

Méthodologie agile



1

PRÉPARÉ ET PRÉSENTÉ PAR :
DR. LAKHROUIT JIHANE

Sommaire

- ▶ Scrum
 - ▶ Introduction
 - ▶ Les piliers de Scrum
 - ▶ Les rôles de Scrum
 - ▶ Les Artefacts de Scrum
 - ▶ Le cycle de vie de Scrum (Sprint, Backlog, User Story)
 - ▶ Les Événements de Scrum
- ▶ Tableaux Kanban
- ▶ XP
 - ▶ Les concepts XP
 - ▶ Cycle de Vie XP
- ▶ Scrum Vs XP
- ▶ Rappel & Conclusion

Scrum

Introduction

- ▶ Scrum est le fruit d'une réflexion en 1986 réalisé par Takeuchi et Nonaka.
- ▶ Scrum n'a pu être formalisé qu'en 1996 lors de l'ACM conférence (Object-Oriented Programming, Systems, Languages and Applications) avec la publication du premier article définissant cette méthode entre Ken Schwaber et Jeff Sutherland



Les trois piliers de Scrum:

- ❖ **La transparence:**

Scrum met l'accent sur le fait d'avoir un langage commun entre l'équipe et le management, Ce langage commun doit permettre à tout observateur d'obtenir rapidement une bonne compréhension du projet

- ❖ **L'inspection :**

Scrum propose de faire le point sur les différents artefacts produits, afin de détecter toute variation indésirable. Ces inspections ne doivent pas être faites trop fréquemment, ou par un inspecteur mal formé : cela nuirait à l'avancement du projet.

- ❖ **L'adaptation :**

le processus doit être adapté. Scrum fournit des rituels, durant lesquels cette adaptation est possible. Il s'agit de la *réunion de planification de sprint*, de la *mélée quotidienne*, de la *revue de sprint* ainsi que de la *rétrospective de sprint*.

Les Rôles de Scrum :

➤ **ScrumMaster:**

- ▶ Bonne communication.
- ▶ Savoir résoudre des conflits et des problèmes afin de ne pas gêner l'équipe par rapport aux éléments extérieurs.
- ▶ Savoir mener sans imposer ses idées puisque chaque décision relève de l'équipe complète.

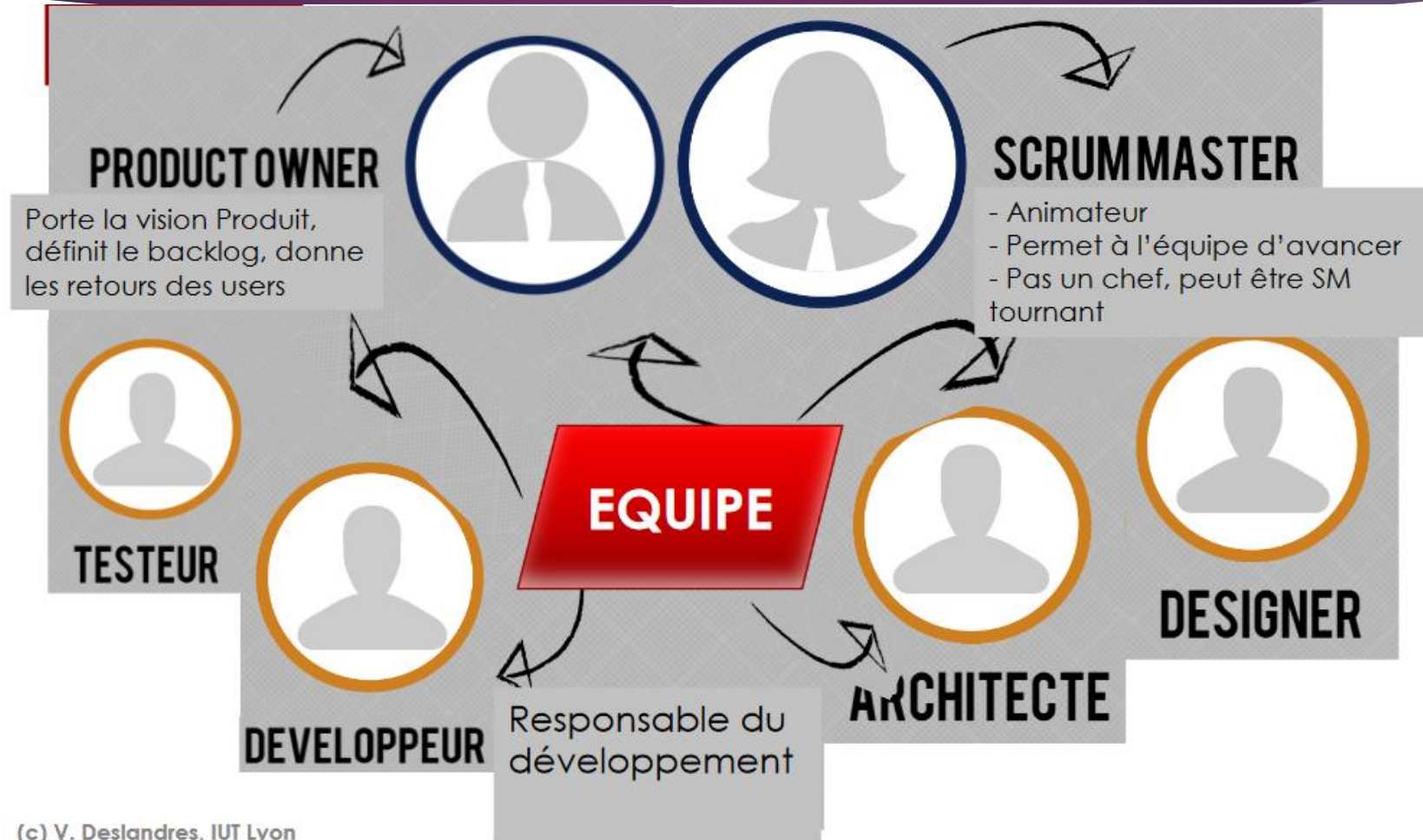
➤ **Directeur de produit ou Product owner :**

- ▶ Bonne communication.
- ▶ Savoir exprimer ses besoins sans ambiguïté

➤ **Equipe:**

- ▶ Savoir communiquer avec le reste de l'équipe et le directeur de produit.
- ▶ Ne pas s'imposer au sein de l'équipe. Tout le monde est au même niveau.

Les Rôles de Scrum :



Les Rôles de Scrum :

8

3 Rôles Scrum Team plus 3 Rôles organisationnels

► Product Owner

- • Il pilote le projet d'un point de vue métier
- • Il communique une vision claire du produit et définit ses caractéristiques
- • Il accepte ou rejette le produit à la fin de chaque Sprint
- • Il s'assure que l'Équipe se concentre sur les items du Backlog de plus forte valeur ajoutée
- • Il a le même objectif que l'Équipe
- • Il est responsable du Retour sur Investissement et des livraisons.
- Entretien le Product Backlog
- Établir et maintenir le Plan de Livraison

► Scrum Master

- • Protège l'équipe des turbulences
- Il n'est pas un membre de l'Équipe
- Il optimise la productivité de l'Équipe
- Il contrôle l'"Inspect-&-Adapt" de l'Équipe
- Il assure que les idéaux "agiles" soient bien compris et respectés par tous les participants au projet.
- Il n'est pas responsable des livrables.
- Il anime les réunions
- **Il est le gardien du processScrum**

► Scrum Team

- Elle délivre le produit et elle est responsable de sa qualité
- Elle travaille avec les utilisateurs-finaux, le client, le Product Owner pour comprendre les exigences-métier.
- Elle travaille continuellement avec le Product Owner pour définir la direction stratégique du Produit.

Les Rôles de Scrum :

9

3 Rôles Scrum Team plus 3 Rôles organisationnels

► Le Client

- Il commande le produit
- Il paye le développement du produit
- Il donne des feed-back et des révisions
- Il contracte l'organisation pour le développement de son produit
- Typiquement, il s'agit d'un responsable qui achète un développement de produit par un sous-traitant.

Dans les projets internes, il s'agit principalement du sponsor au projet, c'est à dire la personne validant le projet et le budget.

