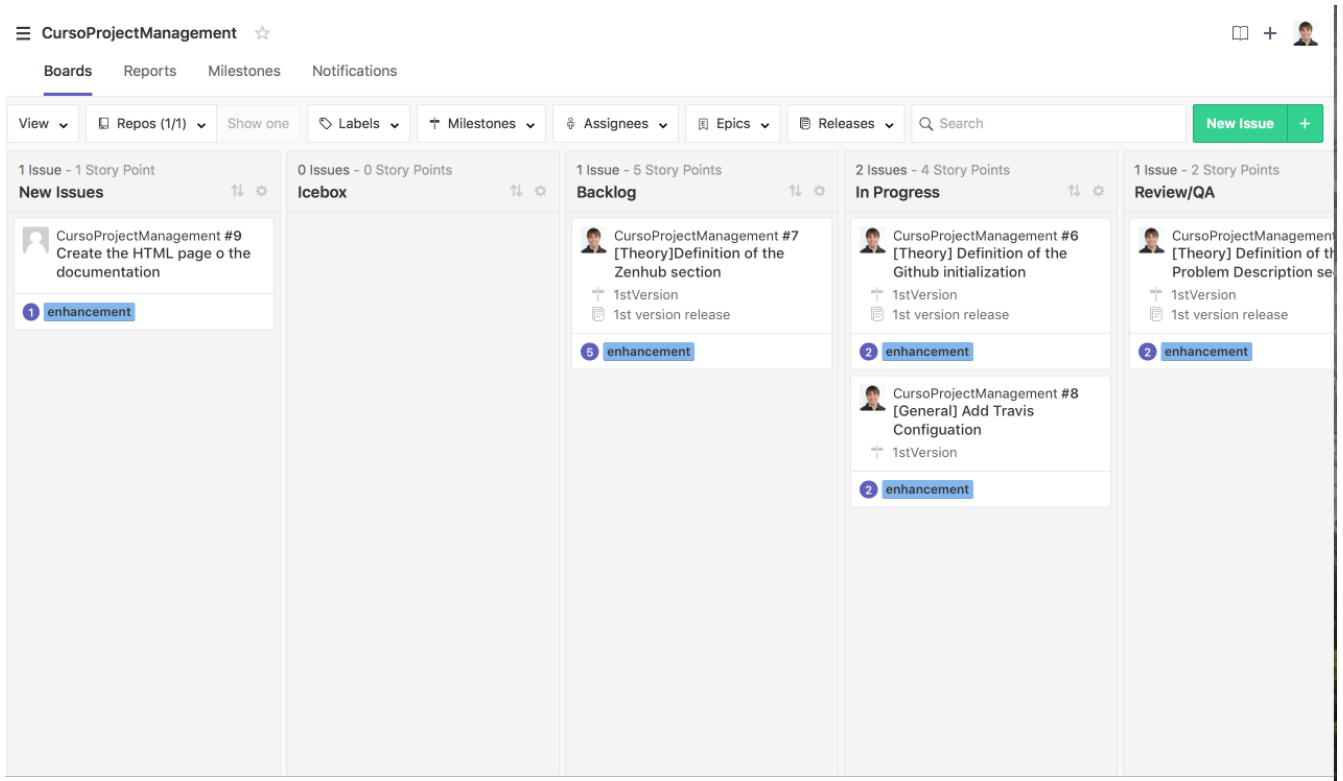


Figure 13. General Overview of the board

For creating a milestone, we have to click on the button "+" at right of "new issue". Once clicked, we select "milestone" to create a new one and, the milestone form appears. In there, we select the title and the description at same as the duration. Once created, We have our first sprint defined!!!

