

Développement de Logiciel

Par : Abderrazak Sahraoui

Sommaire

- Génie Logiciel
- Cycle de vie de logiciel
- Cycle de développement de logiciel
 - Cascade
 - V
 - Itératif et Agile
- Processus unifié
 - Modélisation UML
- Scrum

Génie Logiciel

- Génie Logiciel (anglais software engineering)

Le terme « génie » désigne l'art de l'ingénieur

Le terme génie logiciel désigne l'ensemble **des procédures et des activités de conception et de mise en œuvre** tendant à rationaliser la production du logiciel et son suivi.

Le génie logiciel est l'activité par laquelle le code source d'un logiciel est spécifié puis produit.

Le génie logiciel est né suite à la crise du Logiciel dans les années 1960 et toutes les difficultés observées quant-au respect des coûts, des délais, du cahier des charges et du niveau de qualité.

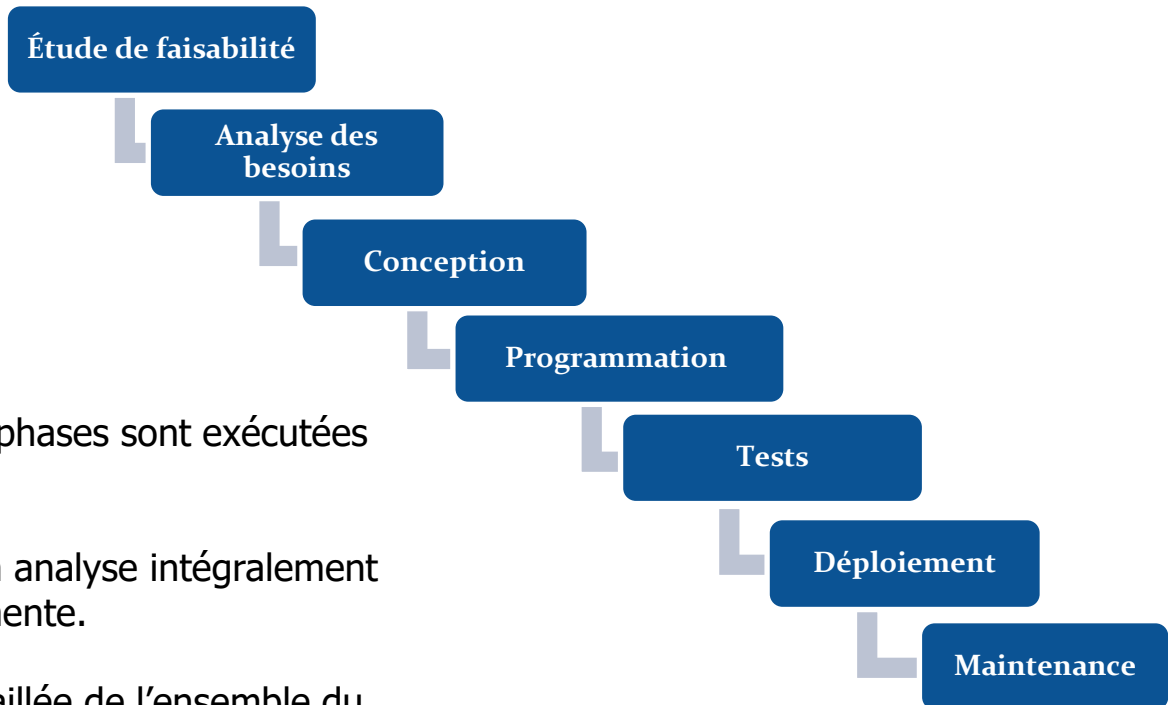
« Le génie logiciel touche au cycle de vie des logiciels. Toutes les phases de la création d'un logiciel informatique y sont enseignées : l'analyse du besoin, l'élaboration des spécifications, la conceptualisation du mécanisme interne au logiciel ainsi que les techniques de programmation, le développement, la phase de test et finalement la maintenance. »

[Source Wikipedia](#)

