



PRÉSENTATION DES MÉTHODES AGILES ET SCRUM

Les méthodes agiles sont des méthodologies essentiellement dédiées à la gestion de projets informatiques.

Elles reposent sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client.

Elles permettent notamment d'impliquer l'ensemble des collaborateurs ainsi que le client dans le développement du projet.

Les méthodes agiles se reconnaissent toutes dans les valeurs suivantes.

L'équipe et la communication avant les outils et processus : dans la vision agile, l'équipe est bien plus importante que les outils ou les procédures de fonctionnement.

Il est préférable d'avoir une équipe soudée et dont les membres communiquent entre eux, composée de développeurs de niveaux différents, plutôt qu'une équipe composée d'experts qui travaillent de manière isolée.

La communication est donc une notion fondamentale dans un contexte de développement agile.

L'application avant la documentation : il est primordial que le projet fonctionne, c'est la priorité avant toute chose.

La documentation technique et les autres outils (de tests, de rapports) constituent une aide précieuse mais ne sont pas une fin en soi.

Une documentation précise est utile comme moyen de communication.

Il est parfois préférable de simplement commenter abondamment le code lui-même, et surtout de transférer la totalité des compétences et connaissances du métier à l'ensemble des collaborateurs de l'équipe.

La collaboration avant la négociation : le client doit être impliqué dans le développement.

Le fournisseur ne doit pas se contenter de négocier un contrat au début du projet, puis de refuser l'évolution des besoins du client.

Le client doit collaborer avec l'équipe et fournir des comptes rendus réguliers sur l'adaptation du logiciel à ses attentes.

5

L'acceptation du changement et la flexibilité avant la planification : la planification initiale et la structure du projet doivent être flexibles afin de permettre les évolutions attendues par le client.

En effet, les premières livraisons du projet donnent très souvent suite à des demandes d'évolution.

6

Le manifeste Agile

Le manifeste Agile (ou « agile manifesto ») est un texte apparu en 2001 et rédigé par 17 experts du développement logiciel. Ce texte reprend les 4 valeurs communes des méthodes agiles et les dérive en 12 principes précisés ci-dessous.

1. La plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
2. Il faut accueillir positivement les changements et les nouveaux besoins, même lorsqu'ils arrivent tardivement dans un projet. Les processus agiles exploitent la flexibilité au changement afin de fournir un avantage compétitif pour le client.
3. Il faut livrer régulièrement un logiciel opérationnel (utilisable en production) avec des cycles courts (idéalement entre deux et ⁷ quatre semaines).

4. Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble au quotidien et tout au long du projet.
5. Il faut réaliser les projets avec des personnes motivées, leur fournir des environnements adaptés à leur besoin ainsi que le soutien dont ils ont besoin et leur confiance pour atteindre les objectifs fixés.
6. Le dialogue en face à face entre les différents acteurs d'un projet agile est la méthode la plus simple et la plus efficace pour transmettre l'information et la connaissance entre ces derniers.

7. L'aspect opérationnel d'un produit est la principale mesure d'avancement de ce dernier.

8. Les processus agiles doivent amener à un rythme de développement soutenable pour l'équipe et constant (il ne doit pas y avoir de période de forte montée ou baisse de charge de travail ayant des impacts significatifs sur l'équipe).

9. La recherche de l'excellence et de la performance conceptuelle et technique renforce l'agilité d'un produit.

9

10. Simplifier le travail en minimisant le nombre de tâches inutiles et redondantes est essentiel.

11. Les meilleures solutions logicielles émergent d'équipe auto-organisées tant au niveau de la clarté des spécifications, que de la conception et de la mise en place d'architectures performantes et efficaces.

12. L'équipe doit réfléchir à des moyens, à intervalles réguliers, pour devenir davantage efficace et mettre en pratique ces nouvelles méthodes une fois décidées.

10

Présentation de la méthode Scrum

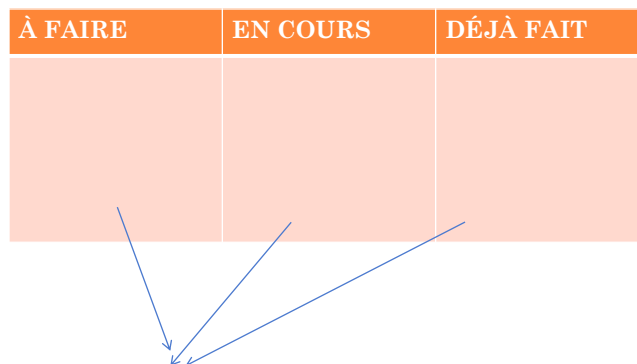
La méthode **Scrum** est une méthode agile, créée en 2002, dont le nom est un terme emprunté au rugby qui signifie « la mêlée ».

