



Gestion de projet

Introduction à Scrum

Nicolas PARIS



[nicolas.paris-
ibm@free.fr](mailto:nicolas.paris-ibm@free.fr)

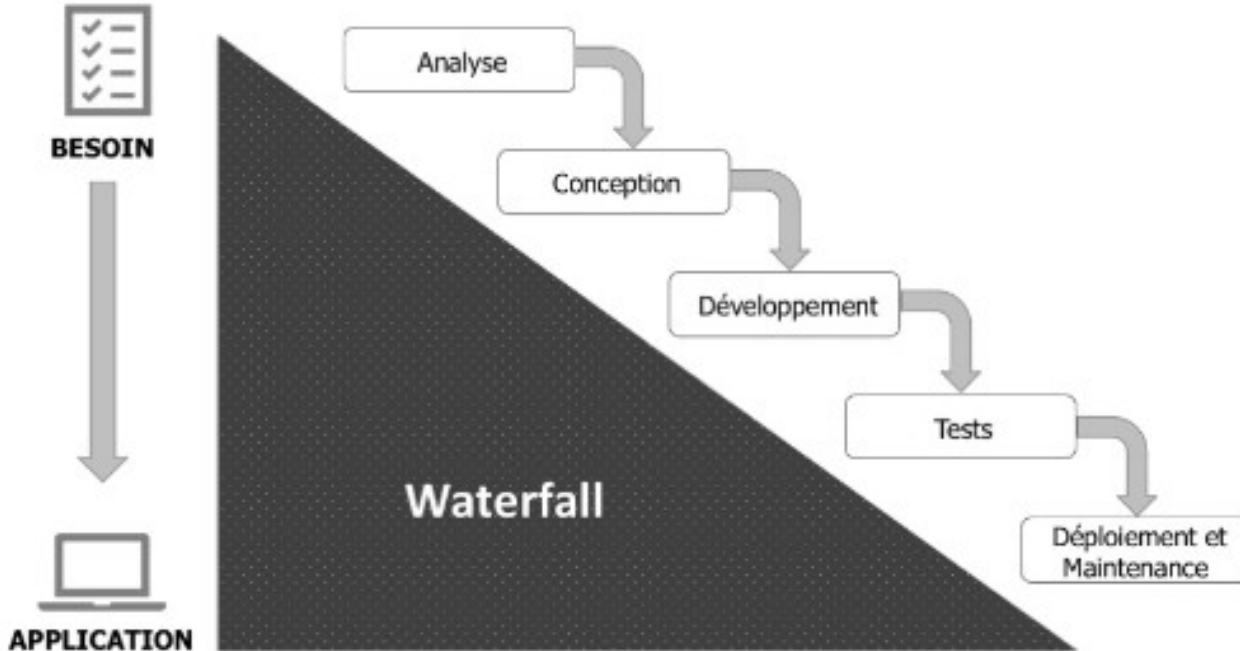


Sommaire

- ❑ Gestion de projet : méthodes classiques
- ❑ Définition de l'agilité
- ❑ Qu'est ce que Scrum ?
- ❑ Les phases de Scrum
- ❑ Etre agile avec Scrum / compléter et étendre Scrum
- ❑ Conclusion

Méthodes classiques (autres que Agiles...)

Modèle en Cascade



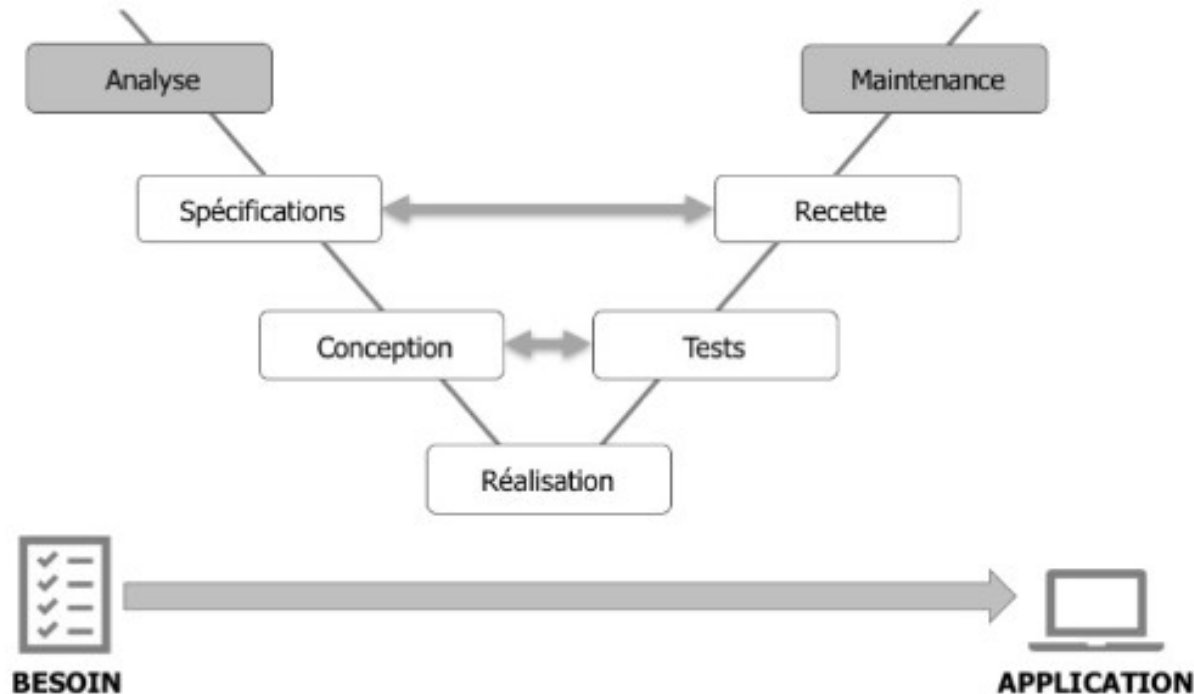
- Phases séquentielles
- Tous les besoins recueillis lors d'une 1^{ère} phase
- Une fois les anomalies corrigées, intégration globale

Remarques :

- Analyse complète des besoins peut s'avérer longue
 - Risque de besoin ayant changé entre temps
 - Modifications non prévues (non planifiées)
- Aucune souplesse dans la réalisation
 - Une fonctionnalité dépend des réalisations précédentes
 - Tout changement est complexe, car impacte les livrables passés (régression), en cours et futurs

Méthodes classiques (autres que Agiles...)

Modèle (cycle) en V

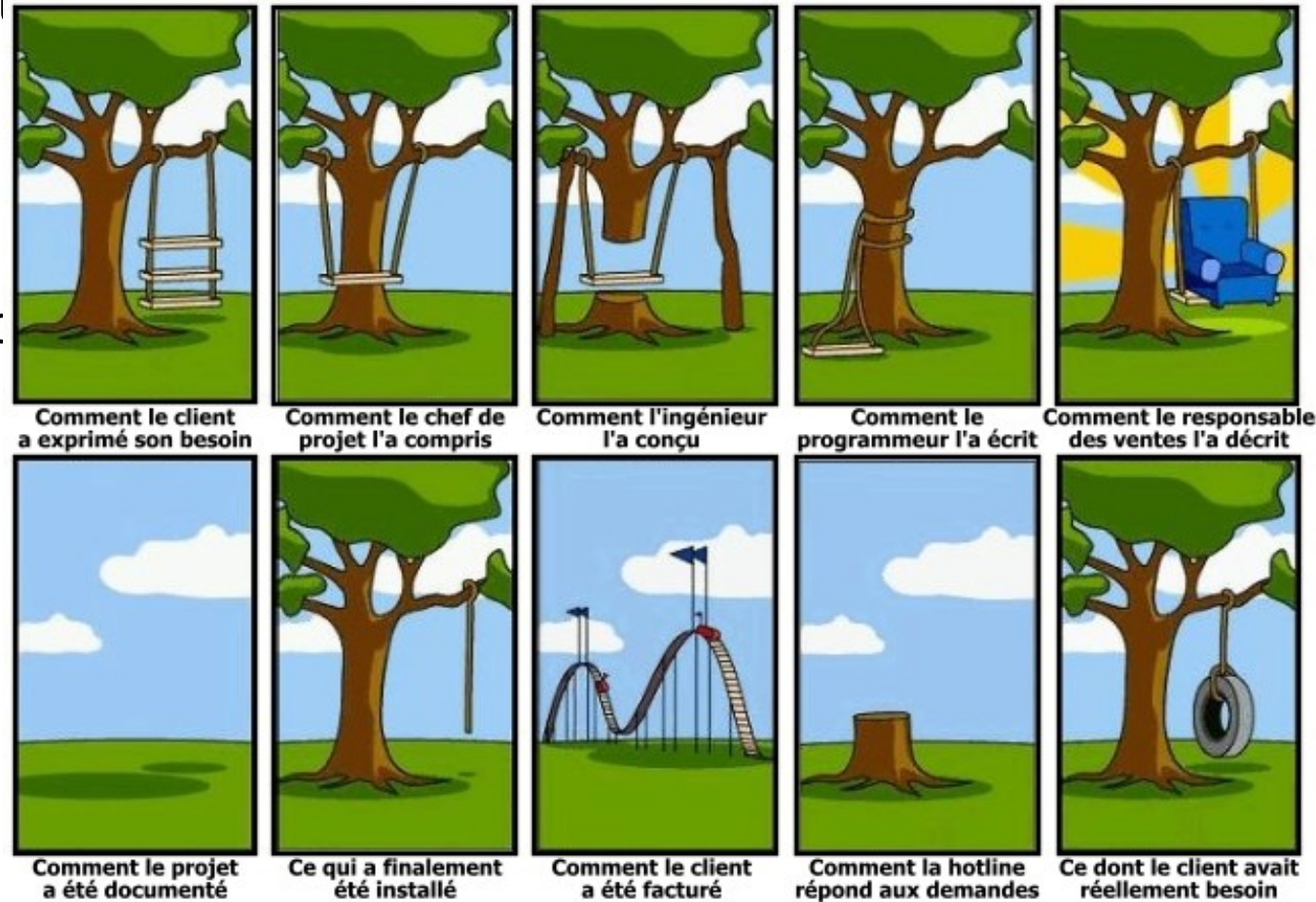
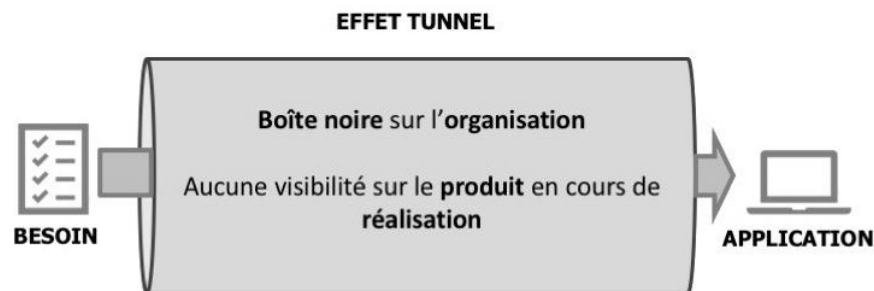


- Pallier le problème de réactivité du modèle en cascade
- Limiter le retour aux étapes précédentes
- Les phases de la partie montante renvoient de l'information sur les phases en vis-à-vis lorsque des anomalies sont détectées
- Amélioration du logiciel

Méthodes classiques (autres que Agiles...)