Cycle de vie de Logiciel

- La construction d'un logiciel comporte généralement différentes activités telles que **l'étude de faisabilité, l'analyse des besoins, la conception, la programmation, les tests, le déploiement et la maintenance**
- **L'étude de faisabilité** permet de déterminer si le logiciel peut être réalisé : s'il est possible d'apporter une solution technique au problème posé, en tenant compte du système informatique à disposition.
- **L'analyse des besoins** permet de produire la spécification fonctionnelle qui servira de référence pour la conception et la programmation.
- **La conception** consiste à choisir les technologies et les outils qui devront être utilisés
- **La programmation** consiste à créer des programmes exécutables en se servant des langages de programmation.
- **Les tests** consistent à simuler des scénarios d'utilisation en vue de vérifier le fonctionnement correct du programme.
- **Le déploiement** consiste à installer le logiciel et ses équipements dans un environnement de production accessible aux usagers.
- **La maintenance** du logiciel englobe les travaux de modification effectués *a posteriori*, après la distribution du logiciel. Elle désigne les modifications apportées à un logiciel, après sa mise en œuvre, pour en corriger les fautes, en améliorer l'efficacité ou autres caractéristiques, ou encore adapter celui-ci à un environnement modifié

Cycle de développement en cascade



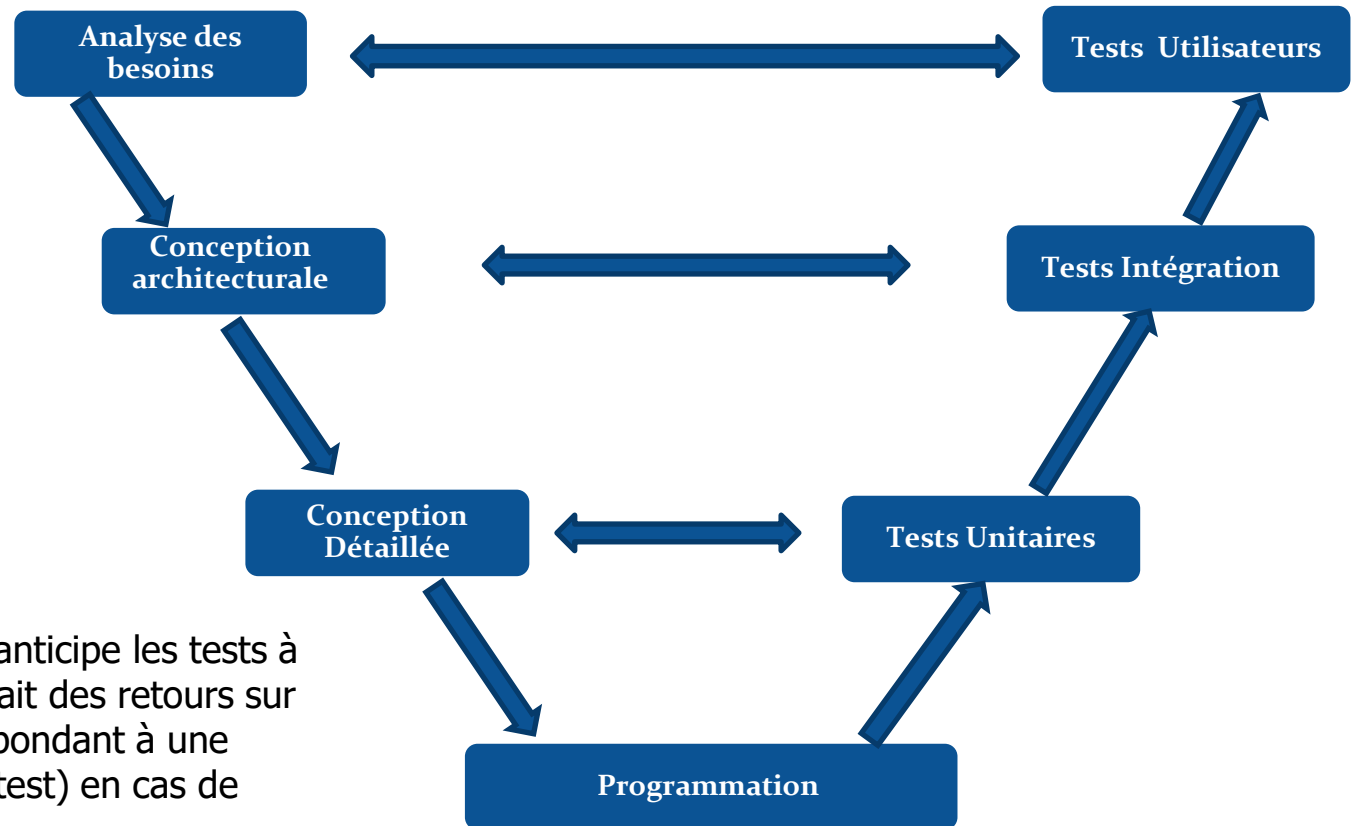
Dans ce modèle, les différentes phases sont exécutées séquentiellement

