



CESI RILDEVOPS 2019-2020

Cycle de dev. d'un logiciel

PRODUCTION DE LOGICIEL

Qu'est-ce qu'un logiciel ?

Application (Store), SiteWeb, Jeux,
Micro-Contrôleur, etc.

Produire => Code Source, Exécutable



Participants (rôles)

L'utilisateur

Le développeur

Le propriétaire



PROJET LOGICIEL - COMMENCEMENT

1. L'**utilisateur** a besoin d'un logiciel
2. Le **propriétaire** demande au développeur de réaliser le logiciel
3. Le **développeur** réalise le logiciel qui correspond au besoin de l'utilisateur



PROJET LOGICIEL – EXPLOITATION

4. Le **propriétaire** déploie le logiciel afin que celui-ci soit utilisé.
5. L'**utilisateur** utilise le logiciel, il rencontre des anomalies ou aimerait voir des évolutions
6. Le **propriétaire** demande au développeur de corriger les anomalies ou de développer les évolutions
7. Le **développeur** corrige les anomalies et réalise les évolutions

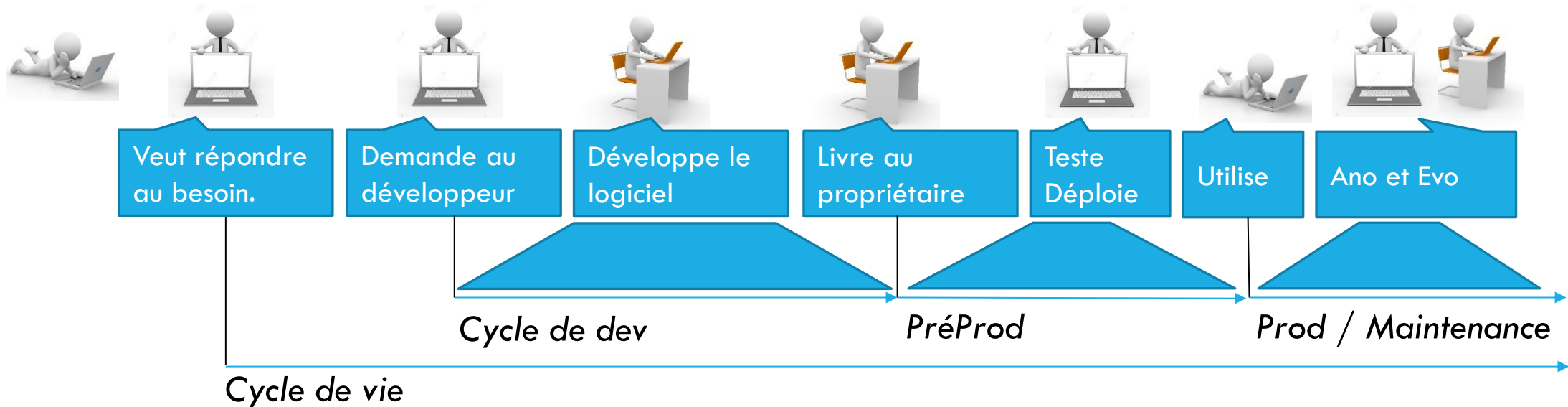


PROJET LOGICIEL — FIN

- 8. L'**utilisateur** n'utilise plus le logiciel,
- 9. Ou le **propriétaire** arrête le logiciel
- 10. le **développeur** ne sert plus à rien,



CYCLE DE VIE – CYCLE DE DÉVELOPPEMENT



DIFFÉRENTS TYPES DE PROJETS

Projet de mise en production

- Objectif : développer et assurer la mise en production
- Démarre dès l'expression du besoin
- Termine lorsque l'utilisateur peut utiliser le logiciel

Projet de développement

- Objectif : livrer la première version utilisable par l'utilisateur
- Démarre à partir de l'expression du besoin par le propriétaire
- Termine lors de la livraison par le développeur

TMA (Pas vraiment un projet)

- Objectif : assurer la maintenance d'une application mise en production
- Démarre lorsque l'utilisateur peut utiliser le logiciel
- Termine lorsque le logiciel n'évolue plus



PROJET DE DÉVELOPPEMENT (ET TMA – EVO)