► Le Manager

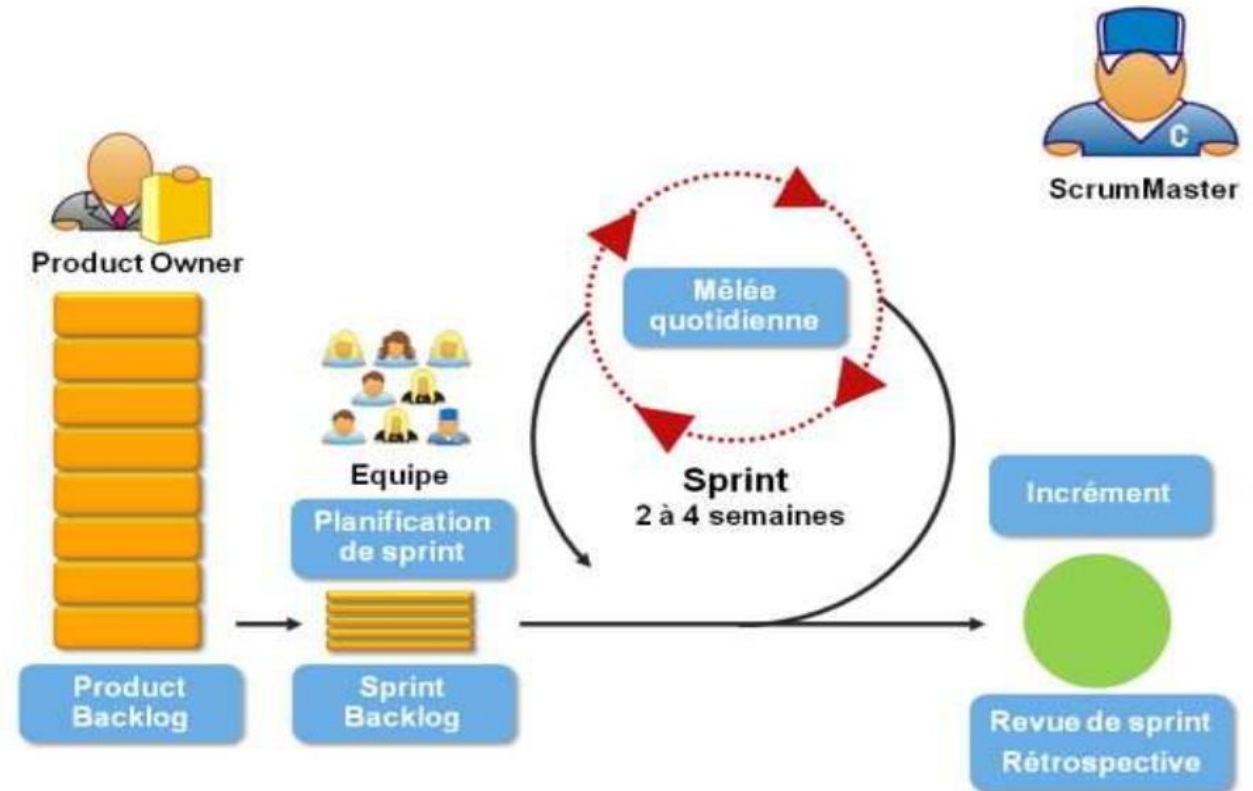
- Il s'assure que l'organisation puisse survivre en cas de défaillance.
- Il crée des règles et des lignes directrices.
- Le management, la gestion, est primordial dans tout projet Scrum. Il permet à l'Équipe de constituer un environnement optimal pour le déroulement du projet Scrum.
- Le manager donne de la structure et de la stabilité. Il travaille avec le ScrumMaster pour réorganiser l'organigramme de la structure et donner de la guidance si nécessaire

► L'Utilisateur Final

- Ce rôle peut être joué par un grand nombre de personnes.
- L'Utilisateur final est celui qui connaît les besoins et avec cette connaissance, il définit le produit en disant à l'équipe ce dont il a besoin comme fonctionnalités.
- Il connaît ses besoins et ses exigences
- Il donne son feed-back lors des revues
- Il participe au Sprint Planning 1

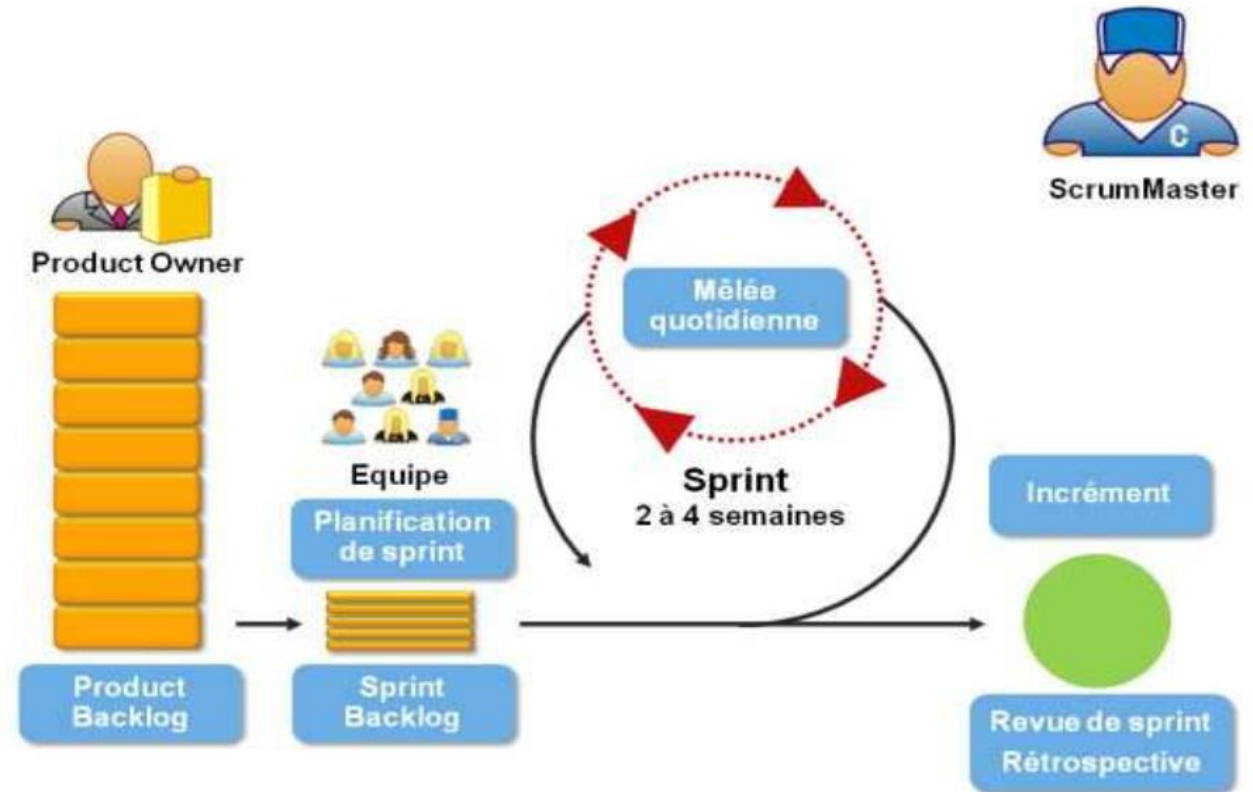
Cycle de vie de Scrum

- La méthode SCRUM implique que le projet progresse à travers la mise en place de séries de « sprints ».
- À chaque lancement d'un sprint, une réunion de planification est organisée afin que chaque membre de l'équipe puisse s'engager sur le nombre de tâches qu'il pourra exécuter, ainsi que sur la création du « sprint backlog », **qui est la liste globale des tâches à réaliser lors du sprint.**

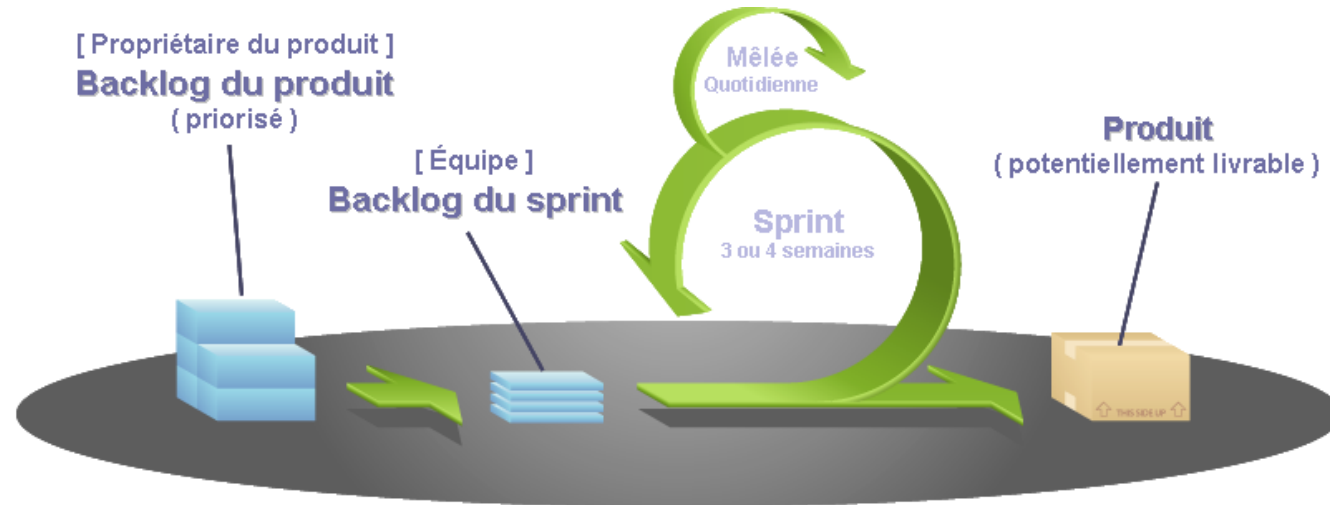


Cycle de vie de Scrum

- Chaque jour du sprint, tous les membres de l'équipe (ainsi que le responsable produit et le SCRUM Master) doivent assister à la réunion SCRUM quotidienne.
- La fin d'un sprint est marquée par une session de débriefing permettant de présenter le travail achevé au responsable produit, et de partager des informations pouvant influencer sur le sprint suivant. *Revue de Sprint*



