

Le petit guide de l'outil Agile à l'attention de l'audacieux SCRUM Master

Gitlab

Projet GL
22 juin 2020

Promotion alternant 2022 - équipe 3 :

D'EMMANUELE Valentin

PEREIRA Nelson

VIGNAL Nathan

LIN William

Organisation AGILE avec Gitlab	3
Informations complémentaires sur les issues, USs et UCs:	4
Guideline générale	4
Board	6
Milestone	10
Burndown Chart	11

Organisation AGILE avec Gitlab

Un panneau d'organisation ou board existe sur gitlab celui ci prend en entrée des Issues qui seront l'équivalent des User stories (US) et des Use cases (UC).

Le planning poker qui permet d'attribuer des poids, des points de difficultés à des tâches est équivalent au weight des issues

Le burndown chart est généré automatiquement par gitlab en déclarant des milestones, représentant des sprints.

Informations complémentaires sur les issues, USs et UCs:

Les USs et les UCs n'existant pas réellement sous gitlab il est possible de créer des relations de dépendance entre les issues. Ainsi une Issue dépendant de plusieurs autres issues sera équivalent un a US et les issues dont elle dépend équivalentes aux UCs.

Il est également possible, pour garder un board clair, de ne créer que les USs et d'y définir les UCs associés dans la description. L'assignation d'un membre se fait alors à l'aide d'une mention du membre.

Guideline générale

Au début de chaque sprint:

- Créer une branche sprint
- Créer un nouveau board
- Créer les issues (USs et les UCs)
- Ajouter les issues dans le board du sprint
- Créer un milestone pour le sprint

Pendant le sprint:

- S'assurer que les issues du board sont dans les listes correspondant à leur état d'avancement.
- Si nécessaire, n'hésitez pas à ajouter un poids à vos issues pour que votre burndown chart soient le plus complet possible.
- S'aider du burndown chart pour vérifier que les délais sont respectés.
- Lors d'un commit, renseignez le numéro de l'issue sur lequel vous travaillez
- Une issue est terminée :
 - Mettre le statut de l'issue à Closed
 - Déplacer l'issue dans la liste adéquate du board

A la fin de chaque sprint:

- S'assurer que toutes les issues réalisés ont été fermés
- S'assurer que les issues du board sont dans les listes correspondant à leur état d'avancement
- Si toutes les issues n'ont pas pu être terminées dans le sprint, vous pouvez déplacer les issues d'un sprint à l'autre, l'état du burndown chart étant gelé lorsque celui-ci est arrivé à terme.

Board

Il s'agit d'un outil d'organisation des projets en listes et cartes comme Trello ou Kanban. Il permet de planifier et organiser un flux de travail d'un projet divisé en plusieurs sprint. C'est pour cela qu'il s'agit d'un outil agile important car il permet d'avoir une vue globale sur l'avancement d'un sprint.

Nous conseillons fortement de créer un board par sprint.