Après l'étude de la faisabilité, on analyse intégralement tous les besoins et on les documente.

Puis, on fait une conception détaillée de l'ensemble du système et on la documente.

Ensuite, on produit le code de tout le système, on valide et on déploie.

Cycle de développement en V



Dans ce modèle, on anticipe les tests à chaque étape et on fait des retours sur chaque étape correspondant à une étape de validation (test) en cas de nécessité.

De cascade vers l'itératif et agile

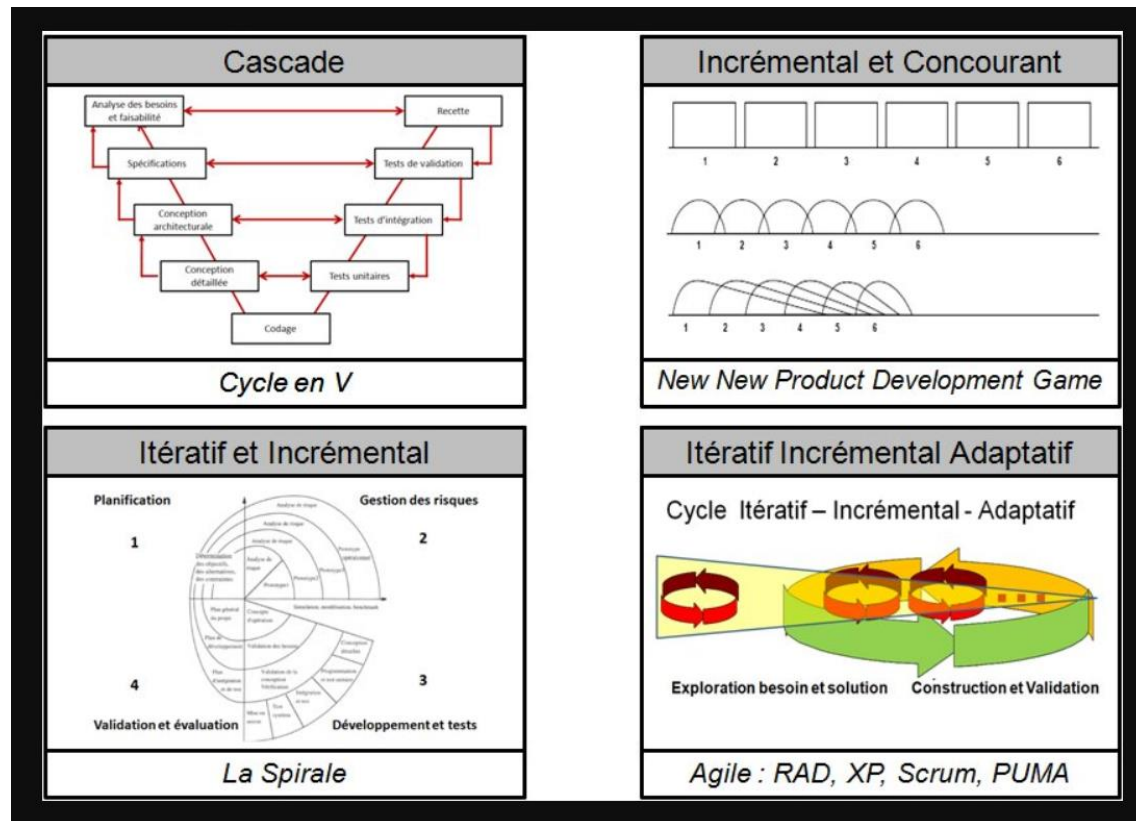
- On reproche au modèle en cascade
 - Sa rigidité vis-à-vis des changements
 - Le long délai avant de voir le produit fini
 - Très faible implication des clients lors des phases de conceptions et de réalisation
- En pratique, il est difficile voire impossible de totalement détacher la phase de conception d'un projet de sa phase de réalisation. On se rend compte souvent au cours de la mise en œuvre que les spécifications initiales étaient incomplètes, fausses, ou irréalisables, sans compter les ajouts de nouvelles fonctionnalités par les clients.
- Un compromis consiste à appliquer un cycle en V le plus court possible et à faire évoluer le projet sous forme de versions, prenant ainsi en compte le fait que le projet ne sera pas parfait et qu'il sera amélioré au fil des versions. Cela permet également de bénéficier du retour d'expérience des versions précédentes. Ce compromis est particulièrement formateur au sein d'un projet