Elle s'appuie sur le découpage des projets en **itérations** encore nommées « **sprints** ».

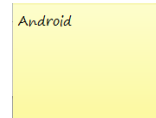
Un **sprint** peut avoir une durée qui varie généralement entre deux semaines et un mois.

11

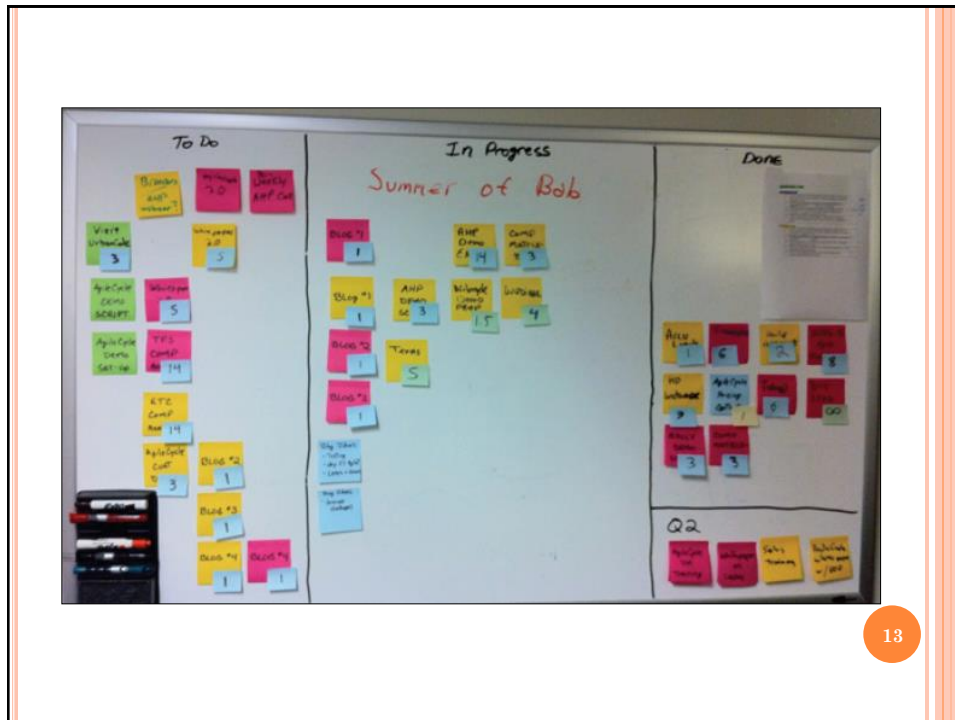
«BACKLOG»



Énoncé : peut être fait pas des «post-it»



12



13

Avant chaque sprint les tâches sont estimées en temps et en complexité .

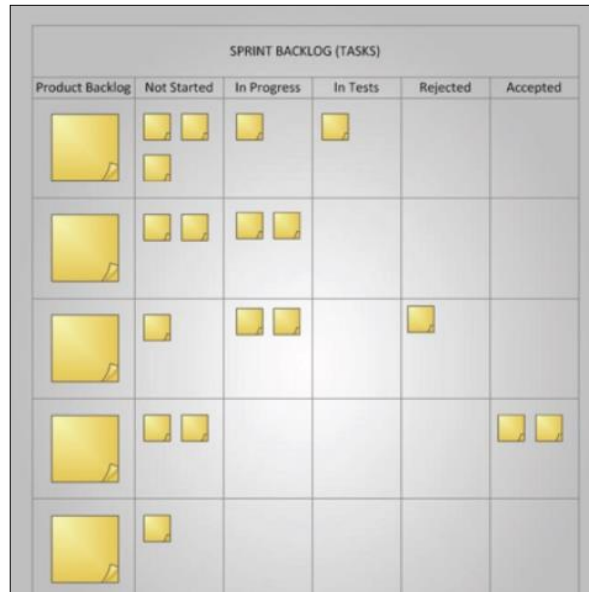
Ces estimations permettent à la fois de planifier les livraisons mais aussi d'estimer le coût de ces tâches auprès du client.

