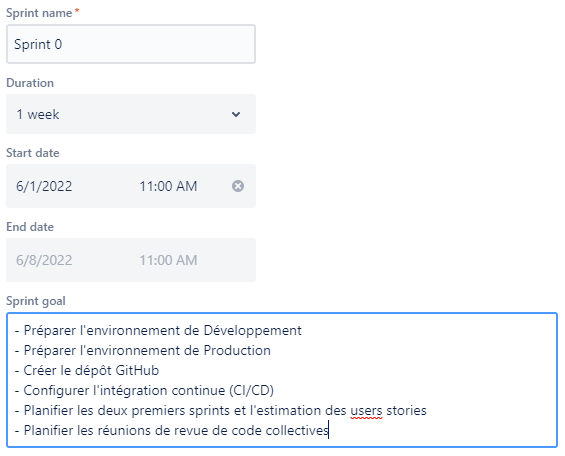
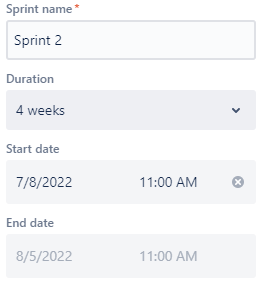
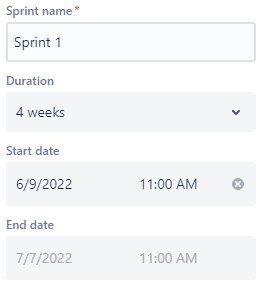
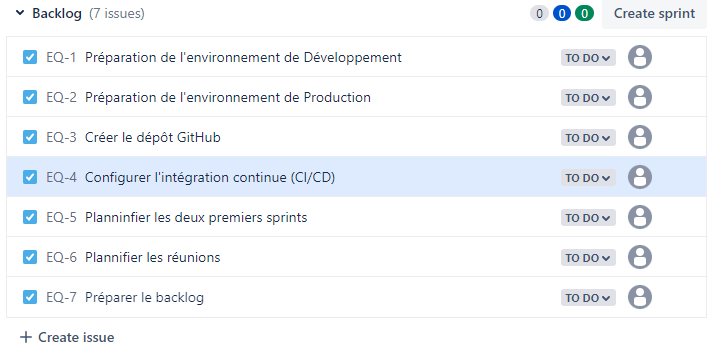
Planifier le sprint zéro qui dure ici une semaine :



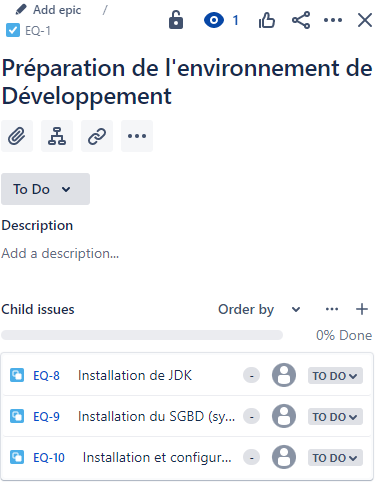
On décide de partir sur des itérations de quatre semaines. On planifie maintenant les deux prochains sprints et pas plus car il peut y avoir des changements entre temps.



On ajoute ensuite des issues (tâches) dans le backlog. (≠ users stories)

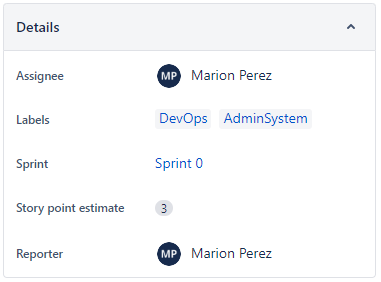


On va ensuite ajouter des sous-tâches au issues que l’on a déjà créées.



Ensuite il faut ajouter des étiquettes aux tâches pour savoir qui doit les réaliser, ainsi que leur assigner un sprint durant lequel elle doivent être faites.

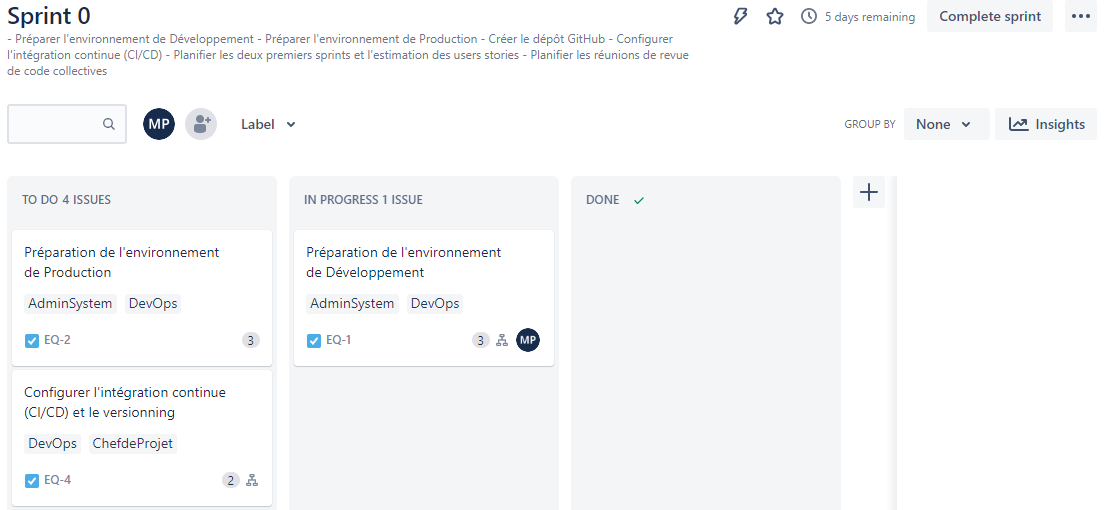
Enfin on doit estimer le temps que va prendre cette tâche à être réalisée en nombre de jours. Admettons qu’on soit dans une équipe de 5 développeurs. Le sprint durant 4 semaines cela fait 25 jours travaillés soit 5 x 25 = 125 jours disponibles pour réaliser les tâches du sprint.



Cela nous permet de savoir combien de tâches (sous-tâches) sont à faire dans le sprint, combien sont en cours de réalisation et combien ont été faites.



On démarre ensuite de sprint ce qui ouvre une nouvelle interface avec les différentes tâches. Il y a une configuration par défaut avec « à faire/en cours/terminée » mais on peut aussi rajouter « en attente de validation ».



Le jour de fin de sprint on s’assure que tout soit fini, et même si tout n’est pas fini, le chef de projet va forcément mettre fin au sprint. Au cours d’une réunion on revient sur le sprint pour savoir pourquoi on n’a pas pu terminer si c’est le cas. On présentera au client ce qui est terminé et les tâches non terminées seront ajoutés en début du prochain sprint. Il n’y a que pour la livraison que l’on ne peut pas se permettre de ne pas avoir terminé.