[Source Wikipédia](#)

De cascade vers l'itératif et agile

Le développement d'un produit logiciel peut être découpé en plusieurs parties qui sont autant de mini-projets. Chacun d'eux représente une itération qui donne lieu à un incrément.

Les itérations désignent des étapes pour les activités, tandis que les incréments correspondent à des stades de développement du produit.



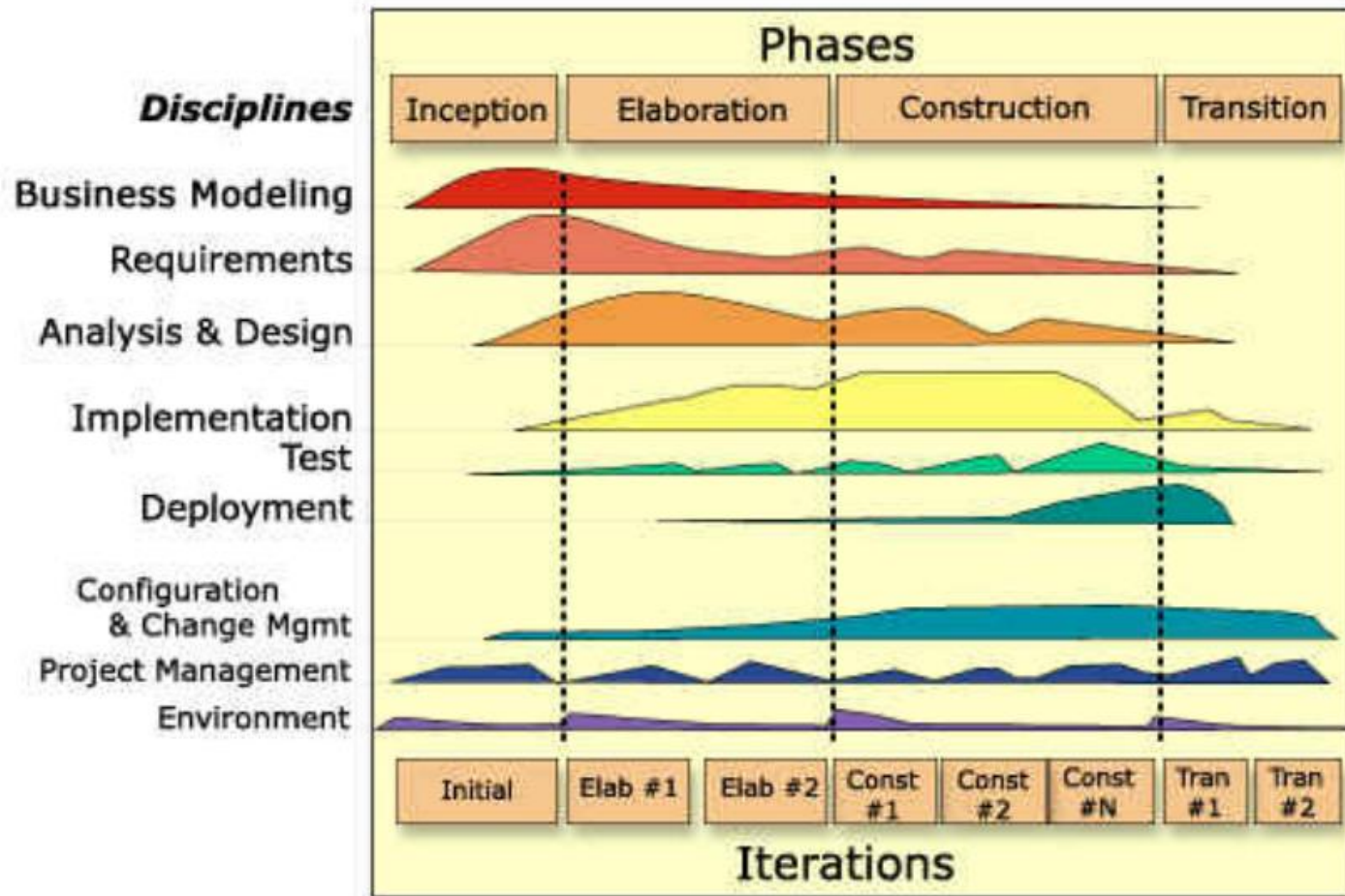
[Source Wikipedia\)](#)

