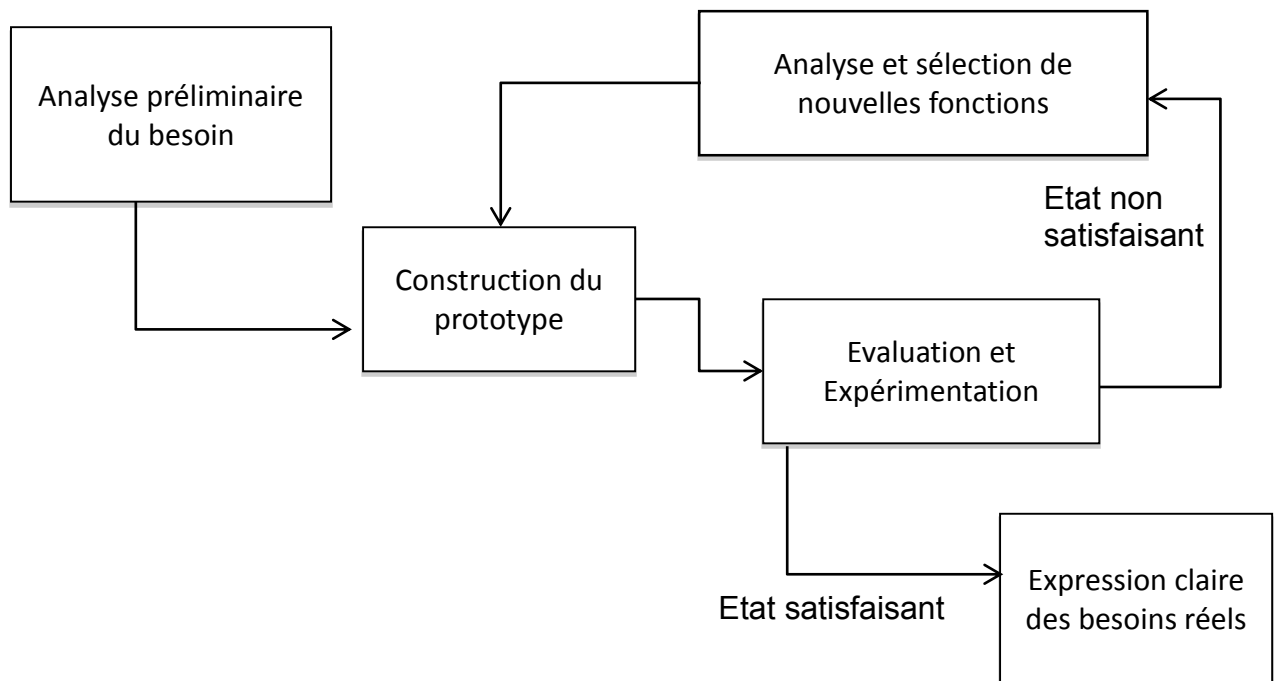


1) Modèles du Cycle de Vie (Suite) : Vers des modèles de plus en plus souples

Modèle par prototypage

Le cycle de vie par prototypage consiste à explorer rapidement et à moindre frais les fonctions, l'architecture et/ou la convivialité d'un logiciel. Il est plus efficace pour les projets où les besoins sont mal définis ou difficiles à élaborer de manière claire et complète dès le début du développement. On part d'une spécification partielle, on développe un premier prototype qu'on expérimente, évalue et enrichit de manière itérative jusqu'à l'obtention d'un état de satisfaction.



On peut distinguer deux types de prototypes :

- Prototype jetable: le prototype ne sert qu'à une exploration ou à une expérimentation,
- Prototype évolutif: le prototype est conçu pour évoluer vers le système final