Les failles du modèle en cascade

- La rigidité de l'approche, tout doit être prévisible
- Prise en compte coûteuse des changements d'exigences
- Une mauvaise communication
- Une levée tardive des facteurs à risque
- Une documentation exagérée
- Le problème de *l'effet tunnel*



Méthodes agiles

- Les méthodes « agiles » proposent une **approche plus pragmatique** que les méthodes dites « classiques »
- Il s'agit de **s'adapter** à un environnement en évolution constante tout en **limitant les facteurs d'échec**
- Les méthodes agiles et classiques se distinguent essentiellement par leurs approches
 - **Approche prédictive** des méthodes classiques : aboutit à la création d'un logiciel développé d'un seul tenant
 - **Approche adaptative** des méthodes agiles : développement par itération avec acceptation du changement en cours de projet
- L'objectif principal est d'**éviter l'effet tunnel**

Définition de l'agilité

- Le **manifeste Agile** (2001)
- Le cadre de la philosophie Agile

4 préceptes

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

12 principes

Priorité : satisfaction utilisateur

Accepter le changement

Livrer fréquemment

Travailler en synergie quotidiennement tout au long du projet

Stimuler la motivation

Favoriser la communication entre tous les acteurs du projet

Un logiciel opérationnel est la principale mesure d'avancement

Les rythmes de production doivent être supportables

L'excellence technique et une bonne conception renforcent l'agilité

Minimiser la quantité de travail inutile

Favoriser l'auto-organisation des équipes des

Méthodes classiques VS Méthodes agiles

APPROCHE INCRÉMENTALE



Définir la cible **précise** (spécifier toutes les caractéristiques de notre Mona Lisa et de son panorama)
et **réaliser parfaitement** ~~chaque~~ des parties dans l'ordre **les unes après les autres**

APPROCHE ITÉRATIVE



Définir une vision **globale** (pour notre exemple : une jeune femme en avant plan d'un paysage vallonné) puis de réaliser au plus vite un **ensemble fonctionnel minimal**. Celui-ci pourra ensuite être *amélioré* - voire même *corrigé* - d'étape en étape

Méthodes classiques VS Méthodes agiles

➤ **Incrémental**

- Avantage : si tout se passe bien, adéquation entre la cible imaginée et le résultat final après 5 h
- Inconvénient : si retard, la Mona Lisa se retrouve à moitié terminée ou sans son bras gauche

➤ **Itératif**

- Avantage : Voir apparaître plus vite notre Mona Lisa
- Avantage : feedback au plus tôt sur votre travail, meilleure adéquation avec vos interlocuteurs, finalisation

➤ **Agile = incrémental + itératif**

- Ne pas réaliser de plan détaillé mais définir à la place une **vision globale** et un premier **résultat fonctionnel minimal** à atteindre. Il constituera le 1er incrément de notre produit.
- Au cours de la réalisation (itérative), **tester à chaque étape** le produit construit avec les personnes qui vont en bénéficier. Déterminer ainsi s'il est suffisamment évolué pour être considéré comme terminé ou s'il faut l'améliorer
- S'il y a besoin d'ajouter de nouvelles fonctionnalités, **créer** alors **de nouveaux incréments**. Les réaliser les uns après les autres et dans un ordre logique qui permette de toujours avoir un résultat cohérent et utile à présenter à l'utilisateur si le temps venait à manquer pour terminer.
- Elle y mélange alors les 2 approches **en réalisant son produit de manière itérative et en l'enrichissant de manière incrémentale**.

Faire adhérer le client à l'agilité...



Méthodes agiles

✧ **La communication est de meilleure qualité**

- ▣ L'utilisateur a la possibilité de clarifier ses exigences au fur et à mesure
- ▣ Le client reçoit des preuves tangibles de l'avancement du projet

✧ **La visibilité est meilleure**

- ▣ Le client peut ainsi visualiser les travaux plus régulièrement, au fil de l'eau, sans attendre la fin du projet

✧ **Les risques sont détectés plus tôt**

- ▣ Grâce aux activités de développement précoces, les risques sont détectés tôt et résolus rapidement

✧ **Acceptation du changement**

- ▣ Accueillez le changement à bras ouverts plutôt que de le craindre et de le combattre

✧ **Un produit de haute qualité**

- ▣ Grâce à la sélection des fonctionnalités à implémenter en priorité
- ▣ Grâce au Feedback permanent du client
- ▣ Grâce à des campagnes de test et au contrôle qualité au cours de chaque itération
- ▣ Les défauts peuvent être détectés et corrigés immédiatement

Les critères qui poussent vers l'agilité

- Le périmètre fonctionnel du projet n'est pas très clair et risque de bouger en cours de projet
- Il y a de très forts risques de ne pas réussir facilement à répondre au besoin du client et il peut être salubre de valider régulièrement avec le client ce qui est réalisé par l'équipe
- Il y a de très forts risques techniques et il peut être salubre d'avoir la capacité de traiter ces risques techniques par une validation technique régulière du produit
- Il est nécessaire de livrer très rapidement une première version quitte à livrer une première version ne contenant que les fonctionnalités primordiales

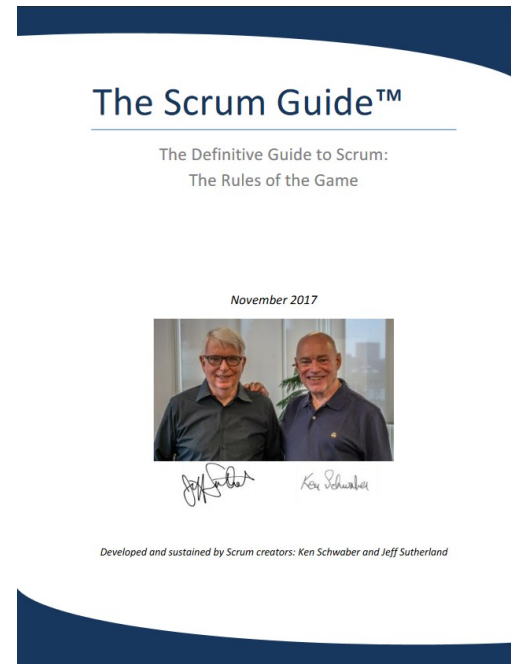
Quelques méthodes agiles

- **RAD** (1991) : introduit le concept de semi-itératifs (les itérations ne concernent que les développements, pas la conception)
- **DSDM** (milieu des années 1990) : propose un processus de production spécifique et 9 principes à respecter.
- **Extreme Programming** (1996) : pousse à l'extrême les bonnes pratiques de réalisation d'un projet
- **FDD** (1997) : propose des itérations en 5 étapes allant de la conception à la réalisation pour chacune d'elles
- **Crystal** (2004) : ensemble des méthodes, chacune étant adaptée à une taille d'équipe et à une criticité de projet

Qu'est ce que SCRUM ?

- Scrum a été initié en 1995 par **Ken Schwaber** et **Jeff Sutherland** (deux des signataires en 2001)

- Les bases de Scrum sont définies dans un document de référence : **le guide Scrum**



- Scrum est un cadre de travail (Framework) et non une méthode. Il met à disposition une boîte à outils permettant de structurer la méthode agile de l'utilisateur par leurs approches
- Scrum = « mêlée de rugby » : tout le monde pousse en même temps dans le même sens lors de la mêlée

Les valeurs de SCRUM

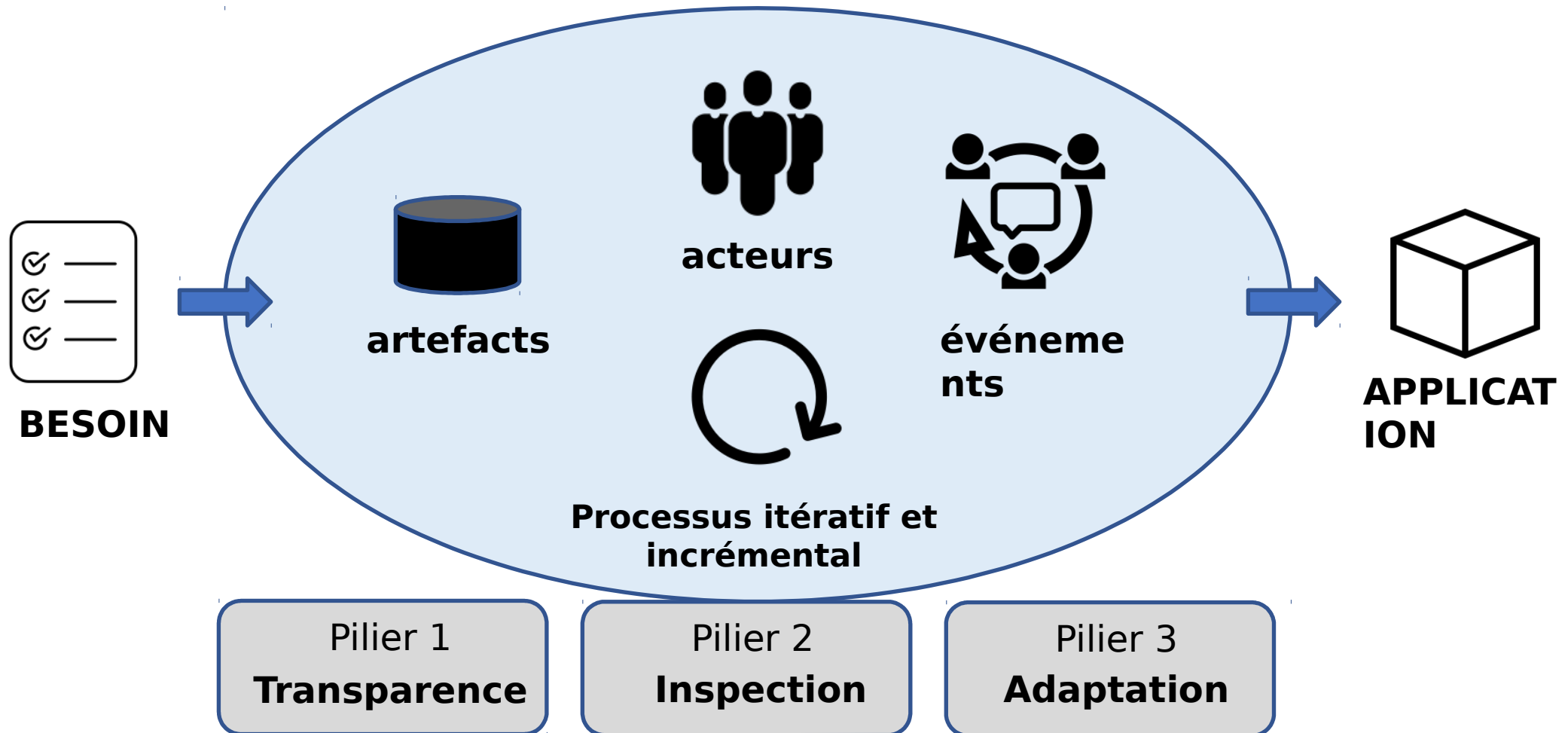


Les objectifs de SCRUM (1/2)

- Mettre à disposition des moyens de gestion de projets informatiques complexes de façon itérative
- Proposer un contrôle empirique du processus de production logiciel. Il implique de se baser sur l'expérience acquise pour améliorer l'organisation et le produit réalisé
- Chaque itération doit permettre de :
 - ❑ Ajouter de la valeur au produit
 - ❑ Préciser la définition du produit
 - ❑ Améliorer le processus de production
- Il repose sur 3 piliers :
 - ❖ **Transparence** : communication constante / compréhension partagée par tous les membres
 - ❖ **Inspection** : afin de détecter les écarts par rapport aux attentes
 - ❖ **Adaptation** : pour réduire le risque et mettre en place une démarche d'amélioration

Les objectifs de SCRUM (2/2)

- Le cadre de travail Scrum repose sur la définition d'**acteurs**, d'**événements**, d'**artefacts** (du travail ou de la valeur) et d'un **processus** spécifique



Scrum : rôles et acteurs

Product Owner (PO)



Rôles :

- Responsable du produit
- Validateur de toutes les décisions autour du produit et du contenu du product backlog
- Traducteur, il assure la transparence (compréhension par tous) des besoins exprimés

Compétences et qualités :

- Rigueur sur le maintien à jour des définitions du produit
- Capacité d'immersion pour assurer une compréhension optimale du métier et du besoin
- Ecoute et ouverture auprès des autres membres de l'équipe et de leurs propositions
- Esprit de synthèse



✓ 1 seul PO

✓ Plaque tournante de toutes les discussions et décisions autour du produit

Scrum : rôles et acteurs

Scrum Master (SM)



Rôles :

- Responsable de la méthodologie
- Organisateur chargé d'accompagner les équipes dans l'utilisation des composantes de Scrum (événements, artefacts, ...)
- Facilitateur, aidant à fluidifier la communication

Compétences et qualités :

- Connaissance et expérience de Scrum
- Capacité à transmettre
- Intelligence relationnelle pour créer une synergie dans l'équipe



- ✓ 1 seul SM
- ✓ Le rôle de SM peut être tournant : à chaque nouveau projet, un nouveau SM est désigné

Scrum : rôles et acteurs

Equipe de développement



Rôles :

- Responsable de l'incrément, du sprint backlog et donc de la manière dont sera réalisée l'application
- Chargée de l'estimation de la charge et du temps de réalisation des items
- Réalisation de la solution technique

Compétences et qualités :

- L'équipe de développement est pluridisciplinaire. Elle regroupe toutes les compétences techniques nécessaires à la réalisation du logiciel : développeurs, architectes, concepteurs, ...



