



Smart It Partner
We Build Smart Solutions For You

Framework Scrum

Préparation à la Certification

Dr. Ing Mohamed Amine MEZGHICH

Maître assistant en informatique à l'ENSI

Consultant en IT, Coach Agile & Devops

Contact: amine.mezghich@gmail.com



17/05/2024

COURTE BIOGRAPHIE

Mohamed Amine MEZGHICH est né en 10 Juillet 1985, à Bizerte Tunisie, Disposant d'une thèse de Doctorat, d'un master de recherche et d'un diplôme d'ingénieur obtenus respectivement en 2014, 2010 et 2009,

- ☐ Occupant actuellement le poste de maître assistant à l'ENSI (Ecole Nationale des Science l'Informatique, La Manouba, Tunisie depuis 2014.
- ☐ Chercheur senior en computer vision et data science, membre du Laboratoire Cristal, Pôle GRIFT de l'ENSI
- ☐ Fondateur de la boîte de développement Smart IT Partner depuis Juillet 2018
- ☐ Consultant Sénior Certifié en Java, Web, Scrum Master, Product Owner, Devops, Huawei AI.....
- ☐ Références : Organismes de Formations en France [PLB, ORSYS, Global Knowledge, ISIKA, IPSSI], Entreprises [VERMEG, SOPRA HR, GFI]
- ☐ Plus de 20 publications scientifiques
- ☐ Plus de 10 Certifications en IT
- ☐ Membre actif de la fédération Tunisienne de la Chasse sous-marine

Contact :

- ☐ LinkedIn : <https://www.linkedin.com/in/mohamed-amine-mezghich/>
- ☐ ResearchGate: <https://www.researchgate.net/profile/Mohamed-Mezghich/>
- ☐ Email perso: amine.mezghich@gmail.com
- ☐ Email pro: amine.mezghich@ensi-uma.tn ma.mezghich@smart-it-partner.com
- ☐ Mobile : +216 51 36 36 34 +216 98 787 521



- Le plan de formation
- Objectifs de la formation
- Public concerné
- Connaissances requises
- Documents utiles

- 1) Introduction au processus de développement logiciel
- 2) Agilité et méthodes agiles
- 3) Scrum
- 4) Outils Scrum

- Comprendre l'agilité
- Appréhender les principales approches agiles
- Connaitre les pratiques d'un projet agile
- Comprendre le Framework agile Scrum

Architectes, chefs de projets, acteurs de projets, directeurs de projets ou toute personne devant intervenir dans un projet Agile, en tant que contributeur ou manager

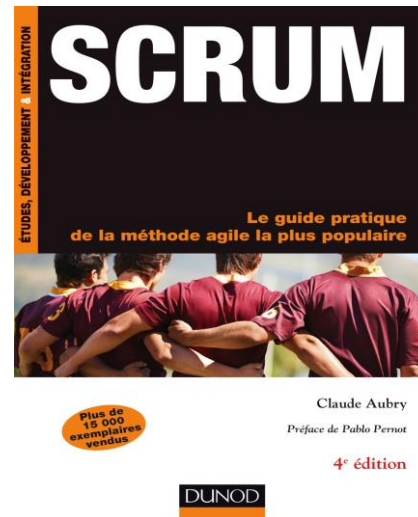
Connaissances requises

Connaissances de base en informatique.

Références bibliographiques

L'Agiliste : Guide de démarrage Scrum:

<https://agiliste.fr/guide-de-demarrage-scrum/>





Smart It Partner
We Build Smart Solutions For You

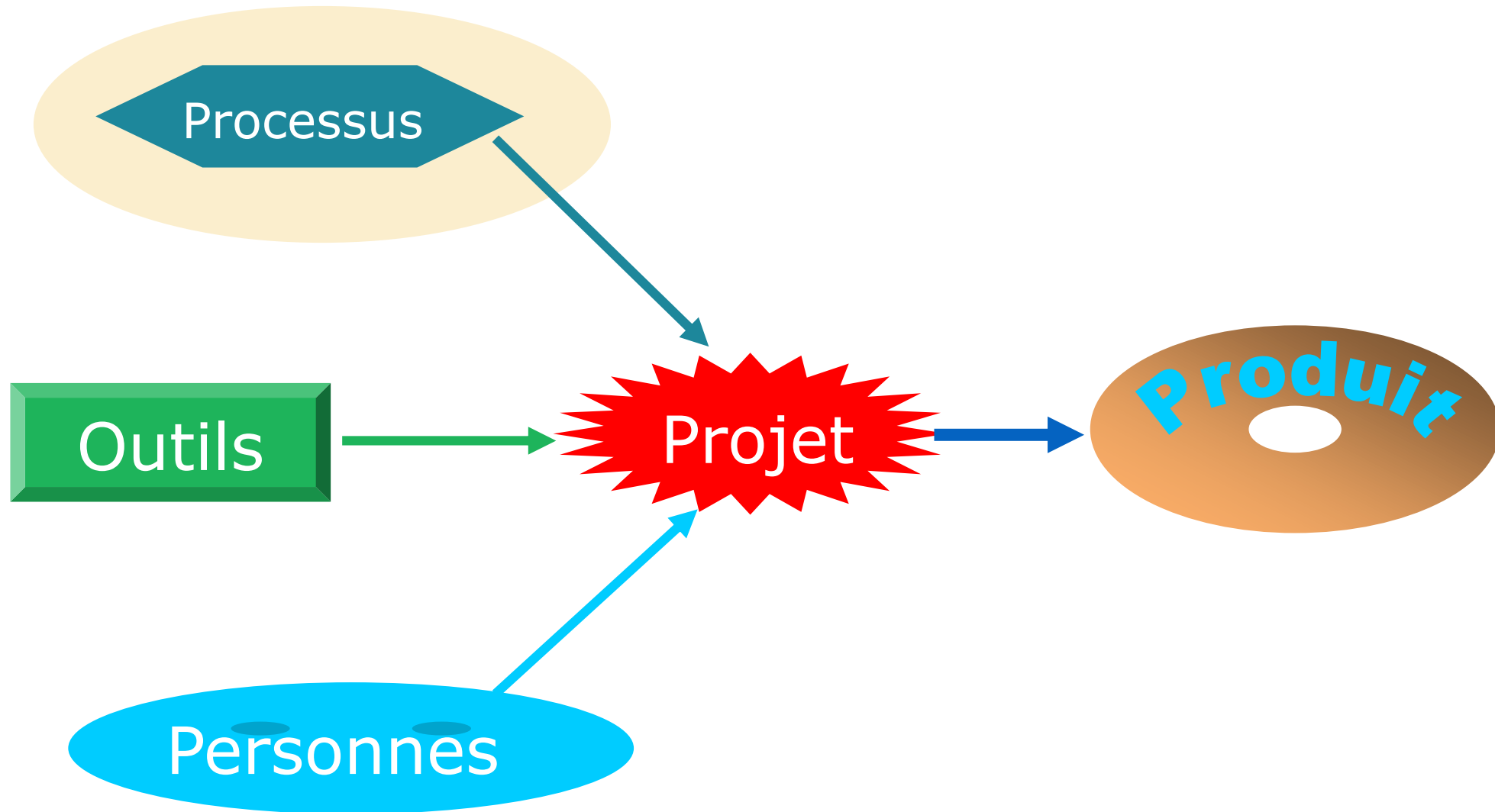
I. Introduction au processus de développement logiciel

- Développement logiciel
- Organisation d'un projet de développement logiciel
- Processus de développement logiciel
- Caractéristiques des processus de développement logiciel
- Principaux modèles de processus de développement logiciel

Développement logiciel

- ✓ Logiciels à développer de plus en plus complexes.
- ✓ Capacité de production logicielle inférieure à la demande (crise du logiciel).
- ➡ **Nécessité de mettre en place une organisation adéquate dans les projets de développement logiciel**

Organisation d'un projet de développement logiciel

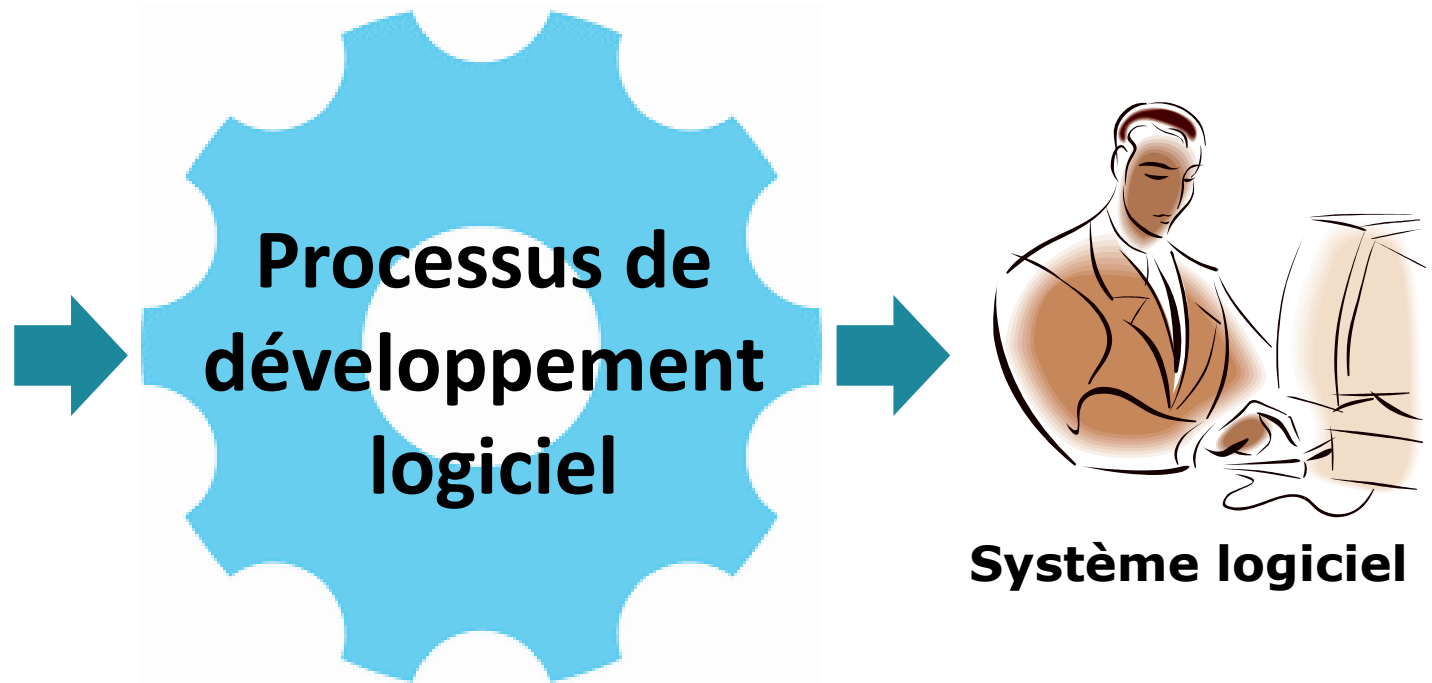


Processus de développement logiciel

Un processus de développement logiciel est un **ensemble d'activités** permettant de transformer les **besoins des utilisateurs** en un **système logiciel**.



