

Scrum



V. Deslandres, IUT - Université Lyon1

(librement adapté de la présentation du **DR. HATEM BEN STA**)



Plan



- **Introduction à Scrum**
- **Esprit des méthodes agiles**
- **Scrum**
 - Principes clés
 - Les rôles
 - Planifier un projet
 - Cycle Scrum et cérémonies
 - Indicateurs de pilotage
 - Techniques d'ingénierie logicielle
 - Passage à l'échelle
- **Perspectives**
- **Conclusion**

Introduction à Scrum

3

- Scrum terme en anglais signifiant *mêlée*, notamment en rugby
- Scrum désigne une méthode agile dédiée à la gestion de projet
- **Objectifs:**
 - Satisfaire au mieux les besoins du client
 - Maximiser les chances de réussite du projet
 - 1986 : « The New product development game »
 - 2001 : le Manifest Agile



Plan



- **Introduction à Scrum**
- **Esprit des méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Esprit des méthodes agiles (1/2)

5

- Une méthode agile est une approche **itérative** et **incrémentale**, qui est menée
 - dans un esprit collaboratif
 - avec juste ce qu'il faut de formalisme
- Elle génère un produit de **haute qualité** tout en prenant en compte l'évolution des besoins des clients
- Concepts formalisés en 2001 par le Manifeste Agile.

Les méthodes agiles. (2/2)



Les **4 valeurs** du Manifeste Agile:

- **L'équipe** : Personnes et interactions plutôt que processus et outils
- **L'application** : Logiciel fonctionnel plutôt que documentation complète
- **La collaboration** : Collaboration avec le client plutôt que négociation de contrat
- **L'acceptation du changement** : Réagir au changement plutôt que suivre un plan.

Plan



- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Scrum – Principes clés



Scrum est une méthode agile qui permet de produire une grande valeur métier dans un temps court.

Méthode **itérative** et **incrémentale**:

- Réalisation d'un ensemble de fonctionnalités par itération
- Itération d'une durée fixe (d'2 à 4 semaines) → **sprint**
- Livraison d'un *incrément* de produit, partiel mais *fonctionnel*, par itération

Participation du client:

- Définition des fonctionnalités prioritaires
- Ajout de fonctionnalités en cours de projet (pas pendant un sprint !)

Plan



- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Scrum – Les rôles



- L'équipe Scrum :
 - Le Product Owner (PO)
 - Le Scrum Master
 - Les Développeurs (*spécialistes* du développement logiciel : architectes logiciel, graphistes, codeurs, testeurs, etc.)
 - Taille idéale : 4/5 personnes

- Les parties prenantes :
 - Tous ceux qui ont un intérêt dans le projet
 - Sont conviés à certaines cérémonies Scrum

+ Certifications (ITIL, etc.)₁₀

Plan

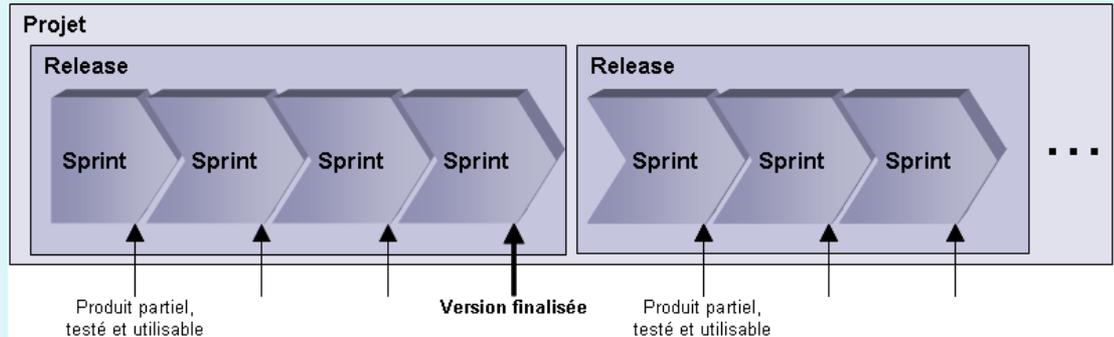


- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Scrum – Planifier un projet