✓ Scrum préconise entre 3 et 9 membres pour optimiser la réactivité, la capacité à produire et la communication

Scrum : rôles et acteurs (hors équipe Scrum)

- ❑ Les commanditaires (clients) qui constituent les initiateurs du projet



- ❑ Les futurs utilisateurs de l'outil



- ❑ Les financeurs (sponsors) qui apportent le support financier au projet



Scrum : les artefacts

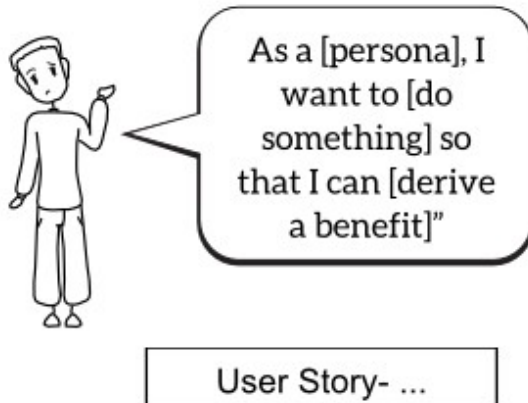
Product Backlog



- Ce carnet porte la définition du logiciel tout au long du projet
- Il est maintenu par le PO et liste les attentes du produit
- Unique et seule référence sur le produit
- Partagé à toute l'équipe
- Evolue en même temps que le projet, maintenu à jour
- L'élément de base est l'item, la **User Story** étant la plus utilisée

Scrum : les artefacts

User Story



- Une fonctionnalité à développer
- Un titre qui évoque son contenu (« Lister les commandes »)
- Une estimation de l'effort (story points) proposée durant un « Poker Planning »
- Une affectation : qui fait le développement, fixée par le SM
- Un objectif : En tant que ... je souhaite / je veux ... afin de ...
- Les règles métier : rédigées avec précision par le PO
- Les critères d'acceptation sous la forme : Etant donné que ... quand un (événement survient) ... alors (décrire le résultat attendu)
- Pas trop longue, sinon à redécouper

Scrum : les artefacts

Exemple de Product Backlog :

Scrum Framework



Scrum Artifacts-Product Backlog (PBL)-Example:



Project: Shopping Website

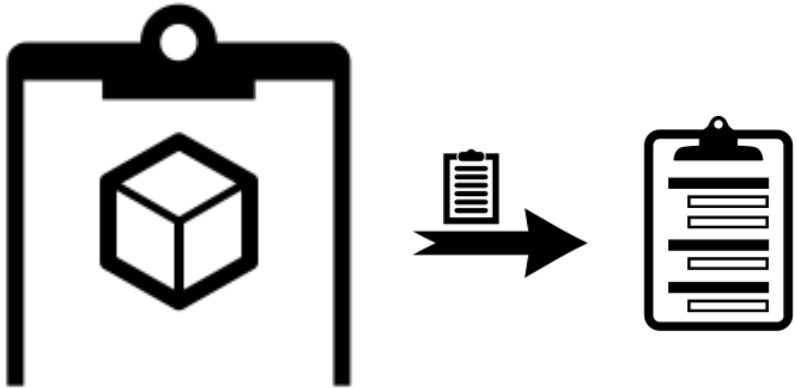
Prior ity	Product Backlog Items	User Story #	User Story	Story Point	Estimate (Hours)
1	Database Creation	9	As an operations engineer, I want to be able store all customer information, so that I can serve to customers.	40	240
2	Login Page	15	As a site member, I want to login the site, so that I can do online shopping.	20	160
3	Category Page	23	As a site member, I want to be able to look for different categories of brands, so that I can choose what I want.	100	400
4	Payment Process	18	As a site member, I will be able to make payment, so that my deliveries can be shipped.	40	240
5	Contact Page	3	As a site member, I want to be able to find contact information of the site, so that in case I need, I can contact.	13	80
6	Banner Area	1	As a marketing personnel, I want to be able to make advertisement, so that I can attract visitors	8	40

TP 1 : Concevoir un product Backlog

Concevoir le Product Backlog de votre projet tutoré

Scrum : les artefacts

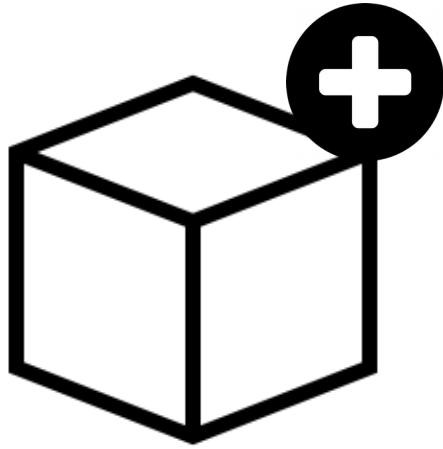
Sprint Backlog



- Reprend du Product Backlog les User Story sélectionnées pour être développées pendant un Sprint
- Il est maintenu par l'équipe de développement
- Un Sprint Backlog est mis en place pour chaque Sprint et évolue tout au long des développements

Scrum : les artefacts

Incrément de produit



- Concaténation d'un ensemble de User Story du Product Backlog intégrées au logiciel
- Réalisé au cours d'un sprint, il comprend les User Story réalisées et terminées au cours du sprint ainsi que les User Story terminées durant les sprints précédents
- La responsabilité de l'incrément revient à l'équipe de développement

Scrum : les artefacts

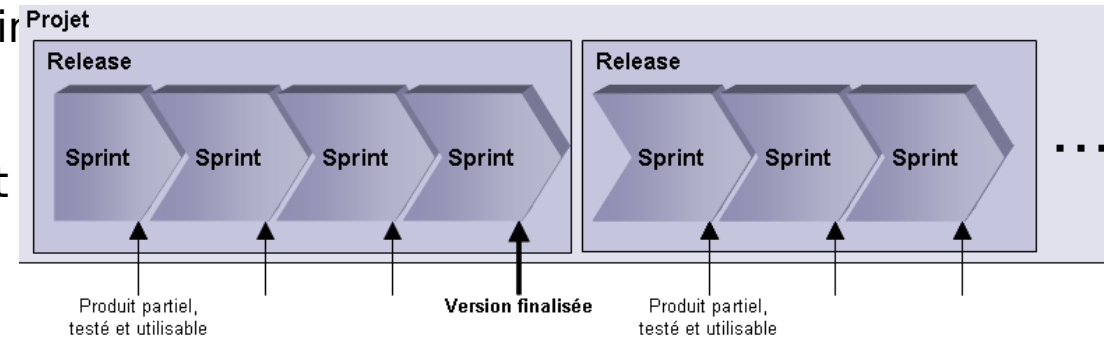
Definition of Done (DoD)



- Définie en début de projet pour que tous les acteurs aient la même conception de ce que signifie avoir terminé une tâche
- Une tâche non terminée lors d'un sprint :
 - ✓ N'est pas prise en compte dans le calcul de la vélocité
 - ✓ N'est pas présentée lors de la revue de Sprint
 - ✓ Ne fait pas partie de l'incrément en fin de Sprint

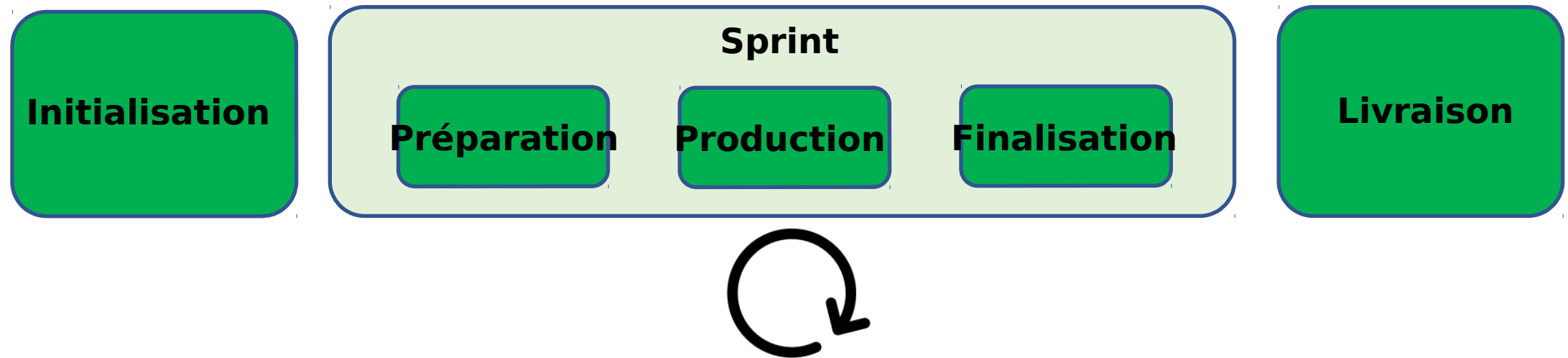
Scrum : les événements (ou cérémonies)

- Les événements de Scrum entrent dans une logique de **TimeBox**
- Une TimeBox doit respecter deux principes : **Limiter le temps à passer sur un événement donné et n'accepter aucun dépassement**
- La *préparation de sprint*, le **sprint**, le *Daily Meeting*, la *revue de sprint*, la *rétrospective de sprint* sont des événements
- Le **sprint** est la base de découpage des projets sous Scrum
- Objectif d'un sprint : proposer une itération supplémentaire qui ajoute de la valeur au produit
- Durée préconisée de 4 semaines maximum, tous les sprints
- Aucun temps mort entre deux sprints, un nouveau sprint démarrant directement à la fin du précédent



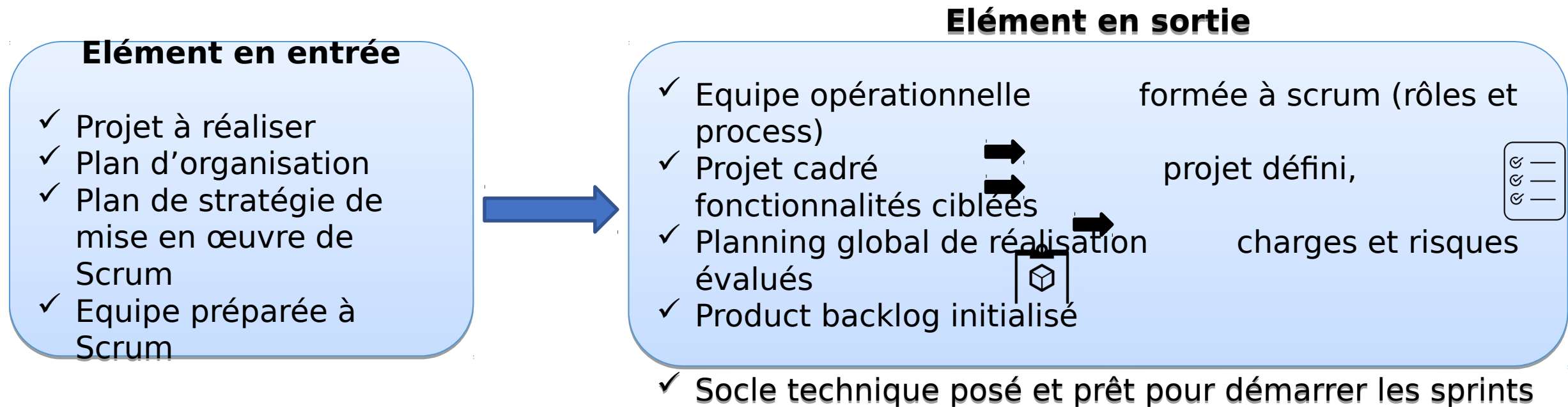
Scrum : le processus

- Cinq grandes phases dans le processus global de Scrum



Scrum : la phase d'initialisation

Cette phase (appelée aussi Sprint 0) permet de poser toutes les bases du projet qui vont permettre ensuite de démarrer sereinement l'enchaînement des sprints