Les Artefacts de Scrum



COPYRIGHT © 2005. MOUNTAIN GOAT SOFTWARE

Sprint

BackLog du
Produit

BackLog du
Sprint

Release
Burn down
Chart

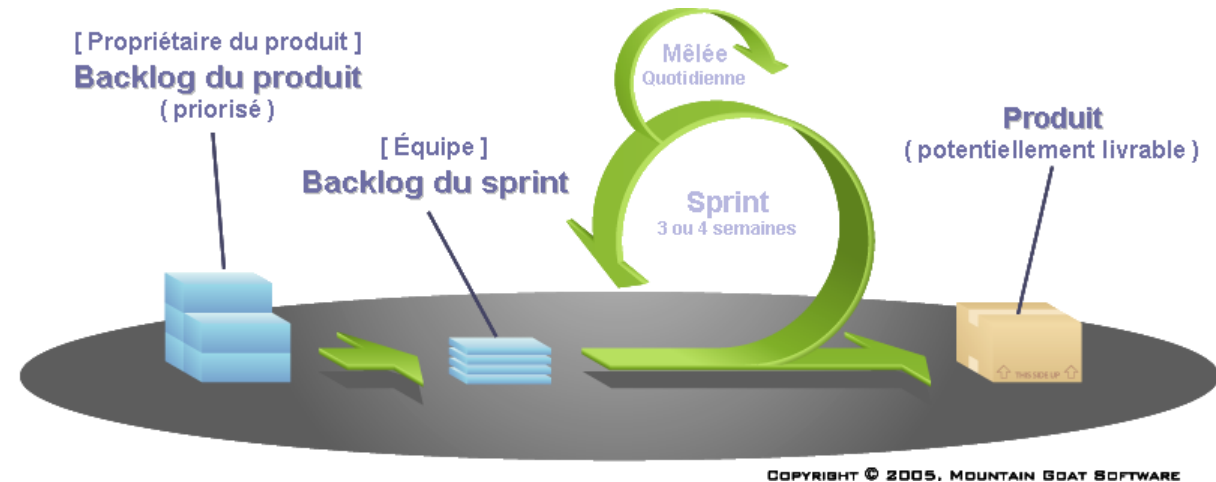
Cycle de vie de Scrum : Sprint

Le sprint est une période d'un mois au maximum, au bout de laquelle l'équipe délivre un incrément du produit, potentiellement livrable. Une fois la durée choisie, elle reste constante pendant toute la durée du développement. Un nouveau sprint démarre dès la fin du précédent.

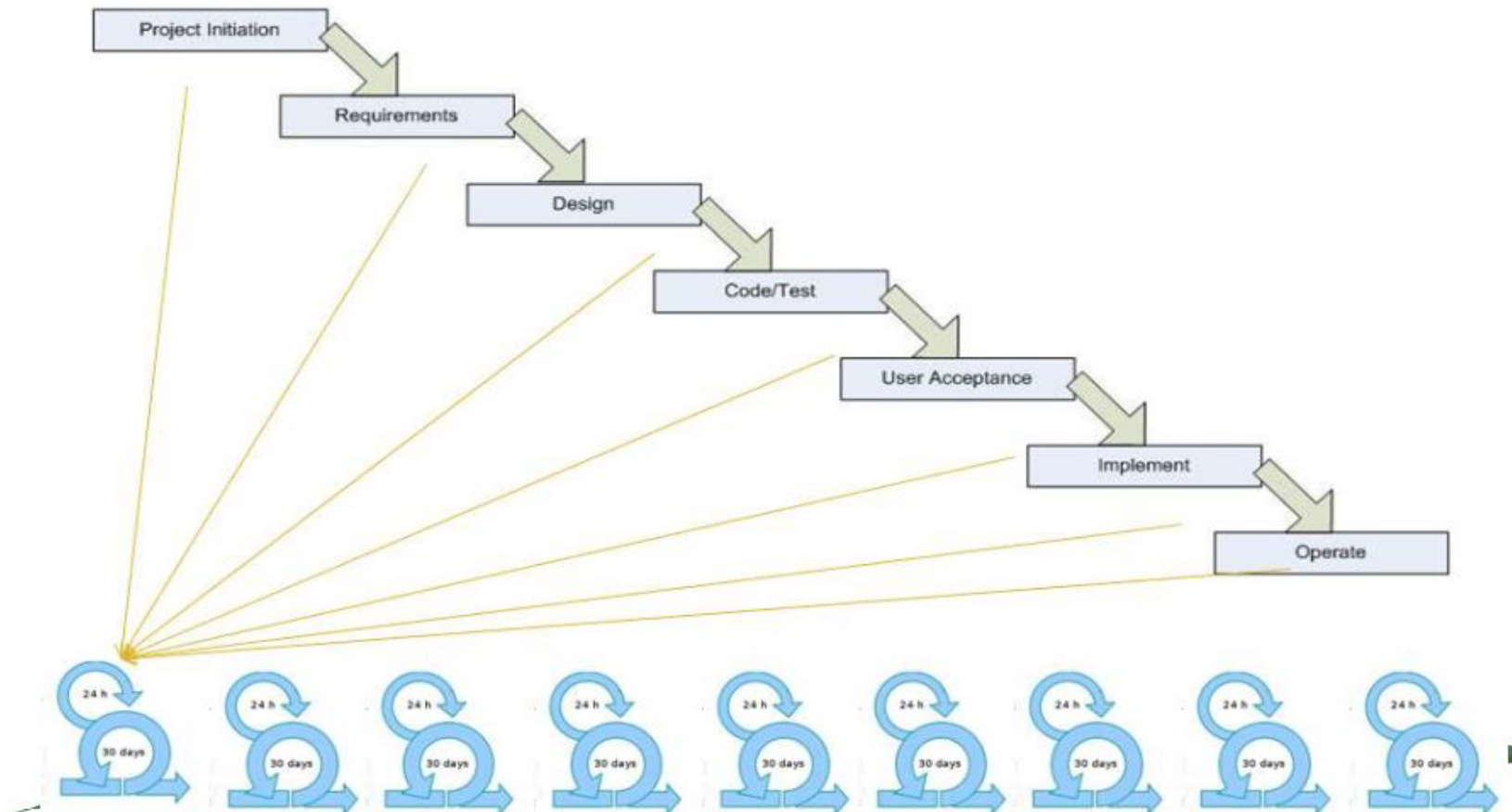
- ▶ le but du sprint ne peut être modifié ;
- ▶ la composition de l'équipe reste constante ;
- ▶ la qualité n'est pas négociable ;
- ▶ la liste d'items est sujette à négociations entre le *propriétaire du produit* et les *développeurs*.

Cycle de vie de Scrum : Sprint

- Sprint = itération
- Durée fixe (entre 2 et 4 semaines)
- Cette durée est en fonction du projet et de la maturité de l'équipe avec l'approche Scrum
- Sprint 0 (mise en place des technologies) et au dernier Sprint (livraison finale)
- Leur durée peut être différente de celle autres sprints



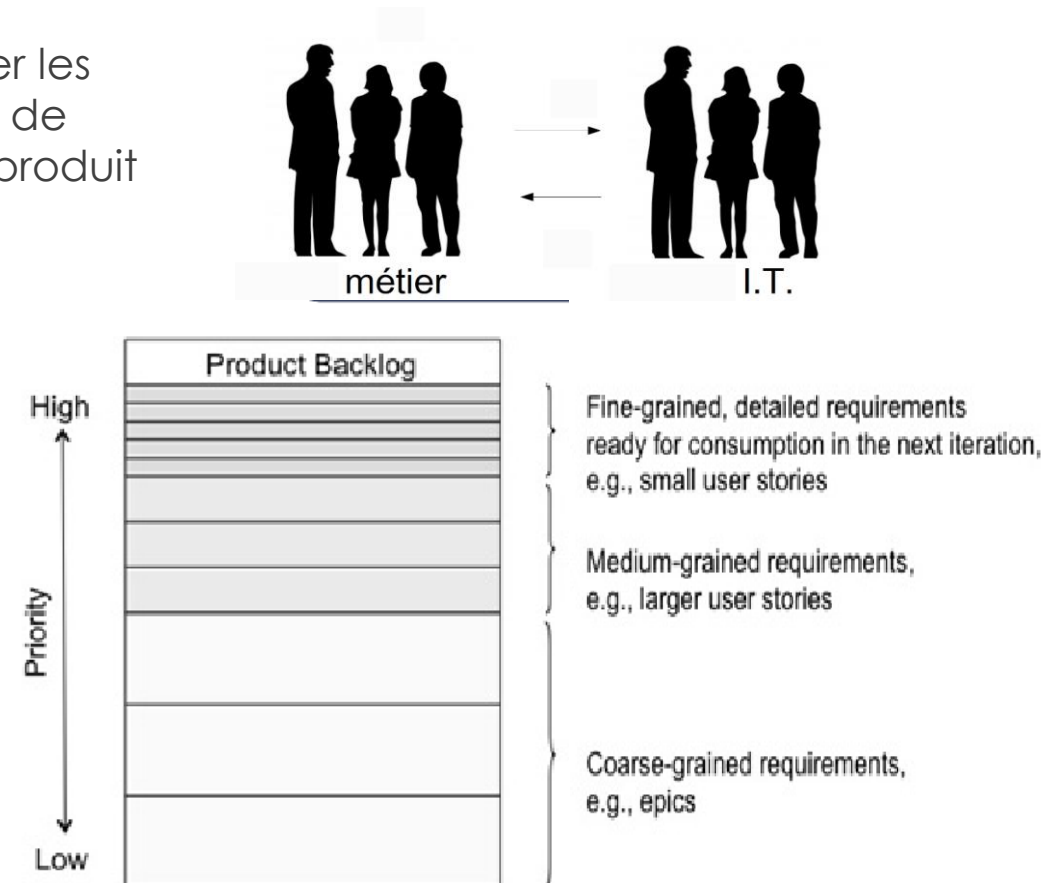
Scrum Vs Modèle en Cascade



Cycle de vie de Scrum : BackLog Produit

- ▶ Scrum utilise une approche fonctionnelle pour récolter les besoins des utilisateurs. L'objectif est d'établir une liste de fonctionnalités à réaliser, que l'on appelle carnet du produit BackLog Produit.
- ▶ Contient des User Stories (Scrum)
- ▶ DEF User Story
 - ▶ Une demande fonctionnelle d'un utilisateur
 - ▶ Apporte de la valeur business au produit
 - ▶ Ecrite en langage naturel

Le Product Backlog répond aux questions suivantes:
Quoi? Quand? Pour Qui?