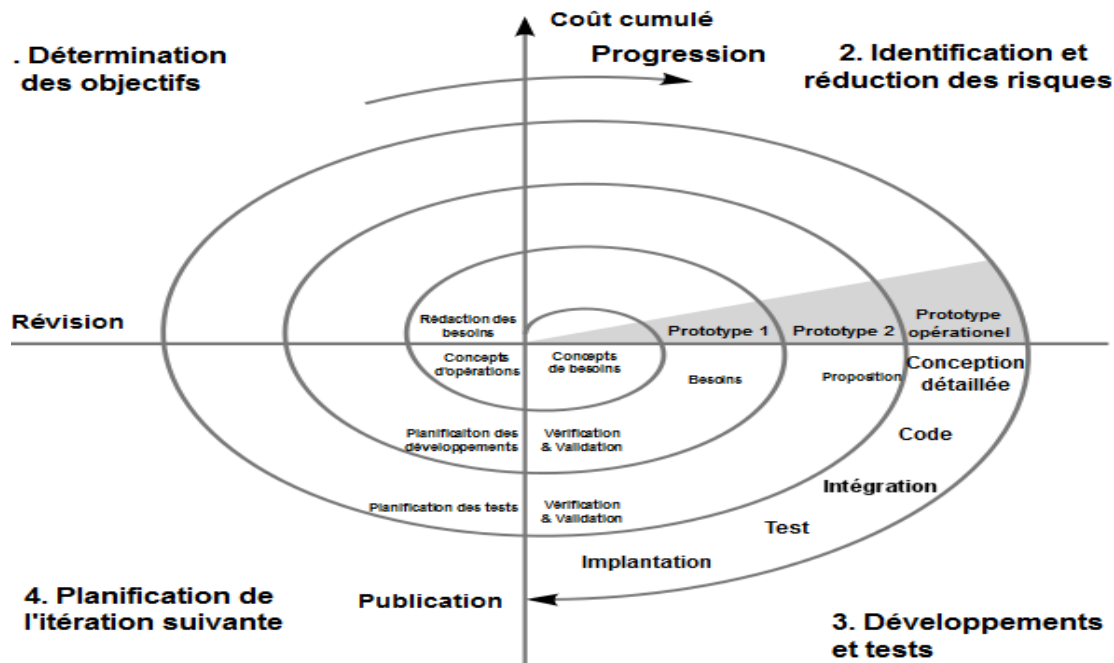
Modèle incrémental

C'est un Modèle constitué d'une succession de cycles en cascade. Les activités des phases doivent constituer une suite logique des phases précédentes. Par exemple, la spécification de la phase 2 (2^{ème} cascade) complète la spécification élaborée dans la phase 1 (1^{ère} cascade en partant du haut vers le bas). Chaque phase vient incrémenter les résultats des travaux d'une phase de même nature d'un cycle parallèle. Chaque incrément donne un produit fonctionnel

Modèle en spirale

Il Constitue le résultat de l'affinement du modèle en V, enrichi de l'expérience des divers modèles de cycle de vie et de l'analyse des risques. On distingue quatre phases dans le déroulement du cycle en spirale :

1. détermination des objectifs, des alternatives et des contraintes ;
2. analyse des risques, évaluation des alternatives ;
3. développement et vérification de la solution retenue ;
4. revue des résultats et vérification du cycle suivant.



Les 3 modèles de cycles de vie (Prototypage, par incrément et spirale) s'inscrivent dans le cadre d'une démarche de développement itérative et incrémentale.

Conclusion : L'examen des modèles de cycle de vie étudiés précédemment et les pratiques de développement soulèvent quelques remarques:

- Certains modèles de développement se basent sur une hypothèse pas trop réaliste: on sait ce qu'on veut. Or le client ne le sait pas toujours de manière précise et complète
- La documentation à outrance dans les étapes de développement consomme de l'énergie et cache l'information utile
- Le manque de collaboration entre les utilisateurs (client ou commanditaire) et l'équipe projet pendant le processus de développement (se voient au début et à la fin du projet). Il en résulte un manque de réactivité de la part de l'équipe projet à des changements qui peuvent survenir pendant le développement.

Par ailleurs, l'apparition de nouvelles classes de problèmes compliqués voir complexes (donc ne pouvant être résolus que de manière incrémentale), la volonté de rendre le développement efficace en prenant en compte les besoins du client tout au long du processus de développement (et non seulement au début et à la fin) ont conduit à l'émergence de nouveaux modèles de développement qualifiés d'agiles.

2) Méthodes de développement Agiles

Définition: Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme.

Un certain nombre d'approches partage cette définition: Crystal, **Scrum**, eXtreme Programming. Ces approches reposent sur quatre principes que l'on peut résumer ainsi :

- **Importance des Individus et de leurs interactions** plus que les processus et les outils
- **Logiciel qui fonctionne** plutôt qu'une documentation exhaustive

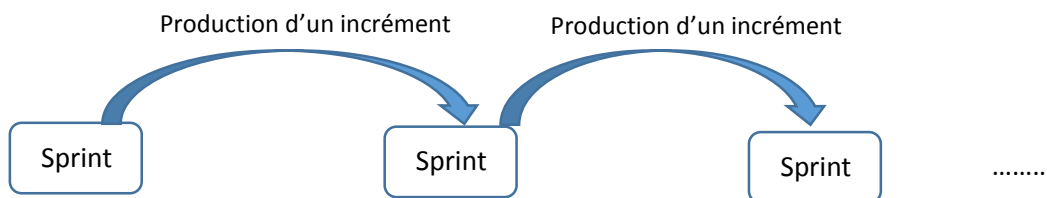
- **Collaboration avec le client** plutôt que négociation de contrat
- **Réponse au changement** plutôt que le suivi d'un plan prédéfini

Attention : ces principes ne signifient pas qu'il faut laisser tomber ce qui n'est pas souligné. Ils signifient seulement que tout en les maintenant (les processus, la documentation,), il faut privilégier ce qui est souligné. Nous allons présenter l'approche Scrum, un processus agile mettant en œuvre ces principes.