Initialisation

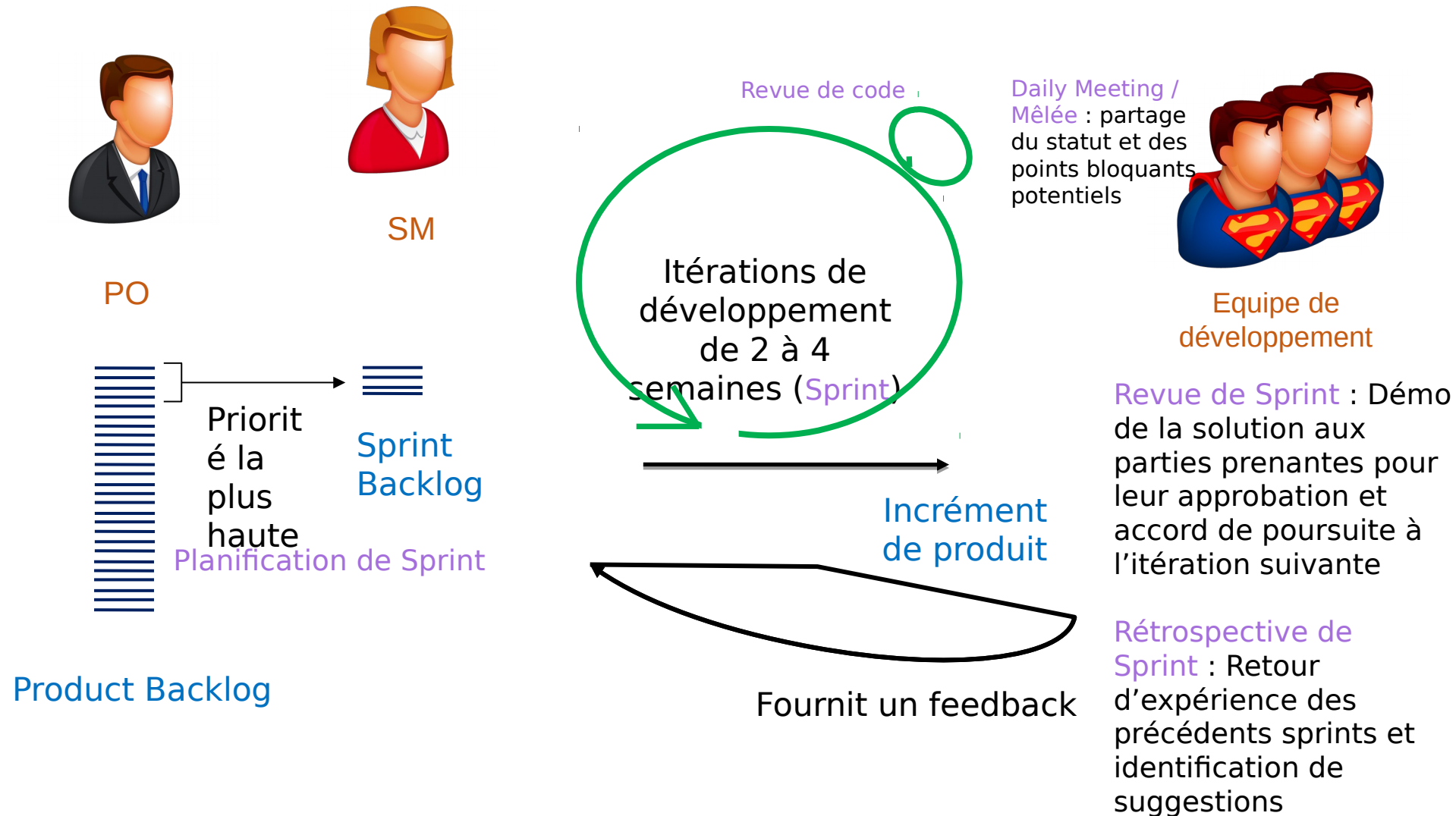
Préparation

Production

Finalisation

Livraison

Scrum : les phases de préparation, production et finalisation



Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : la phase de préparation

Cette phase permet de préparer la production en mettant en place le Sprint Backlog



Initialisation

Préparation

Production



Finalisation

Livraison

Scrum : la phase de préparation




La phase de préparation comprend deux étapes

Préparer le Sprint

- ❑ Objectifs : préciser le contenu du prochain sprint
- ❑ Sortie : liste de User Story respectant les critères INVEST (Independent, Négociable, Valuable, Estimable, Size, Test)
- ❑ Outils / support : **MoSCoW** (**M**ust, **S**hould, **C**ould, **W**on't) fix
- ❑ Qui ?  



Organiser le Sprint

- ❑ Objectifs : valider le contenu du prochain sprint et définir les différentes tâches à réaliser pour chaque User Story
- ❑ Outils /support : Poker planning,
- ❑ Événement de *planification de Sprint*
- ❑ Qui ?   

Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : la planification de Sprint

Quand : au démarrage du sprint

Durée : 8 heures maximum pour un sprint d'un mois

Objectif : Sélectionner dans le Product backlog les US à réaliser, les découper en tâches, évaluer le travail (poker planning) et prioriser

Acteurs : le SM organise, le PO aide à la décision des US. L'équipe de développement sélectionne les US et les découpe en tâches.

Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : le poker planning (1/4)

- ❑ Produire des estimations de l'effort sur les User Story à développer présentes dans le Product backlog
- ❑ Participants avec des niveaux d'expérience et d'expertise différents
- ❑ Favorise les échanges entre le responsable de produit et l'équipe de développement
- ❑ L'estimation se fait en unités d'œuvre intitulées Story Point
- ❑ La suite de Fibonacci est utilisée pour les évaluations (plus le scénario est gros, moins l'évaluation est précise) et des symboles équivoques (0 = US déjà développée, etc...)



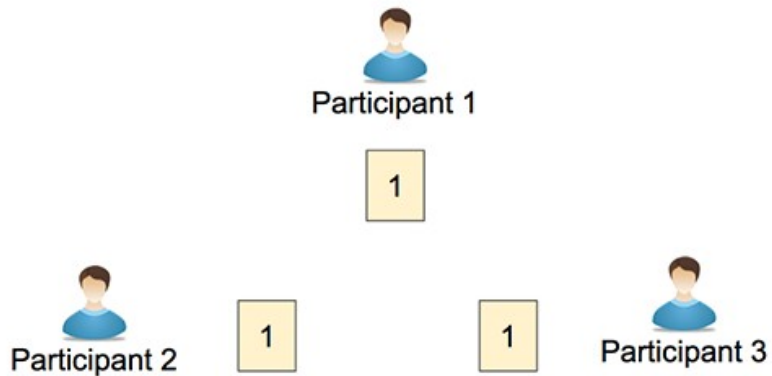
Scrum : le poker planning (2/4)

1. Les participants s'installent autour d'une table, de façon à ce que tout le monde puisse se voir
2. Le PO explique à l'équipe une fonctionnalité / User Story



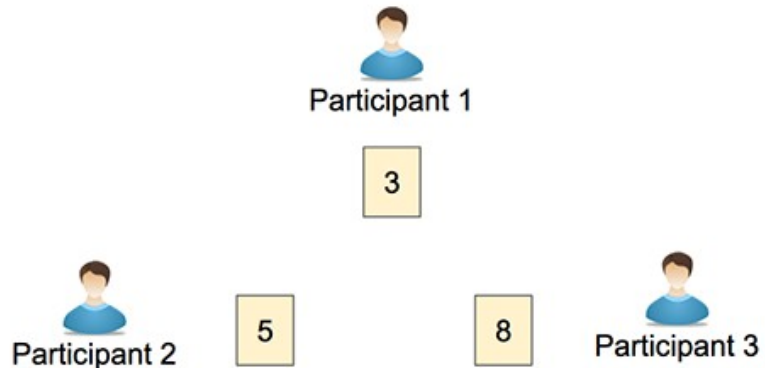
3. Les participants posent des questions au PO, discutent du périmètre de la User Story, évoquent des conditions de satisfaction qui permettront de la considérer comme terminée
4. Chacun des participants évalue l'effort de développement de la User Story, choisit la carte qui correspond à son estimation et la dépose, face vers le bas, sur la table devant lui
5. Au signal du SM, les cartes sont retournées en même temps. 4 Scénarios possibles

Scrum : le poker planning (3/4)



Entente parfaite

- Tous les participants votent la même complexité, on valide alors cette complexité 1 sur la user-story



Consensus 1

- Tous les participants ne votent pas la même chose mais l'écart reste correct (soit pas plus de 2 rangs d'écart entre la plus petite complexité et la plus grande). On validera alors la complexité 5 soit celle du milieu



Initialisation

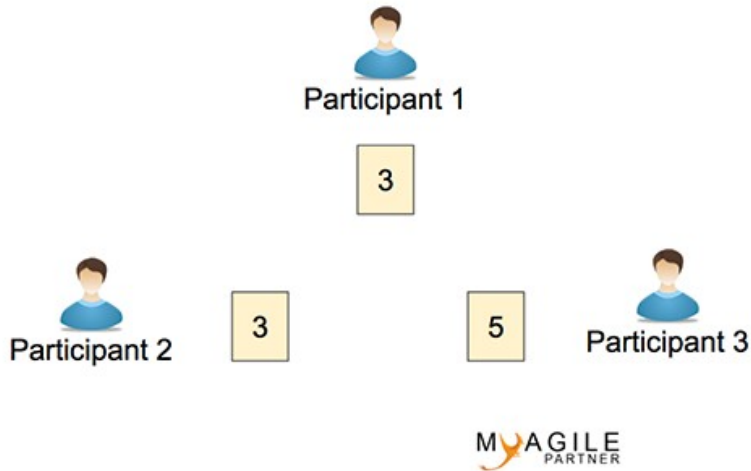
Préparation

Production

Finalisation

Livraison

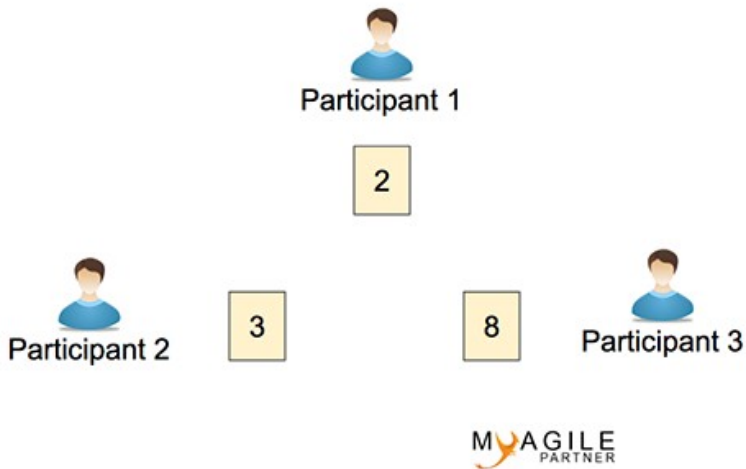
Scrum : le poker planning (4/4)



Consensus 2

- Tous les participants ne votent pas la même chose mais l'écart reste correct. On validera alors la complexité 5 soit la plus haute même si seul 1 participant a mis cette complexité

A revoter



- Tous les participants ne votent pas la même chose et l'écart est trop grand
- Après une bonne discussion, les développeurs revoteront la complexité de la User Story afin de se retrouver dans l'un des 3 premiers cas cités
- Si il n'y a pas d'unanimité, on demande aux deux acteurs ayant produit les évaluations extrêmes d'expliquer leurs points de vues respectifs. La discussion reprend
- On répète le processus d'estimation jusqu'à l'obtention de l'unanimité

Scrum : la phase de production

Cette phase lance la réalisation de l'ensemble des tâches définies dans le Sprint Backlog



Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : la phase de production

Découpage de la production

- Dans une méthode itérative, développer un logiciel nécessite de découper ce qui va être produit
- Ce découpage fonctionnel se fait sur plusieurs niveaux de granularité (de portfolio à tâche) :
 - ❑ **Portfolio** : vision globale du produit qui peut regrouper plusieurs applications
 - ❑ **Application** : focalisé sur un seul logiciel découpé en thèmes
 - ❑ **Thème** : module d'une application qui regroupe plusieurs fonctionnalités
 - ❑ **Epic** : définition grossière d'une ou plusieurs fonctionnalités qu'il faudra préciser ou redécouper en plusieurs US
 - ❑ **User story** : définit une fonctionnalité au format « en tant ...je peux ...dans le but de... »
 - ❑ **Tâche** : découpage d'une user story qui précise les actions à effectuer lors de la réalisation

Initialisation

Préparation

Production

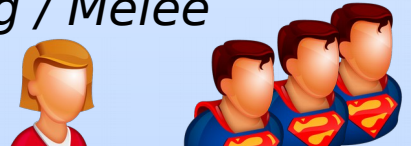
Finalisation

Livraison

Scrum : la phase de production

La phase de production comprend deux étapes

Etat des lieux

- ☐ Objectifs : rester au contact du produit
- ☐ Outils /supports : scrum board, mesures et suivi (vélocité , burndown et burnup chart, team mood)
- ☐ Événement de *Daily Meeting / Mêlée*
- ☐ Qui ? 