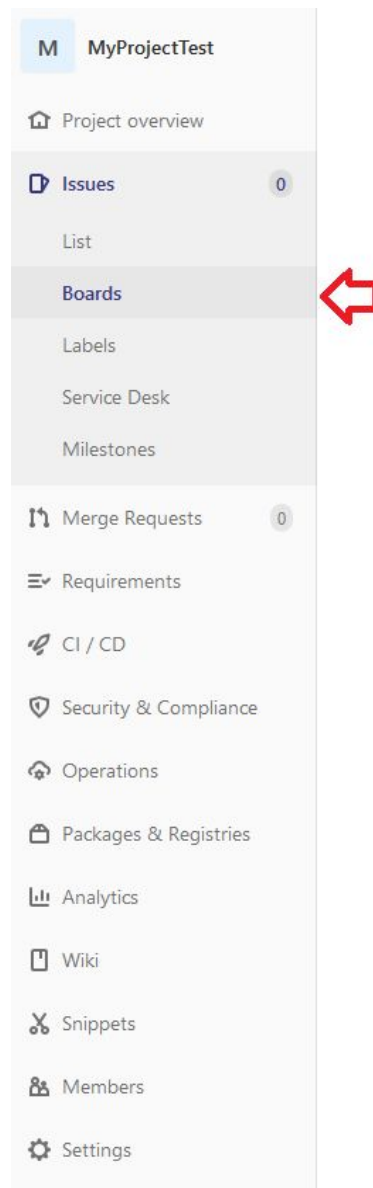


figure 1 - Menu actif Boards

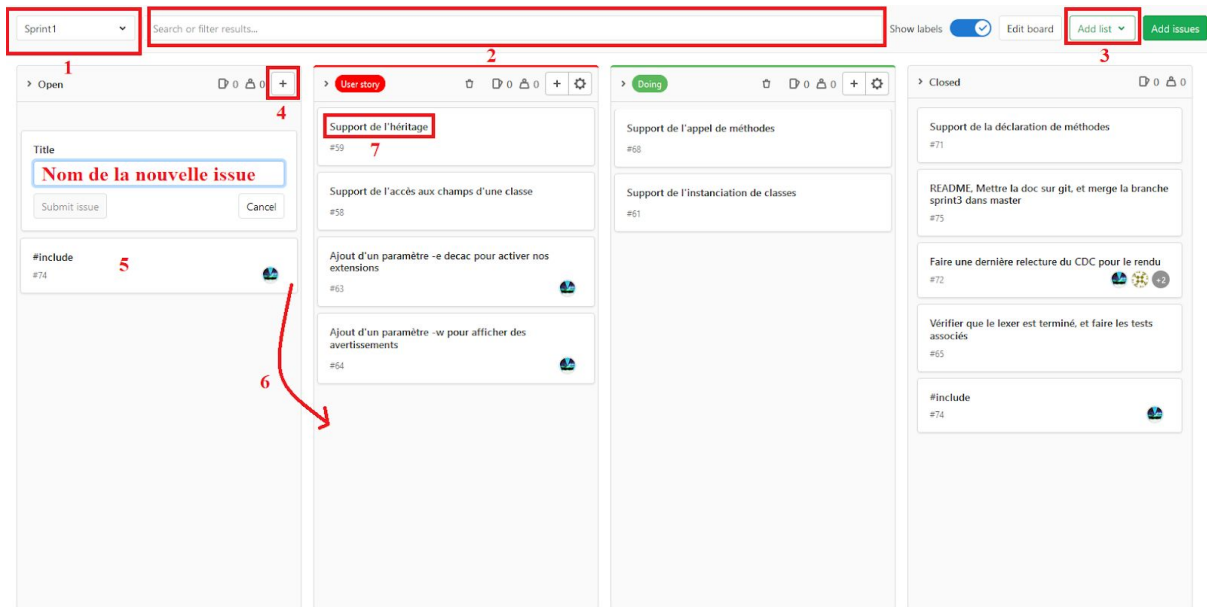


figure 2 - Aperçu du board d'un sprint


1. Créée ou sélectionne un board
2. Filtre pour trouver les issues correspondant aux critères sélectionnés
3. Ajout d'une nouvelle colonne
4. Ajout d'une nouvelle issue
5. Aperçu rapide et édition d'une issue (voir figure 3)
6. Glisser et déposer
7. Accès à l'issue (voir figure 4)

#include
#74

×

Assignee

Edit



⇒ Associer la personne qui s'occupe de cette issue

Epic

Edit

None

Milestone

Edit

Sprint 3

⇒ Associer l'issue au nom du Burndown chart correspondant

Time tracking

?

No estimate or time spent

Due date

Edit

None

Labels

Edit

None

Weight

Edit

None

⇒ Associer un poids à cette issue selon son importance. Par exemple, est-elle longue ou courte

Notifications

☒

figure 3 - Aperçu rapide d'une issue d'un board

Afficher un String Litteral



En tant qu'utilisateur du compilateur, **je souhaite** pouvoir afficher des chaînes de caractère en les écrivant directement dans mon programme deca. **Afin de** transmettre des informations à l'utilisateur de mon programme.

1

Linked issues ⓘ 4 +

Is blocked by

Lexen pour lire un programme hello world avec des commentaires #2	🕒 Sprint 1 🧑🏻	×
Parser_hello_world #5	🕒 Sprint 1 🧑🏻	×
Reproduire_video_intro #4	🕒 Sprint 1 🧑🏻	×
Generate IMA bytecode for #6 #9	🕒 Sprint 1 🧑🏻	×

2

figure 4 - Description et dépendances d'une issue

1. Ecrire la user story
2. Mettre des dépendances entre les issues. Ci-dessus, l'issue ne peut pas être close tant que les quatres autres n'ont pas été pas terminées.