Besoins utilisateurs



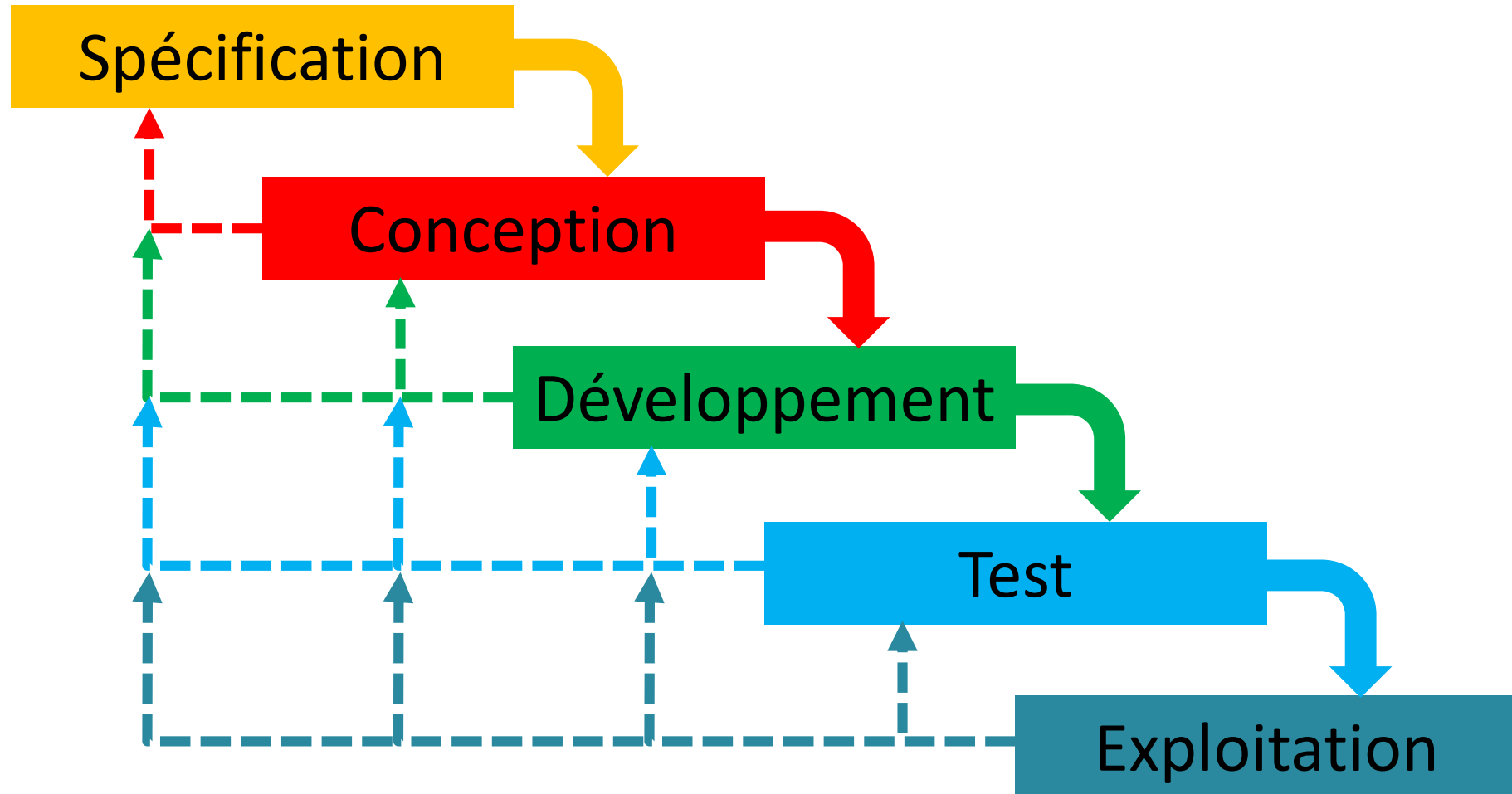
Un processus de développement logiciel doit être **générique**, c'est à dire qu'il doit pouvoir s'adapter à :

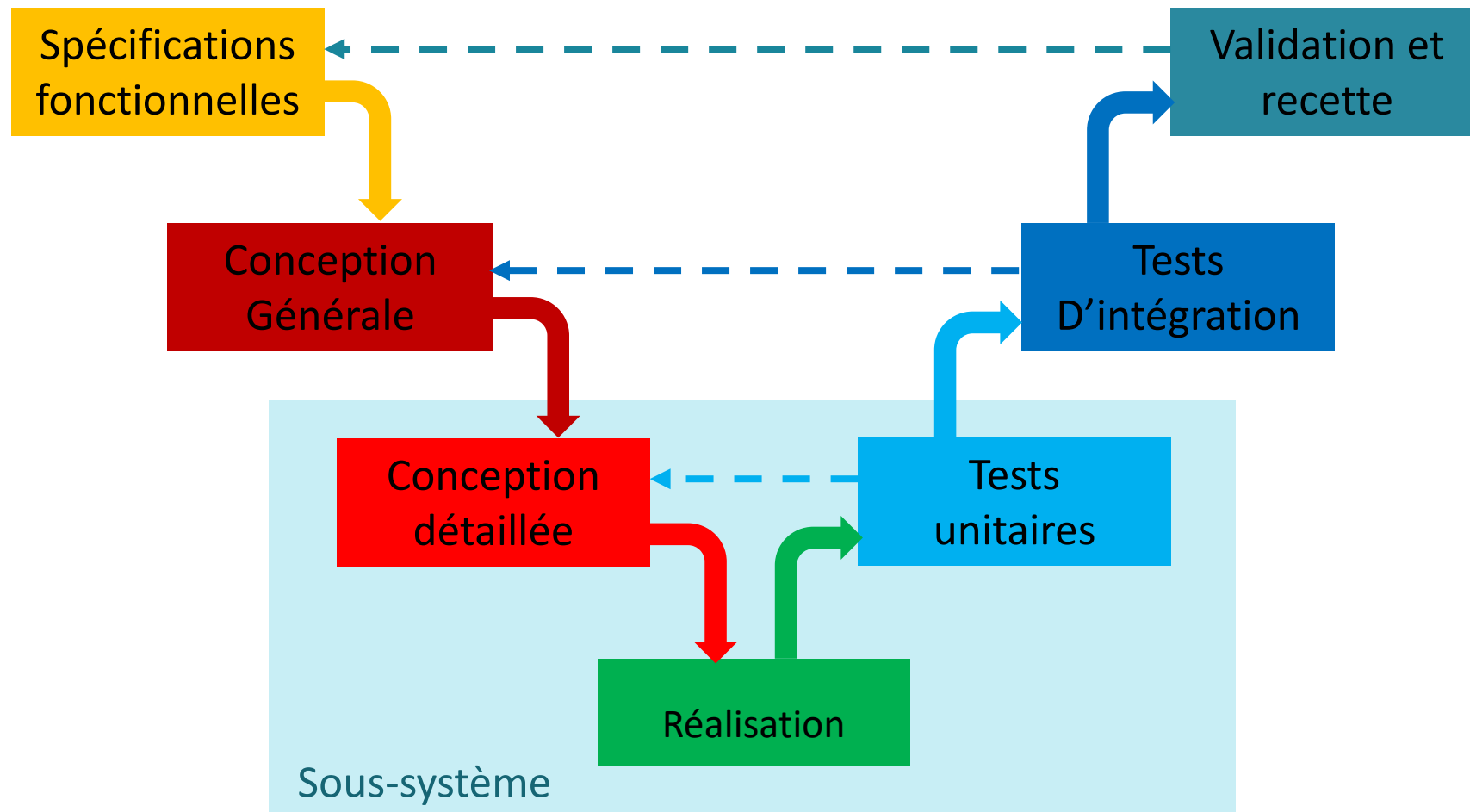
- Tout type de logiciel,
- Tout type d'entreprise,
- Toute taille d'entreprise,
- Toute taille de projet et
- Tout niveau de compétence.