Transformation*

- ☐ Objectifs : réalisations techniques des tâches et tests fonctionnels
- ☐ Qui ? 

* Pas abordé dans ce cours, mais dans vos TP de programmation ;-)

Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : le Daily Meeting (Mêlée quotidienne)



- Le sprint est ponctué chaque jour par un point d'équipe appelé Daily meeting, organisé par le SM
- Celui-ci a pour objectif de communiquer sur le projet, rythmer les développements, donner une vision globale à toute l'équipe et anticiper les problèmes
- Chaque daily Meeting dure au maximum 15 minutes (réunion debout) durant chaque membre de l'équipe de développement répond à trois questions :
 - **Qu'est ce que tu as réalisé hier ?**
 - **Que vas-tu faire aujourd'hui ?**
 - **Est-ce que tu as rencontré des problèmes ?**

Initialisation

Préparation

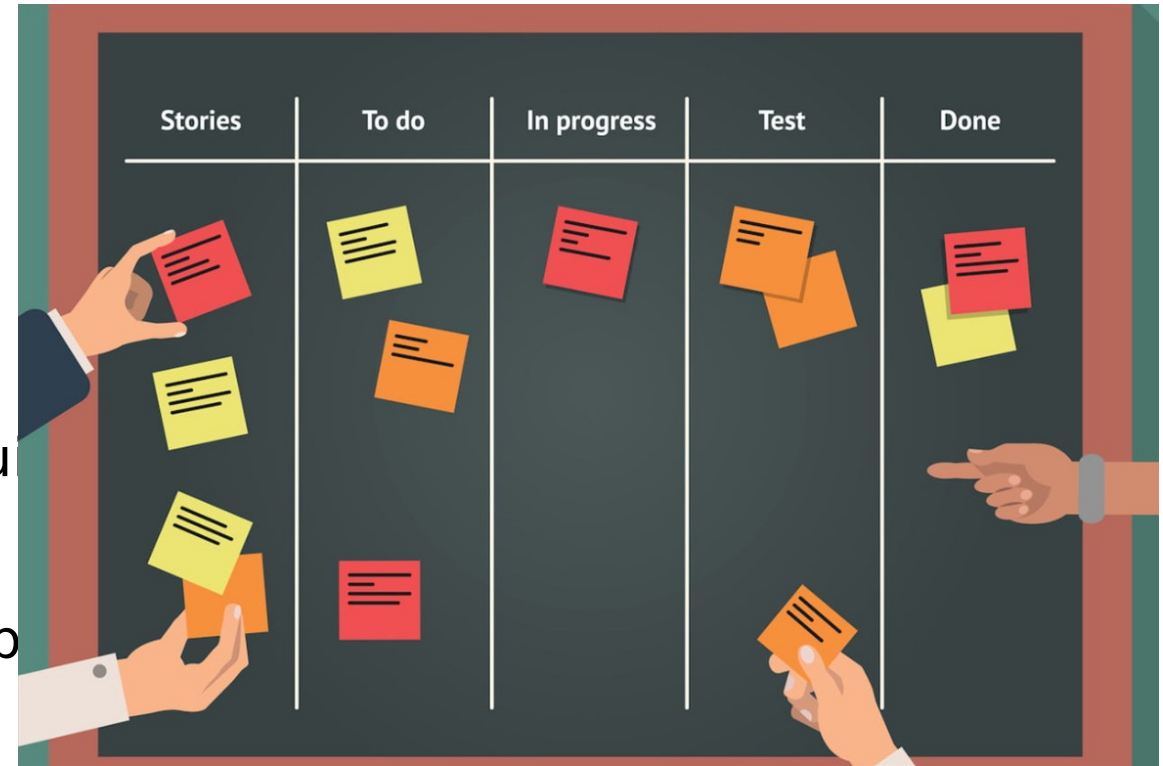
Production

Finalisation

Livraison

Scrum : le scrum board

- Un des éléments de communication Scrum
- Visible en permanence de l'ensemble de l'équipe
- Vue d'ensemble de la progression des développements
- Tableau découpé au moins en trois parties (à faire, en cours, fait) sur lequel on appose des post-it
- Chaque post-it est attribué à une User Story du sprint en cours
- En fonction de l'état d'avancement d'une tâche, on déplace le post-it



Initialisation

Préparation

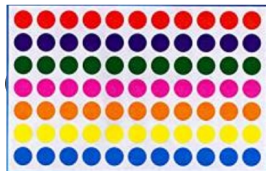
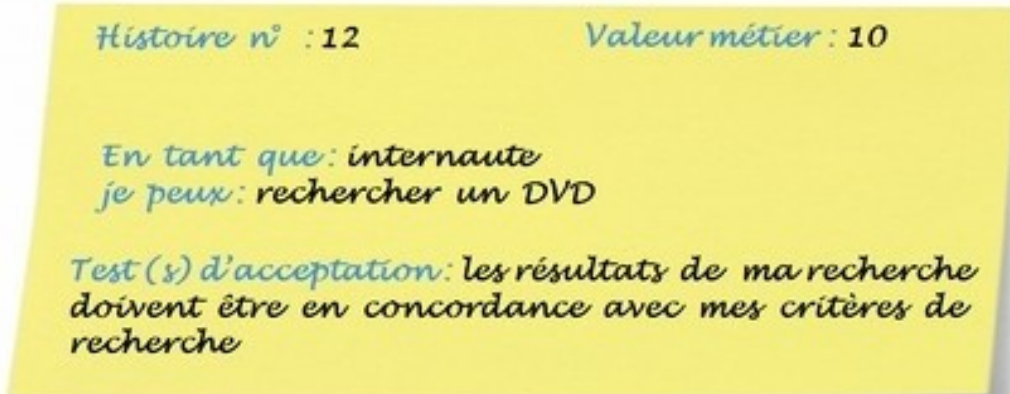
Production

Finalisation

Livraison

Scrum : le scrum board

- ✓ Reprend les éléments de la User Story associée
- ✓ De couleur et de tailles différentes (sprint différents, épics ou user Story,...)
- ✓ Utilisation de marqueur et de feutre pour être visible de loin (outil avant tout visuel !)
- ✓ Possible de mettre pastille ou initiales personne en charge de la User Story



Initialisation

Préparation

Production

Finalisation

Livraison

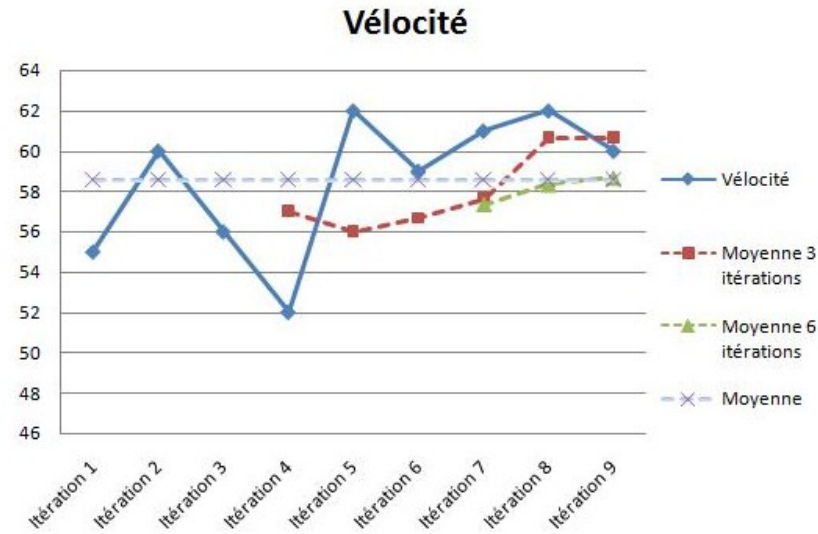
Scrum : mesures et suivis

- Au cours du sprint, il est impératif d'assurer le suivi au quotidien et donc de mettre en place des moyens de mesure : **avancement produit** et **moral de l'équipe**
- Le but est de suivre l'évolution du projet, de détecter des anomalies, de lever les alertes et d'être pro-actif dans la recherche d'amélioration
- Les principaux indicateurs d'avancement (*vélocité*, *burndown* et *burnup Chart*) fondent leurs calculs sur les story point
- Il est de bonne pratique de positionner sur le scrum board ces différents indicateurs



Scrum : la vélocité

- La vélocité est un indicateur pour suivre le travail réalisé d'une équipe sur plusieurs sprints
- Vélocité d'un sprint = **somme des story points des User Story terminées durant ce sprint**



- Pas une courbe de prod. croissante !
- Chaque sprint est différent : présence équipes, séminaire, périodes stratégiques (petites US rapides), ...

Initialisation

Préparation

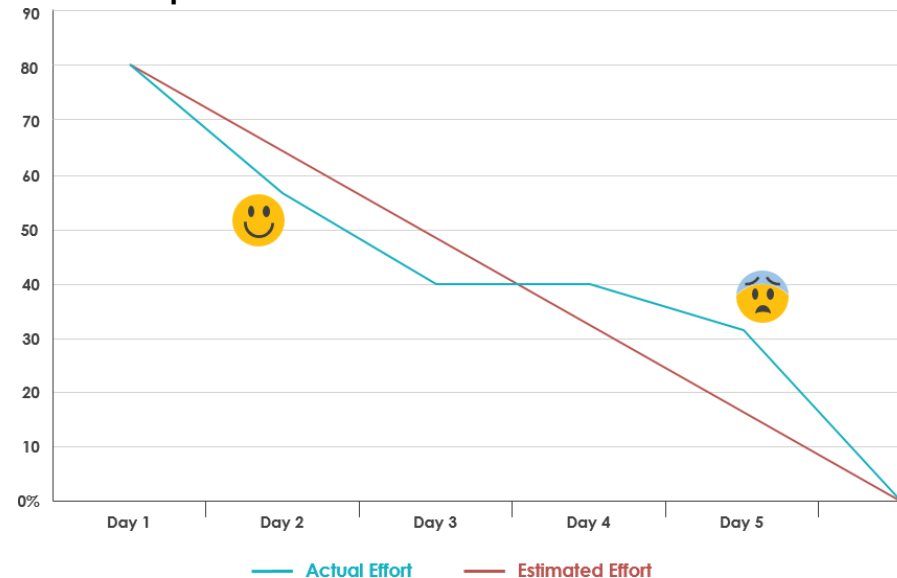
Production

Finalisation

Livraison

Scrum : burndown chart

- Indicateur symbolisé par une représentation graphique qui propose 2 courbes à tendance décroissante au cours du temps



- Effort idéal / estimé : courbe représentant le restant à réaliser dans le cas idéal
- Effort réel : courbe représentant ce qu'il reste véritablement à réaliser
- L'objectif est que le réel reste proche de l'idéal

Initialisation

Préparation

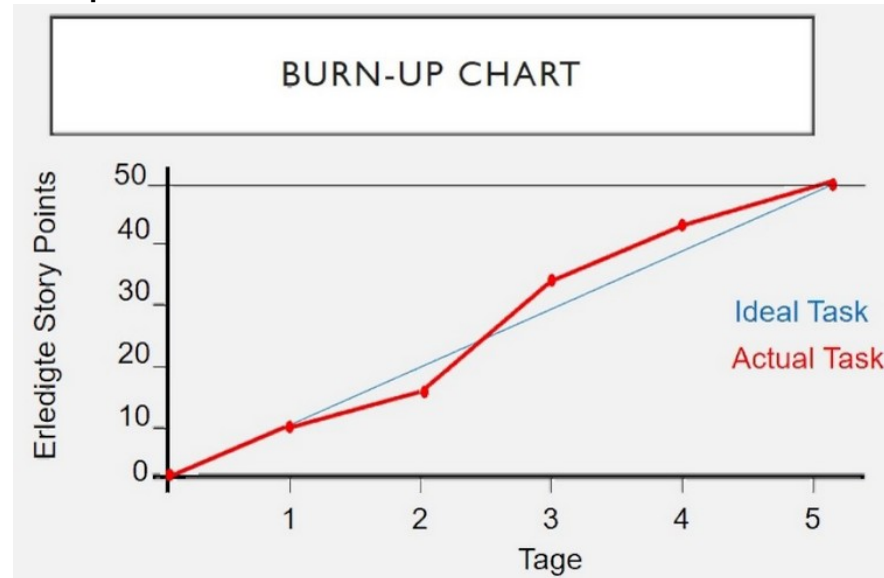
Production

Finalisation

Livraison

Scrum : burnup chart

- Indicateur symbolisé par une représentation graphique qui propose 2 courbes à tendance croissante au cours du temps



- Nombre de points idéal / estimé : courbe représentant le nombre de story point pour finir le sprint
- Effort réel : nombre de story point réalisés
- L'objectif est que la seconde rejoigne la première sans la dépasser

Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : Team Mood

- Indicateur qui représente l'état d'esprit de l'équipe
- Peut être représenté par une note ou une représentation iconographique (météo, smiley, ...)

contact@teammood.com contact@teammood.com
to me

Bonjour Nicolas,

Alors, comment s'est passée cette journée ?