Cycle de vie de Scrum : User Story

Format du User Story

► Format :

« En tant que client, je souhaite pouvoir ajouter un produit dans mon panier afin de pouvoir l'acheter »

En tant que... (rôle)
Je peux (tâche)
Afin de (but)

Combien d'User Stories ?

- Au max 20 US au début, pour respecter le vœux agile « peu de stock »
- Principe : « être précis à court terme, grossier à long terme »
- Permet d'estimer le « reste à faire »
- Ne pas les oublier plus tard
- Une US = un résumé formaté, pour avoir une vision rapide de la demande
- Elle sera détaillée ensuite avec l'équipe

Cycle de vie de Scrum : User Story

Granularité des User Stories ?

- ▶ Une user story doit pouvoir être implémentée en une itération (un sprint)
- ▶ Une itération doit comporter entre 4 et 6 User Stories

Comment obtenir les US ?

- ▶ C'est le problème du Product Owner
- ▶ Cela prend entre 2 jours et plusieurs semaines

En tant que... (rôle)
Je peux (tâche)
Afin de (but)

Cycle de vie de Scrum : User Story

20

Exemples d'User Stories ?

ETQ Assureur, je
peux récupérer des
contrats
d'assurance sur le
site web afin de
vérifier leur
précision et leur
légalité

ETQ Assureur, je peux
chercher les infos du
permis de conduire du
candidat chez la
préfecture concernée
afin de vérifier leur
exactitude

ETQ responsable
d'hotel je peux
ajouter de nouvelles
chambres.

ETQ responsable
d'hotel je peux
vérifier la
disponibilité des
chambres de mon
hotel pour une
période précise.

En tant que... (rôle)
Je peux (tâche)
Afin de (but)

Cycle de vie de Scrum : User Story

Règles pour écrire des US correctes

- ▶ REGLE 1 : **rester SIMPLE**
- ▶ REGLE 2 : parler du **QUOI** (**pas du COMMENT**)
- ▶ REGLE 3 : rester dans le périmètre du projet (vision!), et dans le champ de responsabilités de l'organisation / du service
- ▶ REGLE 4 : lever **l'ambiguïté des termes**
- ▶ REGLE 5 : **pratiquer si possible la réécriture des US**

En tant que... (rôle)
Je peux (tâche)
Afin de (but)

Cycle de vie de Scrum : User Story

Règles pour écrire des US correctes

En tant qu'internaute, je peux naviguer sur le site, saisir mes informations personnelles et celles du véhicule, et soumettre une demande en ligne, Afin d'obtenir une couverture d'assurance automobile

- ▶ Trop de 'choses à faire' mentionnées" pas simple
- ▶ REGLE : pas de User Story composée
- ▶ Éviter les ET (et OU)
 - ▶ pour 'je peux'
 - ▶ sur les données : données personnelles, données véhicules

En tant qu'internaute, je peux naviguer sur le site, Afin de choisir la couverture d'assurance automobile qui me convient

En tant qu'internaute, je peux saisir mes informations personnelles et celles du véhicule, Afin de comparer les offres d'assurance automobile

En tant qu'internaute, je peux soumettre une demande d'assurance automobile en ligne, Afin d'obtenir un contrat

Cycle de vie de Scrum : User Story

Lever l'ambiguïté

En tant que Resp. du Stock,
je peux commander la
bonne quantité de produits
que nous allons vendre

Afin d'éviter d'avoir des
coûts de stock trop élevés

Qu'est-ce qui est ambigu ici ?

- « la bonne quantité » : quelle valeur ? Quelle unité ? (produit unitaire, palette, chargement camion,...)
- « coûts trop élevés » : idem, subjectif --> valeur seuil
- « nous allons vendre » : quand ? demain, la semaine prochaine ?

Cycle de vie de Scrum : User Story

- ▶ BACKLOG de PRODUIT: Je retiens...
- ▶ Les Features (fonctionnalités) sont déclinées en User Stories, et les US en tâches
- ▶ Les US suivent le format: ETQ...(rôle) je peux...(action) afin de ...(but)

RAPPELEZ-VOUS

- Une user story doit être implémentée **sur une seule itération**
- Un sprint doit comporter **au moins 4 US**
- J'applique les principes **KISS** pour écrire les US
- La méthode de **Réécriture** permet de **valider** mes US

Cycle de vie de Scrum : User Story

► Granularité du besoin

■ Vocabulaire SCRUM

- « **Thème** » : domaine Produit, grosse Feature
- « **Feature** » : grosse fonctionnalité Produit
- « **Epic** » : grosse story
- « **User Story** » : besoin élémentaire d'un Utilisateur
- « **Task** » : tâche (technique) pour faire la US

Un backlog de produit

Élément de backlog	Estimation
Un invité peut faire une réservation	3
En tant qu'invité, j'annule une réservation	5
En tant qu'invité, je change les dates d'une réservation.	3
En tant qu'employé de l'hôtel, je produis les rapports de revenu par chambre	8
Améliorer la gestion des exceptions	8
...	30
...	50

Sprint Backlog

07

Tâches pour transformer le Product Backlog en fonctionnalité-produit

Tâches estimées en heures

Une tâche ne doit pas nécessiter plus d'un jour ou deux pour être Done.

Les grandes tâches sont découpées ultérieurement dans le Sprint

Les membres de la Dev. Team s'engagent sur les tâches une fois que le Sprint a démarré

Les tâches ne sont jamais assignées

Le Sprint Backlog est revu journallement

De nouvelles tâches sont identifiées, d'autres sont modifiées ou supprimées.

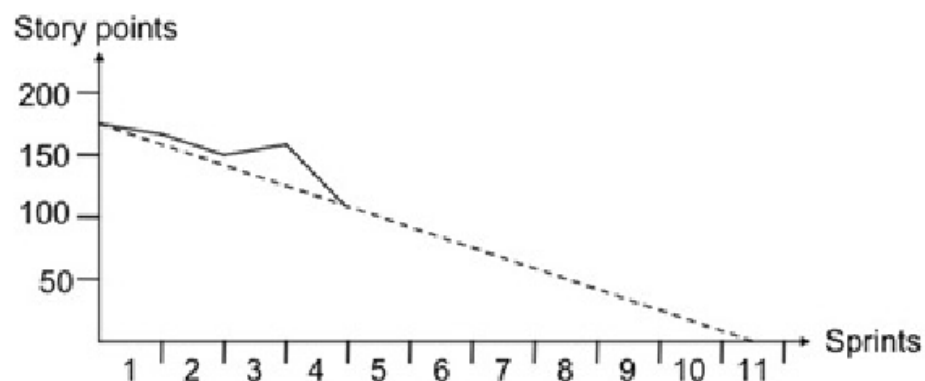
Les heures restantes de travail pour chaque tâche sont mises à jour

Les équipes sont mesurées en fonction de leurs engagements

Et pas sur le temps nécessaire pour les réaliser

Release Burn down Chart

Exemple de Burndown Chart



Le Release burndown rend les tendances des progrès visibles.

Le rapport est basé sur les informations suivantes:

- le reste-à-faire du Product Backlog pour transformer la Vision en un produit gagnant.
- Le nombre de Sprints nécessaires ou restants.
- La vélocité.

Le Release burndown regarde le passé pour comprendre ce que l'avenir est susceptible de détenir. Nous déterminons le taux d'avancement des sprints passés.