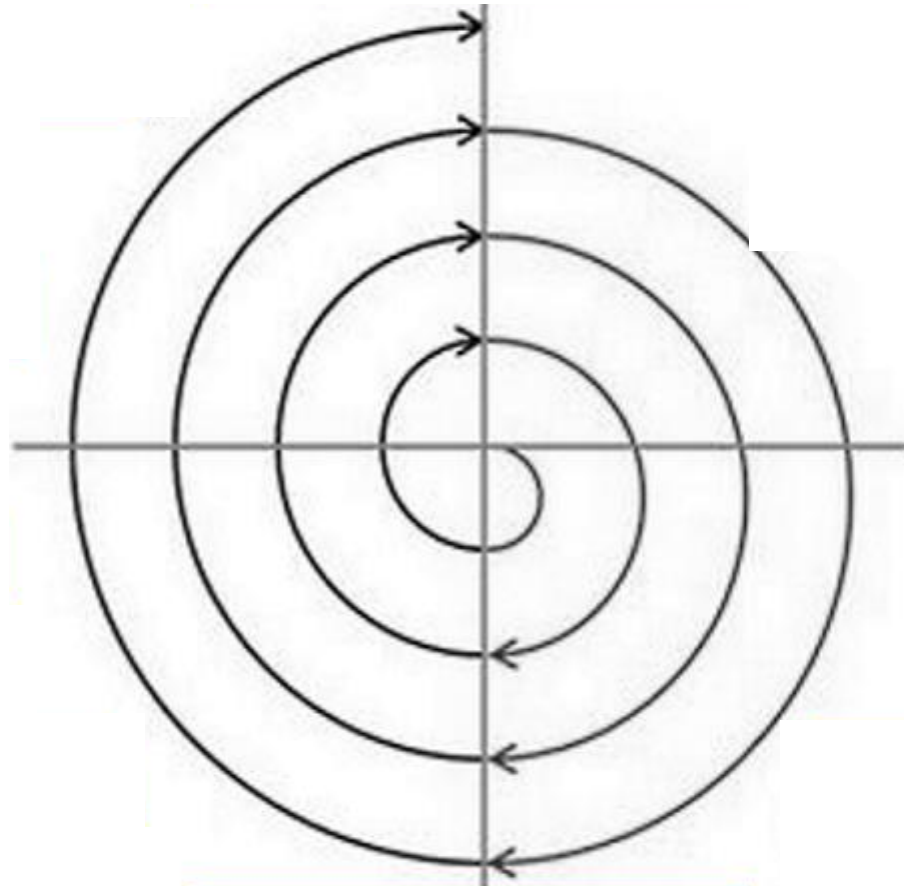
Principaux modèles de processus de développement logiciel

Le processus de développement logiciel peut être réalisé selon l'un des modèles suivants :

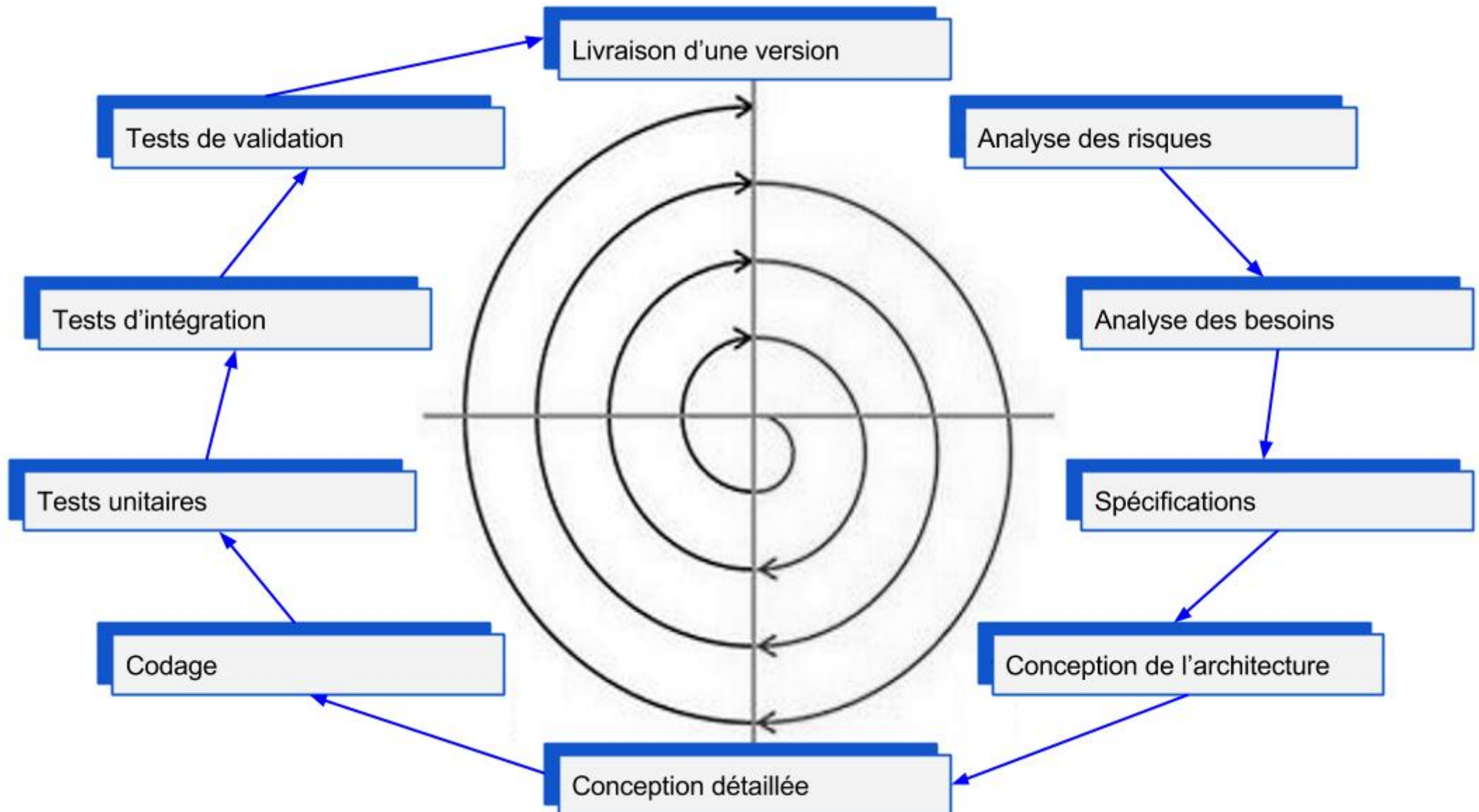
- Modèle en cascade
- Modèle en V
- Modèle en spirale
- **Modèle Agile**







Modèle en spirale



Atelier I : Etude comparative des Processus de développement logiciel

Pour chacun des processus de développement logiciel présentés, identifier :

- 1) Les principaux avantages
- 2) Les principaux inconvénients
- 3) Les cas d'application



Smart **It** Partner
We Build Smart Solutions For You

II. Agilité et méthodes agiles

- Introduction
- Les 4 valeurs de la culture agile
- Les 12 principes des méthodes agiles
- Les bénéfices des méthodes agiles
- Implication des équipes dans les méthodes agiles

- Constatation d'un malaise dans le processus de développement logiciel
- Selon Standish Group (1998) :
 - 25%** des projets respectent les délais et les budgets
 - 45%** dépassent le budget et sont en retard
 - 28%** sont abandonnés ou voient leur périmètre largement restreint
- Les difficultés de la conjoncture rendent les relations entre maître d'ouvrage et maître d'oeuvre difficiles

Principaux facteurs d'échec des projets :

- Difficultés de communication
- Mauvaises compréhension des besoins
- Insuffisance de l'architecture
- Absence de maturité des outils utilisés
- Manque de formation
- Cadre contractuel inadapté
- Insuffisance des tests
- Effet tunnel

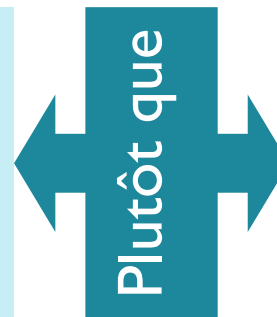
- Les méthodes classiques (1^{ère} et 2^{ème} génération) ne sont plus adaptées
- ➔ Nécessité d'une nouvelle génération de méthodes adaptées et souples : les méthodes **AGILES**.

Les 4 valeurs de la culture agile

La **culture** agile se base sur quatre valeurs qui peuvent être résumés comme suit :

Chaque fois que cela est possible, il est préférable de mettre en valeur :

- Les **individus** et leurs **interactions**
- Des **produits** et **services opérationnels**
- La **collaboration** avec les clients
- L'**adaptation** au changement



- les **processus** et les **outils**
- une **documentation** exhaustive
- la **négociation** contractuelle
- le suivi d'un **plan**

Manifeste pour le développement Agile de logiciels

Nous découvrons comment mieux développer des logiciels
par la pratique et en aidant les autres à le faire.
Ces expériences nous ont amenés à valoriser :

Les individus et leurs interactions plus que les processus et les outils
Des logiciels opérationnels plus qu'une documentation exhaustive
La collaboration avec les clients plus que la négociation contractuelle
L'adaptation au changement plus que le suivi d'un plan

Nous reconnaissons la valeur des seconds éléments,
mais privilégions les premiers.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas



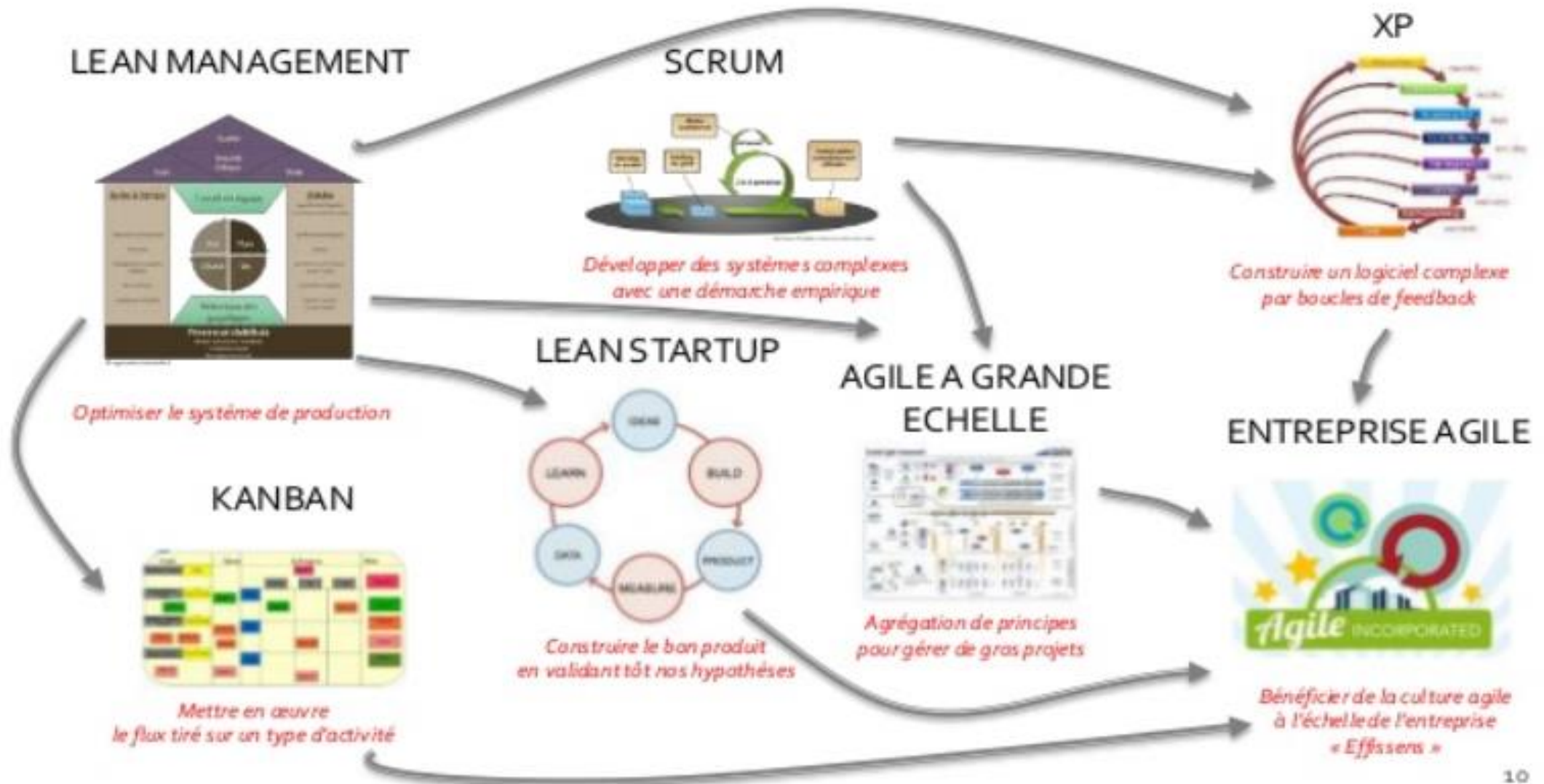
12 principes des méthodes agiles

1. Satisfaire le client, par des livraisons rapides
2. Accueillir les nouvelles demandes pour maintenir l'avantage métier
3. Livrer des versions fréquemment, toutes les deux semaines
4. Client et Développeurs travaillent ensemble quotidiennement
5. Impliquer des acteurs motivés, leur donner les moyens de travailler et leur faire confiance
6. Le meilleur moyen de communiquer ? Le face à face

12 principes des méthodes agiles

7. Des **versions qui marchent** sont les meilleures preuves d'avancement
8. Avoir un **rythme** de projet **soutenable** par tous : sponsor, client, développeurs
9. Continuellement être attentif à l'**excellence** technique et de conception
10. Viser la **simplicité**, c'est-à-dire minimiser l'effort
11. Les meilleures solutions naissent **d'équipes autonomes**
12. Régulièrement l'équipe réfléchit à **s'améliorer** et met en œuvre ses idées

Les mouvements agiles



Lean :

- Le Lean Management a été créé par Toyota au Japon et introduit au sein de ses usines dans les années 1970.
- Il s'agit d'une méthodologie qui se concentre sur une gestion sans gaspillage.

Lean startup :

- La méthode initiée par Eric Ries en 2008,
- C'est une manière d'entreprendre, basée sur des itérations :
 - Lancer une offre minimum,
 - Tester le marché,
 - Ajuster son produit,
 - Recommencer jusqu'à trouver la meilleure idée à vendre.

Design Thinking :

- C'est une approche développée à Stanford dans les années 80
- A pour but d'appliquer la démarche d'un designer pour répondre à un problème ou à un projet d'innovation.
- Cette manière de réfléchir et d'innover s'appuie de manière très importante sur des retours d'utilisateurs.

Scrum :

- C'est une méthode agile dédiée à la « gestion de projets ».
- Considérée aussi comme un Framework de management de projets, qui a pour objectif d'améliorer la productivité de l'équipe.
- Proposée par Jeff Sutherland et Ken Schwaber en 1995.

Kanban :

- Méthodologie issue de l'industrie automobile au Japon.
- Créée par Taiichi Ōno pour Toyota en 1950 dans le but d'optimiser sa capacité de production.

XP :

- L'eXtreme Programming ou XP est une méthode de gestion de projets,
- Applique à l'extrême les principes du [développement agile](#), c'est-à-dire se concentrer sur les besoins du client, mettre en place un développement itératif et l'intégration continue.
- Créée par Kent Beck en 1996

Bénéfices des méthodes agiles

Bénéfices financiers :

- Création rapide de valeur
- Meilleure gestion du rendement de l'investissement
- Possibilité de commencer le projet avec un budget limité

Bénéfices organisationnel :

- Plus grande autonomie des développeurs
- Meilleure collaboration entre les équipes
- Leadership plus mobilisateur

Bénéfices stratégiques :

- Meilleure gestion globale des risques
- Meilleure réactivité pour ajuster le produit aux réalités du marché
- Émergence d'une culture d'entreprise Agile

Implication des équipes dans la démarche agile

Cette implication se base sur trois piliers :

- **Un objectif commun** : pour fonctionner efficacement, une équipe doit avoir un objectif commun qui rassemble tous les acteurs. Cet objectif commun permet d'avoir au sein de l'équipe l'inspiration, la motivation et la recherche du succès.
- **L'alignement des efforts** : une foi que l'équipe a un but et est motivée, il faut que l'effort collectif soit le plus efficace possible. Ceci nécessite d'avoir un sentiment d'unité, de la confiance et de l'engagement.
- **Une connaissance du contexte** : pour avoir un travail d'équipe efficace, il doit y avoir une connaissance du contexte partagée par tous. Cela permet de comprendre les risques globaux de l'entreprise autour du projet et d'en reconnaître l'existence.

L'eXtreme Programming (XP)

- Méthode de développement itérative.
- Proposée par Kent BECK et Ron JEFFRIES.
- Ensemble cohérent de bonnes pratiques.
- Privilégie la production du code à celle des documents et des modèles.

Le développement se fait selon les étapes suivantes :

- Une phase d'**exploration** courte (~un mois) permettant au client de fournir une liste de **user stories** (courtes descriptions des besoins du client)
- Une suite d'itérations imbriquées :
 - Itérations de **développement** consistant à réaliser des fonctionnalités demandées par le client (1 à 4 semaines).
 - Itérations de **livraison** composées de 2 à 3 itérations de développement.

L'analyse et la conception ne peuvent être séparées de l'implémentation car :

- Le client peut changer d'avis en cours du projet.
- Risque de non compréhension des spécifications par les développeurs
- Risque de départ de membre de l'équipe

- **Assurer un feedback constant :**
 1. Développement piloté par les test
 2. Le jeu de la planification
 3. Le client sur site
 4. Programmation en binôme
- **Mettre en place un processus fluide et continu :**
 5. Intégration continue
 6. Amélioration du code par la relecture
 7. Itérations de livraisons courtes

➤ Favoriser une compréhension partagée :

- 8. Conception simple
- 9. User de la métaphore
- 10. Propriété collective du code
- 11. Conventions de codage

➤ Assurer le bien-être de l'équipe :

- 12. Un rythme de travail continu sans surcharge

1. Le modèle en cascade fait partie des :

- a. Modèles linéaires
- b. Modèles itératifs et incrémentaux

2. Le modèle en V exclut la possibilité de rejet du logiciel à la fin du projet :

- a. Oui
- b. Non

3. Le modèle en spirale est conseillé pour tous les types de projets :

- a. Oui
- b. Non

4. Lequel des éléments suivants ne constitue pas un facteur d'échec des projets :

- a. L'effet tunnel
- b. Le suivi d'une démarche itérative et incrémentale
- c. La difficulté de communication

5. Lequel des principes suivants ne fait pas partie de la culture agile :

- a. Privilégier la négociation contractuelle à la collaboration avec le client
- b. Privilégier l'adaptation au changement au suivi d'un plan
- c. Privilégier les individus et leurs interactions aux processus et outils



6. Les méthodes agiles sont utilisables :

- a. Uniquement dans les projets informatiques
- b. Dans tout type de projet

7. L'XP se base sur :

- a. La spécialisation des membres de l'équipe projet
- b. Le partage de la connaissances du projet par plusieurs acteurs

8. L'XP suit une démarche itérative et incrémentale :

- a. Oui
- b. Non

9. Selon la culture agile, le meilleur moyen de communiquer est :

- a. La documentation
- b. Le face à face

10. Quelle caractéristique doit avoir une équipe performante ?

- a. Une communication efficace
- b. L'écoute des directives du manager
- c. Contourner les erreurs



Smart **It** Partner
We Build Smart Solutions For You

III. SCRUM

- Présentation
- Les piliers de Scrum
- Fonctionnement général de Scrum
- Les évènements dans Scrum
- L'équipe Scrum
- Les éléments de la méthode
- Expression des besoins dans Scrum
- Estimation des users stories
- Éléments de conduite de projet

Présentation de Scrum

- Scrum est apparue en 1986 dans une publication de **Hiroataka Takeuchi** et **Ikujiro Nonaka** qui décrit une nouvelle approche **rapide** et **flexible** pour le développement de **nouveaux produits**.
- Ils comparent cette nouvelle méthode au **rugby à XV** : *une **équipe** qui avance ensemble et est toujours prête à **réorienter** le projet au fur-et-à-mesure de sa progression*
 - ➔ Scrum = Mêlée



Smart It Partner
We Build Smart Solutions For You

Présentation de Scrum



- Scrum se base sur les principes des **méthodes agiles** et notamment **la participation active du client** tout au long du projet.
- Scrum est la méthode agile la plus utilisée actuellement.
- Scrum est considérée comme :
 - un **état d'esprit**,
 - Un **Framework**,
 - Une **méthodologie**

Scrum se base sur les 3 piliers suivants :

- **La transparence** : elle garantit que tous les indicateurs relatifs à l'état du développement soient visibles de tous ceux qui sont intéressés par le résultat.
- **L'inspection** : elle est faite à des moments typiques (quotidiennement et à chaque sprint).
- **Les ajustements** : ils sont faits pour minimiser les futures déviations. Ils sont nécessaires si l'inspection met en évidence que certains indicateurs sont en dehors des limites acceptables.

Scrum : Un framework de gestion de projets

Scrum est considérée comme un framework de gestion de projet se basant sur les éléments suivants :

- **Des rôles :** Ce sont les différents acteurs qui participent au projet.
- **Des évènements :** Le cycle de vie d'un projet Scrum est rythmé par un ensemble de réunions définies avec précision et limitées dans le temps.
- **Des artefacts :** Tout au long d'un projet Scrum, les différents acteurs produisent un ensemble de livrables dont le contenu est bien défini.
- **Des règles :** Comme dans toute méthode, il existe dans Scrum un ensemble de règles qui définissent le comportement des acteurs, le déclenchement des évènements et le contenu des livrables. Ces règles ne sont pas figées et font l'objet d'une amélioration continue.