De cascade vers l'itératif et agile

- L'approche itérative ne planifie pas tout le projet en détail au cours de sa phase de création; elle s'engage simplement sur les premières étapes. Puis les itérations suivantes sont planifiées au fur et à mesure de façon progressive et incrémentale et en tenant compte des changements de cap qui peuvent se produire durant le projet.
- On organise les itérations de façon à réduire au maximum les incertitudes et les risques qui peuvent compromettre la réussite du projet.

Processus Unifié

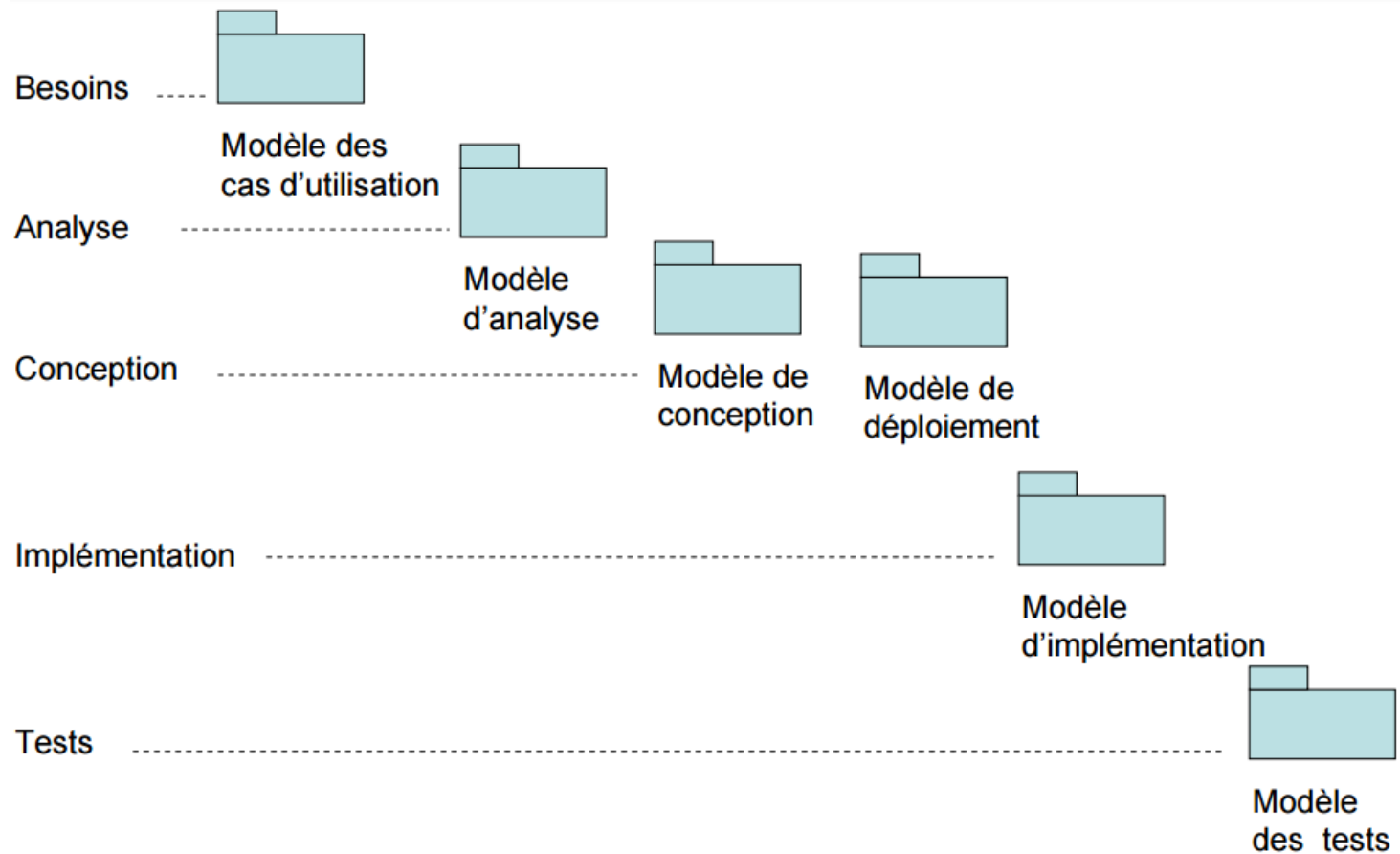
Le cycle de développement du processus unifié prévoit 4 phases et plusieurs itérations de développement où l'on procède à des activités d'analyse conception et programmation tout au long des itérations suivant l'itération de la phase création (Inception)