Milestone

Milestone aussi appelé jalon est un moyen de suivre la progression d'un sprint via la construction d'un burndown chart.

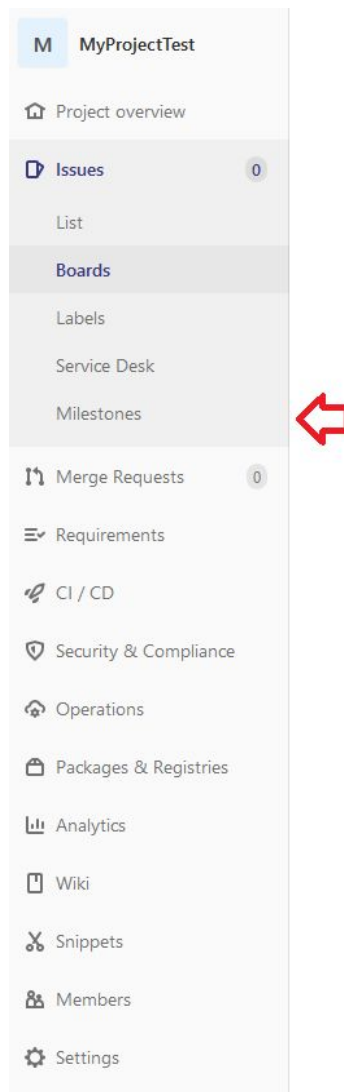


figure 5 - Menu actif Milestones

New Milestone

Title

Start Date [Clear start date](#)

Due Date [Clear due date](#)

Description

1

Write milestone description...

Markdown is supported [Attach a file](#)

2

Create milestone [Cancel](#)

figure 6 - Création d'un milestone

1. Mettre un nom identique au board correspondant par souci de clarté
2. Choisir les dates de début et de fin du sprint

Burndown Chart

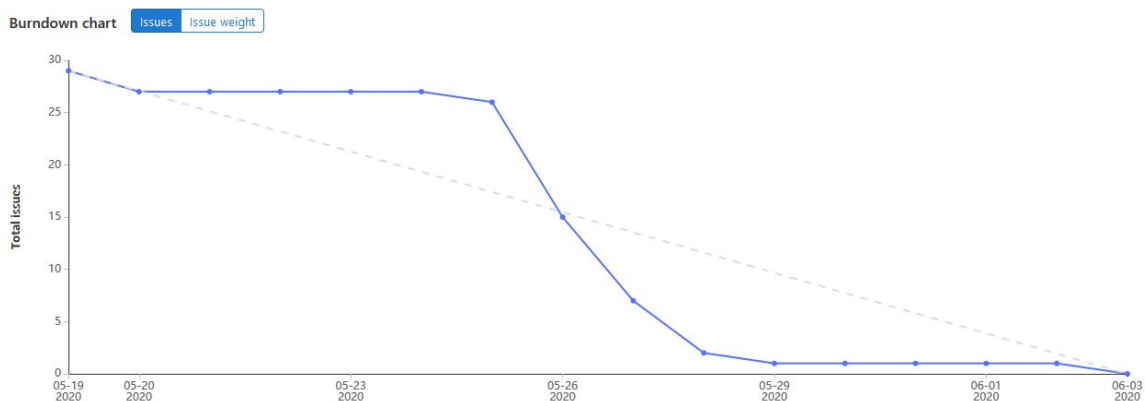


figure 7 - Burndown chart d'un sprint

Le burndown chart représente l'évolution de la charge de travail restante (ici d'un sprint) en se basant sur les issues associées.

Un point au dessus du niveau de la droite veut dire qu'il existe du retard. A l'inverse, un point en dessous du niveau de la droite veut dire qu'il existe de l'avance.

La droite représente une progression idéale où la quantité de travail effectué par jour est constante.

NB:

- le burndown chart de Gitlab ne prend pas en compte le temps passé sur une issue.

- Si le découpage / la granularité de vos issues est très fine, il est négligeable de donner un poids aux issues.
- Il est tout de même intéressant d'essayer de donner un poids aux issues et également d'annoter le temps passé sur celle ci afin de s'entraîner à la prévision.