L'utilisateur a un besoin (même s'il ne le sait pas)

Le propriétaire **exprime** clairement ce besoin

- **Cahier des charges**

Le développeur **spécifie** ce qu'il est capable de produire

- **Réponse au cahier des charges (Spec)**

Le propriétaire accepte la proposition du développeur

- **Cycle de dev**

Une fois terminé, le développeur livre le logiciel au propriétaire

- **Livraison**



PARTICIPANTS AU PROJET DE DÉVELOPPEMENT



MOA (Maîtrise d'Ouvrage) : Celui qui possède le résultat du projet, et qui paye
=> Le propriétaire

MOE (Maîtrise d'Œuvre) : Celui qui réalise le logiciel => Le développeur

L'Utilisateur n'apparaît pas réellement dans le projet (il est représenté par la MOA)



Client, un terme trop ambiguë

- *Le client du propriétaire est l'utilisateur*
- *Le client du développeur est le propriétaire*



PROJET DE DÉVELOPPEMENT : EXEMPLE

Les enseignants et les chercheurs ont besoin d'un site web pour organiser la fête de la science

- Utilisateurs = Enseignants et Chercheurs

Le CNRS veut répondre à ce besoin en demandant à plusieurs sociétés de développer ce site web

- Propriétaire = le CNRS

La société JeanB réalise le site web qui est utilisé par les enseignants et les chercheurs

- Développeur = JeanB



LES 3 GRANDES PHASES DU CYCLE DE DEV

Analyse (Analysis)

- La MOE spécifie ce qu'elle est capable de développer et chiffre le coût

Conception (Design)

- La MOE organise son développement, c'est-à-dire précise les gros modules à développer et précise les tâches

Développement (Dev)

- La MOE développe les composants définis dans la conception et *in fine* livre le résultat