Merci de cliquer sur le terme qui vous semble le plus approprié :

- [Excellente journée](#)
- [Bonne journée](#)
- [Bof, journée moyenne](#)
- [Journée difficile](#)
- [Mauvaise journée](#)

En indiquant votre ressenti, vous pourrez visualiser l'ambiance globale au sein de votre équipe.

A bientôt,
Nicolas, de TeamMood

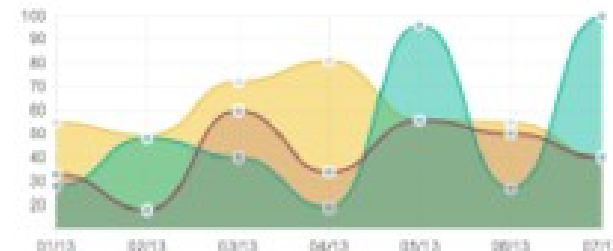
Average team's mood



Yesterday



Today



Initialisation

Préparation

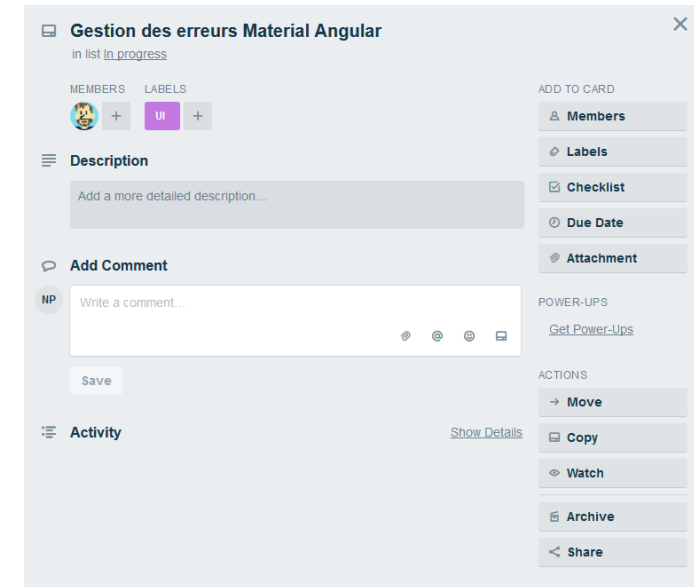
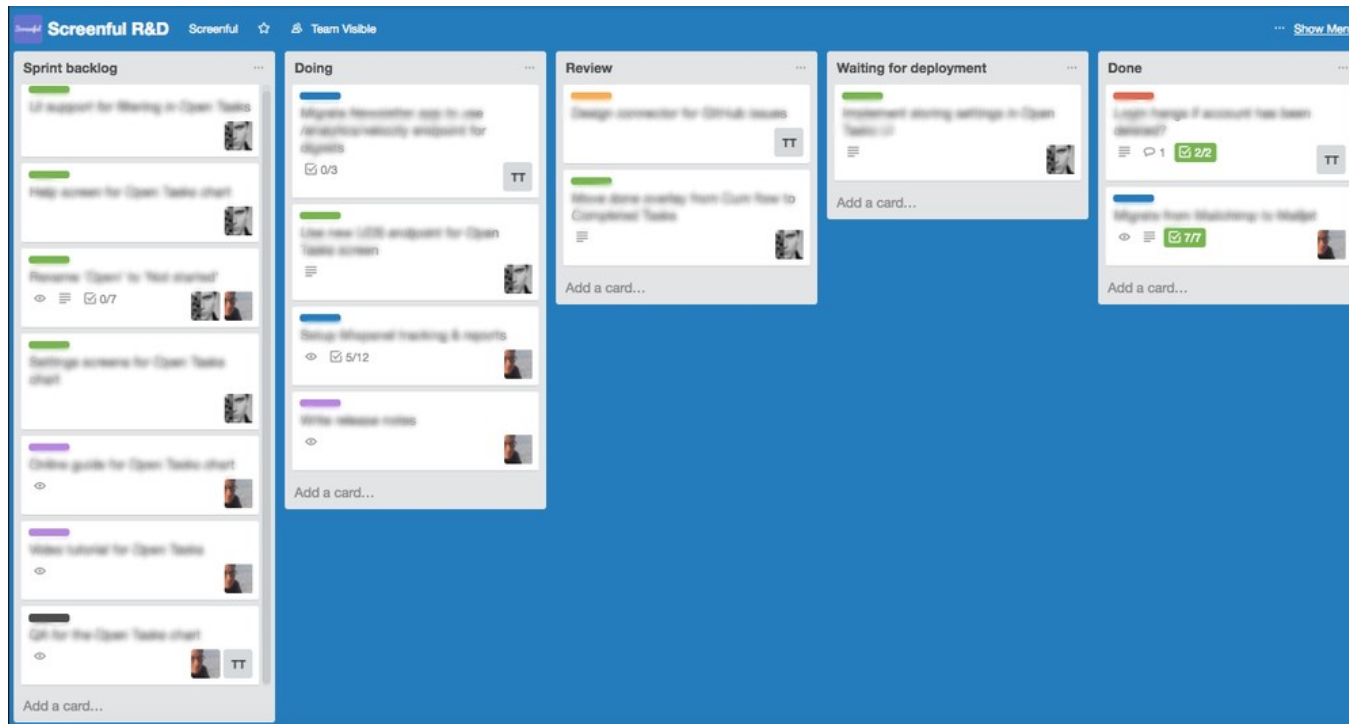
Production

Finalisation

Livraison

Scrum : les alternatives numériques (Trello)

- Equivalent numérique du scrum board et de ses post-it, qui peut être affiché sur n'importe quel écran et partagé par tous les intervenants du projet (délocalisés...)
- Statistiques et mesures générées automatiquement (JIRA, Swift Kanban, Asana, ...)



Initialisation

Préparation

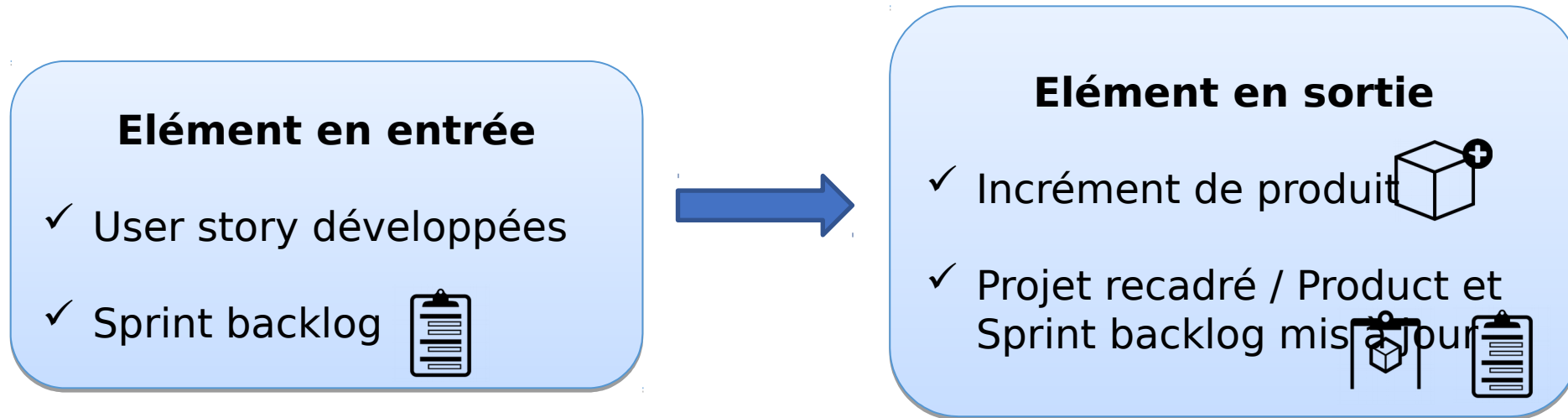
Production

Finalisation

Livraison

Scrum : la phase de finalisation

Cette phase a pour objectif la livraison de fonctionnalités répondant aux exigences



Initialisation

Préparation

Production

Finalisation

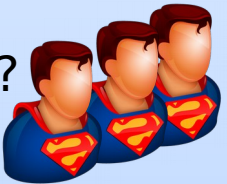
Livraison

Scrum : la phase de finalisation

La phase de finalisation s'articule autour de trois étapes

Revue de Code

- ❑ Objectifs : assurer la qualité du livrable / partager
- ❑ Événement de *revue de code*
- ❑ Qui ?



Revue de Sprint

- ❑ Objectifs : valider les développements
- ❑ Événement de *revue de sprint*
- ❑ Qui ?



Rétrospective de Sprint

- ❑ Objectifs : relever les risques sur le projet et établir un plan d'action
- ❑ Événement de *rétrospective de Sprint*
- ❑ Qui ?



Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : la revue de code et les tests croisés

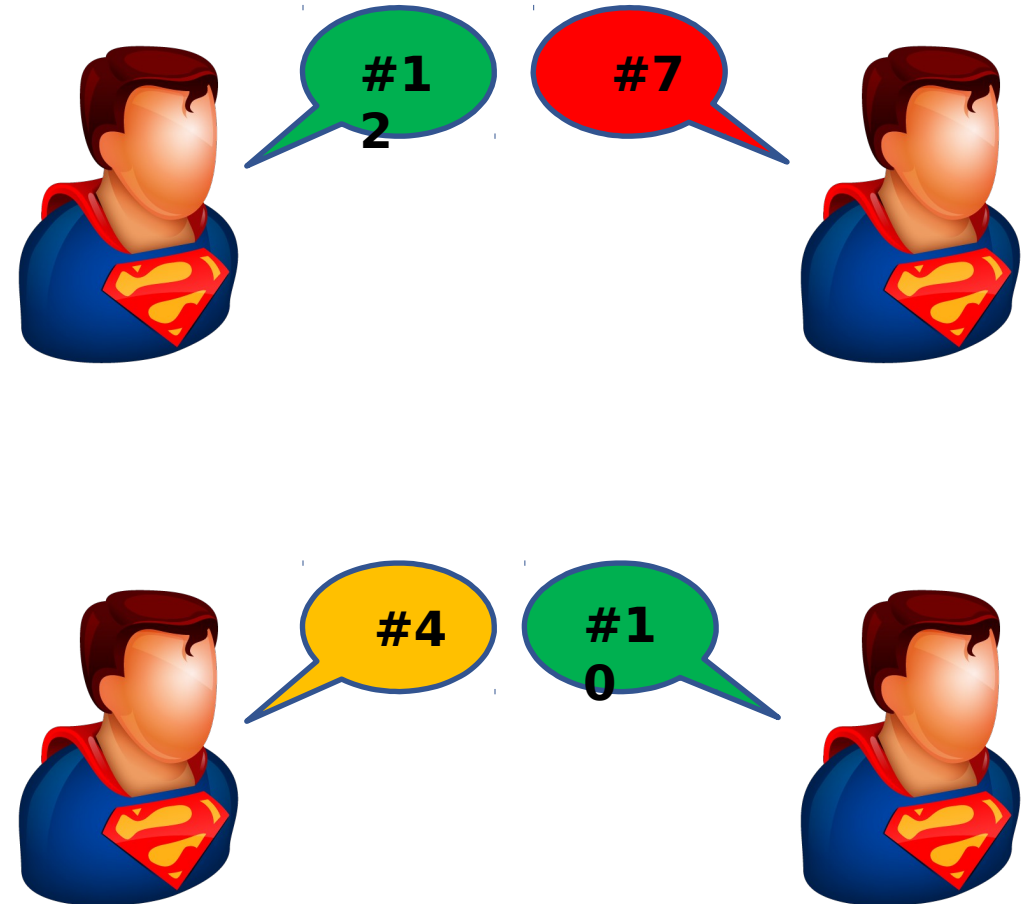
Quand : pendant et à la fin du sprint

Objectif : améliorer la qualité du produit livré et favoriser la collaboration en parcourant le code de chacun