Backlog produit - Site marchand XY					
ID_Item	Titre	Importance	Estimation	Démonstration de la fonctionnalité	Commentaires
SPRINT 1					
1	Besoin 1	130	12	XXXXXXXX	XXXXXXXX
2	Besoin 2	120	9	XXXXXXXX	XXXXXXXX
3	Besoin 3	115	20	XXXXXXXX	XXXXXXXX
SPRINT 2					
4	Besoin 4	110	8	XXXXXXXX	XXXXXXXX
5	Besoin 5	100	20	XXXXXXXX	XXXXXXXX
6	Besoin 6	95	12	XXXXXXXX	XXXXXXXX
SPRINT 3					
7	Besoin 7	80	10	XXXXXXXX	XXXXXXXX
8	Besoin 8	70	8	XXXXXXXX	XXXXXXXX
9	Besoin 9	60	10	XXXXXXXX	XXXXXXXX
10	Besoin 10	40	14	XXXXXXXX	XXXXXXXX
SPRINT 4					
11	Besoin 11	35	4	XXXXXXXX	XXXXXXXX
12	Besoin 12	25	6	XXXXXXXX	XXXXXXXX
13	Besoin 13	10	7	XXXXXXXX	XXXXXXXX
14	Besoin 14	10	11	XXXXXXXX	XXXXXXXX
15	Besoin 15	10	3	XXXXXXXX	XXXXXXXX



Source : <http://fr.wikipedia.org>

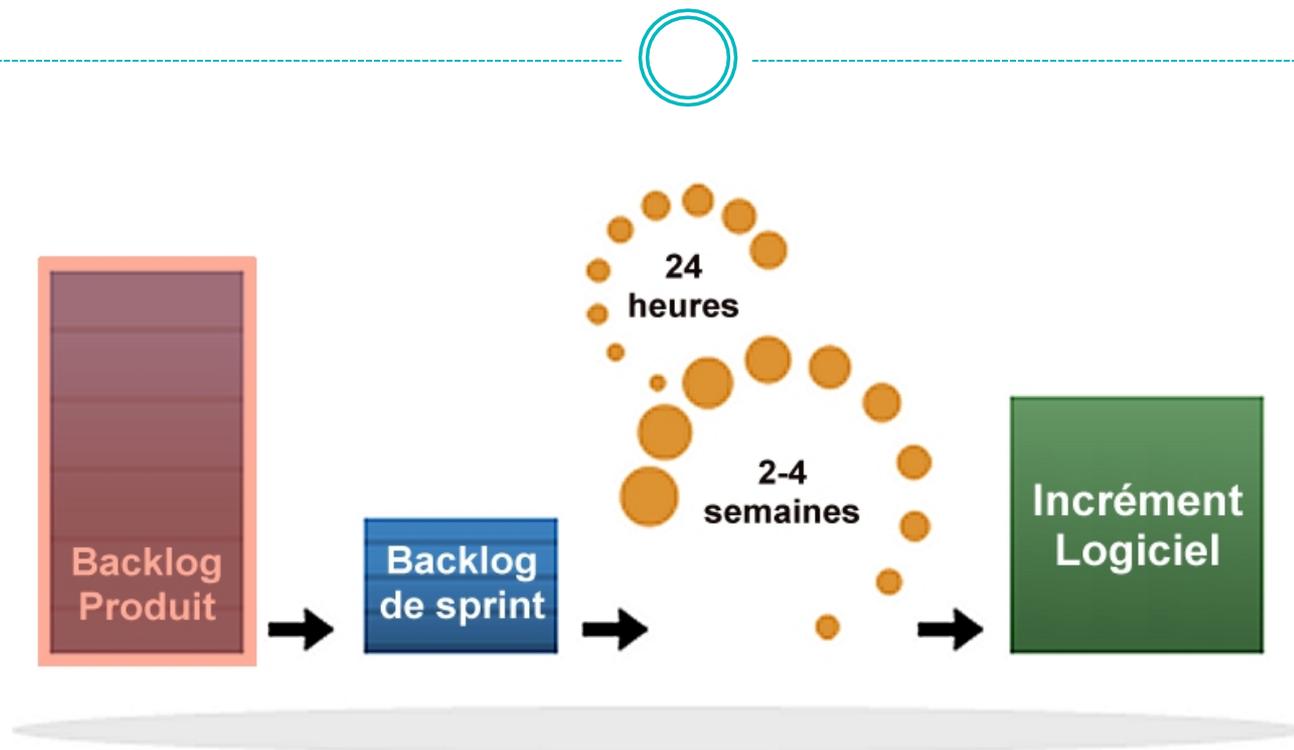
- Constitution du **backlog produit** par le product owner.
- Répartition en **sprints** et en **releases**.