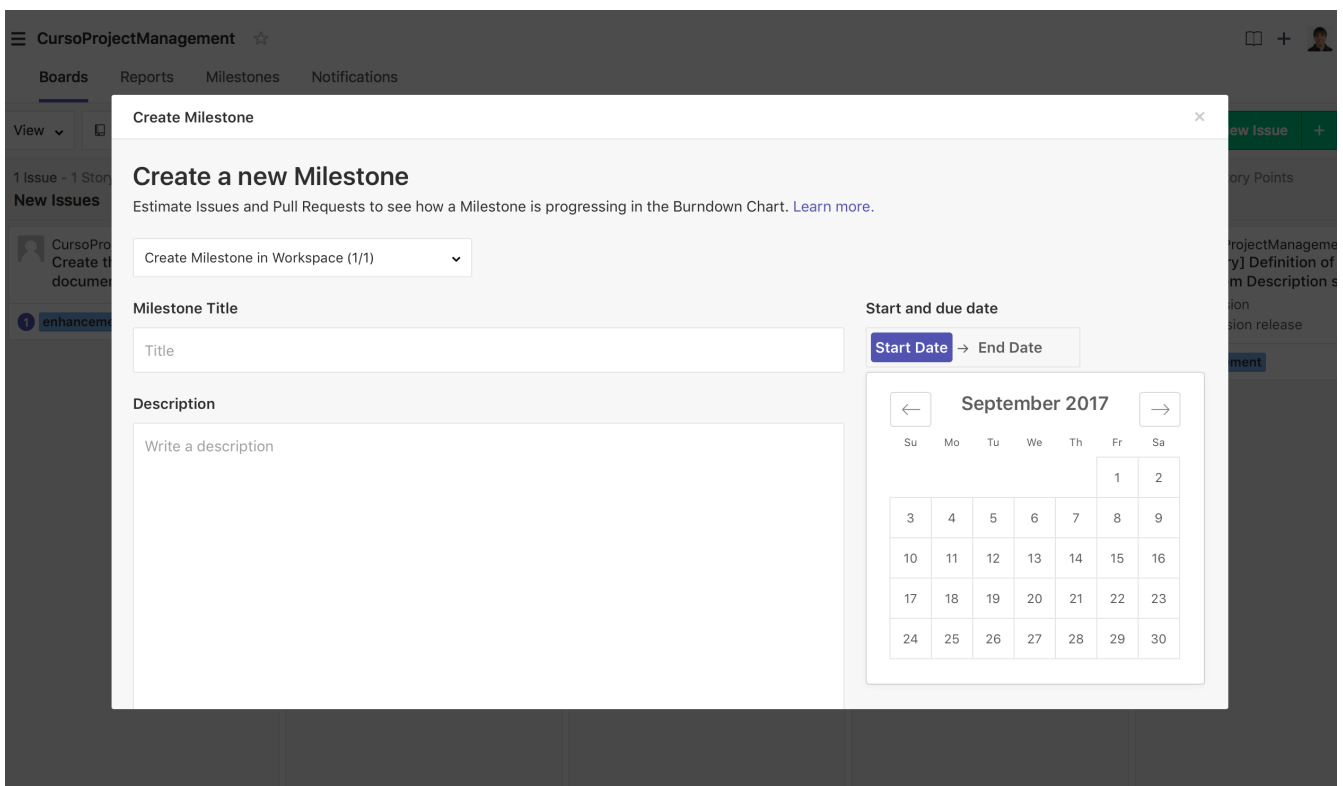


Figure 14. Create new milestone

Just now, we need to assign the tasks to the milestone (task from the sprint backlog) and we have our sprint planned and ready.

## STEP 4. Sprint Planning

The sprint planning corresponds with the issues associated to the milestone. To manage the sprint there exist burn down and velocity graphs just to see how the sprint is going on. These graphs are generated automatically in the Report part of the Zenhub tool.

