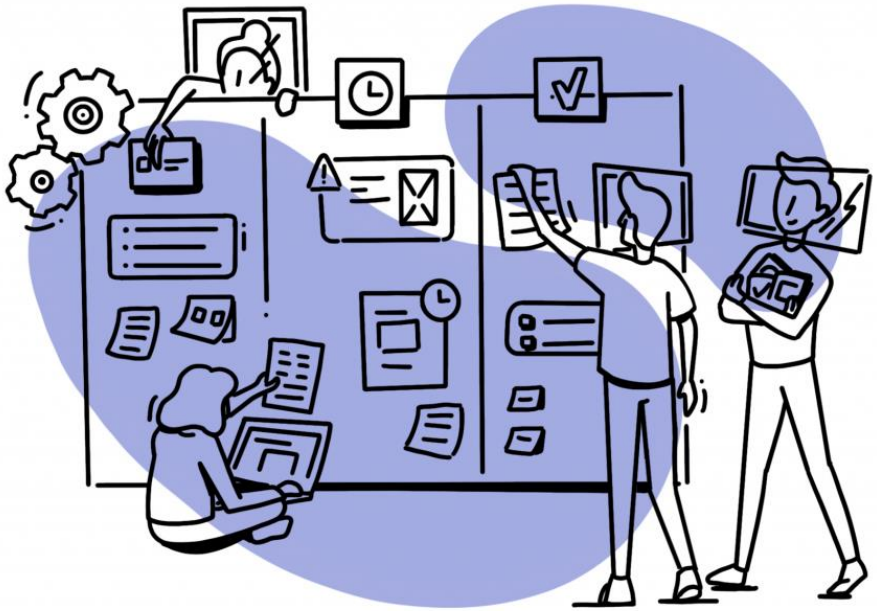


Projet GED'IMAGINATION
Approche agile / Security By Design

BTS SIO2 SLAM

SCRUM = Cadre de gestion de projet agile

Rôle de « coach » : il vérifie que le cadre *Scrum* est appliqué, il veille à la productivité de l'équipe, …

Expert du domaine métier, il définit le besoin et le priorise En interaction avec l'équipe de développement, il assure la bonne compréhension des exigences

Product Owner

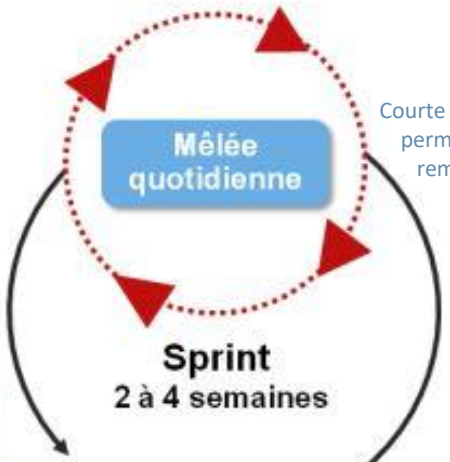
Auto-organisée, multi-compétences, elle est chargée de transformer le besoin exprimé par le PO en fonctionnalités utilisables

Equipe

Product Backlog

Liste priorisée des besoins et exigences attendues (user stories / récit utilisateur)
pouvant évoluer au cours du projet
éléments ajoutés ou supprimés,
éléments décomposés, priorités ajustées…)

En accord avec le PO, l'équipe sélectionne les éléments prioritaires qui seront développés et planifie le sprint



Courte réunions quotidiennes (« stand up meeting ») permettant aux membres de coordonner leurs activités, remonter les difficultés, vérifier l'avancement du sprint

Un sprint permet de développer les fonctionnalités de bout en bout (de la conception aux tests).

Lors de la revue de sprint, un produit partiel opérationnel peut-être montré au client afin de recueillir son feedback

Rétrospective sur le vécu du sprint écoulé dans le but de s'améliorer (processus de travail, organisation, productivité, efficacité, qualité…)

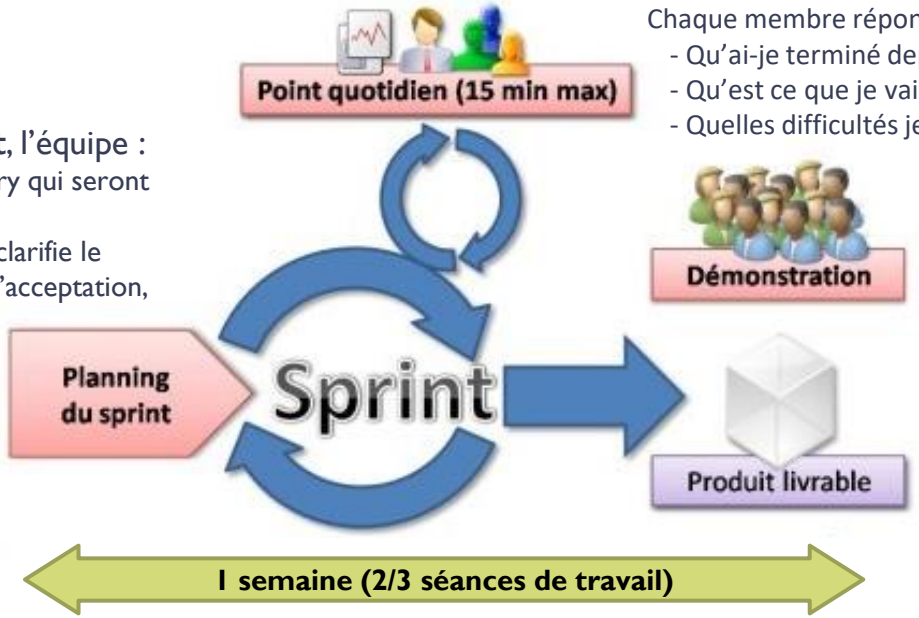
<< Processus de développement itératif et incrémental >>