Analysis



Design



Dev



L'ANALYSE

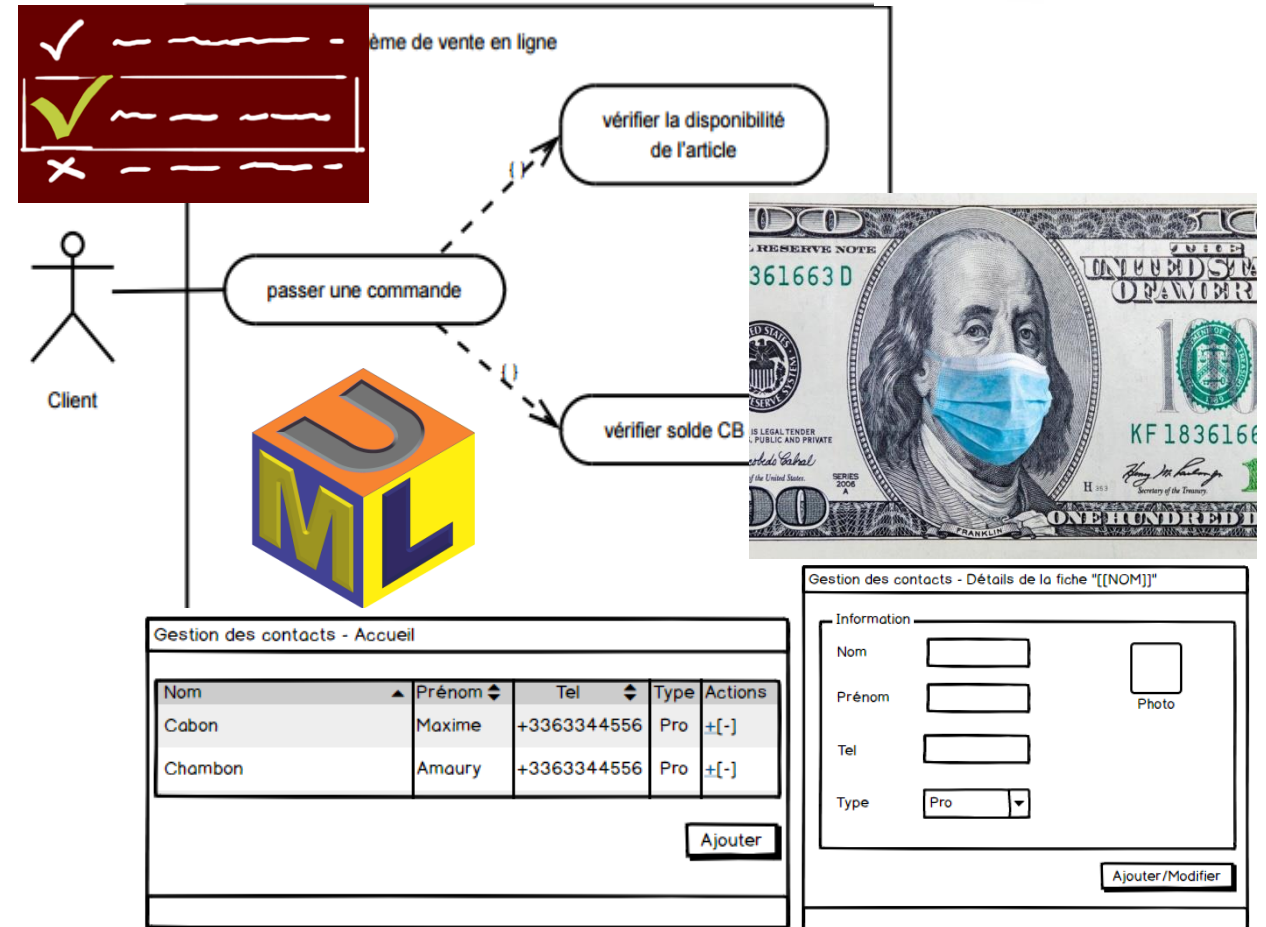
Qui : La MOE

Entrée : Le cahier des charges

Objectif : spécifier ce que la MOE est capable de fournir

Outils : UML, Balsamiq Mockup, Draw.io, Power Point, ...

Sortie : Une spécification de ce qui sera développé





LE CHIFFRAGE

Le chiffrage se fait en jours-homme dans les projets de développement

- 1 jh = 1 homme qui travaille pendant un jours
- 20 jh = 1 mh (mois-homme)
- 12 mh = 1 ah (an-homme)

Idéalement, l'analyse devrait fournir les éléments permettant de chiffrer le coût

Pour autant, trop souvent le chiffrage est réalisé avant l'analyse!

NB : Chiffrer un projet nécessite de l'expérience !!!

Quelques éléments:

- Développer un écran d'un site web par exemple
 - 1,5 à 3 jours-homme pour les écrans simples
- Nombre de ligne de code développé en moyenne par développeur = 150
 - Une classe d'une complexité moyenne nécessite donc au moins un jours

Taille des projets :