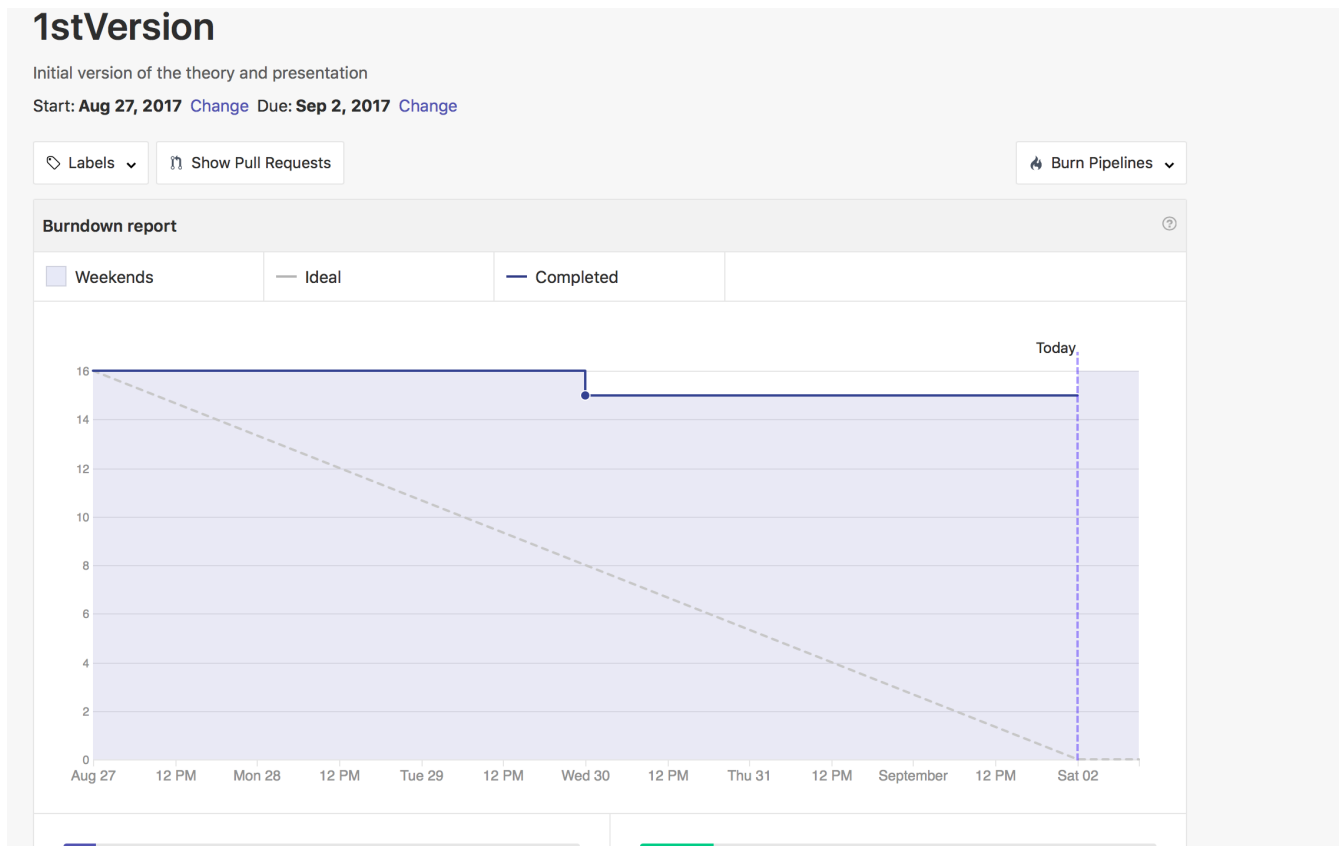


Figure 15. Burndown Report of the sprint

## STEP 5. Product Release

The product release make reference to the end of the sprint and teh presentation of a prototype to the customer for getting their feedback. In the one hand, Zenhub provides an element to manage the product releases and see if the product of the sprint is ging on time or delayed.

