

Projet 13 : Projet Final

Les bonnes pratiques de l'extrem programming

Voici comment j'ai appliqué les bonnes pratiques de l'XP dans mon projet :

1 – Séance de planification

Dans mon idée initiale, le programme contenait trop de fonctionnalités pour être réalisé à temps dans de bonnes conditions, mon mentor m'a donc encouragé à déterminer la MVP (minimum viable product) de mon application. Les tâches prioritaires et secondaires étaient donc bien définies au lancement du projet.

2 – Métaphore

Vu la fonctionnalité principale de l'application (à savoir, gérer le contenu d'un congélateur), les métaphores et les analogies ont été inspirées du lexique de l'alimentaire et de la conservation (à tel point qu'on les a utilisées naturellement avant même d'y réfléchir). Par exemple, lier un produit au compartiment d'un congélateur est résumé par « congeler ».

3 – Convention de codage

Étant seul pour développer, je n'avais pas de convention particulière à respecter. J'ai donc comme d'habitude veillé à ce que mon code soit conforme à la PEP8 et aux bonnes pratiques en général.

4 – Conception simple

La structure de l'application a été pensée pour être appréhendée par n'importe quel développeur. Mon code reste simple et lisible. Mon erreur sur ce point a été de parfois chercher à développer des fonctionnalités secondaires avant de boucler les prioritaires.

5 – Programmation en binôme

Je considère que mon mentor a joué ce rôle de binôme en me remettant souvent sur les bons rails quand je commençais à vouloir tout coder en même temps ! Mais également via ses remarques sur la qualité de mon code, sur l'utilité de certaines fonctionnalités, etc.

6 – Propriété collective du code

Je connais bien évidemment mon code, quant à mon mentor, je l'informais chaque semaine de tout ce qui avait été ajouté dans le programme. L'application est quasiment mise à jour quotidiennement sur GitHub et visible par tous.

7 – Rythme soutenable

Le rythme a été correct durant tout le projet grâce à l'intervention de mon mentor au tout début, concernant la création d'une MVP. J'ai donc pu travailler sereinement sur les fonctionnalités importantes sans avoir à déborder sur mon temps personnel.

8 – Client sur site

Le client étant représenté ici par les personnes m'ayant suggéré de développer une telle application, leurs avis ont souvent été demandés afin de déterminer les fonctionnalités qui leur étaient utiles. La plupart d'entre eux ont également testé les premières versions de l'application sur serveur test et m'ont fait des feedback sur le sujet.

9 – Livraisons fréquentes

Comme il est possible de le constater sur le repo GitHub du projet, je fais beaucoup de push. Selon leurs simplicités, plusieurs fonctionnalités peuvent être incluses dans un seul push mais la majeure partie du temps, une modification équivaut à une version différente.

10 – Tests

Le développement des tests est l'une des parties que j'apprécie le plus. Je commence toujours par établir la liste de tous les éléments à tester puis, lorsque les éléments nécessaires aux tests et à la future fonctionnalité (par exemple, les models de Django) sont en place, je code le test puis passe à la fonctionnalité.

Bien entendu, il m'arrive d'avoir des idées de test après que la fonctionnalité ait été développé. De la même façon, mes tests de parcours utilisateur sont développés une fois le template terminé.

Je fais toujours en sorte d'avoir un large éventail de test afin de cibler au mieux les éventuels problèmes. Ceux-ci m'ont d'ailleurs souvent été très utile lors du développement de l'application.

11 – Remaniement continu

Dès que je trouvais une méthode qui permettait d'optimiser une partie de mon code, je recherchais toutes les parties où cette méthode était applicable pour les remanier.

12 – Intégration continue

L'intégration continue a été mise en place très rapidement après le lancement du projet (juste après la création des serveurs test et production). Recevoir la notification m'indiquant que les tests avaient échoués lors du push me faisait réagir immédiatement et le problème était corrigé rapidement.