

Découpage du projet en sprint

Décembre 2023 – Version : 1.0

Antoine TASSEZ

Introduction

Ce document présente la découpe du projet en sprints, une approche méthodique visant à traduire les objectifs définis dans le cahier des charges en étapes de développement concrètes.

Chaque sprint représente une progression claire et mesurable, permettant à l'équipe de développement de se concentrer sur des aspects spécifiques du projet, tout en garantissant une compréhension approfondie des fonctionnalités et des exigences à chaque étape. Ce découpage en sprints propose une méthodologie efficace pour atteindre les objectifs globaux du projet de manière organisée et structurée.

Table des matières

Introduction.....	1
Prise en main du dépôt.....	1
Réduction des dettes techniques principales.....	2
Amélioration de l'architecture modulaire.....	3
Déploiement, QA, CD.....	3

Prise en main du dépôt.

Le sprint 0 sert à la lecture approfondie du projet et de son cahier des charges afin d'anticiper les axes de développement nécessaires à mener à bien le projet.

À la fin du sprint 0 :

- Le serveur, le linting, et les tests actuels sont tous fonctionnels.
 - La commande ``runserver`` doit s'exécuter sans erreur ni avertissement, et le site doit être accessible en local pour le développeur.
 - Le linting, après exécution, ne doit soulever aucune erreur.
 - La commande ``pytest`` doit s'exécuter sans erreurs ni avertissements.
- Les bonnes pratiques de développement sont tracées et définies.

- Un document, le « Guide de Développement », doit être accessible sur le dépôt. Il permet à tout nouvel arrivant sur le dépôt de se familiariser avec les pratiques de développement et de nommage.
- Le calendrier de développement est défini.
 - Les dates de rendu pour les sprints 0 et 1 sont définies, et les objectifs pour l'ensemble du projet sont mis à l'écrit.
- Le dépôt est prêt à être repris par n'importe quel collaborateur après lecture de son README.md.
 - Le document d'introduction au projet est écrit et permet à tout utilisateur de récupérer le code de la branche principale pour exécuter le site en local et de s'orienter dans le dépôt.
- Le projet est assimilé.
 - Le développeur ou l'équipe de développement a acquis une compréhension approfondie des objectifs, des fonctionnalités, de l'architecture du code, et des pratiques de développement du projet. Cette assimilation garantit une contribution efficace et informée à toutes les phases du développement.

Réduction des dettes techniques principales.

Le sprint 1 sert à installer de solides bases pour le dépôt tout en réduisant les dettes techniques dont il souffre actuellement.

À la fin du sprint 1 :

- La quantité de tests a sensiblement augmenté.
 - Chaque élément du projet doit avoir une suite de tests appropriée. La couverture des tests sur le code doit atteindre au moins 85 % pour considérer cette étape comme accomplie.
- L'ensemble du code est correctement commenté et formaté.
 - L'ensemble des éléments du projet est correctement commenté, respectant les bonnes pratiques de format des docstrings et de la PEP 8.
- Les user stories sont définies et claires.
 - Les User Stories sont consignées sur papier, découpées par page, et les actions des personas sont définies.
- Une documentation du projet est fournie.
 - Une documentation technique du projet ainsi que les outils prévue pour les sprints suivants sont définie intégré à la documentation complète du projet.

Amélioration de l'architecture modulaire

Le sprint 2 a pour but d'améliorer l'architecture modulaire en séparant le monolithe actuel qu'est l'application.

À la fin du sprint 2 :

- Deux applications distinctes sont en place dans le projet.
 - Le monolithe est à présent constitué de deux application distinctes : oc_lettings_site → lettings / profiles
- La migration de la base est effective.
 - L'ensemble des données déjà présentes doit être transféré dans la nouvelle base. Aucune perte de données n'est autorisée.
- Les templates sont organisés de façon claire et cohérente.
 - Après la réorganisation des applications, les templates des pages doivent rester accessibles et organisés de façon logique avec la nouvelle architecture du projet.
- Les URLs et endpoints sont dissociés.
 - Après la réorganisation des applications, le dispatch des URLs doit rester cohérent et en accord avec les modifications effectuées.
- Les documentations d'introduction et techniques sont relues et mises à jour.
 - Fin du sprint, les différentes documentations et commentaires doivent rester fidèle à l'état actuel du projet.
Les commentaires étant normalement mis à jour par le développeur lorsqu'il travail sur le code, cette étape s'assure surtout de la cohérence des documentation techniques.

Déploiement, QA, CD

Le sprint 3 a pour but de mettre en place le site à proprement parler, finaliser le testing et mettre en place le déploiement automatique.

À la fin du sprint 3 :

- Le site est en ligne et consultable depuis son URL dédiée ;
 - Le déploiement du site doit être effectif, une documentation dédiée doit être présente sur le repo.
- Les push sur la branche principale déclenchent le testing et éventuellement le redéploiement du site ;

- La pipeline de test et de conteneurisation doit être correctement mise à jour pour entraîner une étape finale de déploiement.
- La documentation s'actualise automatiquement en fonction des push ;
- En plus des différentes documentations techniques, une documentation en ligne via « ReadTheDocs » est requise par le client. Les éventuelles modifications du code et de sa documentation doivent déclencher la mise à jour de celle en ligne pour s'assurer que celle-ci est toujours correcte et à jour.
- Le projet peut être repris par n'importe quel collaborateur après la lecture du README.md et des documentations fournies.
- Relecture des documentations pour s'assurer qu'elles sont à jour et en accord avec les pratiques de développement en cours. Les détails de mise en place et les paramètres des outils doivent être écrits, détaillés, et accessibles à tout collaborateur.