Les fonctionnalités (encore appelées « **user stories** ») qui font l'objet d'un sprint constituent ce que l'on appelle un « **sprint backlog** » du produit éventuellement livrable à la fin du sprint.

Il est nécessaire de distinguer le «**sprint backlog**» du «**product backlog**» qui lui correspond à l'ensemble des fonctionnalités attendues pour le produit sur l'ensemble des sprints.

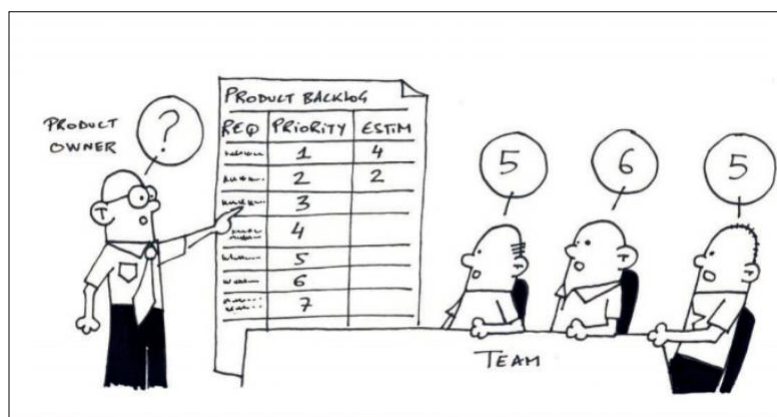
14

«Product Backlog»



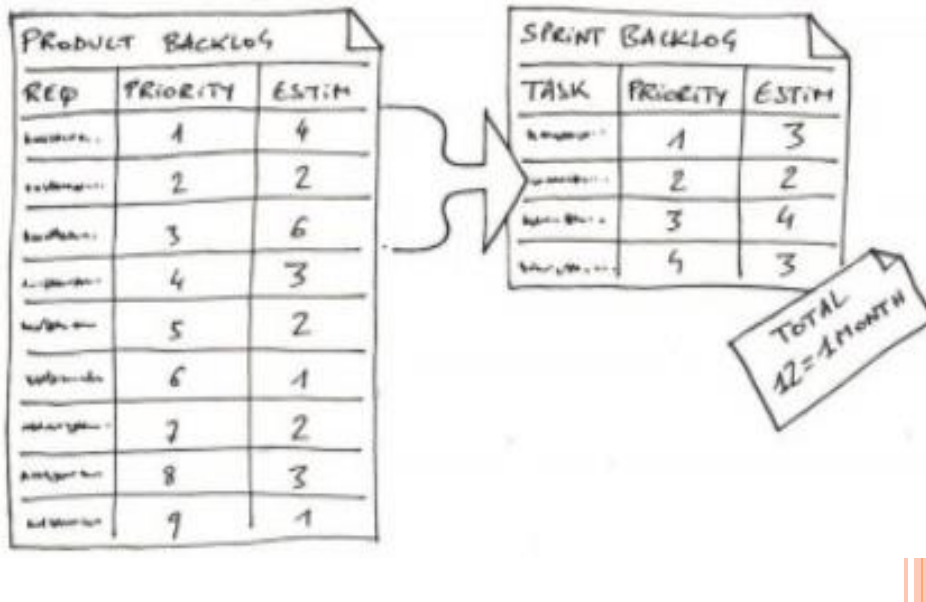
15

Meetings : Planification du sprint



16

Sprint Backlog

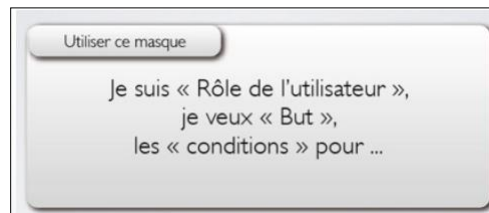


Recueil des différentes tâches, extraites du Product Backlog, que l'équipe s'engage à réaliser lors du Sprint.