Fonctionnement général de Scrum

1. **Scrum** s'appuie sur le découpage des projets en itérations, appelées «**Sprints**» ou **Itérations**, dont la durée varie entre deux et quatre semaines.
2. Le projet commence par la réalisation d'un «**Product Backlog (PB)**» ou **Carnet de produit** composé de l'ensemble des fonctionnalités attendues pour le produit sur l'ensemble des sprints.
3. Chaque fonctionnalité est appelée «**User Story (US)**» ou **Histoire utilisateur**.

Fonctionnement général de Scrum

4. Avant chaque sprint, les tâches sont estimées en temps et en complexité à l'aide de certaines pratiques comme le «**planning poker**». Ces estimations permettent à la fois de planifier les livraisons, mais aussi d'estimer le coût de ces tâches auprès du client.
5. Les fonctionnalités qui font l'objet d'un sprint constituent le «**sprint backlog**».

Fonctionnement général de Scrum

6. Scrum est aussi caractérisée par une « **mêlée** » quotidienne dans laquelle les collaborateurs (chefs de projets, développeurs et responsables fonctionnels) indiquent les tâches qu'ils ont effectuées, les difficultés rencontrées et enfin les tâches à effectuer le jour suivant.

7. A la fin de chaque sprint, une réunion, appelée **revue** est organisée.

Au cours de cette réunion, l'équipe projet présente ce qu'elle a ajouté au produit pendant le sprint et décide si le travail effectué peut être livrée ou pas.

Fonctionnement général de Scrum

8. Après la livraison du sprint, une réunion, appelée **rétrospective** est organisée.

Elle est consacrée à la révision du processus suivi pendant le sprint.

- Au cours de cette réunion, l'équipe projet présente les difficultés rencontrées et les bonnes pratiques découvertes durant le dernier sprint.

Les évènements dans Scrum

- ❑ **Sprint** : Le sprint est un élément clé en Scrum. C'est une période durant laquelle une fonctionnalité (ou incrément du projet) est réalisée. Chaque sprint a une durée de 2 à 4 semaines.
- ❑ Chaque sprint comporte :
 - **Une réunion de planification** dans laquelle l'équipe évalue les tâches à effectuer,
 - De réunions quotidiennes de quelques minutes (**daily Scrum**),
 - Une **période de développement** pendant laquelle le travail de conception et de développement est fait.
 - Une **revue** effectuée à la fin de chaque sprint qui permet de valider en équipe le travail qui a été livré (incrément),
 - **Une rétrospective** de sprint effectuée après la revue dont l'objectif est d'étudier les erreurs commises dans le sprint et les éléments à améliorer pour que l'équipe s'améliore.

L'équipe Scrum est composée des 3 rôles suivants :

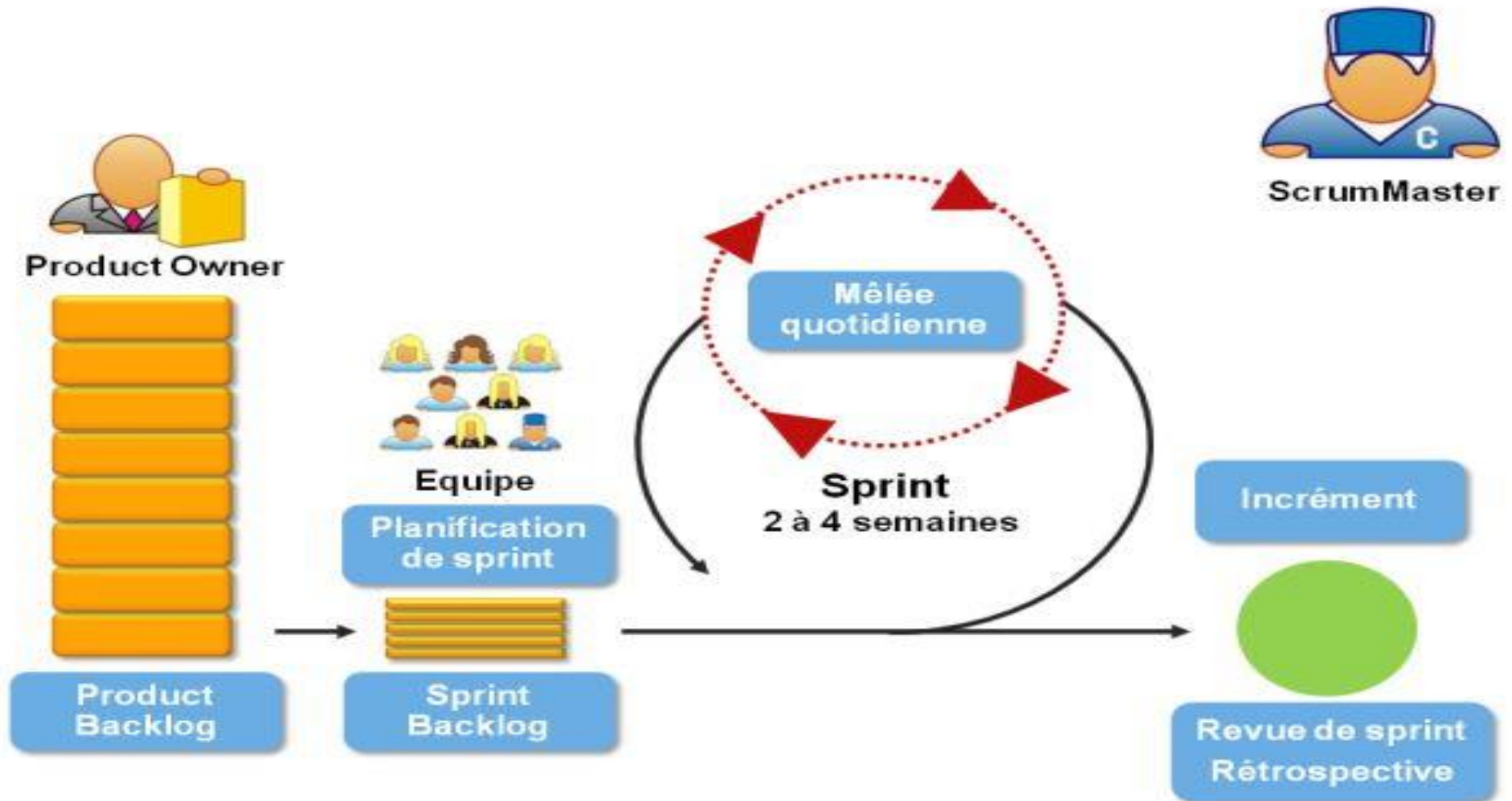
- Le propriétaire du produit (**Product Owner**, ou **PO**)
- Le responsable Scrum (**Scrum Master** ou **SM**)
- L'équipe projet (**The Team**)

- ❑ C'est le 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 sprint.

- ❑ C'est la personne chargée de veiller à la mise en application des principes 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 empêcheraient l'avancement de l'équipe et du projet pendant les différents sprints.

- ❑ Ce sont les personnes chargées de la réalisation du sprint et d'un produit utilisable en fin de sprint.
- ❑ Il peut s'agir de développeurs, architectes, personnes chargées de faire des tests fonctionnels, ...
- ❑ Le nombre de personnes ne doit pas dépasser 9 personnes.

Les éléments de la Méthode



Les éléments de la méthode



Expression des besoins dans Scrum

- ❑ L'expression des besoins sous forme de spécifications fonctionnelles est nécessaire dans tout projet pour traduire les **besoins du client** en fonctions applicatives, et pour guider ensuite les équipes techniques dans leurs réalisations.
- ❑ Contrairement aux démarches classiques, les spécifications fonctionnelles en mode agile **ne vont donc pas être rédigées intégralement dès le début du projet** mais tout au long de son déroulement. Chaque spécification fonctionnelle sera détaillée juste avant son développement (sprint).

L'expression des besoins dans Scrum

- Afin que l'expression des besoins soit efficace, il va falloir beaucoup de **rigueur**, et une **communication** sans faille entre le Product Owner d'un côté, et l'équipe de développement de l'autre.
- Contrairement à une méthode classique, la responsabilité de la rédaction des spécifications fonctionnelles est partagée :
 - Le « Product Owner » va aider à décrire son besoin et les spécifications (**dans son langage métier**) sous forme de User Stories.
 - L'équipe de développement va déterminer l'architecture et les solutions techniques à mettre en œuvre (**dans des termes techniques**).
 - Les testeurs vont identifier les cas limites à vérifier.

Forme générale d'expression d'une US :

- <En tant que> **rôle, utilisateur**
- <Je veux> **besoin, action**
- <Afin de> **bénéfice, valeur métier**

Ceci permet de répondre aux 3 questions :

- **Qui** a fait la demande ?
- **Quelle** est la demande ?
- **Quelle valeur métier** découle de la réalisation de ce besoin ?

Exemple:

En tant que client détenteur d'une carte bancaire
Je veux saisir mes données bancaires
Afin de régler ma commande en ligne

Remarque:

- La formulation initiale de la User Story est une description simple et compréhensible d'une fonctionnalité.
- Une User Story, comme son nom l'indique, est avant tout une **histoire** qui se raconte, crée la discussion et amène l'équipe à confirmer sa compréhension du besoin.

Expression des besoins dans Scrum

Exigences :

- Avec chaque User Story, le PO peut rajouter une ou plusieurs **exigences**.
- Une exigence correspond à une contrainte utilisateur imposée pour que le US permet d'atteindre la valeur métier attendue.
- **Exemple** : Contrôle à effectuer sur le format du numéro de carte bancaire

Expression des besoins dans Scrum

- Une « Bonne » user story doit respecter les principes **I.N.V.E.S.T** suivants :
 - **Indépendante** : peut fonctionner seule, indépendamment des autres user stories,
 - **Negociable** : éviter d'arriver avec une solution 'toute faite'. C'est un des gains de la séance d'estimation que de permettre à l'équipe de co-construire la solution, sur la base d'un besoin correctement décrit,
 - **Valorisable** : porteuse de valeur en soi. Elle doit rendre service à l'utilisateur.
 - **Estimable** : sa faisabilité peut être estimée,
 - **Small** : de la plus petite taille possible pour qu'elle soit réalisable dans un seul sprint,
 - **Testable** : son fonctionnement devra être testable et testé de manière effective. Elle doit avoir des critères évidents d'acceptabilité.

