



Méthode Agile: SCRUM

Formateur : Ahmed Kenore

Sommaire.

☐ Introduction

- ☐ La gestion de projet informatique.
- ☐ Le modèle nominal "en cascade"
- ☐ Le modèle en V
- ☐ L'agilité dans les projets
- ☐ Scrum une méthode agile

☐ Quelques méthodes agiles

- ☐ Le Lean Management
- ☐ Le Kanban
- ☐ La méthode eXtreme Programming(XP)
- ☐ Scrum une combinaison de méthodes

☐ Présentation générale

- ☐ La naissance de Scrum
- ☐ Scrum, une vision globale
- ☐ Cycle de vie de Scrum
- ☐ Les trois piliers de Scrum
- ☐ Coût, délai et périmètre

Sommaire.

☐ Les rôles

- ☐ Le Scrum Master
- ☐ Le Product Owner
- ☐ L'équipe de développement

☐ Artefacts et Activités

- ☐ Product Backlog
- ☐ Rédaction des tests pour les User Stories
- ☐ Priorisation du Product Backlog
- ☐ Estimation et planification de Sprint
- ☐ Planifications de releases
- ☐ La vie d'un Sprint
- ☐ Le suivi
- ☐ La conduite de changement
- ☐ Scrum à l'aide d'un logiciel



Part 1 : Introduction



Qu'est ce qu'un projet ?

❑ Définition

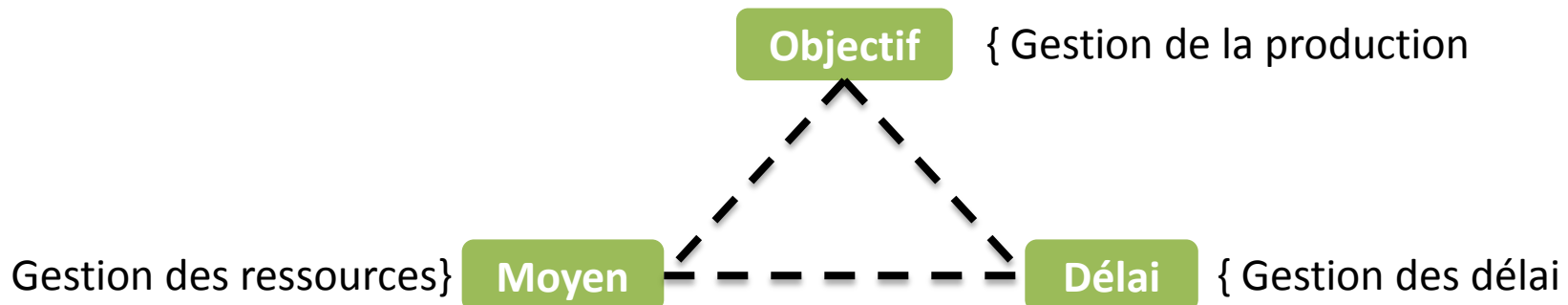
Projet = Entreprise temporaire initiée dans le but de créer un produit, un service ou un résultat unique.

Source : Guide du PMBOK, 2004

Projet = processus unique, qui consiste en un ensemble d'activités coordonnées et maîtrisées comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques telles que les contraintes de délais, de coûts et de ressources.

Source : Norme ISO 10006 : 2003

❑ Le Triangle projet



Introduction : Le modèle nominal "en cascade"



Qu'est ce qu'un modèle nominal ?

Cette méthode consiste à prévoir des étapes séquentielles.

Chaque étape suit la précédente et ne peut exister que si l'étape antérieure est terminée.

Il n'y a donc pas existence de notion de parallélisations d'étapes.

A l'issue de chaque étape des documents sont produits pour en vérifier

Egalement appelé "modèle en cascade" ou "Waterfall" , il fut mis au point dès 1966 puis formalisé aux alentours de 1970.

Les étapes séquentielles appliquées et rencontrées dans la majeure partie des entreprises en termes de gestion de projet sont celles-ci :

Introduction : Le modèle nominal [Cycle de vie d'un projet]

Étape 1

- Posséder une idée de projet (cela va de soi).

Étape 2

- Avoir une expression de besoins rédigée selon un formalisme défini par l'entreprise, recensant les exigences fonctionnelles des utilisateurs

Étape 3

- Rédiger éventuellement un contrat (dans le cas d'une réalisation au forfait) basé sur cette expression de besoins où nous y trouverons les engagements tels que le délai, le coût et ainsi que les responsabilités de chacun (pénalités de retard...).