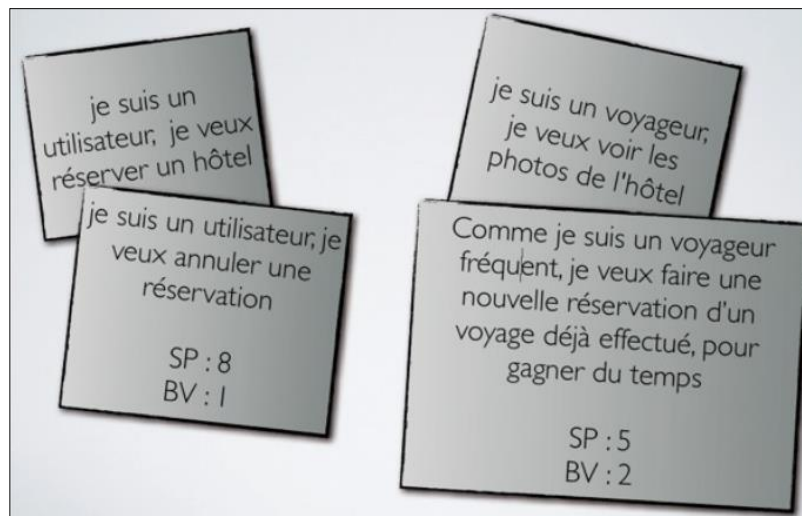
- Le travail n'est jamais assigné par un autre
- L'estimation du reste à faire est ajustée chaque jour
- Si une tâche n'est pas claire, ou trop volumineuse, la décomposer en tâches plus petites.

User stories (Histoires d'utilisateur)

- ❑ Agile est associé à l'expérience utilisateur
- ❑ On raconte une histoire : User stories
- ❑ On montre notre histoire : Prototype



19



20

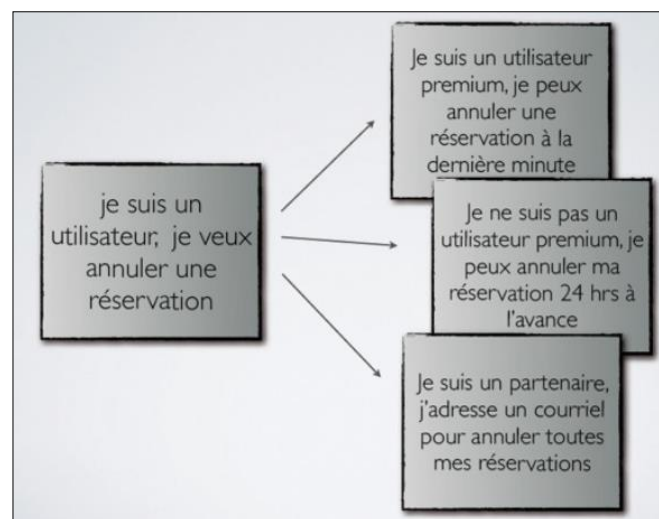
Détails

Je suis un utilisateur, je veux annuler une réservation

- ❑ Le client reçoit un remboursement complet ou partiel
 - Je rembourse directement sur son compte ou à l'intermédiaire
- ❑ Comment doit fonctionner l'annulation d'une réservation ?
 - C'est le même principe pour tous les hôtels ?
- ❑ Un voyageur fréquent peut-il annuler après sa réservation
 - Une confirmation est adressée à l'utilisateur ?
 - Comment ?