# 1st version release

Initial definition of the theory

Start date: Aug 30th, 2017 Desired end date: Sep 1st, 2017

Labels

## Release report

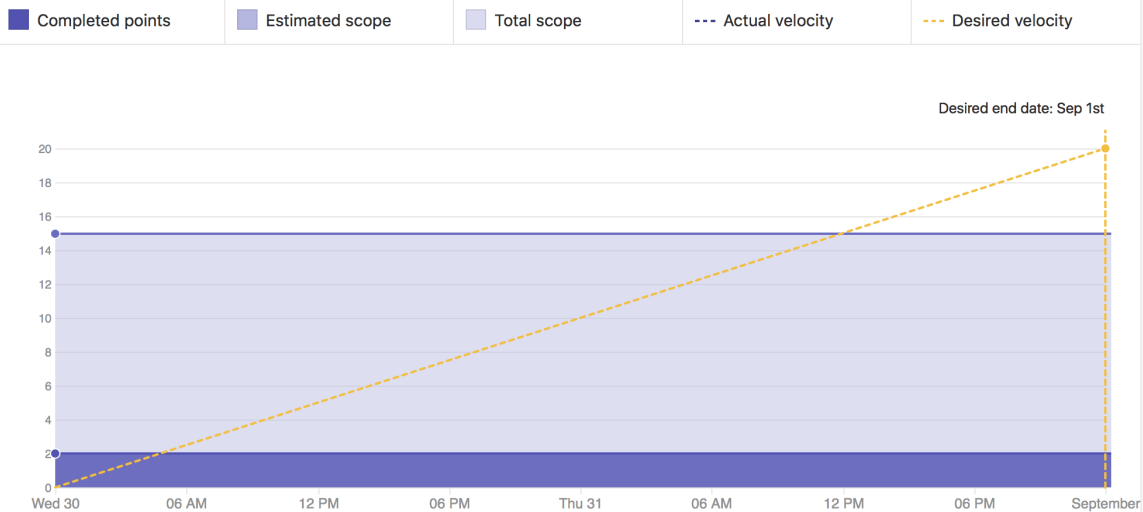


Figure 16. Product Release management graph

To enable this option, we have to create a new release (similarly as the milestone) and assign the issues to that defined release (following similar steps and in the milestones).

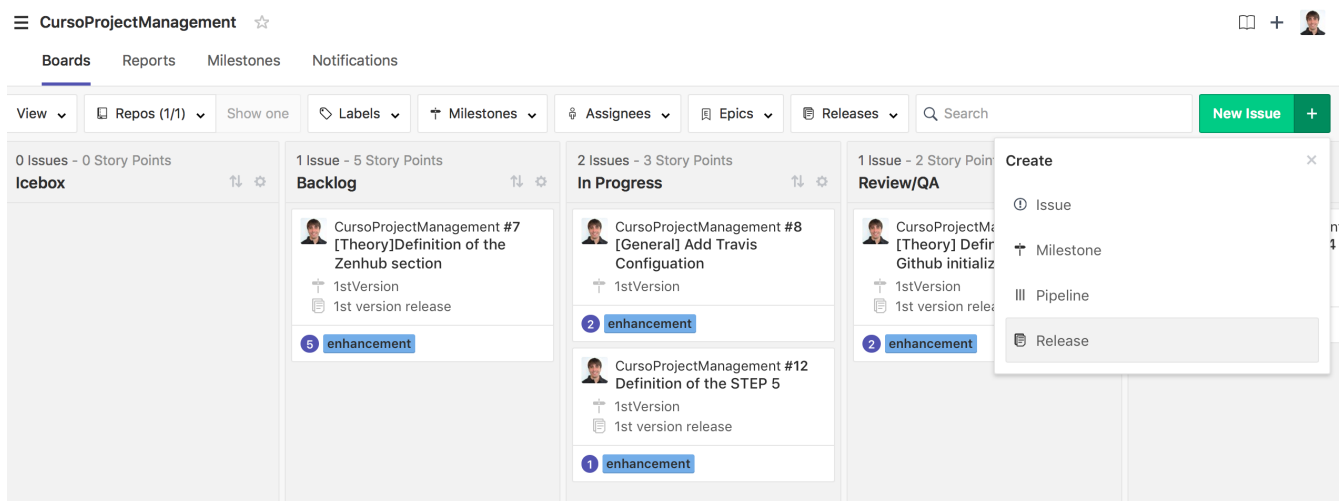


Figure 17. Creation of a new Release in Zenhub

On the other hand, once the sprint is finished, we have to create the release with the corresponding code. For that, Github provide us the necessary tools. For creating a new release, we have to go the repository main page and click on the "Release" tab. Once there, we will click on "new release" and fill the corresponding form.