Étape 4

- Rédiger des spécifications techniques par les concepteurs informatiques, où nous pourrons voir la naissance de diagramme UML (Unified Modeling Language), de schéma de base de données...

Étape 5

- Début des développements informatique sur lecture des spécifications fonctionnelles et techniques


Étape 6

- Recette globale de l'application (tests unitaires, test fonctionnel, tests d'intégration).

Étape 7

- Livraison et installation de l'application

Introduction : Le modèle nominal [Cycle de vie d'un logiciel]

- 
1. **Définition des objectifs**, consistant à définir la finalité du projet et son inscription dans une stratégie globale
 2. **Analyse des besoins et faisabilité**, c'est-à-dire l'expression, le recueil et la formalisation des besoins du client et de l'ensemble des contraintes.
 3. **Conception générale**. Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.
 4. **Conception détaillée**, consistant à définir précisément chaque sous-ensemble du logiciel.
 5. **Codage** (Implémentation ou programmation), soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
 6. **Tests unitaires**, permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémentée conformément aux spécifications.
 7. **Intégration**, dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration consignés dans un document.
 8. **Qualification (ou recette)**, c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
 9. **Documentation**, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
 10. **Mise en production**,
 11. **Maintenance**, comprenant toutes les actions correctives (maintenance corrective)

Introduction : Le modèle nominal "en cascade"

❑ **Avantage du modèle en cascade**

Le planning est établi à l'avance et le chef de projet sait précisément ce qui va lui être livré et quand il pourra en prendre livraison

❑ **Inconvénients du modèle en cascade**

Le principal inconvénient est la très faible tolérance à l'erreur (les anomalies sont détectées tardivement) qui induit automatiquement un coût important en cas d'anomalie