Phases du processus unifié

- Création
 - Préciser la portée et le rôle du produit
 - Identifier et mitiger les risques les plus importants
 - Valider la rentabilité et la viabilité du produit
- Élaboration
 - Saisir l'essentiel des besoins
 - Créer l'architecture de référence
 - Réduire les risques de moindre gravité
- Construction
 - Développer le système complet
 - S'assurer que le produit peut être utilisé par les clients
- Transition
 - S'assurer que l'on dispose d'un produit prêt à être livré à l'ensemble des utilisateurs
 - Former les utilisateurs

La modélisation dans le processus unifié



UML (Unified Modeling Language)

- Le processus unifié utilise UML pour modéliser les différents livrables de l'analyse, la conception et le déploiement.
- UML est un langage graphique de modélisation et de communication sur les divers aspects d'un logiciel durant les étapes de :
 - Processus métier ;
 - Besoins des utilisateurs ;
 - Architecture logiciel ;
- UML ne définit pas comment élaborer les modèles mais ses auteurs préconisent une démarche :
 - Itérative et incrémentale ;
 - Piloté par les besoins des utilisateurs ;
 - Centré sur l'architecture.

Les Modèles UML

- UML fournit une panoplie de diagrammes permettant de représenter diverses projections d'une même représentation.
- Les diagrammes sont organisés en vue statique et vue dynamique.
- Les diagrammes de la vue statique représentent la structure du système :
 - diagrammes d'objets
 - **diagrammes de classes**
 - **diagrammes de cas d'utilisation**
 - diagrammes de composants
 - diagrammes de déploiement
- Les diagrammes de la vue dynamique montrent le comportement (fonctionnement) du système :
 - **diagrammes d'activités**
 - **diagrammes de séquence**
 - **diagrammes de collaboration**
 - diagrammes d'états-transitions

Le client dans les méthodes agiles

- Les méthodes de type Développement rapide d'applications ou agile manquent parfois de traçabilité ce qui nécessite d'impliquer le client à la fois en termes de conception et également en termes juridiques : avec un cycle en V, le client est censé recevoir ce qu'il a commandé alors qu'avec des méthodes « agiles », le client devient co-développeur et intervient donc au niveau du projet.