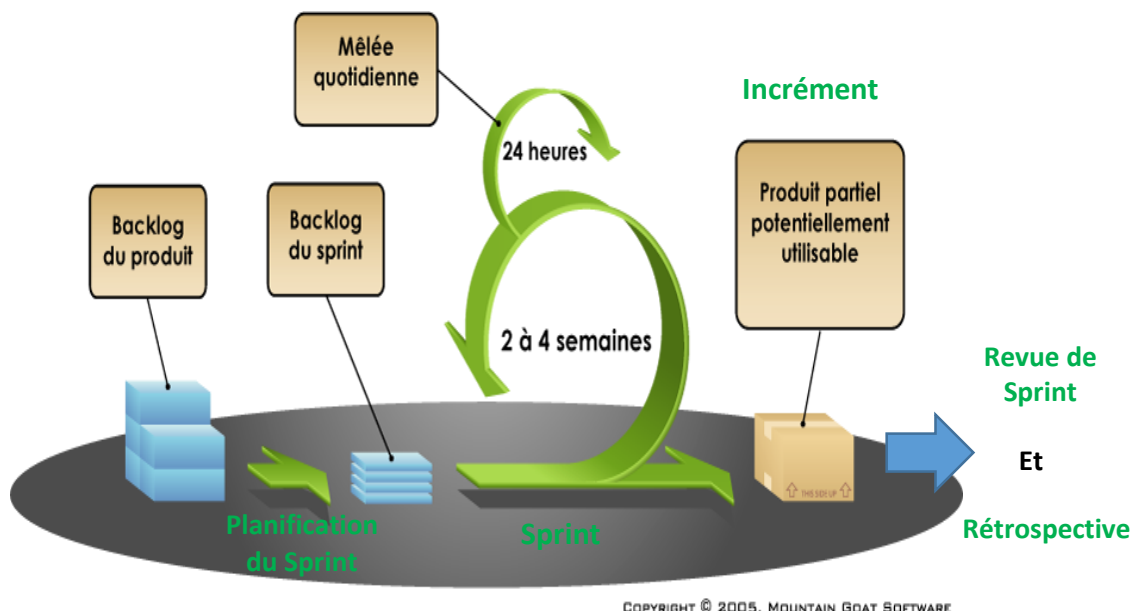
3) Approche Scrum

3.1) Vue globale

Cette approche reprend le cycle de développement itératif et incrémental en le structurant autour de ce que l'on appelle les sprints (voir figure ci-dessous). Un sprint est un cycle de développement qui aboutit à la création d'un incrément du produit, potentiellement livrable.



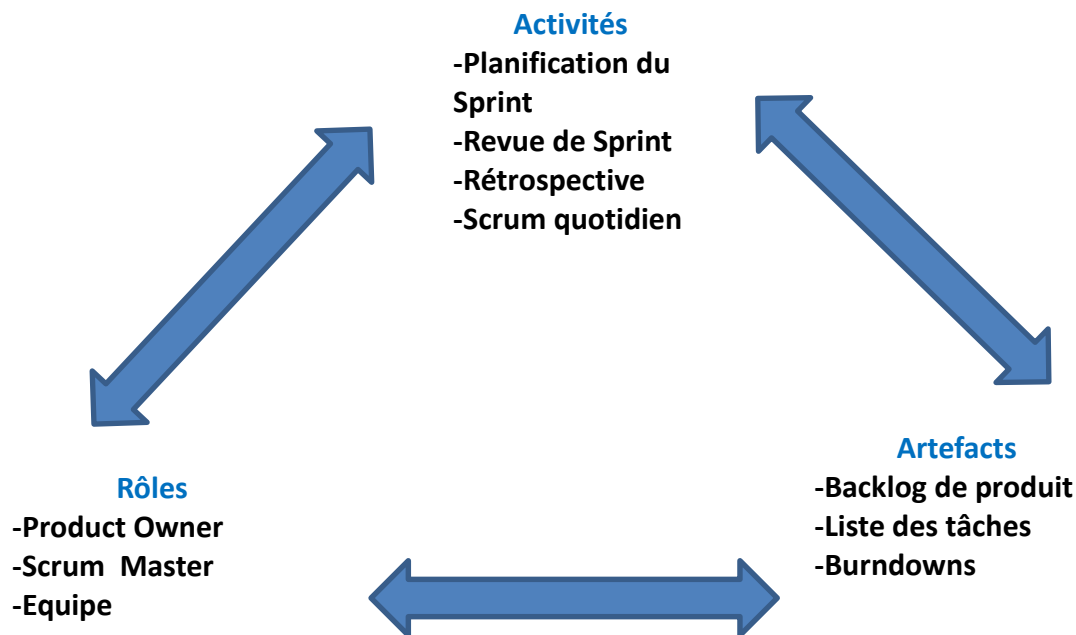
Bien évidemment, durant le sprint, le développement peut s'articuler autour des activités de spécification, conception, codage et test. La figure suivante donne une vue globale du processus Scrum.