21

LES DÉTAILS SONT AJOUTÉS EN PETITES USER STORIES



22

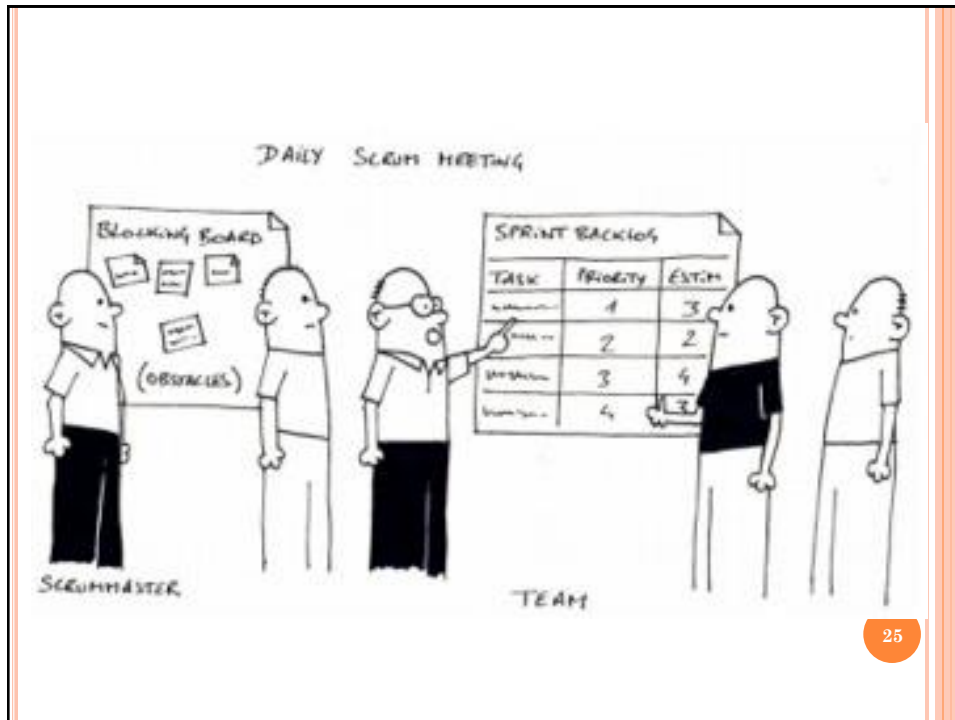
La méthode Scrum est aussi caractérisée par une « **mêlée** » quotidienne, encore appelée « **morning** » ou « **stand up** », dans laquelle les collaborateurs (chefs de projets, développeurs et responsables fonctionnels) indiquent tour à tour:

1. les tâches qu'ils ont effectuées la veille,
2. les difficultés rencontrées
3. et enfin ce sur quoi ils vont poursuivre leur travail le jour suivant.

23

- ❑ Tous les Jours
- ❑ 15 minutes (time boxed)
- ❑ Debout
- ❑ Pas fait pour résoudre les problèmes
- ❑ Tout le monde est invité
- ❑ Seuls les membres de l'équipe peuvent parler
- ❑ Permet d'éviter l'organisation d'autres réunion

24

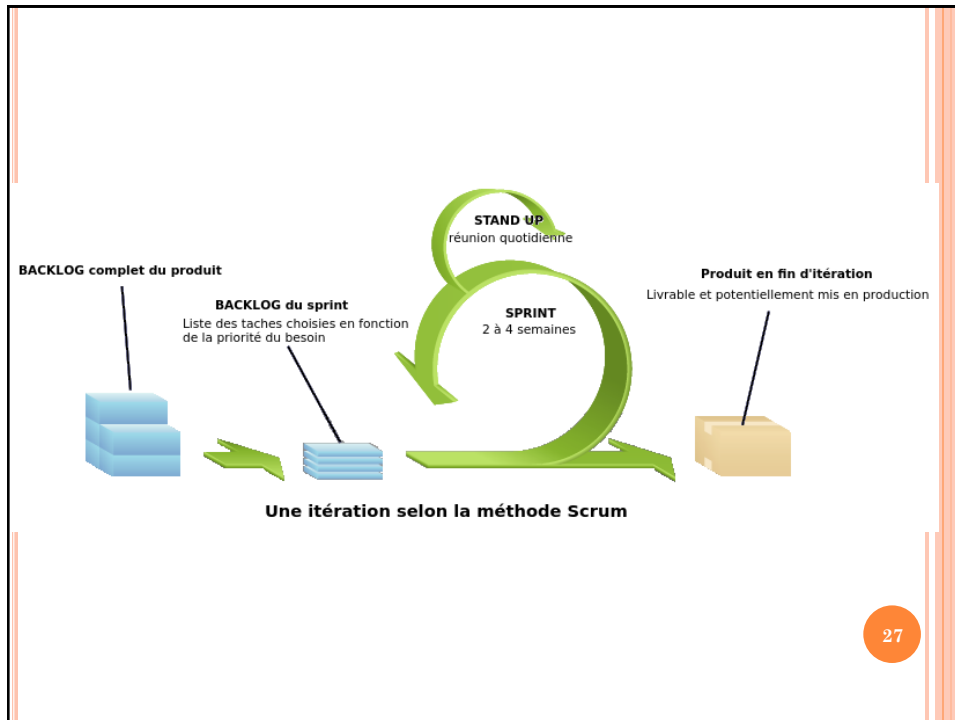


25

Cela permet :

1. d'évaluer l'avancement du projet,
2. de mobiliser des ressources là où cela est le plus nécessaire,
3. mais aussi de venir en aide aux collaborateurs rencontrant des difficultés lorsque celles-ci ont déjà été rencontrées auparavant par d'autres membres de l'équipe.