Plan



- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Scrum – Cycle (1/5)

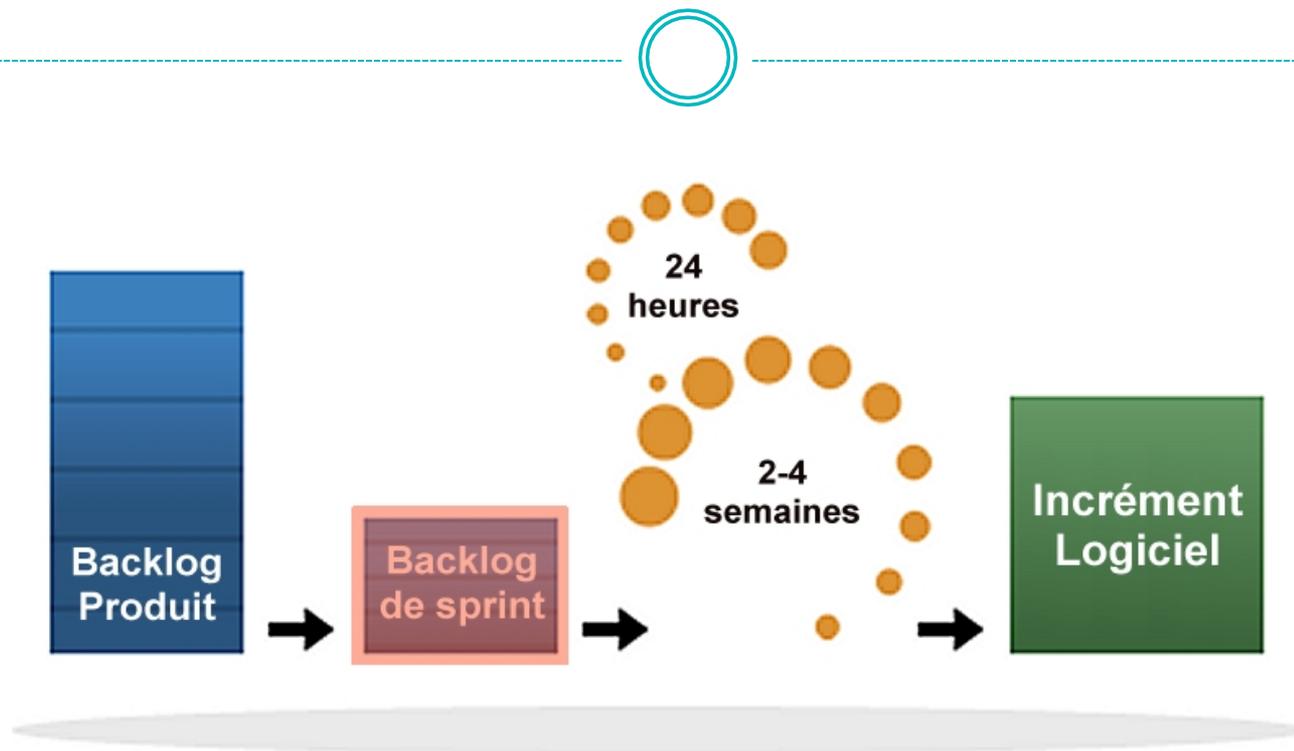


Source : www.scrumalliance.org

1. Backlog produit (ou catalogue des besoins)

- Besoins priorisés par le product owner
- Besoins évalués par l'équipe

Scrum – Cycle (2/5)