Source Wikipédia

Approche centrée utilisateur

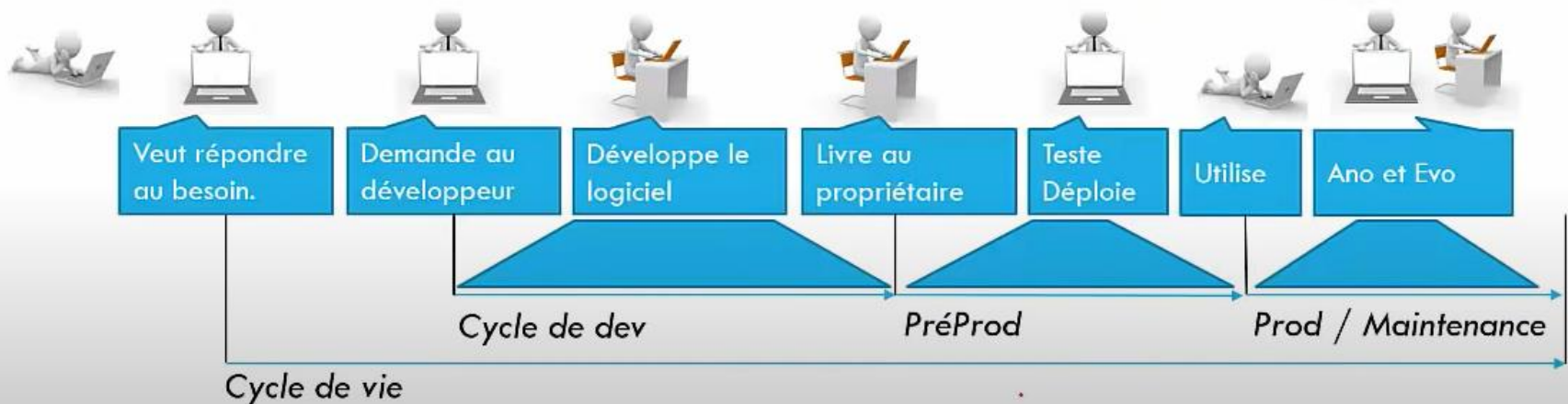
La démarche de développement centrée utilisateur a pour objectif **d'impliquer l'utilisateur du logiciel tout au long du cycle de vie du logiciel dont il a besoin**. Cela nécessite de :

- aller au contact des utilisateurs ;
- recueillir leurs besoins ;
- comprendre leurs objectifs et leurs attentes ;
- concevoir et évaluer avec eux tout au long du projet de l'application, à travers une démarche participative ;
- procéder par validation itérative.

Les bénéfices espérés :

- limiter les risques d'erreur ;
- réaliser des produits utiles (répondent au besoins), efficaces (atteignent les objectifs) et faciles d'usage.

CYCLE DE VIE — CYCLE DE DÉVELOPPEMENT



https://www.youtube.com/watch?v=vhpcngRVE_A&t=749s

(Auteur : Xavier Blanc, Durée 38 mn)

Sommaire Vidéo Cycle de vie et Cycle de développement

Types logiciel : application, site web, logiciel système, logiciel de développement

Participants à un projet de développement : utilisateur, propriétaire, développeur, (parties prenantes)

Étapes Cycle de vie : cahier de charge, développement, déploiement, utilisation, évolution,

Rôles : Maître d'ouvrage (Propriétaire), Maître d'œuvre (Développeur)

Activités Développement : Analyse, Conception, Réalisation

Activité Analyse : Spécification des requis et chiffrage de l'effort pour estimer la durée du projet et le budget.

Activité Conception : Choix d'architecture, décomposition en modules, spécification des composants, spécification interactions entre modules et composants,

Types d'architecture : 3 tiers (Présentation, Traitement, Base de données), en couche, par service, par entrepôt.

Outils de collaboration et de planification : Planification et répartition des tâches (Kanban), Codage collaboratif (Git) ,

Cycles de développement Cascade : Les activités se font en séquence ; analyse et documentation exhaustive de tous les besoins , ensuite conception et documentation de la totalité de la solution, puis réalisation, tests et déploiement

Cycle de développement en V : anticipation des tests utilisateur, tests d'intégration et tests unitaires lors de l'analyse, la conception et la conception détaillée. Retour sur les étapes analyse, conception et conception détaillée si les tests révèlent des anomalies.