Notre gestion de projet agile

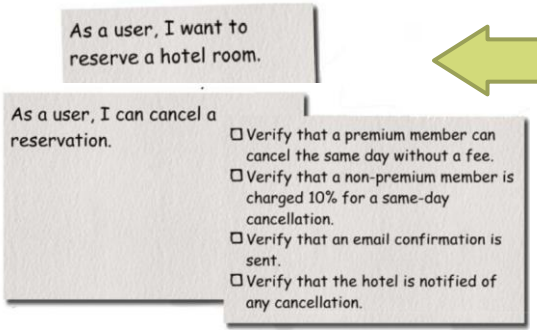
- Déroulement d'un sprint

Une mêlée au début de chaque séance
Chaque membre répond à 3 questions :
- Qu'ai-je terminé depuis la dernière mêlée ?
- Qu'est ce que je vais faire d'ici la prochaine mêlée ?
- Quelles difficultés je rencontre ou je pense rencontrer ?

Au début d'un sprint, l'équipe :
- sélectionne la/les user story qui seront développées
- affine chaque user story : clarifie le besoin, définit les critères d'acceptation, découpe en tâches
- réfléchit à la solution d'implémentation (conception fonctionnelle et technique)



Revue en fin de sprint
- Démo des fonctionnalités développées
- Feedback de l'équipe et du PO
- Améliorations possibles pour le prochain sprint



Approche *Security By Design*

- Evaluation des risques à partir des récits utilisateurs (*User Story*)

  Une **Evil User Story** (*Abuser Story*) décrit la réalisation d'un scénario de risque à travers l'identification d'une **source de risque** (attaquant externe, collaborateur malveillant), **exploitant une vulnérabilité**, occasionnant un **impact sur la valeur métier**

  Une **Security User Story** décrit les mesures de sécurité à mettre en place pour éviter / limiter le risque. Les *Security User Stories* peuvent venir alimenter le *Product Backlog*.

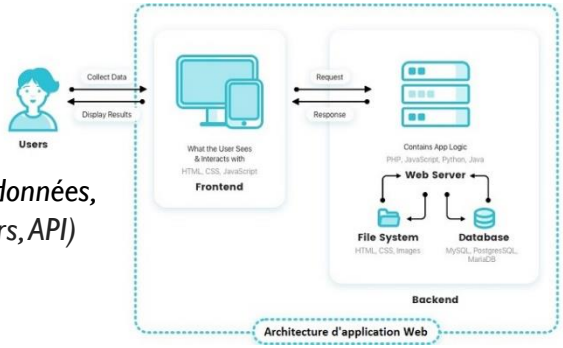
Exemple

Les **User Stories** sont les exigences logicielles **centrées sur la valeur** exprimées en **conversation** par les utilisateurs	*« En tant que **(rôle utilisateur)**, je veux **(activité)**, afin de **(valeur métier)** »*
	« En tant qu'**utilisateur**, je veux **renseigner mes identifiants**, afin **de me connecter à l'application**»
Les **Evil User Stories** mettent en évidence l'impact métier d'une activité malveillante ciblant le produit	*« En tant que **(utilisateur malveillant)**, je veux **(activité malveillante)**, afin de **(impact métier)** »*
	« En tant qu'**attaquant**, je veux **essayer de deviner le mot de passe d'un utilisateur** en envoyant de très nombreuses requêtes d'authentification en parallèle **pour me connecter à sa session** »
Les **Security User Stories** décrivent les mesures de sécurité à implémenter pour mitiger les risques.	*« En tant que **(rôle squad)**, je veux **(activité)**, afin de **(éviter qu'un evil user story se produise)** »*
	« En tant que **développeur**, je veux **mettre en place un mécanisme de blocage des comptes utilisateurs** après 5 tentatives **pour éviter les attaques par bruteforce**

Sprint 0

- Proposer un schéma d'architecture de la solution et analyser les risques

Architecture de la solution
Utilisateurs (acteurs),
périmètre fonctionnel, flux de données,
architecture technique (serveurs, API)
…



Analyse des risques
Identifier les menaces (extérieures et internes)
[où-suis-je vulnérable ? comment et par où peut-on m'attaquer ?]

- Préparer et configurer votre environnement de développement en appliquant les bonnes pratiques de sécurité

https://lincnil.github.io/Guide-RGPD-du-developpeur/

- Documenter

Livrable Sprint 0 : schéma d'architecture + environnement de développement (choix techniques / mesures de sécurité mises en place)