Sprint Burn-down Charts

- Le Sprint Burn-down chart montre
 - *combien d'efforts a été déployé en travaillant sur la tâche contenue dans le Sprint Backlog*
 - *Et compare cela à la dépense idéale*

Sprint Backlog for j_sprint (team: b-team) (displaying some of 2 items)

Start: 16.04.2009 End: 28.04.2009 Status:

Sprint Burndown Chart



- Le tableau donne une tendance qui indique si l'équipe est susceptible de respecter son engagement (indicateur avancé)

Événements:

Événements : la **mêlée** quotidienne

Daily Meetings

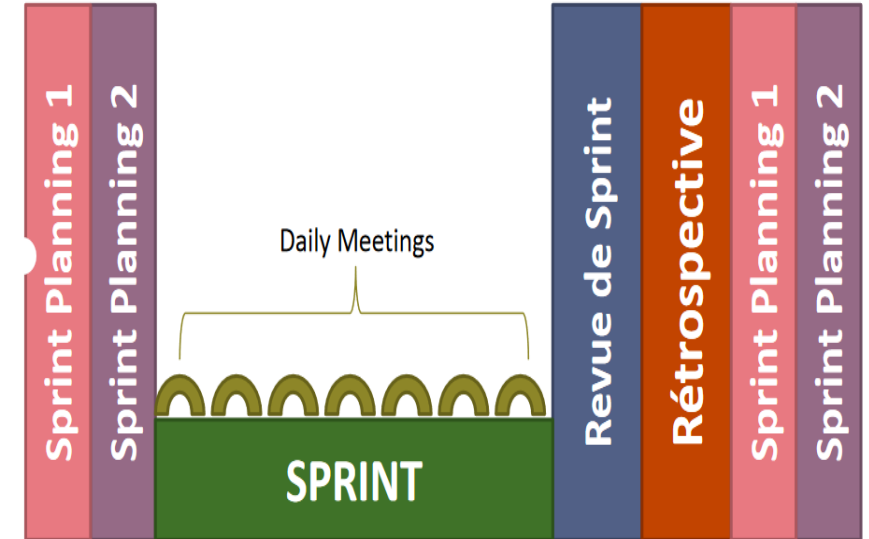
31

Au quotidien, une réunion, la **mêlée** quotidienne **Daily Meetings** permet aux développeurs de faire un point de coordination sur les tâches en cours et sur les difficultés rencontrées, Cette réunion dure 15 minutes au maximum. À tour de rôle, chaque membre répond à 3 questions :

Qu'ai-je fait hier ?

Que dois-je faire aujourd'hui ?

Quelles sont les difficultés rencontrées ?



Organisateur: l'équipe
Participants: l'équipe (actif), le ScrumMaster (passif), Product Owner (passif)
Durée: 15 min

Evènement : **Sprint Planning Meeting**

32

La réunion de planification (*Sprint Planning Meeting*) se passe en deux temps. Dans la première partie, l'équipe de développement cherche à prévoir ce qui sera développé durant le prochain sprint. A l'entrée de cette phase, l'équipe doit avoir à sa disposition le carnet du produit priorisé, l'incrément réalisé à la dernière itération, la capacité de production de l'équipe lors des dernières itérations, ainsi que la capacité de production prévue pour la prochaine itération. L'équipe et le propriétaire du produit cherchent alors à déterminer le but du sprint.

Dans un second temps, l'équipe se focalise sur la manière dont ses membres atteindront le but du sprint. Ils prévoient alors ce qui sera fait pendant le sprint, construisant ainsi une estimation. Cela constitue l'engagement de l'équipe

Sprint Planning Meeting

Sprint Planning Meeting

- Organisateur: Product Owner
- Participants: l'équipe (actif), le ScrumMaster (passif)
- Durée: 8 heures pour un Sprint de 4 semaines

- **2 PARTIES:**
 - Le QUOI?
 - Le COMMENT?
- **LE PRODUCT OWNER:**
 - *Présente le Product Backlog priorisé par le client et/ou les utilisateurs*
 - *Présente le Release Plan Initial*
 - *Présentation de la Vision*
- **L'ÉQUIPE:**
 - *Estime le Product Backlog en fonction de sa faisabilité (estimation fonctionnelle)*
 - *Découpe le Product Backlog en Sprint Backlogs avec le Product Owner*
 - *Découpe le Sprint Backlog en tâches*
 - *Estime le Sprint Backlog*
- **LE PRODUCT OWNER ET L'ÉQUIPE:**
 - *Définissent l'objectif du Sprint*
 - *Valident la Definition of Done*

Evènement : **Sprint Planning Meeting**

- ▶ Quelle interface devons-nous designer?
- ▶ Quelle architecture devons-nous créer?
- ▶ Quelles tables devons-nous actualiser?
- ▶ Quels composants devons-nous mettre à jour ou créer?

Evènement : La Revue de Sprint

35

À la fin du sprint, tout le monde se réunit pour effectuer la revue de sprint, qui dure au maximum 4 heures. L'objectif de la revue de sprint est de valider le logiciel qui a été produit pendant le sprint.

- ▶ Présentation (par l'équipe)
- ▶ Feedback (par l'utilisateur final)
- ▶ C'est l'inspect-&-adapt de l'utilisateur permettant la création ou le changement des items du Product Backlog

- C'est l'inspect-and-adapt des utilisateurs, du client et du management
- L'équipe présente les résultats du Sprint
- Utilisateurs/Client/Management expriment leurs remarques et trouvent un compromis avec l'équipe
- Le Product Owner valide ou rejète les items du Sprint Backlog en fonction de la Definition of Done
- C'est le Product Owner qui a toujours le dernier mot...

Evènement : La Revue de Sprint

36

Organisateur: Product owner

- Participants: l'équipe(actif), le ScrumMaster (passif), le Management(actif), le client (actif),
- les utilisateurs (actifs)
- Durée: 4 heures pour un Sprint de 4 semaines

- C'est l'inspect-and-adapt des utilisateurs, du client et du management
- L'équipe présente les résultats du Sprint
- Utilisateurs/Client/Management expriment leurs remarques et trouvent un compromis avec l'équipe
- Le Product Owner valide ou rejète les items du Sprint Backlog en fonction de la Definition of Done
- C'est le Product Owner qui a toujours le dernier mot...

La rétrospective du sprint

La rétrospective du sprint est faite en interne à l'équipe (incluant le ScrumMaster). Elle dure 3 heures pour un sprint d'un mois, et réduit selon la durée du sprint.

L'objectif est de comprendre ce qui n'a pas bien marché dans le sprint, les erreurs commises et de prendre des décisions pour s'améliorer. Il est tout à fait possible d'apporter des aménagements à la méthode Scrum dans le but de s'améliorer. Il faut être très vigilant à ne pas retomber dans les pratiques rigides des méthodes plus classiques.

- Organisateur: ScrumMaster
- Participants: l'équipe (actif), le ScrumMaster (actif),
- le Product Owner (actif en sa qualité de membre de l'équipe)
- Durée: 3 heures pour un Sprint de 4 semaines

La rétrospective du sprint

Nous faisons un point après l'action en nous posons deux questions:

- Qu'est-ce qui a bien fonctionné?
- Que devons-nous améliorer?
- Objectifs:
 - Apprendre du passé pour préparer l'avenir
 - Améliorer la productivité de l'équipe

- Analyse du Process Scrum:
 - *Comment cela c'est passé pendant le Sprint*
 - *Comment s'améliorer*
- Points principaux de vérification:
 - *La communication dans l'équipe*
 - *Les relations entre les membres de l'équipe*
 - *Les process et les outils*
 - *Les besoins en formation*

Le Jargon

39

Sprint:	Est une itération
Backlog:	Est une liste de tâches ouvertes
Product Backlog:	Est une liste d'items ouverts pour livrer le produit
Sprint Backlog:	Est une liste de tâches ouvertes attribuées au Sprint
L'EQUIPE ou la Development TEAM:	C'est l'équipe de développement
La Scrum Team:	C'est l' EQUIPE + le ScrumMaster + le Product Owner
Estimation Meeting	C'est la réunion d'estimation
Sprint Planning Meeting	C'est la réunion de planification de Sprint
Daily Scrum ou Stand-up Meeting	C'est la réunion journalière de 15' où l'EQUIPE inspecte et adapte, coordonne son effort.
Sprint Review ou Revue de Sprint	C'est la réunion de fin de Sprint où tous les acteurs du projet se retrouvent pour inspecter les livrables du Sprint.
Rétrospective	C'est la réunion d'inspection et d'adaption de la Scrum Team.

