

TD2 : Conduite de projets Besoins et tâches

Compétences

- Transformer un besoin en exigence
- Transformer une exigence en une ou plusieurs tâches
- Implémenter les tâches

Propriétaire — Développeur

Vous trouverez ici une vidéo d'introduction à la conduite de projet faite par Xavier Blanc, Professeur en Informatique à l'université de Bordeaux.

CYCLE DE VIE — CYCLE DE DÉVELOPPEMENT

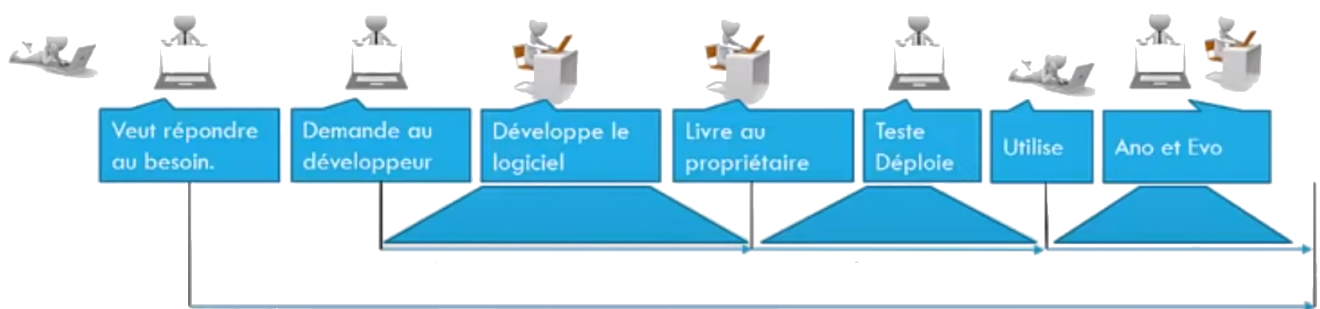


FIGURE 1 – Extrait de la vidéo "Conduite de projet" de Xavier Blanc.

1. À partir des documents fournis, indiquez sur l'image ci-dessus les grandes étapes d'un projet logiciel, à savoir :
 - cycle de vie,
 - cycle de développement,
 - pré-production,
 - production et maintenance.
2. Une première étape à franchir pour entrer dans le cycle de développement est celle de la transformation des **besoins** (exprimés par le propriétaire dans le cahier des charges) en **exigences** par le développeur. C'est une étape importante car elle engage le développeur sur ce qu'il est capable de faire.

À partir du cahier des charges et en vous inspirant de l'exemple fourni dans *besoin.md*, rédigez les exigences correspondantes aux besoins suivants :

- enregistrer les données acquises lors du pointage dans un fichier au format csv,
- mise en place d'une échelle par l'utilisateur.



FIGURE 2 – Cycle de développement en cascade.

Cycle de développement

De la conception aux tâches

Objectif : Décomposer le logiciel en gros modules, préciser les tâches permettant la construction de ces modules. L'identification des tâches permet :

- de mesurer le travail à réaliser,
 - d'organiser le travail dans une équipe,
 - de maîtriser les délais et les risques.
1. À partir du document `tache.md`, écrivez les tâches correspondant aux exigences de la première partie.
 2. Faites ensuite un tableau dans lequel vous indiquerez :
 - le numéro de l'exigence,
 - un numéro pour chacune des tâches ainsi que sa description,
 - le nom du responsable de cette tâche,
 - le temps estimé pour réaliser cette tâche,
 - son état d'avancement.
 3. Dans un premier temps, nous ne nous occuperons pas de la partie IHM, c'est-à-dire de l'interface graphique qui permet à l'utilisateur d'interagir avec le logiciel. Nous allons commencer le code de notre application en définissant les classes : `Point` et `FileRepo`. Associez à chacune des classes un fichier python rangé correctement dans une arborescence correspondante au *Pattern* : *MVC*.
 4. Implémentez chaque classe en vous appuyant sur les tâches à réaliser.
 5. Dans un fichier `test.py`, écrivez une méthode `randomPoints(n:int)->list` qui prend en argument un entier `n` et qui renvoie un tableau contenant `n` points tirés aléatoirement.
 6. À l'aide de ce tableau de Points, testez la/les méthode(s) de votre classe `FileRepo`.

Renforcer ses compétences

Les fichiers : https://book.pythontips.com/en/latest/open_function.html?highlight=write%20funcion

Les exceptions : <https://book.pythontips.com/en/latest/exceptions.html>