Scrum Mémento

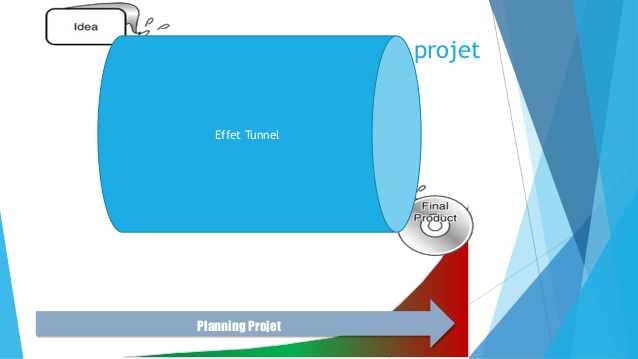
**Les méthodes agiles**

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’échecs.

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 a été mis en place en 2001 par 17 signataires issus du monde de l’informatique et des méthodes. Il définit 4 préceptes et 12 principes à respecter pour rester dans le cadre de la philosophie agile.

Quelques méthodes :

En dehors de Scrum, plusieurs méthodes et cadres de travail font partie de l’écosystème agile :

* **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 de méthodes, chacune 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 qui feront partie en 2001 des 17 signataires du Manifeste agile.

Les bases de Scrum sont définies dans un document de référence : le Guide Scrum (<http://www.scrumguide.org>).

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.

**Les fondamentaux.**

Objectif de Scrum

Scrum met à disposition des moyens permettant de gérer des projets informatiques complexes de façon itérative. Chaque itération doit permettre :

* D’ajouter de la valeur au produit
* De préciser la définition du produit
* D’améliorer le processus de production

Empirique

Scrum propose 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. Il repose sur 3 piliers :

* **Transparence** par une communication constante et une compréhension partagée par tous les membres de l’équipe
* **Inspection** pour détecter les écarts par rapports aux attentes
* **Adaptation** pour réduire le risque et mettre en place une démarche d’amélioration

Au-delà du contrôle empirique, le cadre de travail Scrum repose sur la définition d’acteurs, d’événements, d’artefacts et d’un processus spécifique.

**Rôles et acteurs**

**Scrum Master (SM)**

Rôles

* Responsable méthodologie
* Organisateur chargé d’accompagner les équipes dans l’utilisation des composantes de Scrum (évènements, 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.

Au sein de l’équipe

* Un seul SM par équipe Scrum
* Le rôle de SM peut être « tournant » : à chaque nouveau projet, un nouveau SM peut être désigné.

**Product Owner (PO)**

Rôles

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

Compétences et qualités

* Rigueur sur le maintien à jour des définitions du produit
* Ecoute et ouverture au près des autres membres de l’équipe et de leurs propositions
* Esprit de synthèse

Au sein de l’équipe

* Un seul PO par équipe Scrum
* Plaque tournante de toutes les discussions et décisions autour du produit

**Equipe de développement**

Rôle

* 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 …

Au sein de l’équipe

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

**Les autres parties prenantes (stackholder)**

Le PO, le SM et l’équipe de développement constituent l’équipe core (noyau) de Scrum. Celle-ci intervient sur le projet conjointement aux autres parties prenantes :

* Les commanditaires (client) qui constituent l’initiateur du produit
* Les utilisateurs à qui est destiné l’outil
* Les financeurs (sponsors) qui apportent le support financier au projet

**Evénements**

Les événements de Scrum entrent dans une logique de Timebox. Chacune de ces boîtes doit respecter 2 principes :

* Limiter le temps à passer sur un événement donné
* N’accepter aucun dépassement

**Sprint**

Le Sprint est à la base du découpage des projets sous Scrum. L’objectif à la fin de chaque Sprint est de proposer une itération supplémentaire qui ajoute de la valeur au produit.

Les caractéristiques du Sprint sont :

* Une durée préconisée de 4 semaines maximum
* Tous les sprints d’un projet ont la même durée
* Aucun temps mort entre 2 sprints, un nouveau démarrant directement à la fin du précédent

En plus du développement des fonctionnalités de l’itération, chaque sprint est ponctué par les 3 événements suivants :

**Sprint planning**

* Quand : au démarrage du sprint
* Durée : 8 heures maximum pour un sprint d’un mois
* Objectif : sélectionner dans le product backlog les items à réaliser, les découpages en tâches, évaluer le travail et le prioriser
* Acteurs : le SM organise, le PO aide à la section des items, l’équipe de développement sélectionne les items et les découpes en tâches

**Sprint Review**

* Quand : à la fin du sprint
* Durée : 4 heures maximum pour un sprint d’un mois
* Objectif : présenter l’itération et démontrer que chaque item développé répond aux attentes. Seuls les items terminés sont présentés
* Acteurs : Le SM organise, l’équipe de développement présente les développements, le PO valide ou non les items et maintient le product backlog en fin de review. Les parties prenantes principales assistent à la review et participent à la validation

**Sprint Retrospective**

* Quand : à la fin du sprint
* Durée : 3 heures maximum
* 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.

Il est à noter que le PO peut annuler un sprint en cours si le Sprint Goal est obsolète. Cela implique :

* La mise à jour du Product Backlog
* La mise en place d’une Sprint Retro
* L’enchainement immédiat avec le prochain Sprint

**Daily Scrum / Daily meeting / Daily Stand up meeting / La mêlée quotidienne**

Le Sprint est ponctué chaque jour par un point d’équipe : la mêlée quotidienne. Celle-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 mêlée dure au maximum 15 minutes durant lesquelles chaque membre de l’équipe de développement est sensé répondre à 3 questions :

* Qu’est ce que j’ai réalisé hier ?
* Que vais-je faire aujourd’hui ?
* Quels sont mes obstacles ?

Si chacun peut assister à la mêlée, seuls les développeurs ont droit à la parole. Le SM peut être invité, par l’équipe de développement, à participer.

**Artefacts**

Les artefacts « représentent soit du travail, soit de la valeur » (cf. guide Scrum) et permettent de fournir une base de travail commune à l’ensemble des acteurs.

**Le Product Backlog**

Ce carnet porte la définition du logiciel tout au long du projet. Il est maintenu par le PO, liste les attentes sur le produit et doit respecter les contraintes suivantes :

* Il est unique
* Il est la seul référence sur le produit
* Il est partagé à toute l’équipe
* Il évolue en même temps que le projet et est maintenu à jour

L’élément de base du product backlog est l’item qui se définit par :

* Une description
* Une estimation de l’effort
* Une estimation de la valeur apportée au logiciel
* Une priorité

Le récit d’utilisateur (la user story) est le format d’item le plus couramment utilisé.

**Le Sprint Backlog**

Le Sprint Backlog reprend du carnet de produit les items sélectionnés pour être développés pendant le Sprint. Il est maintenu par l’équipe de développement qui le met en place et le fait évoluer tout au long des développements. L’élément de base est la tâche qui est un découlement plus fin de l’item.  
Celle-ci se définit par :

* Une description
* Une estimation de l’effort
* Une priorité

**Incrément de produit / Product Increment**

Un incrément peut être vu comme la concaténation d’un ensemble d’items du backlog intégrés au logiciel. Il comprend :

* Les items réalisés et terminés lors de ce Sprint
* Les items réalisés lors des Sprints précédents.

La responsabilité de l’incrément revient à l’équipe de développement.

La définition du fini (definition of done ou DOD) a toute son importance. Elle se définit en début de projet pour que l’ensemble des acteurs ait 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 review
* Ne fait pas partie de l’incrément en fin de Sprint.