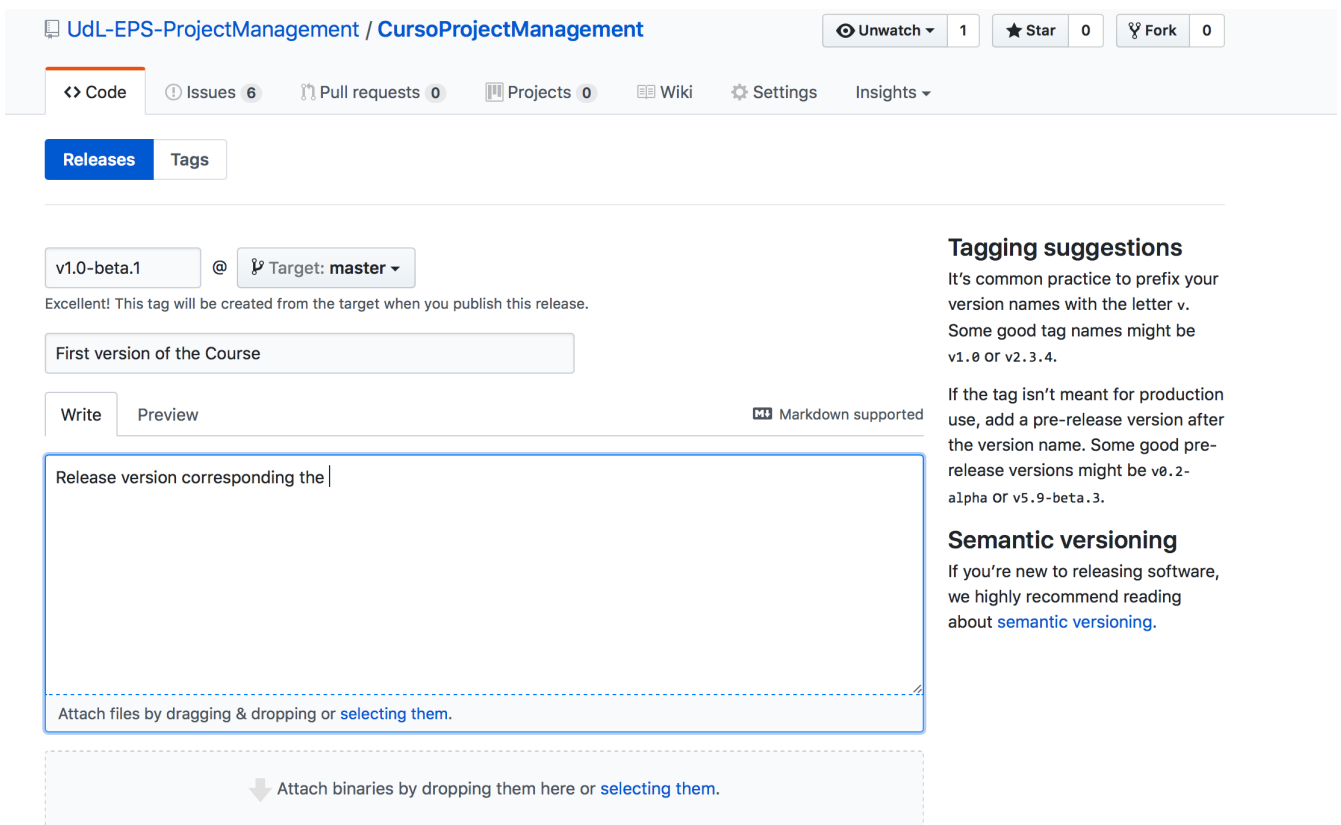


Figure 18. Creation of a new Release in Github

Once completed the form, we click on the button and then, a code zip will be created with the master branch code. This will appear in the releases list.