Partant d'une liste incomplète de fonctionnalités pouvant répondre aux besoins du client (**Backlog du produit**), l'équipe de développement choisit lors de la réunion de **planification du sprint**, les éléments qui seront réalisés ou traités durant ce sprint. Ces éléments constituent alors le **backlog du sprint**. En général, un sprint peut durer de 2 à 4 semaines et termine par un

produit apportant une valeur ajoutée au client. Une réunion rapide et productive est prévue quotidiennement (mêlée quotidienne de l'ordre de 15mn) pour coordonner les membres de l'équipe sur les tâches en cours et sur les difficultés rencontrées. A la fin du sprint, une réunion est tenue pour montrer le travail réalisé (les fonctionnalités demandées par le client), obtenir un retour des participants sur les tâches réalisées, et identifier les nouveaux items à ajouter au product backlog.

Nous allons maintenant donner une définition aux différents concepts de l'approche, résumés par le schéma ci-dessous :



Nous allons préciser chaque concept.

3.2) Backlog de produit

C'est la liste des fonctionnalités (items, User stories)

- Une liste de tout ce qui va entraîner du travail pour l'équipe
- Exprimée de telle façon que chaque élément apporte de la valeur ajoutée aux utilisateurs ou clients du produit
- Les priorités sont définies par le Product Owner et revues à chaque sprint
-

Un exemple de ce que peut contenir un backlog est donné par le tableau suivant, concernant un projet de développement d'un système de réservation de chambres d'hôtel.

| Elément du backlog | Estimation |
|--|------------|
| En tant qu'invité, j'annule une réservation | 3 |
| En tant qu'employé de l'hôtel, je peux produire les rapports de revenu par chambre | 8 |
| | |

L'approche Scrum utilise la notion de « User stories » pour définir les fonctionnalités ou les services attendus par le client. Elle propose aussi un style d'expression de ces fonctionnalités

de la forme : **En tant qu'**employé de l'hôtel (**rôle**), **je peux** produire (**action**) les rapports de revenu par chambre **afin de (but)** disposer d'un tableau de bord sur le rendement des chambres.

3.3) Les rôles

Product Owner C'est le représentant du client (des utilisateurs) dans l'équipe : Définit les fonctionnalités du produit, choisit la date et le contenu de la release, ajuste les fonctionnalités et les priorités à chaque sprint si nécessaire, et accepte ou rejette les résultats.

ScrumMaster Il incarne le management du projet : fait appliquer par l'équipe les valeurs et les pratiques de Scrum, facilite une coopération poussée entre tous les rôles et fonctions, et protège l'équipe des interférences extérieures

Equipe Scrum Regroupe toutes les compétences nécessaires au développement: Architecte, concepteur, développeur, spécialiste IHM, testeur, etc. L'équipe s'organise par elle-même. La composition de l'équipe ne doit pas changer pendant un Sprint

3.4) Les activités

Planification du Sprint

- Analyser et évaluer le backlog de produit pour définir le but du sprint
- Créer la liste des tâches à partir des éléments du backlog de produit et estimer le nombre d'heures nécessaires à leurs réalisations.

Scrum quotidien

Il s'agit d'une réunion (Scrum Meeting) pouvant durer de 15 à 30 mn tous les jours pour faire le point sur l'avancement des tâches. Elle peut se structurer autour de trois questions à se poser par chaque membre de l'équipe : Qu'as-tu fait hier ? Que vas-tu faire aujourd'hui ? Y a-t-il un obstacle qui te freine?

Revue de sprint

L'équipe présente ce qu'elle a fait pendant le sprint et réalise une démonstration des nouvelles fonctionnalités.

Rétrospective du sprint

Il s'agit de mener une réflexion sur ce qui marche et ce qui ne marche pas: lister les problèmes rencontrés et les erreurs commises. Elle dure en général de 15 à 30 minutes à la fin de chaque sprint

Bibliographie

Présentation de M. Cohn, Mountain Goat Software, traduite en Français par C. Aubry

A Lightweight Guide to the Theory and Practice of Scrum, Version 2.0. Pete Deemer, Gabriele Benefield, Craig Larman, Has Vodde.

Scrum Handbook, Jeff Sutherland's