26



Les rôles de la méthode Scrum

La méthode Scrum définit trois rôles pour un projet.

1. **Le product owner** : il s'agit du représentant officiel du client au sein d'un projet Scrum.

Il est l'interlocuteur principal du Scrum Master et des membres de l'équipe. Il définit les besoins du produit et rédige les spécifications. Il peut se faire aider de responsables fonctionnels pour la rédaction des spécifications.

Il est également chargé de définir et prioriser les **users stories** pour chaque sprints.

28

Définit les fonctionnalités du produit

- ❑ Choisit la date et le contenu de la livraison
- ❑ Responsable du retour sur investissement
- ❑ Définit les priorités dans le backlog en fonction de la valeur « métier »
- ❑ Ajuste les fonctionnalités et les priorités à chaque sprint si nécessaire
- ❑ Accepte ou rejette les résultats

29

2. Le scrum master : il s'agit d'une personne chargée de veiller à la mise en application de la méthode et au respect de ses objectifs.

Il ne s'agit pas d'un chef de projet, mais d'une personne chargée de lever les obstacles éventuels qui empêcherait l'avancement de l'équipe et du projet pendant les différents sprints.

30

Représente le management du projet

- ❑ Responsable de faire appliquer par l'équipe les valeurs et les pratiques de Scrum
- ❑ Résout des problèmes
- ❑ S'assure que l'équipe est complètement fonctionnelle et productive
- ❑ Facilite une coopération poussée entre tous les rôles et fonctions
- ❑ Protège l'équipe des interférences extérieures

31

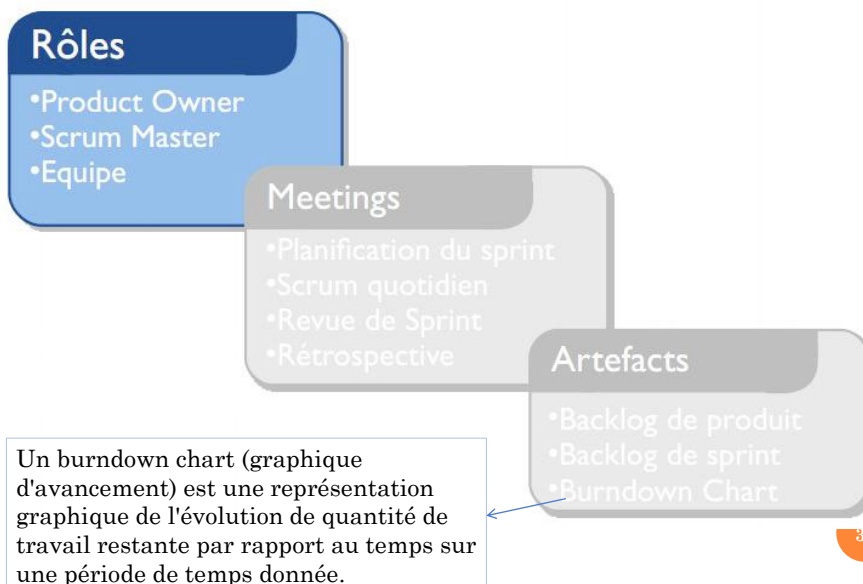
3.L'équipe (« team members ») : ce sont les personnes chargées de la réalisation du sprint et d'un produit utilisable en fin de chaque sprint.

Il peut s'agir de développeurs, testeurs, architectes...

32

- ❑ De 5 à 10 personnes
- ❑ Regroupant tous les rôles
- ❑ Architecte, concepteur, développeur, spécialiste IHM, testeur, etc.
- ❑ A plein temps sur le projet, de préférence exceptions possibles (administrateur, ...)
- ❑ L'équipe s'organise par elle-même
- ❑ La composition de l'équipe ne doit pas changer pendant un Sprint

33



34

Meetings : Rétrospective

- ❑ A la fin de chaque sprint
- ❑ Permet de réfléchir régulièrement à ce qui marche et ce qui ne marche pas.
- ❑ Dure en général de 15 à 30 minutes.
- ❑ Fait à la fin de chaque Sprint.
- ❑ Toute l'équipe participe.
 - Scrum Master
 - Product Owner
 - Équipe
 - Éventuellement d'autres intervenants

35

Pendant la phase d'analyse

Lors de cette phase, il faut se poser trois grandes questions.