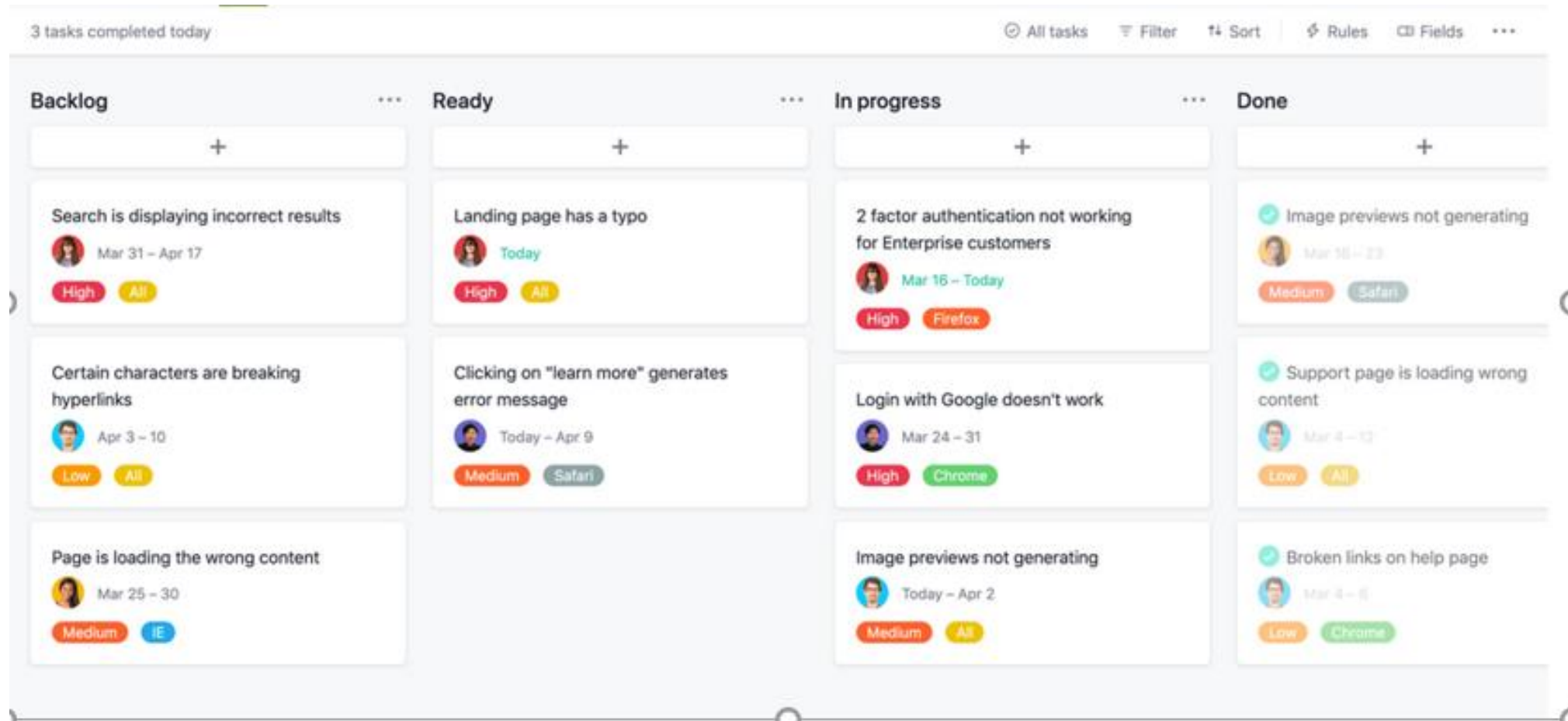
Méthode Kanban

- ▶ De nos jours, les tableaux Kanban prennent généralement la forme de tableaux virtuels composés de colonnes qui représentent les diverses étapes de travail. Ceci étant, il est tout aussi possible de dessiner un tableau Kanban sur un tableau blanc et d'utiliser des post-its ! Chacune des cartes du tableau représente une tâche qui avance d'une étape à l'autre jusqu'à son accomplissement.
- ▶ En règle générale, les équipes qui adoptent la méthode Kanban collaborent sur un tableau unique, bien que les tâches soient habituellement attribuées à leurs différents membres.

Les pratiques centrales de la méthode Kanban

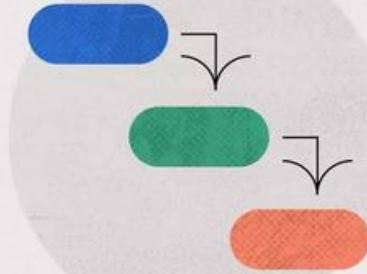
41

- L'un des nombreux avantages de la méthode Kanban consiste en la possibilité « physique » de visualiser la progression du travail au fur et à mesure des étapes.
- Une carte, qui correspond à une tâche, commence son parcours du côté gauche du tableau Kanban et, à mesure du travail mené par votre équipe, se déplace pas à pas jusqu'à atterrir dans la colonne **Terminé**. Non seulement cette façon de procéder vous donne un bon aperçu de la manière dont les tâches progressent d'étape en étape, mais elle vous permet de savoir où vous en êtes, en un seul coup d'œil et en temps réel.



- ▶ Dans un tableau Kanban, les colonnes représentent les différentes étapes par où passe le travail. Les colonnes que vous choisirez de créer seront fonction de votre équipe, mais voici quelques exemples de colonnes très souvent utilisées :
- ▶ **Backlog, Boîte de réception** ou **Nouveau** : voici la colonne qui accueillera tout nouveau travail, avant son attribution à quelconque membre de l'équipe.
- ▶ **Prêt** ou **Prioritaire** : déplacez les tâches vers cette colonne lorsqu'elles sont prêtes à être lancées.
- ▶ **En cours** : il s'agit de l'ensemble des tâches en cours d'exécution. Selon les besoins propres à votre équipe, peut-être souhaitez-vous décomposer cette colonne en diverses sous-colonnes. À titre d'exemple, une équipe de création de contenu pourrait choisir des colonnes relatives aux phases d'**Ébauche**, de **Révision** et de **Modification**, alors qu'une équipe d'ingénierie préférera des colonnes intitulées **Développement**, **Test** et **Déploiement**.
- ▶ **En attente** : déplacez une tâche vers cette colonne si un quelconque élément empêche sa progression.
- ▶ **Terminé** : comme son nom l'indique, les tâches atterrissent dans cette colonne une fois terminées.

Après avoir décrit les caractéristiques de chacune des méthodologies et structures de travail énoncées, nous allons désormais pouvoir les comparer pour vous aider à choisir celle qui conviendra le mieux à votre équipe et lui permettra d'atteindre ses objectifs.



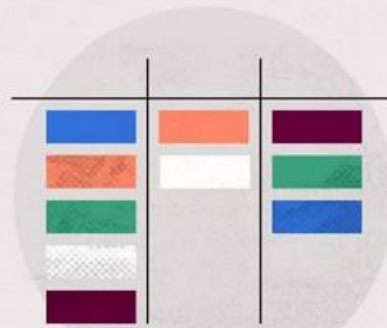
Waterfall

Le **modèle Waterfall** divise chaque projet en plusieurs phases et les **organise selon un ordre séquentiel**.



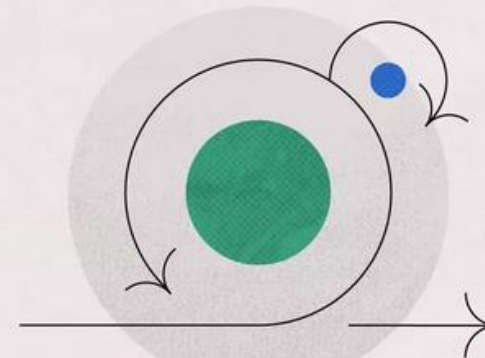
Agile

La **méthodologie Agile** est une méthodologie itérative qui divise les projets en plusieurs phases dynamiques, couramment appelées « sprints ».



Kanban

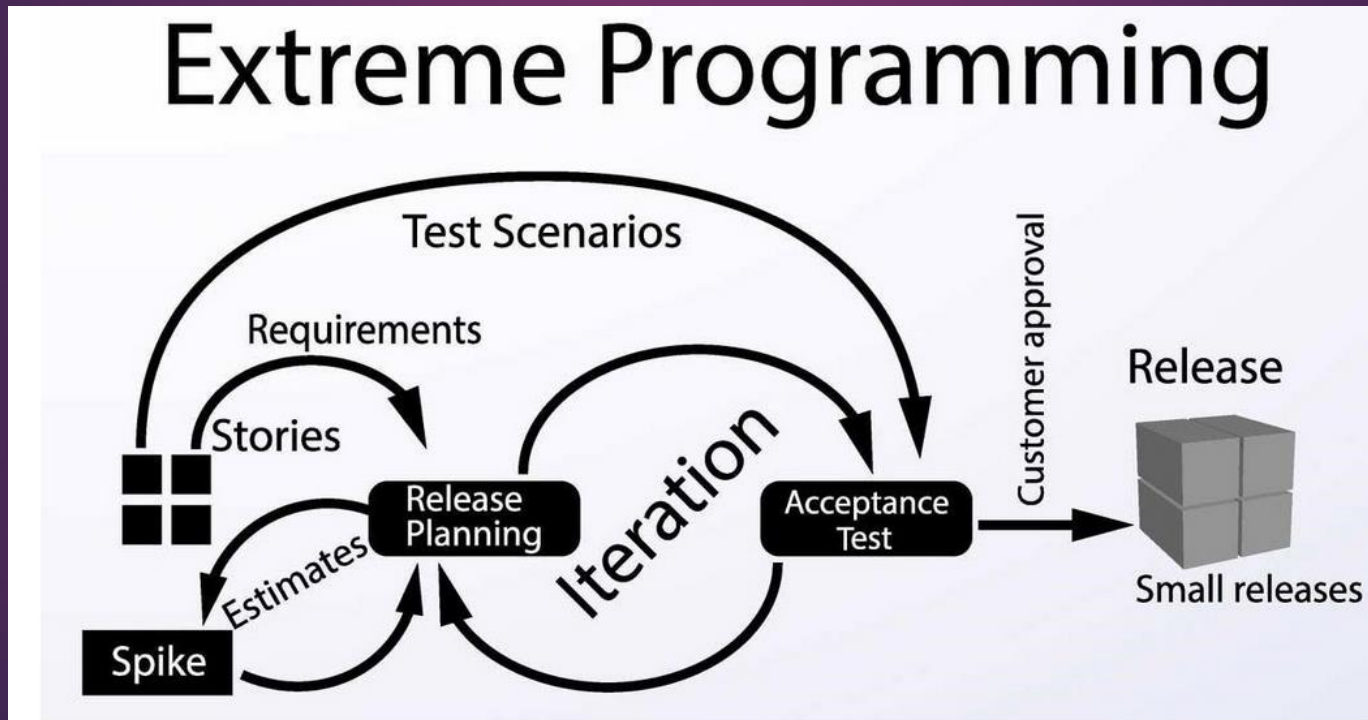
Le Kanban fait référence à une **stratégie visuelle de gestion de projet** qui vise à présenter les activités sous forme de tableau.