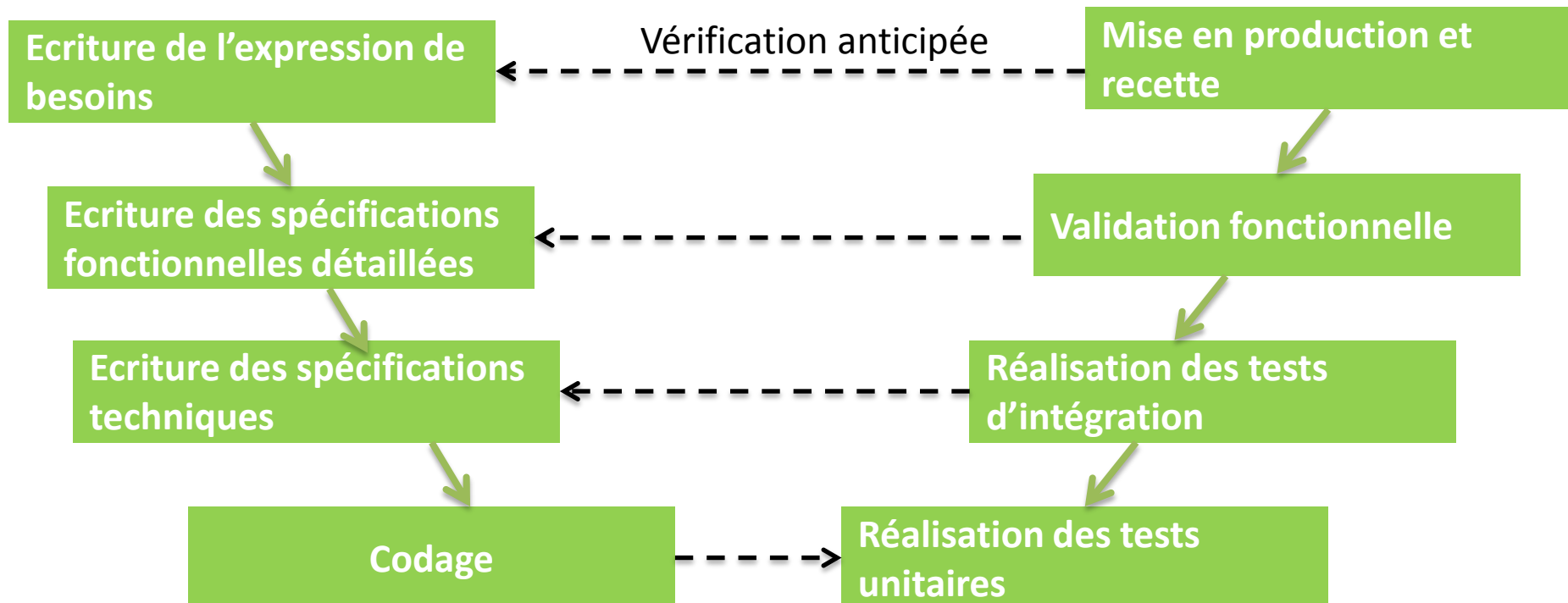


Qu'est ce qu'un modèle en V ?

□ En Théorie

Le modèle en V part du principe que les procédures de vérification de la conformité du logiciel aux spécifications doivent être élaborées dès les phases de conception.

Elle comporte 8 étapes décrites ci dessous:



Introduction : Le modèle en V

Étape 1 : Ecriture de l'expression de besoins : Le client exprime son besoin, en décrivant les fonctionnalités correspondant au produit fini tel qu'il peut l'imaginer.

Étape 2 : Ecriture des spécifications fonctionnelles détaillées : c'est le cahier des charges exact du produit final, tel que le désire le client. Il doit couvrir l'intégralité des cas d'utilisation du produit, en expliquant ce qu'il doit faire. On y trouve les règles de gestion, des copies d'écrans... (cela nécessite plusieurs dizaines de pages et quelques mois d'écriture).

Étape 3 : Ecriture des spécifications techniques : c'est une traduction des spécifications fonctionnelles en termes techniques (mise en place de l'architecture, design de la base de données, choix du langage...). Cette tâche est généralement réalisée par les ingénieurs concepteurs.

Étape 4 : Codage : c'est la phase de réalisation du produit.

Introduction : Le modèle en V

Étape 5 : Réalisation des tests unitaires : ces tests permettent de s'assurer que chaque brique fonctionnelle (parfois appelé module) est en ordre de marche sans bug apparent.

Étape 6 : Réalisation des tests d'intégration : ce sont là les premiers tests grandeur nature du produit fini. On s'assure que le produit livré est conforme aux recommandations décrites dans les spécifications techniques.

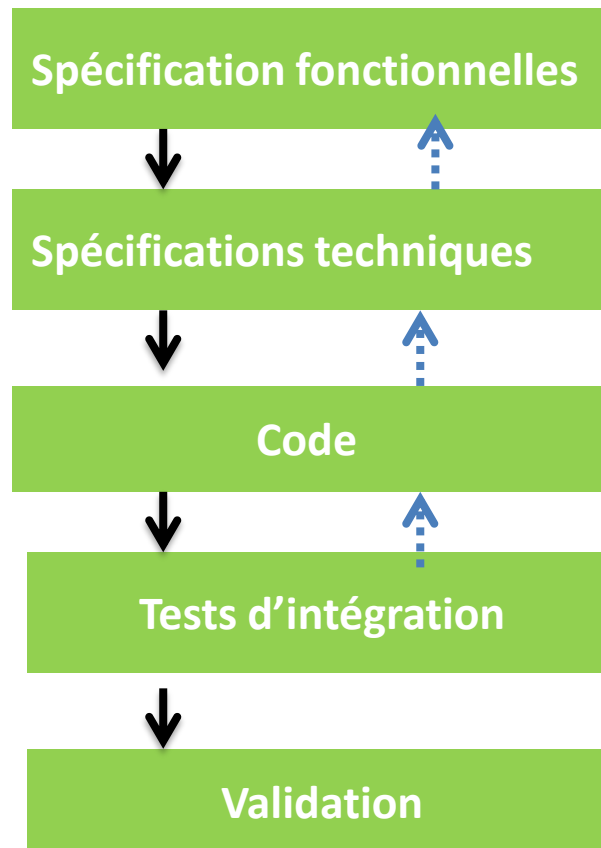
Étape 7 : Validation fonctionnelle du produit. Le produit doit être conforme à l'ensemble des règles fonctionnelles décrites dans les spécifications fonctionnelles.

Étape 8 : Mise en production et recette : le produit est vérifié une dernière fois dans un environnement dit de "préproduction", avant d'être installé sur l'environnement de production. Le client procède alors à la recette, pour vérifier que son expression de besoins est respectée.