Cycle de développement Itératif et Agile : produit livré progressivement en plusieurs incréments à travers plusieurs itérations. Dans chaque itération, on procède à des activités d'analyse, conception et réalisation. D'une itération à l'autre, on peut accepter des changements aux requis de la part du client propriétaire du produit. de

Le Guide Scrum (résumé)



- Piliers
 - Transparence,
 - Inspection
 - Adaptation
- Participants
 - Product Owner
 - Development Team
 - Scrum Master
- Événements
 - Users Stories
 - Sprint Planning
 - Sprint
 - Daily Scrum
 - Sprint Review
 - Incrément
 - Sprint Retrospective

<https://www.youtube.com/watch?v=2wRGhYQAnOc>

(Auteur : Denis Voituren, durée 45 mn)

Sommaire Vidéo SCRUM

Piliers : **transparence** (processus visibles à tous les participants, définition standard de terminé), **Inspection** (prévoit des tâches de vérification de l'avancement du projet) , **Adaptation** (ajuster le développement selon les priorités du client (Propriétaire du produit))

Propriétaire produit : représentant du client et responsable du carnet des fonctionnalités que l'on veut développer dans le produit. Chaque fonctionnalité est exprimée selon une formulation de User story (récit utilisateur) dans laquelle on doit préciser l'acteur de la fonctionnalité, décrire ce que fait la fonctionnalité et son objectif

Équipe développement : équipe multidisciplinaire (analyste, architecte, développeur, testeur, utilisateurs). Doit planifier avec le Propriétaire produit , **la liste des fonctionnalités à développer à chaque sprint**, déterminer les tâches de développement de chaque fonctionnalité, définir les **critères selon lesquels le développement d'une fonctionnalité est dit terminé (Definition of Done)** et doit exécutés les activités de développement planifiés et produire l'incrément du sprint et déployer le livrable au client.

Scrum Master : coach SCRUM et facilitateur il doit veiller à l'application du processus SCRUM durant les sprints, faire le suivi et la **résolution des obstacles** rapportés durant les scrums (mélés) quotidiens, et suivre le **burn chart** de l'équipe (nombres d'heures de développement consommées vs nombre d'heures planifiées)

Étapes : **Sprint 0 :** création du **carnet des fonctionnalités du produit** (Product backlog), **priorisation des fonctionnalités** et **chiffrage**, **Sprint 1** (planification du carnet des fonctionnalités du sprint et des activités de développement, développement des fonctionnalités, scrums quotidien , revue de sprint en fin de projet, déploiement au client si l'incrément est livrable, rétrospective en fin de sprint, Sprint 2, Sprint3 etc....

Scrum (quotidien 15 mn à la même heure) : chaque membre doit rapporter les activités **complétés**, activités **en cours**, activités à **venir**, **obstacles**, **burndown chart**.

Revue de Sprint (en fin de sprint, 2 à 4 heures) : valider le développement des fonctionnalités du sprint, décider de l'incrément (livrable ou non au client) replanifier les fonctionnalités non terminées s'il y en a.

Rétrospective de Sprint (en fin de sprint, 2 à 4 heures) : vérifier les conditions et l'environnement de travail, identifier les points satisfaisants et les points à améliorer.

Scrum



<https://www.youtube.com/watch?v=ZBaCbdWYLeA&t=2620s>

(Auteur Carl Gilbert : Durée : 55 mn)

Résumé succinct Vidéo SCRUM

Cycle en Cascade : Contenu à livrer fixe, temps et coût variable,

Agile : temps et coût fixes, Contenu variable

Processus de démarrage au Sprint 0 : 3 livrables

Vision du produit : quoi et pourquoi (objectifs), production de maquette / visual box

Backlog du produit : Liste de toutes les fonctionnalités en forme de user stories (rôle, quoi, justification) avec priorité (en fonction de la valeur) et effort (heure personne ou en points de complexité)

Plan de livraison : fixer durée sprint (1 semaine à 4 semaines), déterminer capacité de l'équipe (heures personnes), déterminer user stories de chaque sprint, organiser livraisons

Principes Agile et récits utilisateurs

Agile - 12 mn



Principes - 9 mn



User story - 7 mn



Splitting user stories - 9mn



<https://www.youtube.com/watch?v=EDT0HMTDwYI>