Scrum

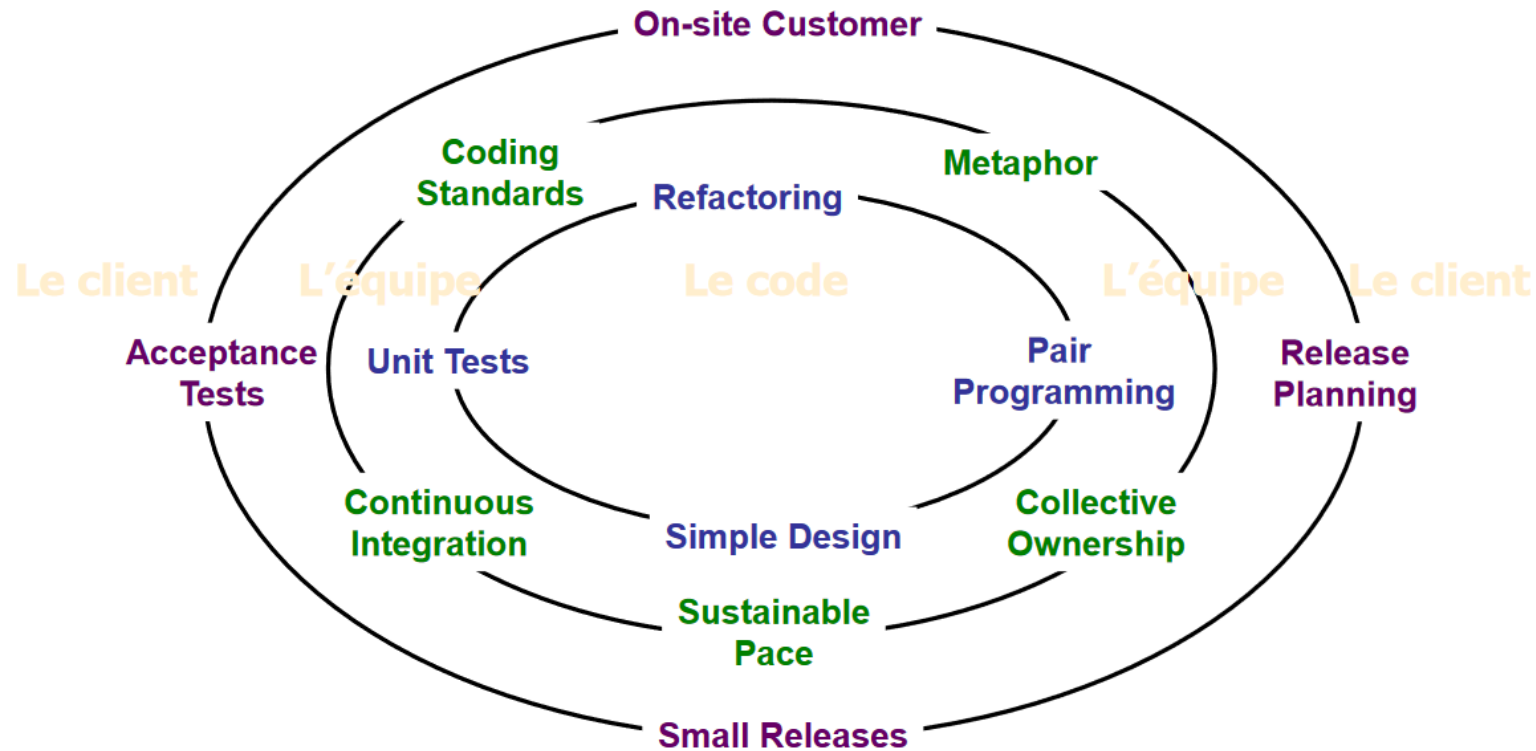
Scrum est une **structure Agile** qui facilite la collaboration au sein des équipes et les aide à réaliser des tâches à haute valeur ajoutée.

XP Extreme programming



Les « cercles de vie » en XP

45



Les « cercles de vie » en XP

46

- ▶ Client sur site
- ▶ Définit les besoins, les fonctionnalités, définit les priorités et répond aux questions des développeurs
- ▶ Une interaction quotidienne de vis-à-vis
- ▶ Réduit la quantité de documentation écrite
- ▶ Et le coût élevé de sa création et de sa maintenance
- ▶ Planification de Release
- ▶ Où « jeu du planning »
- ▶ Le « client » XP doit définir la valeur métier des fonctionnalités
- ▶ User Stories
 - ▶ Les développeurs (pas seulement un CP) estiment les fonctionnalités
 - ▶ A partir de ces informations, le client et les développeurs effectuent une sélection en fonction du ratio coût / bénéfice
 - ▶ Permet une priorisation optimale des fonctionnalités
- ▶ Livraisons courtes
- ▶ Mettre en production un système simple le plus rapidement possible puis appliquer des cycles de livraisons très courts
- ▶ Exemple
 - ▶ Releaser tous les 2 ou 3 mois
 - ▶ Chaque release contient plusieurs itérations
 - ▶ Établir un « rythme »
 - ▶ Feedback régulier pour le client et l'équipe
 - ▶ Permet d'évaluer la véritable valeur du produit dans son environnement réel

Les « cercles de vie » en XP

47

► TDD

- TESTS D'ACCEPTANCE : on demande aux clients de fournir des tests d'acceptance avant les développements (idéalement automatisés)
- TESTS UNITAIRES : les développeurs écrivent d'abord des tests puis créent le logiciel qui répond aux exigences capturées dans les tests
- Un développement piloté par les exigences garantit que les fonctionnalités essentielles seront fournies

► Refactoring

- Les développeurs restructurent le système sans en modifier le comportement pour supprimer les duplications, faciliter la communication, simplifier ou améliorer la flexibilité
- Petites étapes
 - Coder, tester, refactorer, tester, coder, refactorer

► Programmation en binôme

- Le code est écrit par 2 développeurs sur 1 seule machine L'un la tactique, l'autre la stratégie
- Le binôme devrait être « dynamique »
- Les membres en binôme changent de rôle toutes les 30 à 60 minutes Et sur tous les types de tâches
- L'expérimentation montre une réelle efficacité
- Les bénéfices
 - Revue de code continue
 - Partage continu du métier
- Amélioration continue des compétences techniques (Java, Design Patterns, Refactoring, IDE...)
- Les développeurs ont parfois plus de mal avec cette pratique que les managers
- Tester pour voir si cela fonctionne
- Expérience du développement plus intense

XP Extreme programming

48

► Standards de développement

- Les développeurs produisent du code en conformité avec des règles favorisant la communication au niveau du code
- L'objectif : produire un code auto documenté

► Métaphore

- La vision de « l'architecture » selon XP
- Guide les développements en fournissant une vision commune et unique de la façon dont le système fonctionne
- Définit un vocabulaire et guide l'équipe dans les développements et la communication

► Intégration Continue

- Intégrer et construire le système aussi souvent que possible
- Intégrer à chaque fois qu'une tâche est finie
- Permet de connaître à tout instant le « statut » du système

XP Extreme programming

► Propriété collective

- Tout développeur peut modifier toute portion de code à n'importe quel moment
- Ceci est facilité par l'utilisation de Standards de développement, de TDD et de Pair Programming (Synergie)
- Sécurise l'équipe en terme de vacances, congés, maladie, turn-over
- Les progrès ne s'arrêtent pas sur un composant parce qu'un membre de l'équipe est absent