- Petit : max 2 mh
- Moyen : entre 2 mh et 1,5 ah
- Gros : plus de 1,5 ah

LA CONCEPTION

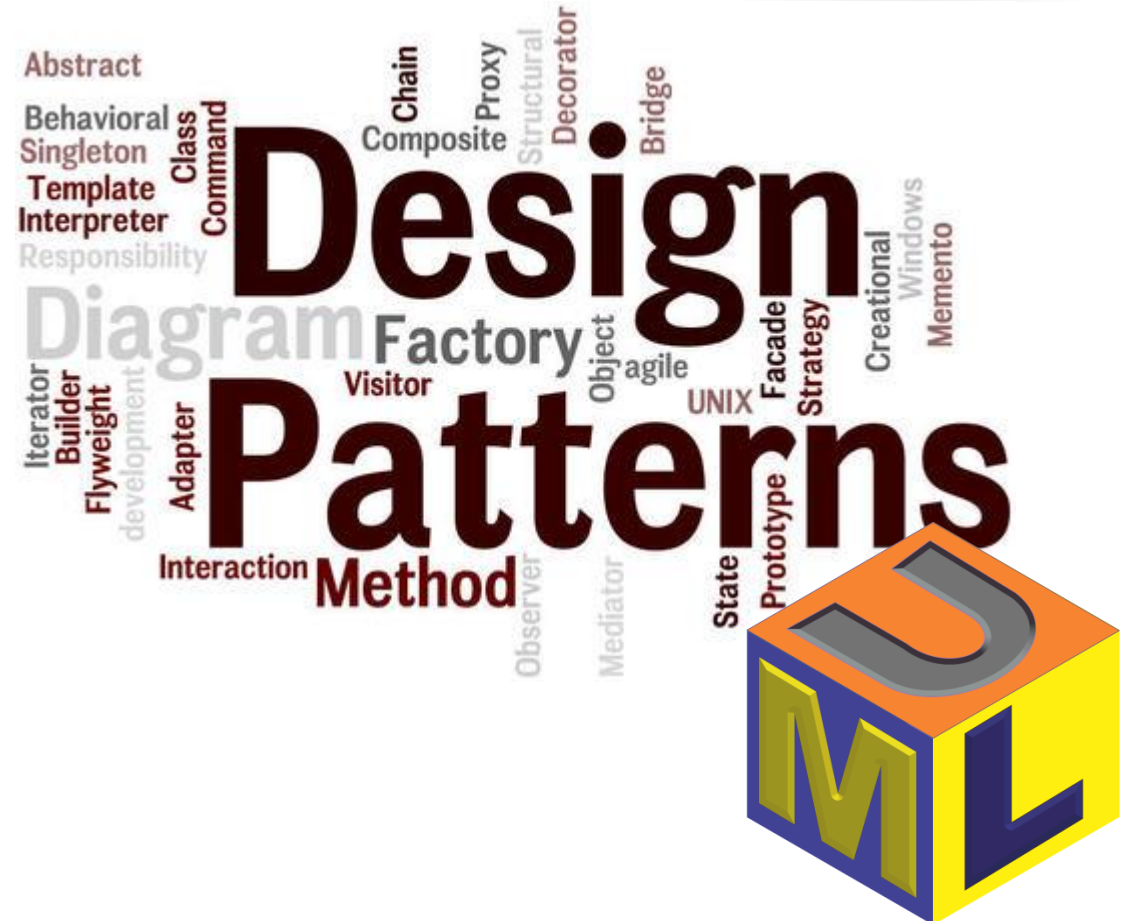
Qui : La MOE

Entrée : La spécification réalisée lors de l'analyse

Objectif : décomposer le logiciel en gros modules, préciser les tâches permettant la construction de ces modules

Outils : UML, les « Patrons de conception », Trello, Kaban, ...

Sortie : Une spécification des composants, de leur manière de communiquer, une liste des tâches.



LA MODÉLISATION



- Mieux comprendre le fonctionnement du système à développer
- Moyen de réduire sa complexité et d'assurer sa cohérence
- Réduire les coûts et les délais de son développement
- Assurer un bon niveau de qualité et de maintenance

Méthodes de modélisation

- Par l'approche fonctionnelle (ou algorithmique)
- Par la perspective orientée objet
- ...

L'ARCHITECTURE



Le choix d'une architecture facilite la mise en place d'une conception

- 3 tiers, En couche, Par Service, Par entrepôt de données, etc.

La définition des modules et de leurs interactions

- Interface, protocole, etc.

La projection d'une architecture sur un socle logiciel permet l'identification des tâches

L'identification des tâches permet de mesurer le travail à réaliser

- (RAF : Reste A Faire)

Cela permet aussi d'organiser le travail dans une équipe

Et de maîtriser les délais et les risques.