Qui ? Qui vont être les acteurs du système ?

Quoi ? Que doit faire le système ?

C'est la phase de spécifications fonctionnelles. C'est dans cette partie que l'on va retrouver toutes les règles de gestion métiers et les cas d'utilisation UML de l'application qui définissent ces spécifications fonctionnelles.

36

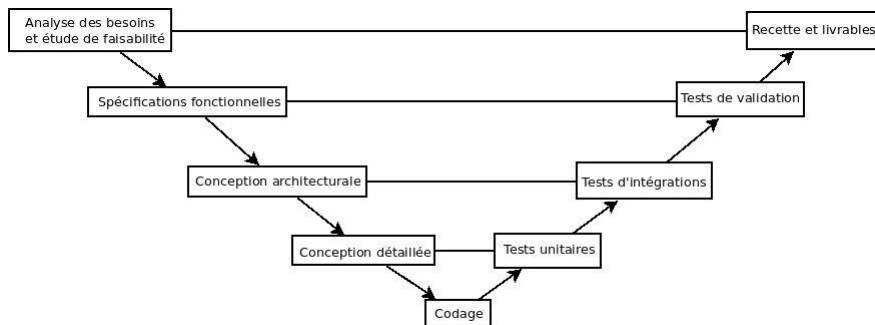
Comment ? Comment ce système va satisfaire les besoins des utilisateurs.

C'est à ce moment que l'on va définir l'architecture, la conception détaillée (diagrammes de classes, scénarios et diagrammes de séquences, modèles conceptuels de données...).

Dans le cas de projets développés à l'aide des méthodes agiles, ces questions se posent à chaque itération qui embarque un lot important d'évolutions.

37

Méthode classique en V



38

Le cycle en V est un des modèles les plus répandus au niveau de la gestion de projet. En observant ces différentes étapes, on peut tirer des méthodes agiles, les avantages suivants :

1 - Le « **delivery** » (livraison d'un produit utilisable) est plus rapide et le produit utilisable très tôt dans le cycle de développement.

Ceci est dû notamment à la plus grande participation du client (ou du destinataire de l'application lorsqu'il s'agit de développement en interne) qui permet de répondre au plus vite à ses attentes.

Mais aussi parce que l'on n'attend pas la réalisation de toutes les fonctionnalités désirées pour utiliser le produit.

39

2 - **Le client a les mains libres.** Une fois qu'il juge que le système possède l'essentiel des fonctionnalités nécessaires, il n'est plus obligé de continuer les itérations suivantes. C'est un gain en matière de coût et de satisfaction du client.

3 - **Les spécifications sont souples.** Lorsque la MOA (voir page suivante) s'aperçoit que certaines fonctionnalités ne répondent pas aux besoins, une fois les tests de validation ou de recettes réalisés, il suffira de prévoir de nouvelles tâches lors de la prochaine itération. Lorsque le projet est réalisé en suivant les phases du cycle en V, cela nécessite souvent de repartir au niveau des spécifications et de suivre à nouveau pas à pas toutes les étapes, ce qui peut prendre un certain temps et surtout un coût important.

40

La **maîtrise d'ouvrage (MOA)**, aussi dénommée **maître d'ouvrage** est l'entité porteuse du besoin, définissant l'objectif du projet, son calendrier et le budget consacré à ce projet.

Le résultat attendu du projet est la réalisation d'un produit, appelé ouvrage.

La maîtrise d'ouvrage maîtrise l'idée de base du projet et représente, à ce titre, les utilisateurs finaux à qui l'ouvrage est destiné.

41

Le terme **maîtrise d'œuvre** (souvent abrégé **MOE**) désigne l'entité retenue par le **maître d'ouvrage (MOA)** afin de réaliser le projet dans les conditions de délais, de qualité ainsi que de coûts fixés par ledit projet, le tout conformément à un contrat.

42