Acteurs : l'équipe de développement

Avantages : détecter des anomalies, appliquer un standard, formation

Alternatives : réunion commune ou choisir une US parmi celles terminées (croisées avec variation)



Initialisation

Préparation

Production

Finalisation

Livraison

Scrum : la revue de Sprint

Quand : à la fin du sprint

Durée : 4 heures maximum pour un sprint d'un mois

Objectif : présenter le sprint et démontrer que chaque User Story développée correspond aux attentes. Seuls les US terminées sont présentées, seules celles validées pendant la revue font partie de l'incrément de produit

Acteurs : le SM organise, l'équipe de développement présente les développements, le PO valide ou non les US et maintient le Product backlog en fin de sprint. Le commanditaire (client) assiste à la revue et participe à la validation

Initialisation

Préparation

Production

Finalisation

Livraison

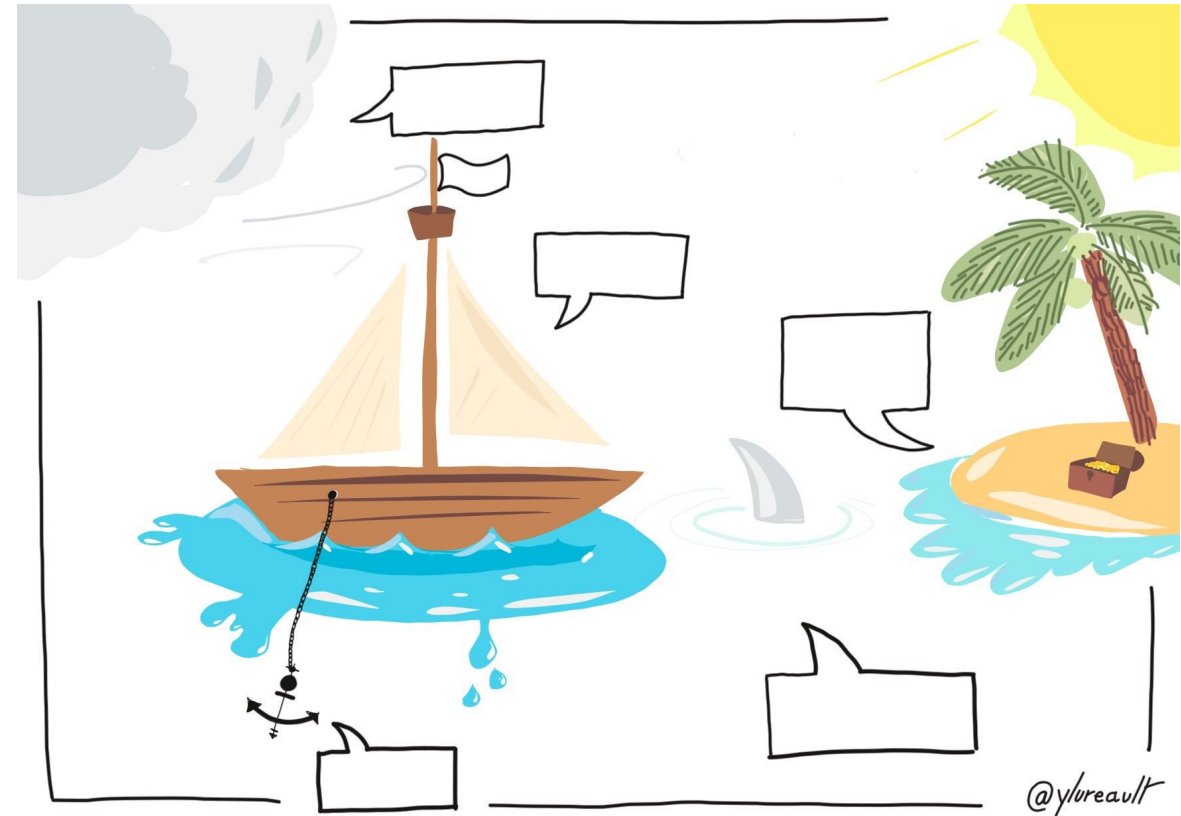
Scrum : la rétrospective de Sprint

Quand : à la fin du sprint

Durée : 3 heures maximum pour un sprint d'un mois

Objectif : présenter ce qui a fonctionné ou non pendant le sprint et établir un plan d'action pour améliorer l'organisation

Acteurs : le SM organise, l'ensemble de la scrum Team échange et intervient



Speed Boat

Initialisation

Préparation

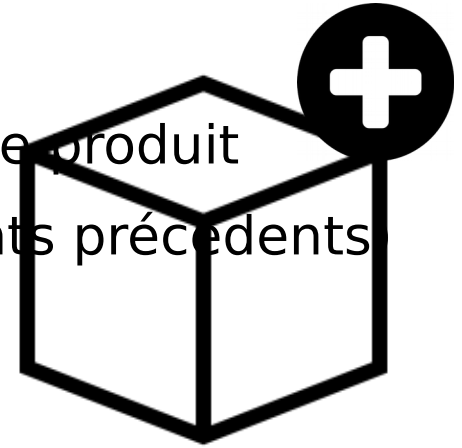
Production

Finalisation

Livraison

Scrum : livraison de l'itération et après...

- Livraison au PO (et donc au commanditaire) du dernier incrément de produit (qui contient les US du dernier sprint ainsi que les US de tous les sprints précédents)
- Deux cas de figure sont à prendre en compte à la fin d'un sprint
 - 1) Vers un nouveau sprint : phase de préparation du sprint suivant
 - 2) Fin du projet : attente de démarrage d'un autre projet



Initialisation

Préparation

Production

Finalisation

Livraison

Etre agile avec scrum

- ✓ Le premier moyen de mettre de l'agilité dans Scrum est d'altérer certains de ses concepts en les modifiant ou en les supprimant
- ✓ Méthodes se basant sur du Scrum sans en être réellement
- ✓ Pour ne pas perdre totalement l'essence de Scrum, il convient de conserver à minima sa philosophie et ses piliers, la notion de sprint et au moins sa rétrospective, La notion d'incrément de produit

Compléter et étendre Scrum

- Le second moyen d'adaptation de Scrum consiste à le compléter avec des méthodes lui permettant d'améliorer son efficacité

Méthode	Principe
Lean Management	Basée sur 7 principes applicables dans une démarche Scrum : éliminer les déchets, améliorer la qualité de production, générer de la connaissance et l'organiser, différer les engagements (en retardant les décisions irréversibles), livrer rapidement, respecter les membres de l'équipe, optimiser l'ensemble et avoir une vision globale du produit
Kanban	Basée sur 5 principes : faciliter la visualisation (tableaux de bord Kanban), Limiter le WIP (Work In Progress), maîtriser les étapes de production grâce à des indicateurs avec la possibilité de stopper plus facilement la production si problème majeur, assurer la transparence règles, notamment celles permettant de définir le changement d'état d'une tâche, s'inscrire dans une démarche d'amélioration continue
Scrumban	Scrum + Kanban Planification du sprint suivant effectué dès lors qu'il reste un nombre de tâches limitées dans le sprint courant (les sprints ont donc des durées variables), une limite de WIP est imposée, les itérations ne doivent pas dépasser 2 semaines, Système de pull : un membre de l'équipe choisit une nouvelle tâche dès qu'il en a terminé une
DevOps	Peut être vu comme une évolution de l'agilité qui permet de réunir dans une même équipe les compétences de développement (Dev) et d'exploitation (Ops) pour optimiser la chaîne de production logicielle. Cette organisation est totalement compatible avec Scrum et impacte directement l'équipe de développement qui doit 1) intégrer les compétences Ops 2) Mettre en place un processus de production unifié, allant du développement jusqu'au déploiement automatisé de l'application
Safe, spotify	Plus grande échelle (Equipe de développement > 9 membres). Safe (comme spotify) introduit les notions de programme et de portefeuille. Le but est de cadencer le travail de plusieurs équipes, centrées chacune sur une brique logicielle, dans l'optique de les aligner sur une politique produit cohérente

Conclusion

- ❑ Approche très itérative de la gestion de projet
- ❑ Cycles de travail courts, à la fois rigoureux et flexibles
- ❑ Respect des rôles de chacun
- ❑ Grande autonomie et liberté à l'ensemble de l'équipe
- ❑ Transparence, inspection et adaptation
- ❑ Adapter l'outil aux besoins du projet
- ❑ Grandes entreprises gèrent des projets en Scrum et accompagnent dans la formation



Quelques outils

Lors de la mise en œuvre de Scrum, il est important de réunir l'ensemble des outils nécessaires pour respecter une bonne organisation de travail

➤ Redmine

- Interface web sécurisée pour gérer les utilisateurs, les demandes, les documents du projet

➤ Jira

- Suite logicielle en mode SaaS – gestion orientée tickets pour gérer les utilisateurs, les demandes, les documents du projet

➤ Trello

- Outil SaaS pour faciliter des outils Scrum comme le ScrumBoard, le Burndown chart, etc...

➤ IceScrum

- Solution logicielle Open Source pour gérer la durée des sprints, estimer la durée des tâches, créer des User Story

➤ ScrumBoard

- Matériel – tableau blanc sur lequel sont collés les post-it

➤ Tableur

- Logiciel – tableur excel, Google Spreadsheet

Formations / Certifications

- Les certifications sont principalement délivrées par :



- Organismes fondés respectivement par Jeff Sutherland et Ken Schwaber, les fondateurs de scrum
- Noms des certifications diffèrent selon les organismes, mais celles-ci restent tout de même assez Proches en termes de reconnaissance



Références

- <http://agilemanifesto.org/iso/fr>
- <https://www.scrumalliance.org>
- <https://www.scrum.org>
- <http://scrummethodology.com/>



KANBAN

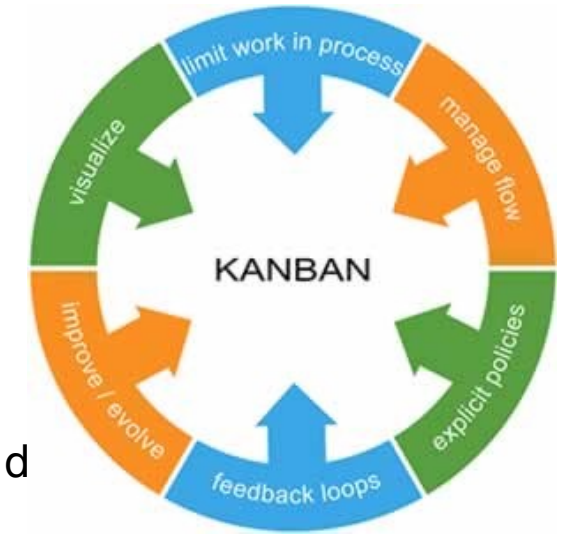


- Créé par Toyota au Japon dans l'industrie des années 60
- Peut se traduire par « étiquette », « panneau »
- A l'origine, système d'étiquettes associées aux pièces des lignes d'assemblage et permettant de respecter une production en flux tendu (juste à temps) != flux poussé ?
- Formalisé pour le contexte logiciel par David J.Anderson en 2010
- Fait partie de l'eco-système des méthodes agiles
- Son but : fluidifier le processus du développement afin de diminuer les délais de production tout en produisant à la demande

KANBAN

On y retrouve la logique de flux tendu qui se traduit par les principes suivants :

- ❑ Faciliter la visualisation du processus au travers d'outils tels que les tableaux d bord Kanban qui permettent de voir l'état des tâches à chaque instant
- ❑ Limiter le WIP (Work in Progress), le nombre de tâches en cours à un instant donné est limité. Aucune nouvelle tâche ne peut être démarrée par l'équipe tant que la limite de WIP est atteinte
- ❑ Maîtriser les étapes de production en assurant le suivi et la mesure. En fonction des résultats, l'équipe prend les mesures nécessaires et peut bloquer la production le temps de résoudre un problème majeur
- ❑ Transparence des règles qui doivent être disponibles et compréhensibles par tous les membres de l'équipe (notamment les règles de changement d'état des tâches)
- ❑ S'inscrire dans une démarche d'amélioration



Différences entre **Scrum** et **Kanban**

(1/4)