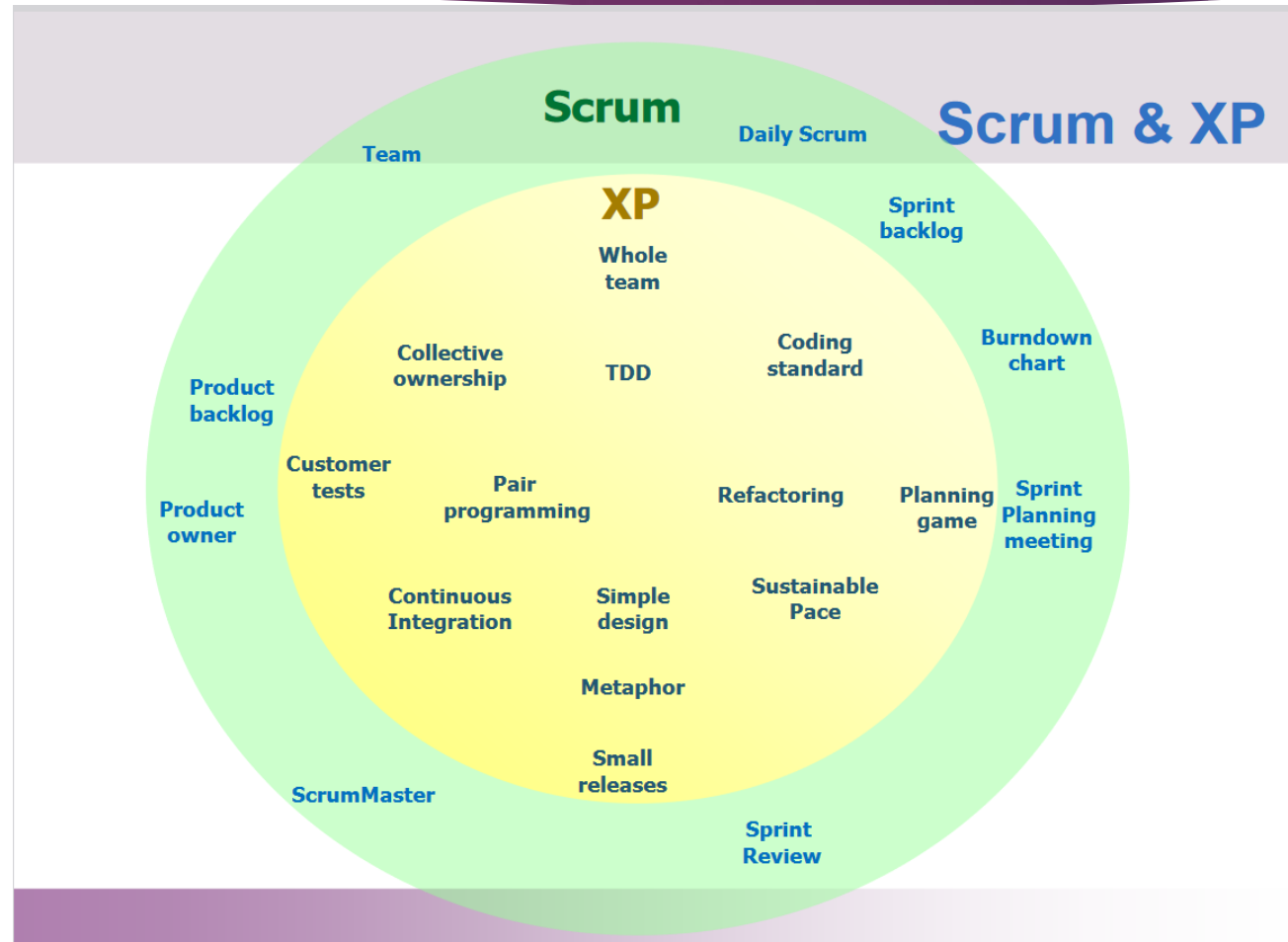
Cycle de vie en XP

1. Le client écrit ses besoins sous forme de scénarios.
2. Les développeurs évaluent le coût de chaque scénario, en collaboration avec le client.
3. Le client choisit les scénarios à intégrer à la prochaine livraison.
4. Chaque développeur prend la responsabilité d'une tâche pour la réalisation d'un scénario.
5. Le développeur choisit un partenaire.
6. Le binôme écrit les tests unitaires correspondant au scénario à implémenter.
7. Le binôme prépare l'implémentation en réorganisant le code existant, puis il procède à l'implémentation proprement dite.
8. Le binôme intègre ses développements à la version d'intégration.

Planification collective

- Le projet est décomposée en une suite de « tours » appelés **itérations** (durée de 1 à 3 semaines selon les projets) et de **livraisons** au rythme d'une livraison toutes les 2 ou 3 itérations en général.
- Le **cycle des livraisons** porte sur la planification des fonctionnalités décrites sous forme de scénarios client (*user stories*).
- Le **cycle des itérations** porte sur la planification des activités (tâches) menées par les développeurs.
- Les **tâches** sont donc les activités des développeurs qui permettent de réaliser un scénario client.

Scrum Vs XP



Signification d'Agile en gestion de projet

- Alors que les méthodes traditionnelles visent à traiter les différentes phases d'un projet d'une manière séquentielle (que l'on nomme aussi cycle de développement en cascade ou encore cycle en V), le principe des méthodes Agiles est de le découper en sous-parties (ou sous-projets) autonomes (on parle également de développement itératif).
- Les parties (itérations) forment le projet dans sa globalité.

Manifeste Agile, Les principes fondateurs

- Le **Manifeste Agile** est une déclaration rédigée par des experts en 2001 pour améliorer le développement de logiciels.
- Les 4 valeurs agiles :



- Le Manifeste définit 12 principes :

- 1 - La priorité n°1 est d'obtenir la satisfaction client au plus tôt par la livraison rapide et régulière de fonctionnalités attendues.
- 2 - Accepter les demandes de changement en cours de projet . Ce sont des opportunités pour donner plus de valeur au projet et coller aux vrais besoins des clients.
- 3 - Mettre en œuvre des livraisons rapides reposant sur des cycles courts (quelques semaines). Ces livrables doivent être opérationnels pour permettre des tests de validation des fonctionnalités attendues.

- 4 - Coopération forte et continue entre les utilisateurs et le développement. A l'inverse des méthodes classiques où les rencontres entre les utilisateurs et la maîtrise d'œuvre interviennent surtout en début et en fin de projet.
- 5 - Donner de l'autonomie à des personnes impliquées et leur faire confiance.
- 6 - Privilégier le face à face comme canal de communication entre les parties. Les interactions sont plus efficaces et plus riches. Tout va plus vite.
- 7 - L'important est d'avoir une application opérationnelle.
- 8 - Avancer avec un rythme constant compatible avec ce que peut produire l'ensemble des acteurs.
- 9 - Focus sur la qualité technique et la qualité de conception pour construire une base solide renforçant l'agilité.
- 10 - Rester simple dans les méthodes de travail : ne faire que ce qui est nécessaire.
- 11 - Une équipe qui s'organise elle-même produit de meilleurs résultats.
- 12 - En revoyant régulièrement ses pratiques, l'équipe adapte son comportement et ses outils pour être plus efficace.

Méthodes Agiles

- La méthodologie Agile se base sur une idée simple. Planifier la totalité de votre projet dans les moindres détails avant de le développer est contre-productif.
- Vous perdez du temps si vous organisez tous les aspects de votre projet en amont. Il est effectivement rare que tout se passe exactement comme prévu. Souvent, des aléas surviennent et vous forcent à revoir votre planification.
- La méthode Agile recommande de se fixer des objectifs à court terme. Le projet est donc divisé en plusieurs sous-projets. Une fois l'objectif atteint, on passe au suivant, et ce jusqu'à l'accomplissement de l'objectif final. Cette approche est plus flexible. Puisqu'il est impossible de tout prévoir et de tout anticiper, elle laisse la place aux imprévus et aux changements.

Quelles sont les principales méthodes Agile ?

- Selon la méthode Agile à laquelle on se réfère, la démarche peut prendre différentes formes, et revêtir un vocabulaire spécifique.
 - **La méthode Scrum et son fonctionnement en sprints**
 - La plus célèbre des méthodologies de gestion de projets déclinées de la méthode Agile relève de la “**Scrum**”, autrement dit la “**mêlée**” dans le langage rugby. Le responsable de projet s’appelle ainsi le “**SCRUM Master**”.
 - Cette approche s’organise autour de cycles courts, qu’on appelle communément des itérations. En langage **Scrum**, une itération se nomme un “**sprint**”. À chaque nouveau sprint, l’équipe projet se rassemble pour lister les tâches à exécuter. Cette liste s’appelle le “**sprint backlog**”.
 - L’ensemble relève d’une logique de développement produit. C’est ce qui explique que la méthodologie **Scrum** se déploie autour d’acteurs spécifiques, comme le **Product Owner**. Des réunions **Scrum** ont d’ailleurs lieu quotidiennement. Il s’agit de courtes périodes d’échange, pendant lesquelles les membres de l’équipe projet communiquent sur leurs avancées et leurs difficultés.
 - **Les autres méthodologies d’inspiration Agile**
 - Si **Scrum** reste la méthode Agile la plus utilisée, elle entre en compétition avec la méthode **Kanban** pour ce qui relève du pilotage de projets dit “**mono-équipe**”.
 - L’approche **Kanban** trouve son origine dans le mot japonais pour “**panneau**”. Elle nous vient des procédures de production de Toyota, appliquées à l’univers de la programmation logiciel. Cette approche consiste à croiser des tâches avec leurs états d’avancement, au sein d’une matrice en colonnes.
 - Le “**Lean Development**” est une méthode proche de **Kanban**. Il s’en différencie seulement par deux objectifs : améliorer les apprentissages des participants et éviter le gaspillage de ressources.

Conclusion

56

- ▶ De façon générale, l'on peut affirmer que le cycle en V se focalise sur le processus, tandis que les méthodes agiles privilégient le produit.
- ▶ Dans le cadre des méthodes agiles (Scrum, XP, RAD, ...), le projet s'affine par itérations, à travers la répétition d'un cycle d'opérations (le sprint dans le cadre de la méthode Scrum). Comme nous l'avons vu, le cycle en V définit l'intégralité du produit final dès les premières étapes, et ne laisse que peu de place à l'adaptation dans la suite du cycle.
- ▶ Les méthodes agiles permettent d'élaborer le produit par incrémentation. On produit un peu plus à chaque fois, morceau par morceau, pour aboutir au résultat final. Le cycle en V concentre au contraire la réalisation de l'ensemble dans une seule phase, qui est intégralement conçue en amont et vérifiée en aval.
- ▶ Ce manque d'adaptation et de flexibilité du cycle en V a précisément conduit à l'émergence des méthodes agiles, en particulier dans le domaine du logiciel et du marketing, pour répondre aux changements de plus en plus rapides des technologies et des demandes des consommateurs.