Introduction : Le modèle en V

❑ En pratique

En pratique on observe une communication ascendante et descendante dans les phases de rédaction des spécifications et du codage.



Introduction : Le modèle en V

❑ Les rôles

❑ MOA(Maîtrise d'OuvrAge)

Elle est en charge de l'aspect fonctionnel du produit.

Elle définit le besoin et veille à ce que ce dernier soit respecté lors des phases de recettes

❑ MOE (Maîtrise d'OEuvre)

Elle est en charge d'écrire les spécifications détaillées, de réaliser le développement sans oublier les tests d'intégration.

Exemple: les chefs de projets, les développeurs et les architectes

❑ Inconvénients du modèle en V

Les utilisateurs (la MOA dans son ensemble) après la rédaction des besoins sont peu ou pas sollicités pendant les phases techniques (conception, développement,...).

Donc un effet tunnel qui fait que le produit développé ne permet pas de prendre en compte de nouvelles exigences.

Introduction : L'agilité dans les projets

❑ La petite histoire

Les valeurs et principes de l'agilité(manifeste agile) ont vu le jour au Etats Unis lors d'une réunion organisée en février 2001, réunion dans laquelle on participée Kent Beck (inventeur de l'eXtreme Programming), Ken Schwaber et Jeff Sutherland (cofondateur de Scrum).

❑ Les valeurs

Le manifeste agile est composé de quatre valeurs :

1. Favoriser les individus et leurs interactions plutôt que la mise en place de processus et d'outils.
2. Favoriser la collaboration avec le client plutôt que la négociation contractuelle.
3. Faire en sorte d'obtenir un logiciel immédiatement disponible à l'instar d'une documentation bien fournie mais qui s'avère être finalement inutile (un utilisateur préférera utiliser le logiciel plutôt que de rêver sur une belle documentation).
4. Être réactif face au changement plutôt que de respecter un plan.

Introduction : L'agilité dans les projets

❑ Les 12 principes

1. Accorder une haute priorité à la satisfaction du client à travers des livraisons de logiciels rapprochées et continues.
2. Accepter le changement de besoins même tard dans le développement.
3. Livrer fréquemment un logiciel qui marche à échéances régulières, de deux semaines à deux mois avec une préférence pour les plus petites périodes de temps.
4. Faire travailler ensemble quotidiennement les utilisateurs et les développeurs.
5. Construire les projets autour de personnes motivées. Leur donner l'environnement et le support dont elles ont besoin et leur faire confiance.
6. Privilégier la communication face à face qui est le moyen le plus efficace pour transmettre de l'information aux équipes de développement.
7. Considérer les versions opérationnelles du logiciel comme étant les mesures principales de progrès.

Introduction : L'agilité dans les projets

❑ Les 12 principes

8. Considérer les procédés agiles comme les moteurs d'un développement viable. Sponsors, développeurs et utilisateurs doivent pouvoir maintenir un rythme constant indéfiniment.
9. Apporter une attention continue à l'excellence technique et à la bonne conception afin d'améliorer l'agilité.
10. Privilégier la simplicité. C'est à dire l'art de maximiser le travail à ne pas faire.
11. Considérer que les meilleures architectures, besoins et conceptions émergent d'équipes auto organisées.
12. Réfléchir à intervalle régulier à la façon de devenir plus efficace et agir sur le comportement de l'équipe en conséquence.

Introduction : Scrum une méthode agile

C'est en 2001 que Ken Schwaber et Mike Beedle publient officiellement la description de leur méthode agile nommée Scrum dans l'ouvrage "Agile Software Development With Scrum".

Le nom Scrum ou bien encore « Mêlée » exprime le concept fort de la méthode s'articulant autour de la cohésion d'équipe.

La méthode Scrum, comme toute méthode agile, s'illustre par le respect des valeurs et principes du manifeste agile :

1. L'existence d'une équipe auto organisée.
2. Des rôles définis : Scrum Master, Product Owner (le client), équipe de développement.
3. Un ensemble de cérémoniaux (Scrum Meeting, revue de sprint...).
4. La présence constante du client.
5. La mise en place de mécanismes favorisant les livraisons fréquentes (Sprint).
6. Une acceptation du changement (mais pas à n'importe quel prix).