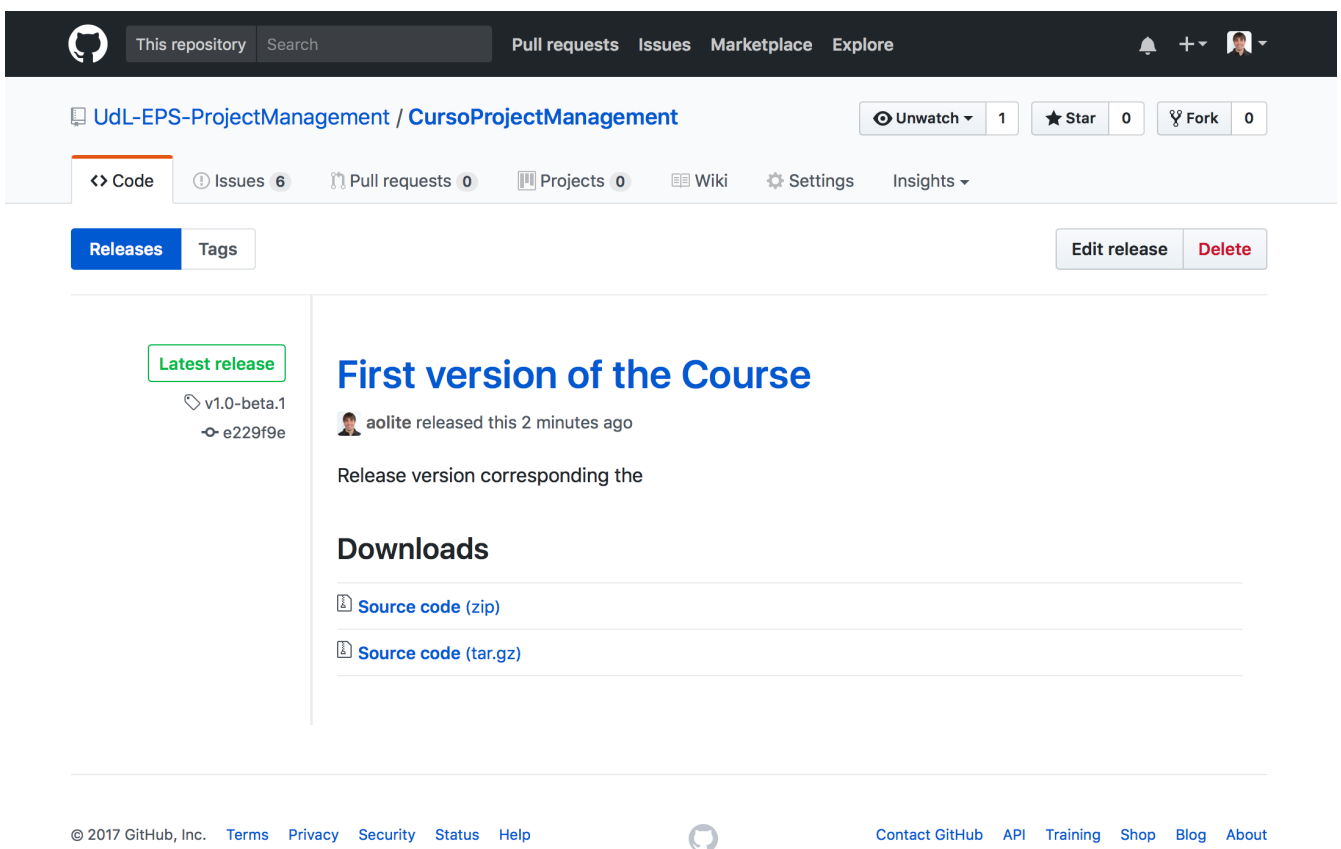


Figure 19. Releases page once the release has been created

# Conclusion

During this part of the course we have learn:

- Agile basics
- Agile implementation in Github
- Agile implementation in Zenhub

# References

Szalvay, V. An Introduction to Agile Software Development (2004). [http://www.danube.com/docs/Intro\\_to\\_Agile.pdf](http://www.danube.com/docs/Intro_to_Agile.pdf)