L'essence du Kanban et ses principes de base convergent en partie avec les bases de Scrum => On y retrouve ainsi le principe d'**empirisme** et donc **d'apprentissage par l'expérience** basé sur les piliers *Transparence, Inspection* et *adaptation*

Il existe cependant un certain nombre de différences :

3 rôles

X rôles

Equipe de développement
= 1 seule entité
autoorganisée aux
multiples compétences

Plusieurs équipes de
développements avec leur propre
compétence et dirigée par des
membres extérieurs

Timeboxing /
cloisonnement temporel
imposé

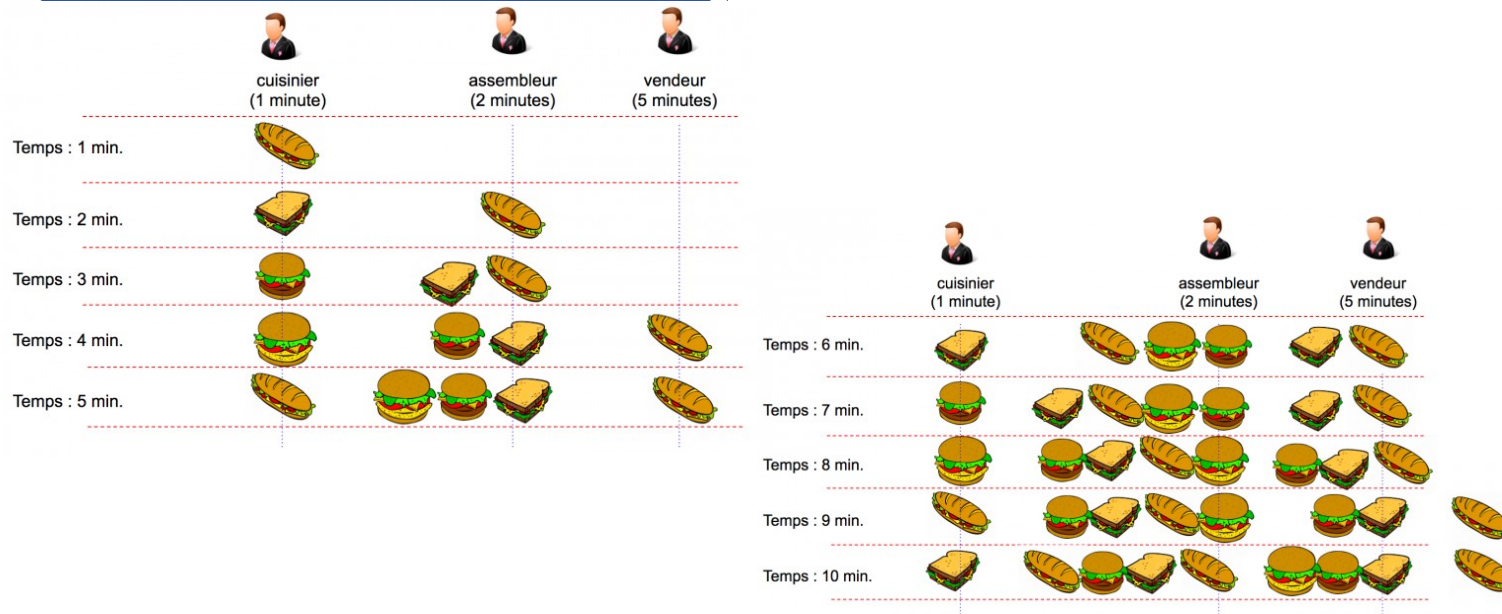
Timeboxing non imposé /
peut évoluer

Différences entre Scrum et Kanban (2/4)

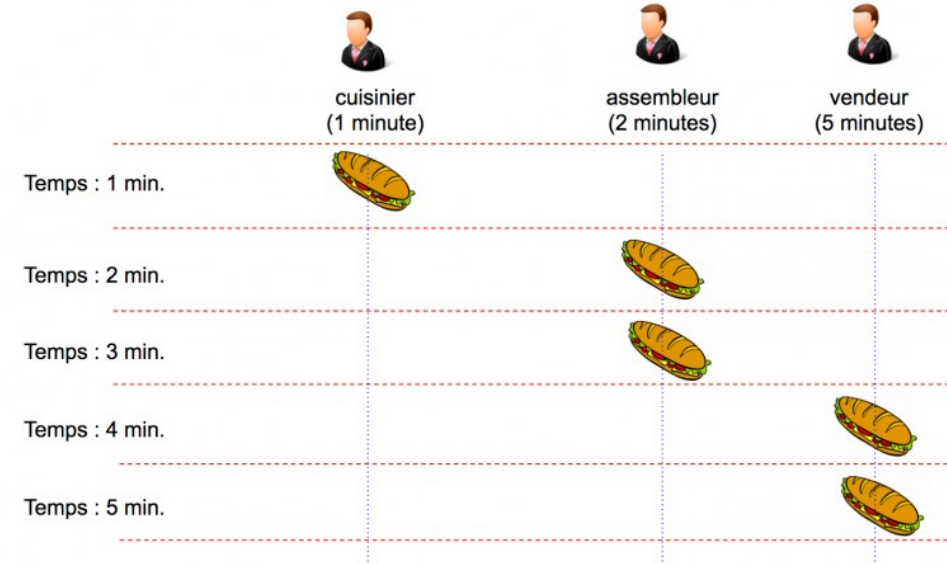
Sprint backlog (et priorisations) défini et extensible

Vision non précise de toutes les tâches du sprint (prises au fur et à mesure)

Flux poussé



Flux tiré ou « Juste à temps » : éviter la surproduction



Différences entre **Scrum** et **Kanban**

(3/4)

Processus entièrement chargé
en début de sprint et réinitialisé
à la fin

Processus en flux continu :
flux tiré avec limitation du
travail à faire / éviter la
surproduction

Nombre de tâches limitées par
Estimations story point (prépa)
et capacité équipe (In Progress)

Limite de WIP (Work In
Progress)

Processus figé (étapes
avec timebox)

(favorise
l'expérimentation et
processus de création

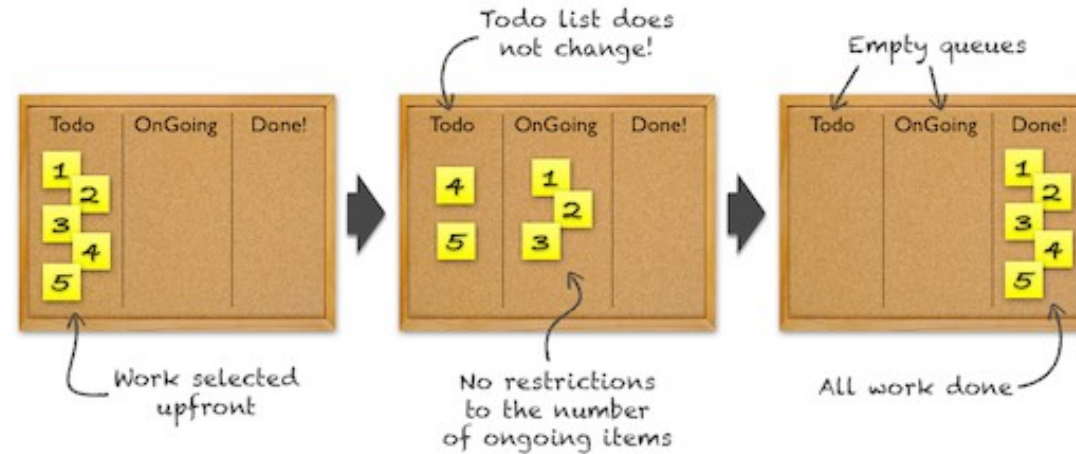
Différences entre Scrum et Kanban

(4/4)

SCRUM BOARD

1 Sprint = 1 Board

1 Board = 1 équipe pluri-disciplinaire

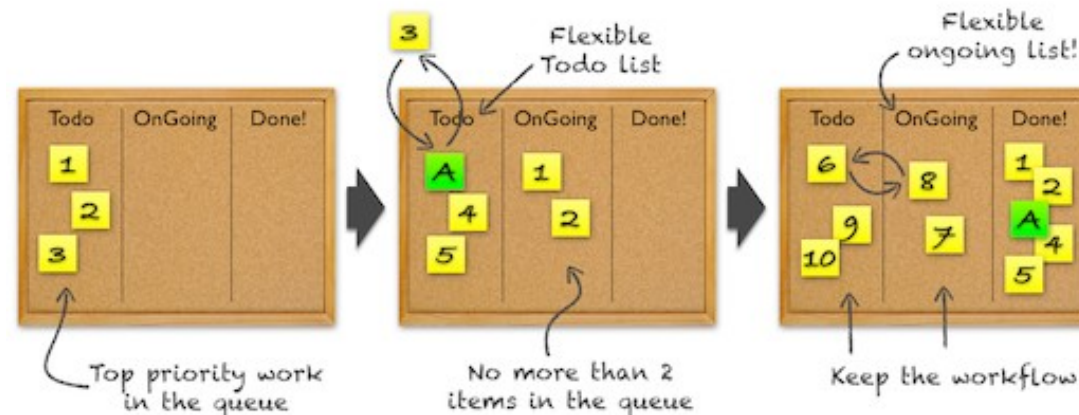


1 projet = 1 Board

1 Board = X équipes mono/pluri-disciplinaire

KANBAN BOARD

WIP non limitées



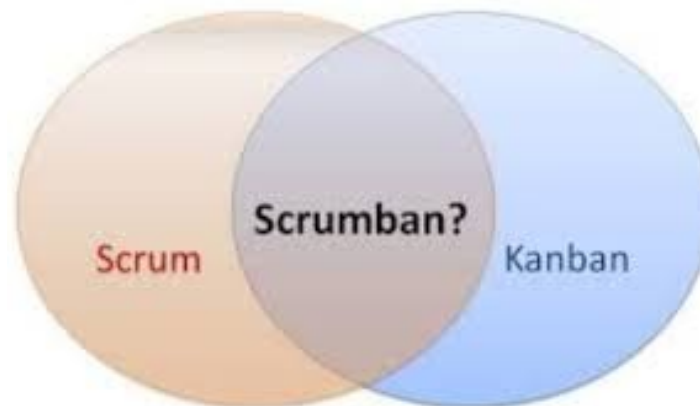
WIP limitées

Différences entre Scrum et Kanban : Bilan

- ❑ Kanban moins directif que Scrum, Scrum plus normatif que Kanban
- ❑ Cependant, les deux méthodes ne sont pas exclusives et peuvent se rapprocher l'une de l'autre
- ❑ Kanban peut être une première étape pour des équipes qui souhaiteraient utiliser Scrum à terme (ou simplement rester sur du kanban)
- ❑ Par exemple il est tout à fait possible avec kanban de faire évoluer son processus en ajoutant au fur et à mesure des spécificités de Scrum telles que : des itérations fixes, la mise en place de PO et SM, etc.

Scrum + Kanban = Scrumban

- ❑ Les méthodes Scrum et Kanban ne sont donc pas exclusives
- ❑ Certains ont imaginé en 2008 une méthode basée sur la fusion de ces deux méthodes : Scrumban
- ❑ Au-delà d'une fusion, il s'agit en fait d'une intersection permettant de faire tendre Scrum vers Kanban



Scrumban

- Objectif : apporter une capacité à mieux gérer la production en flux tendu, dans le cas par exemple où l'équipe croule sous les demandes de changements
- Pas une extension de Scrum : méthode à part entière qui emprunte à la fois à Scrum et à Kanban
- Les différences avec Scrum portent sur les points suivants :
 - ✓ La planification de sprint remplacée par un « **planning on demand** ». Pas à une date fixe mais déclenché lorsqu'il ne reste qu'un nombre limité de tâches « à faire » sur le board pour l'itération courante. En d'autres termes, on démarre le travail sur l'itération suivante lorsque la majorité des tâches du sprint courant sont terminées (et non pas sur la base d'un planning défini)
 - ✓ Mise en place d'une **limite de WIP** comme sur Kanban
 - ✓ Les **itérations doivent rester très courtes** et ne pas dépasser une durée de deux semaines
 - ✓ L'organisation de l'équipe tourne autour d'un système de **pull** qui consiste à ne pas attribuer les tâches aux membres de l'équipe au début de l'itération : les membres de l'équipe qui vont choisir une nouvelle tâche dès qu'ils auront terminé leur tâche en cours