- Un **EPIC** (une **épopée** en français) est une longue histoire (Bin user story).
- Elle sera fort probablement découpée en plusieurs User Stories et sera réalisée en un ou plusieurs sprints.

- ❑ Avant toute planification de réalisation d'un projet, il y a forcément une étape d'estimation.
- ❑ L'évaluation de la réalisation d'une tâche peut être basée sur des notions différentes comme des jours/homme, des heures, des points ou même les tailles (comme pour les t-shirts : XS, S, M, L, XL...).
- ❑ Dans les méthodes agiles, il existe plusieurs méthodes d'estimation des user stories :

- **Planning Poker**
- **Bucket system**
- **Dot voting**
- **T-Shirt Sizes**



Estimation des user stories :

Le Planning Poker

- ❑ Scrum fournit à l'équipe de développement un outil particulièrement efficace pour réaliser des estimations de charges de réalisation : **le planning poker**.
- ❑ Le planning poker est une réunion qui regroupe tous les membres de l'équipe projet. Pendant cette réunion, les participants utilisent un jeu de cartes comportant les valeurs suivantes : 0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100, ∞ et ?
- ❑ Les cartes entre 0 et 100 correspondent à des niveaux de complexités croissants :
 - **0** : il s'agit d'un travail déjà effectué
 - **100** : il s'agit d'une tâche complexe qui a besoin d'être décomposée en deux sprints
 - **∞** : il s'agit d'une tâche très complexe qui nécessite une décomposition en plusieurs sprints
 - **?** : Il s'agit d'un travail impossible à estimer.

Estimation des user stories : Le Planning Poker

Déroulement d'une séance de Planning Poker :

Pour chaque fonctionnalité (user story) :

- 1. Explications du user story** : Le responsable de produit (Product Owner) explique aux autres participants le user story.
- 2. Questions/réponses sur le user story** : Les participants posent des questions, discutent du périmètre du user story, évoquent les conditions de satisfaction qui permettront de le considérer comme « terminé ».
- 3. Estimation individuelle de la complexité** : Chacun des participants évalue la complexité du user story, choisit la carte qui correspond à son estimation et la dépose, face vers le bas, sur la table devant lui.
- 4. Estimation collective** : Au signal du facilitateur, les cartes sont retournées en même temps. S'il n'y a pas unanimité, la discussion reprend. On répète le processus d'estimation jusqu'à l'obtention de l'unanimité.

Le « scrum master » note l'évaluation de la fonctionnalité et passe ensuite à la suivante, jusqu'à ce que l'ensemble de la liste ait été traitée.

Story point et vélocité :

- A la fin du **planning poker**, chaque fonctionnalité envisagée pour le nouveau sprint a donc une évaluation exprimée en nombre de points (Story points).
- Les fonctionnalités les plus complexes ou longues à réaliser ont obtenu des scores importants alors que les plus simples auront des scores compris généralement entre 0 et 3.
- Afin de déterminer quelles fonctionnalités vont pouvoir être embarquées dans le sprint, il faut tenir compte de ce que l'on appelle la « **vélocité** » de l'équipe de développement.
- La vélocité d'une équipe représente le nombre de points qu'elle peut réaliser entièrement durant un sprint.
- Si l'ensemble des fonctionnalités sélectionnées pour le sprint dépasse la vélocité de l'équipe, alors il va falloir que le scrum master réalise un arbitrage afin de déterminer lesquelles devront être reportées à un sprint ultérieur.

Estimation de la valeur métier des user stories

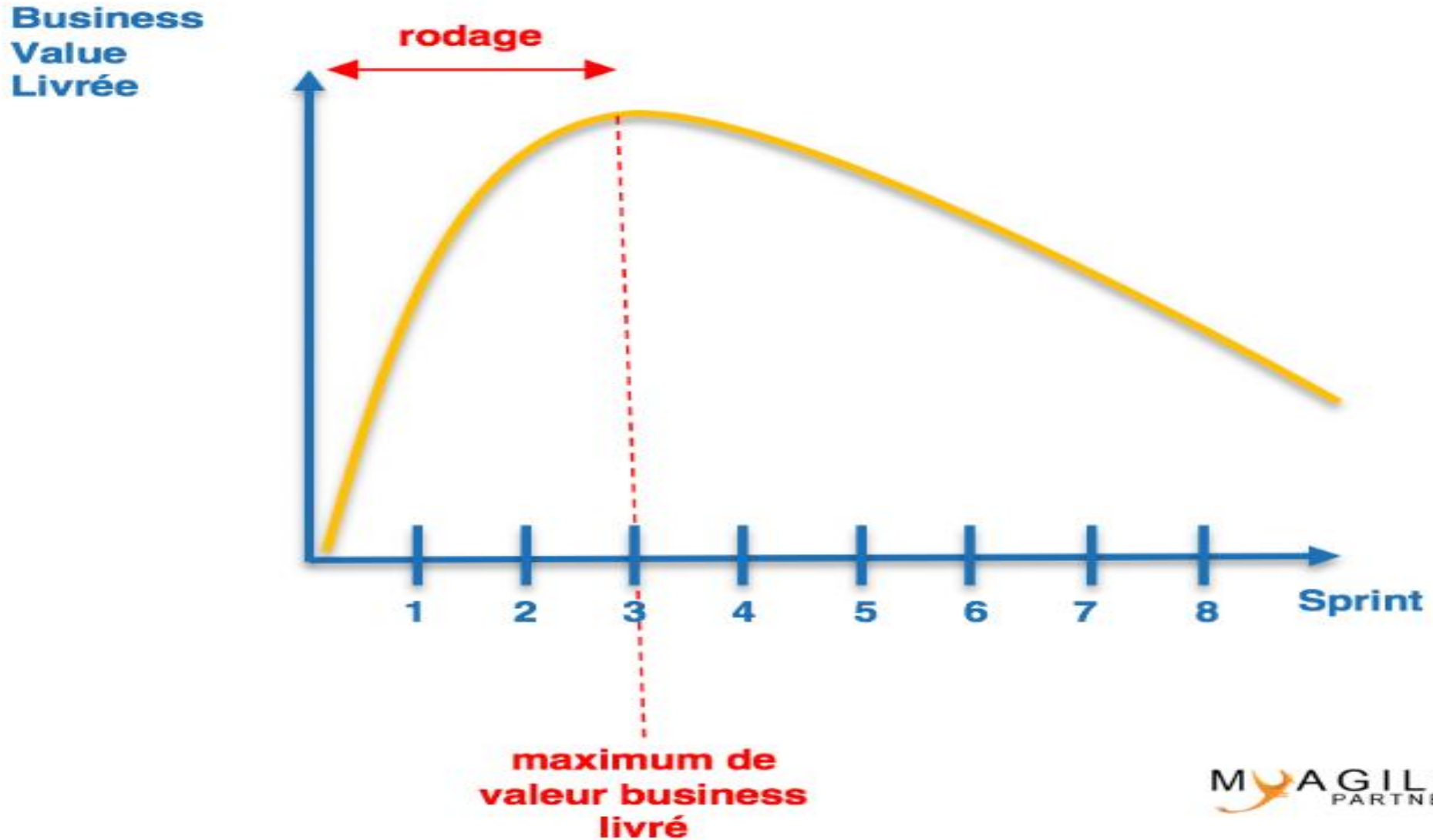
- ❑ Chaque User Story est porteuse de valeur car elle doit rendre service à l'utilisateur.
- ❑ Il est donc conseillé d'évaluer la valeur métier de chaque User Story.
- ❑ L'estimation de la valeur métier d'une user story permet :
 - De mesurer la quantité de "valeur métier" livrée au client en la visualisant sur un graphe après chaque sprint.
 - Prioriser les user stories en comparant la valeur métier de chacune des stories par rapport à son coût

Prioriser les US avec MoSCoW

- On prépare 4 fiches
- En équipe, on place les User Stories sur les fiches :
Soit l'un après l'autre, soit par vote : argumentation, débat
- Max 8 par fiche
- REGLE : tous les items **MUST** devront être finis sur les 2 premiers sprints