LE MODÈLE 3-TIERS



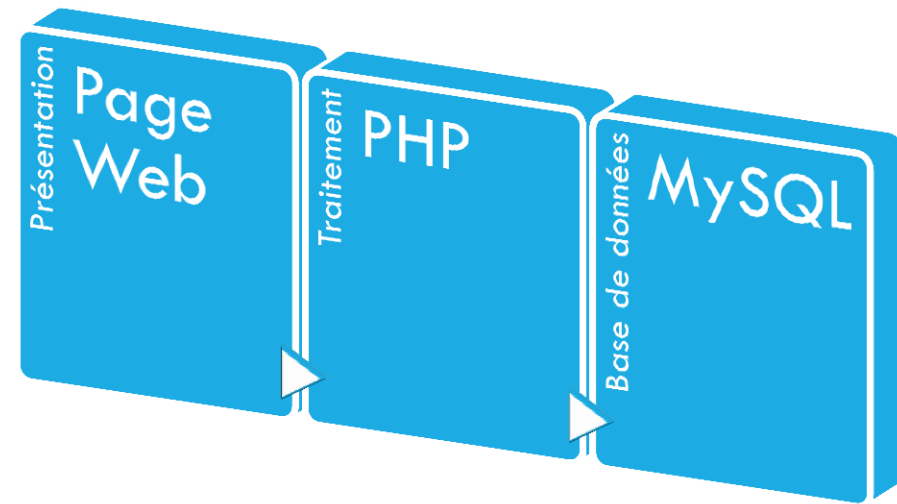
Trois modules en couche

- Base de données
- Traitement
- Présentation

Focus mis sur la protection des données,
et pas sur les performances

Tâches

- Module : BD, Traitement, Présentation
- Interfaces :
 - Traitement <-> BD,
 - Présentation <-> Traitement



LE DÉVELOPPEMENT

Qui : La MOE

Entrée : La définition des modules et la liste des tâches

Objectif : coder les modules et livrer le logiciel

Outils : Langages, outils de versionning et de collaboration, infrastructure matérielle, ...

Sortie : Le logiciel



L'ENVIRONNEMENT DE DÉVELOPPEMENT



Le logiciel est composé d'artefacts logiciels (Fichiers)

Il faut savoir qui peut les éditer, les modifier, les tester, etc.

L'environnement de développement définit ce cadre

Edition Collaborative

- Les artefacts sont-ils partagés en écriture?
- Quelles sont les versions? Comment les retrouver?

Répartition des tâches

- L'affectation des tâches est-elle faite par une personne ou par l'équipe?
- Comment suivre l'avancement?

LE CYCLE EN CASCADE



Analyse



Conception



Développement

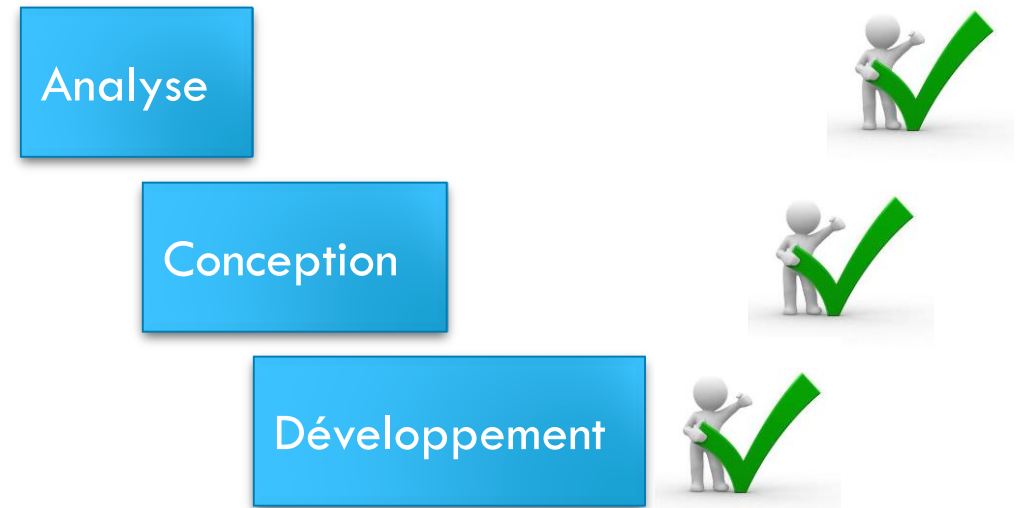
LES TESTS : CYCLE EN V

Tester le logiciel pendant tout le cycle de développement

Selon les phases:

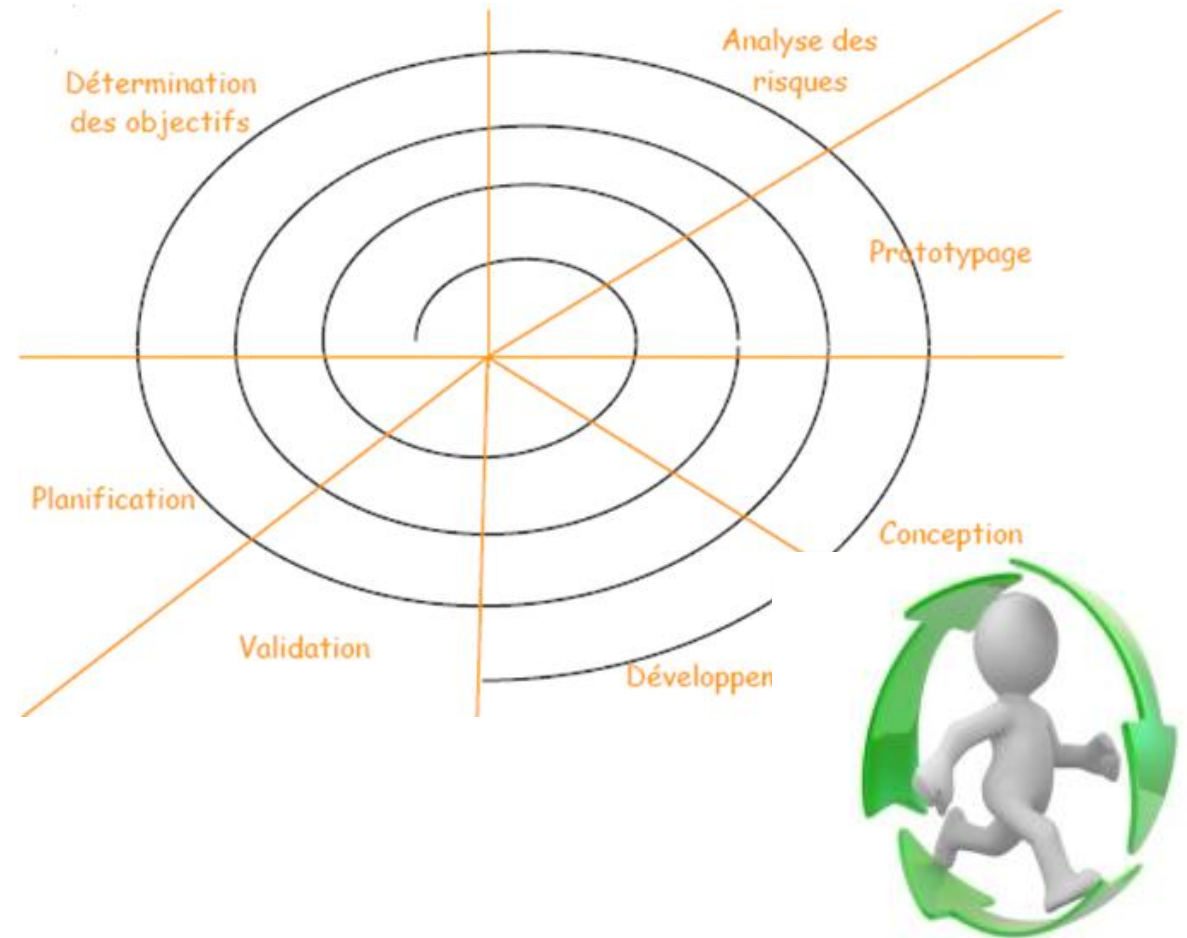
- Analyse : Test de validation (E2E)
- Conception : Test d'intégration
- Dev : Test Unitaire

Rédaction des tests != faire passer les tests



L'ITÉRATION : CYCLE EN SPIRALE

- Les phases Analyse, Conception, Codage doivent être réalisées les unes après les autres (waterfall)
- Pour autant, on peut aussi faire des itérations ne prenant en compte qu'une partie du cahier des charges
- Les nouvelles itérations nécessiteront des aménagements sur ce qui a déjà été réalisé



L'AGILITÉ

- Le cahier des charges contient les besoins exprimés par le propriétaire pour l'utilisateur
- Si ces besoins changent au cours du projet, l'impact peut être important pour le logiciel
- L'agilité considère comme hypothèse de base que les besoins peuvent changer et qu'il faudra y faire face!
- Le propriétaire est responsable des changements
- Les développeurs travaillent pour réaliser les évolutions du besoin



SYNTHÈSE

Cycle de vie

- cycle de dev,
- pré-prod,
- prod (maintenance)

Propriétaire, Développeur, Utilisateur

Cycle de dev

- Analyse, Conception, Dev

Cascade (waterfall)

Test (Cycle en V)

Itération (Cycle en spiral)

Agilité