Estimation de la valeur métier des user stories



Estimation de la valeur métier des user stories

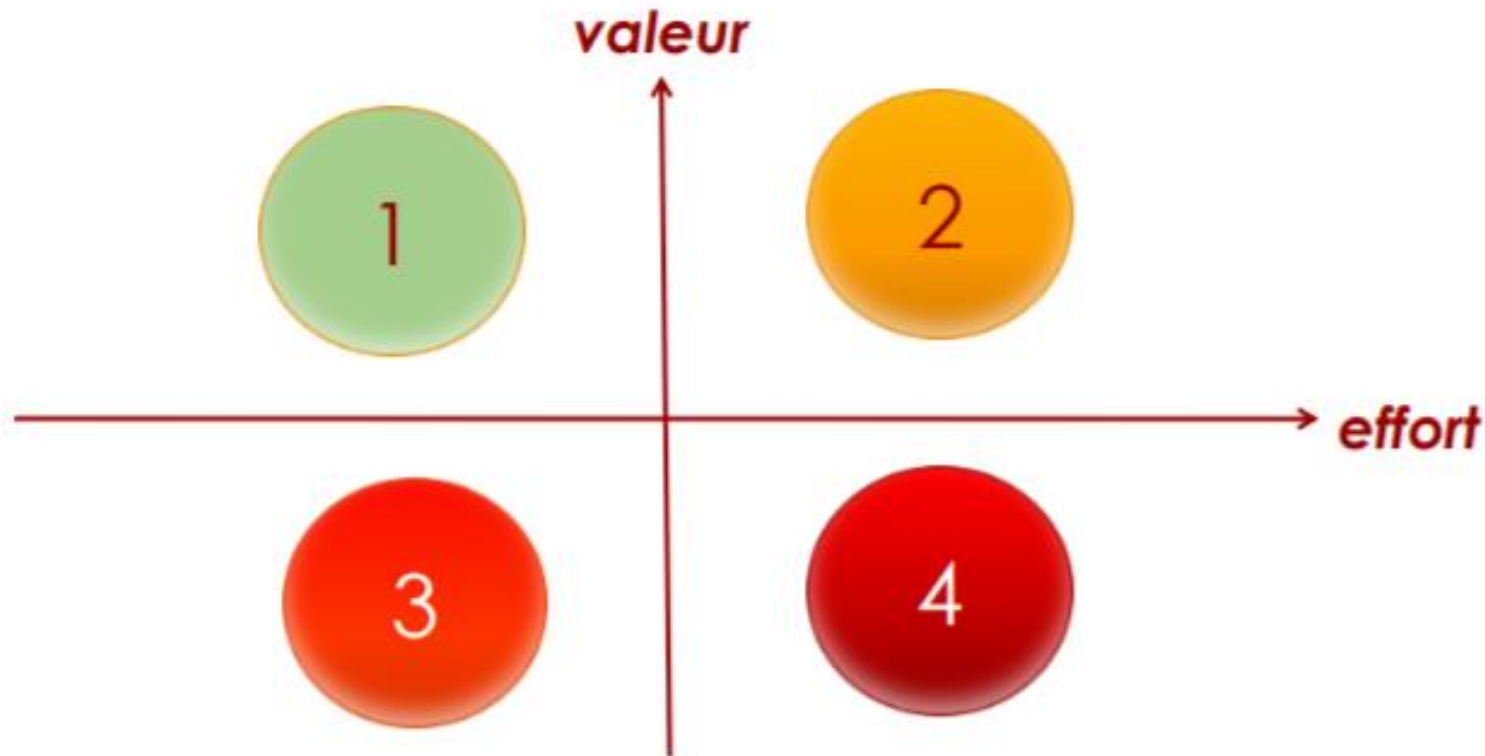
- Une fois la liste priorisée des user stories, on attribue une valeur globale au projet (par ex. 100 ou 500 points) et on répartit les valeurs sur les US ; En restant cohérent avec les priorités



- Pas en fonction de la VALEUR métier seule
- Mais du couple (valeur, effort)
- Objectif : faire ce qui est le plus important et le plus facile d'abord
- méthodes pour hiérarchiser le backlog :
 - Matrice Valeur / Effort
 - ROI

Hiérarchisation du Backlog avec la Matrice Valeur / Effort

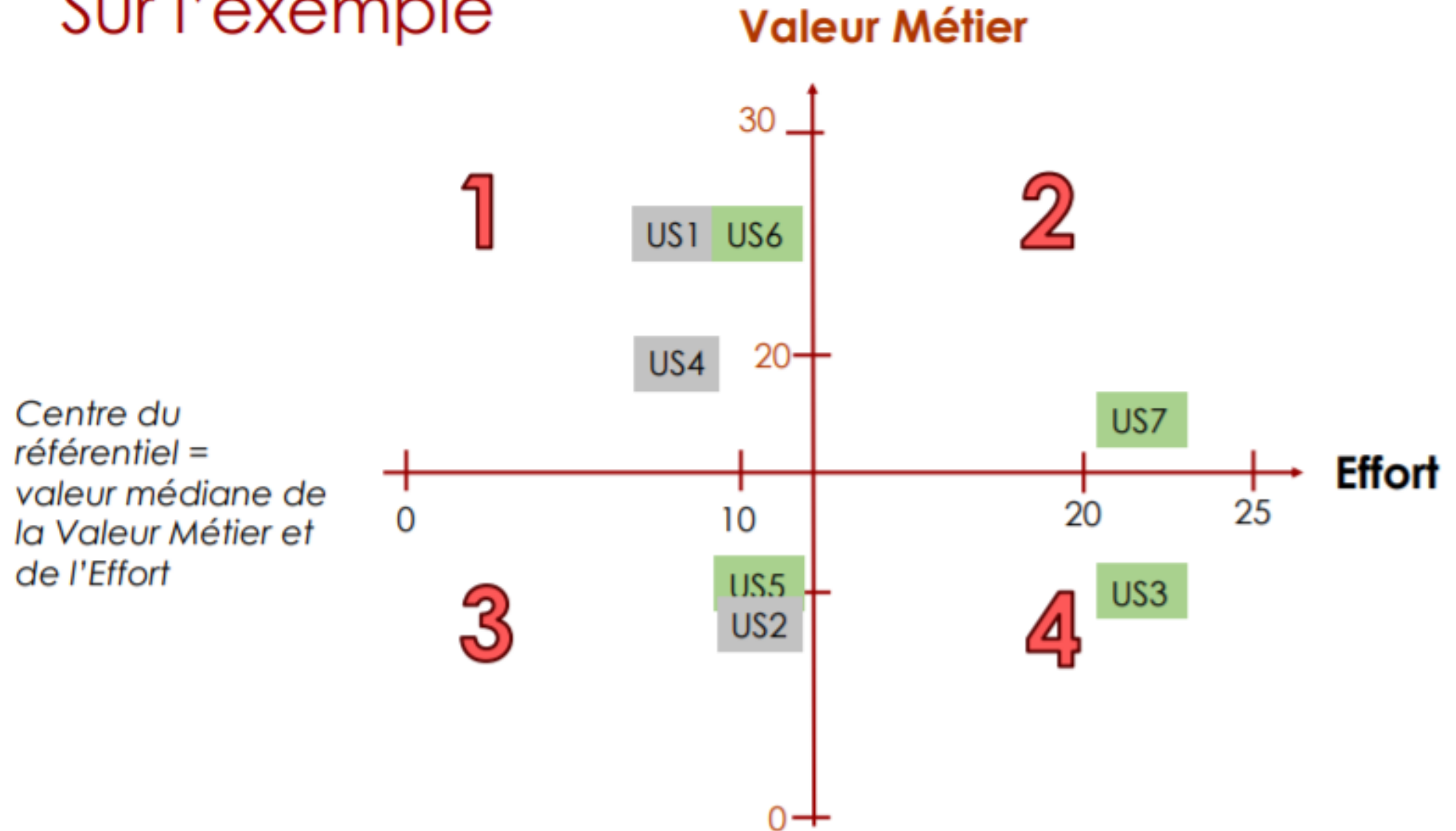
- Méthode la plus utilisée, la plus efficace



Un ex. de Backlog Produit

Backlog item	Acceptance Criteria	Effort	Value
US1 - En tant qu'internaute, je peux réserver l'hôtel en ligne	<ul style="list-style-type: none"> • Un email de confirmation est envoyé • Réservation doit être faite au min 24h avant date 	8	25
US3 - Améliorer la gestion des exceptions	Pas de msg "Exception ..." même en cas de pb	21	10
US4 - En tant que membre, je peux modifier les dates d'une réservation	...	8	20
US6 - En tant que membre, je peux annuler une réservation	<ul style="list-style-type: none"> • Un email de confirmation est envoyé • Ne peut être annulée qu'au moins 15 jours avant la date 	11	25
US7 - En tant que Resp. Hotel, je peux voir les réservations à venir	...	21	15

Sur l'exemple

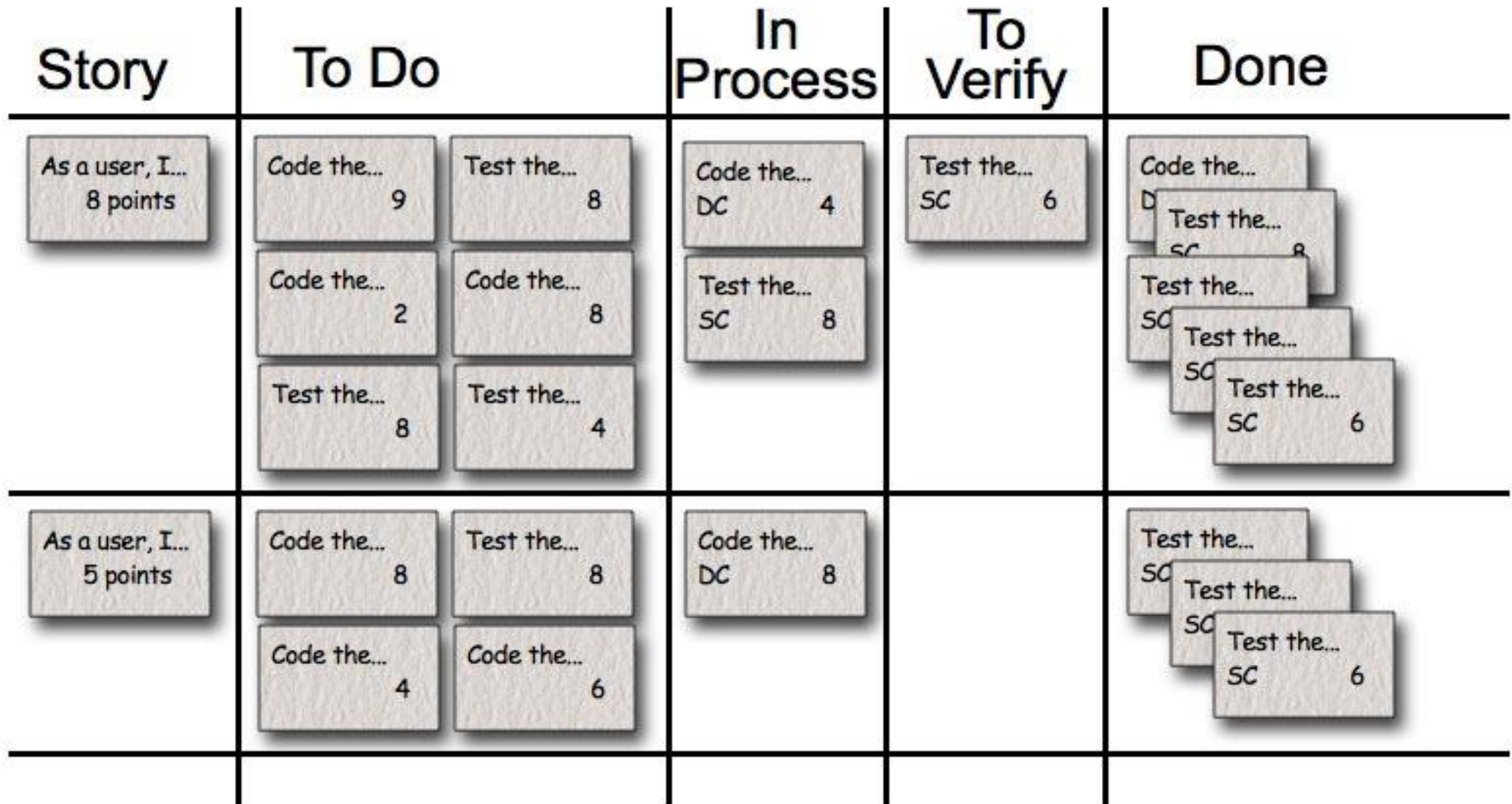


→ Backlog produit hiérarchisé

Priority	Backlog item	Acceptance Criteria	Estimate	Value
1	US1 - En tant qu'internaute, je peux réserver l'hôtel en ligne	<ul style="list-style-type: none"> • Un email de confirmation est envoyé • Réservation doit être faite au min 24h avant date 	8	25
2	US6 - En tant que membre, je peux annuler une réservation	<ul style="list-style-type: none"> • Un email de confirmation est envoyé • Ne peut être annulée qu'au moins 15 jours avant la date 	13	25
3	US4 - En tant que membre, je peux modifier les dates d'une réservation	...	8	20
4	US7 - En tant que Resp. Hotel, je peux voir les réservations à venir	...	21	15
...				
7	US3 - Améliorer la gestion des exceptions	Pas de msg "Exception ..." même en cas de pb	21	10

Éléments de conduite de projet :

Cycle de vie d'une user story



Scrum : Récapitulatif

Roles



Product Owner
Set Priorities
Manage Product Backlog



Scrum Master
Teach Scrum
Manage Process
Protect Team
Enforce Rules
Remove Blocks



Team
Develop Product
Organize Work
Report Progress



Stakeholders
Observe & advise

Artifacts



Product Backlog
List of requirements
Owned by product owner
Anybody can add to it
Prioritized by business value
Can change without affecting the active sprint



Sprint Goal
One sentence summary
Defined by Product Owner
Accepted by Team



Sprint Backlog
Decomposed task list
Driven by a portion of Product Backlog
Owned by Team
Only Team modifies it



Blocks List
List of blocks
& pending decisions
Owned by Scrum Master
Blocks stay on list until resolved



Increment
Version of the product
Potentially shippable
Working functionality
Tested & documented according to project definition of "DONE"

Meetings



Sprint Planning
Part A
Time-boxed to 4 hours
Run by Scrum Master
Declare Sprint Goal
Top of Product Backlog presented by Product Owner to Team
Team asks questions & selects topmost features
Part B
Time-boxed to 4 hours
Run by Scrum Master
Team decomposes selected features into a Sprint Backlog
Team adjusts +/- features by estimates against sprint capacity



Daily Scrum
Time-boxed to 15 minutes
Run by Scrum Master
Attended by all
Stakeholders do not speak
Same time/place every day
Answer 3 questions:
1) What I did yesterday?
2) What I'll do today?
3) What's in my way?
Team updates the Sprint Backlog
Scrum Master updates the Blocks List

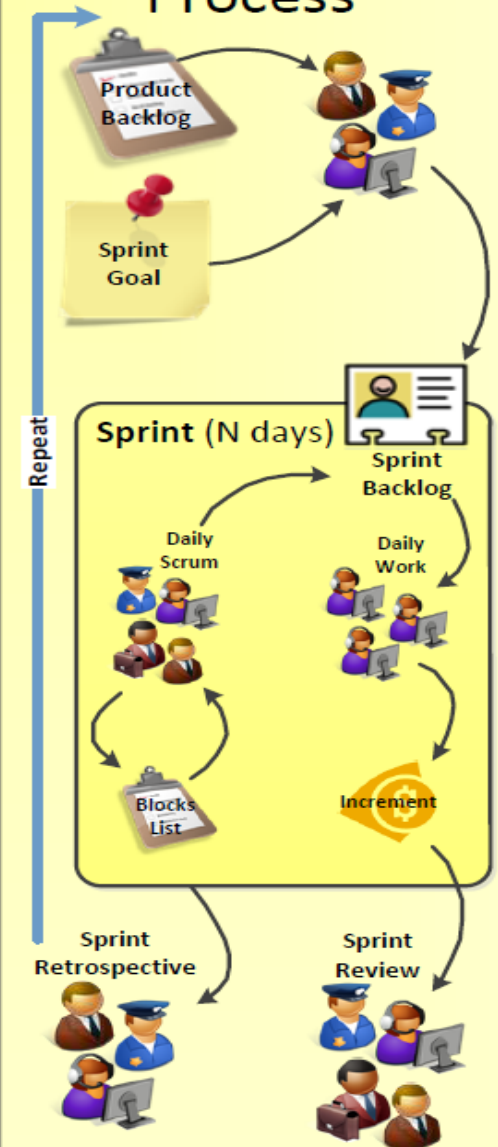


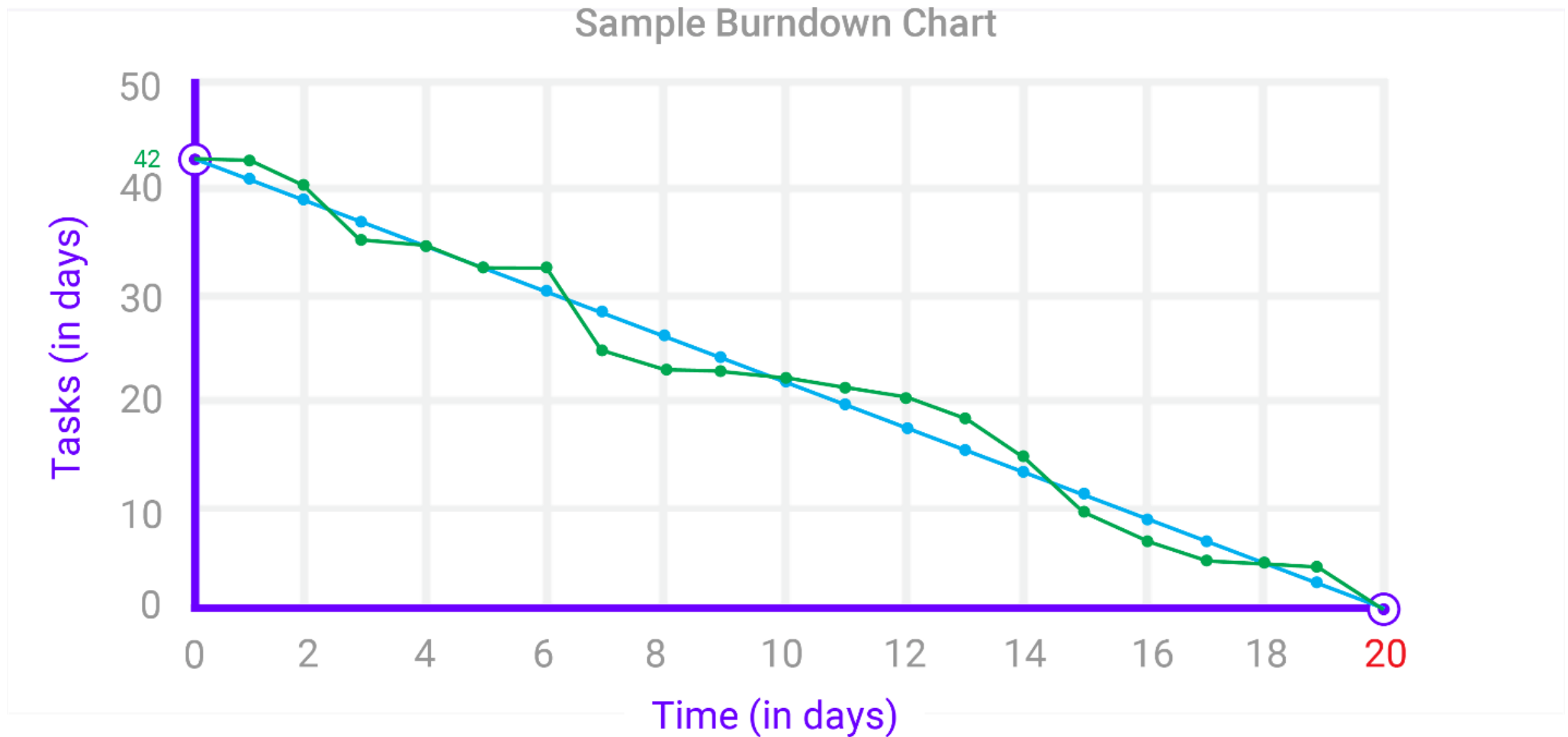
Sprint Review
Time-boxed to 2 or 4 hours
Run by Scrum Master
Attended by all
Informal, informational, Discussion
Team demonstrates increment
All discuss



Sprint Retrospective
Time-boxed to 1 or 2 hours
Run by Scrum Master
Attended by Team and Product Owner
Discuss process improvements, successes and failures
Adjust process



Process





IV. Outils SCRUM

Outils de conduite de projet Scrum

			
Nom	Trello	Asana	Jira
Version Gratuite	Oui	Oui	Oui
Fonctionnalités	<ul style="list-style-type: none"> > Organisez et suivez toutes les tâches, fichiers et informations de vos équipes en un seul endroit. > La recherche est incroyablement rapide et puissante, avec des opérateurs sophistiqués qui vous aident à affiner votre recherche. 	<ul style="list-style-type: none"> > Design flexible, rapide et moderne. Riche en fonctionnalités. La nouvelle vue Chronologie facilite la gestion > Mises à jour automatiques par email > Créer des calendriers et des vues personnalisés. Prise en charge iPhone, site mobile HTML5 	<ul style="list-style-type: none"> > Tableaux Scrum, Kanban personnalisables > Informations de performance en temps réel > Créez des workflows personnalisables, qui s'adaptent à n'importe quelle méthode de travail > Intégrez vos outils de développement pour une traçabilité de A à Z.

1. monday.com
2. [ProjectManager](#)
3. [Trello](#)
4. [Jira](#)
5. [Targetprocess](#)
6. [ClickUp](#)
7. [Vivify Scrum](#)
8. [Axosoft](#)
9. [Scrumwise](#)
10. [Quickscrum](#)