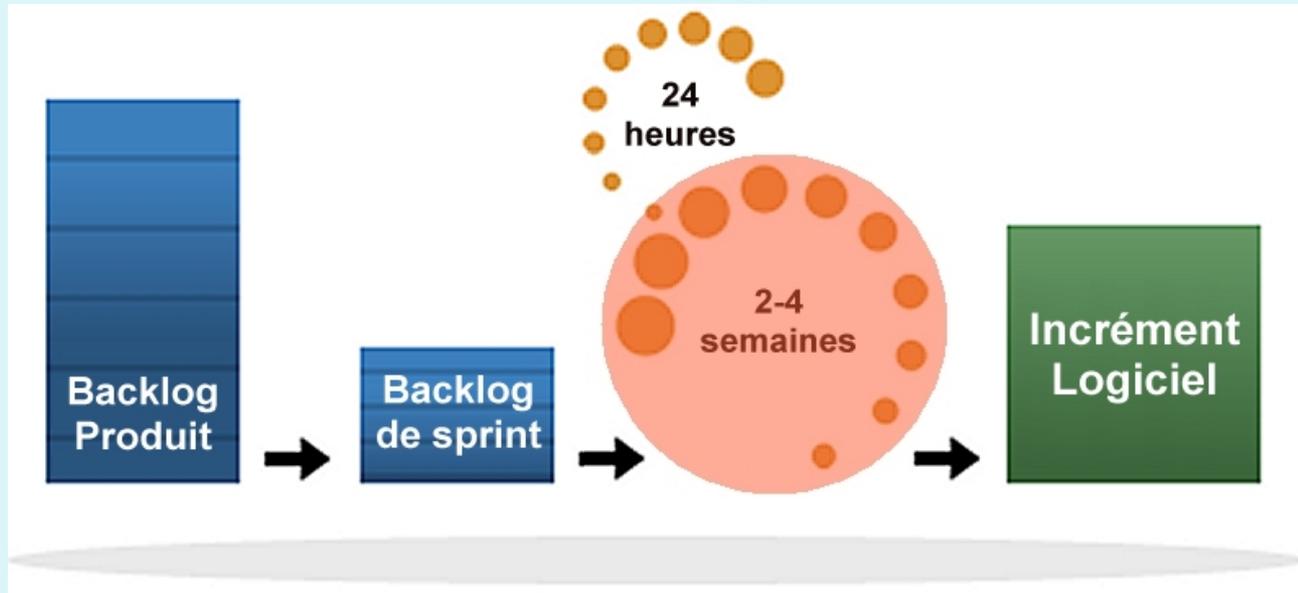
Source : www.scrumalliance.org

2. Backlog de sprint

- Extrait du backlog produit
- Besoins éclatés en tâches

15

Scrum – Cycle (3/5)

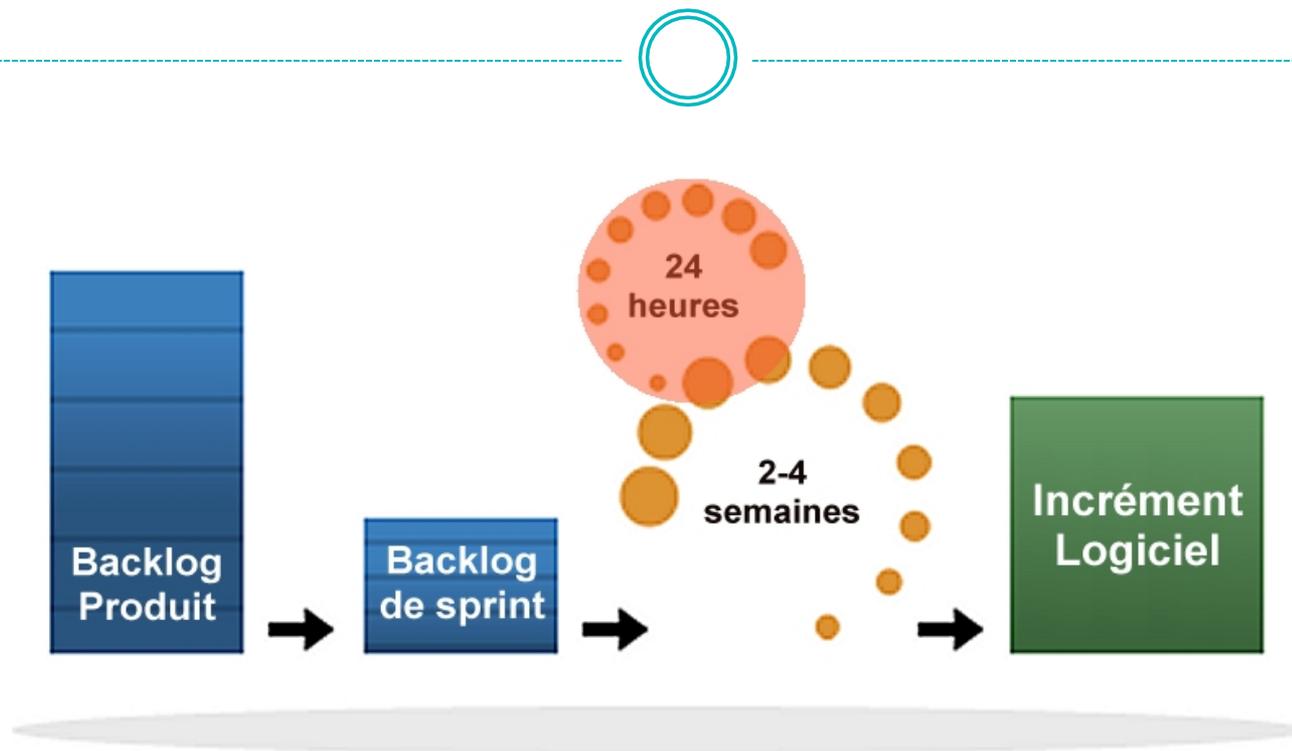


Source : www.scrumalliance.org

3. Sprint

- Développement des fonctionnalités du backlog de sprint
- Pas de modification du *backlog de sprint* possible
- Affinage du BackLog (1x par semaine, 3h max)

Scrum – Cycle (4/5)

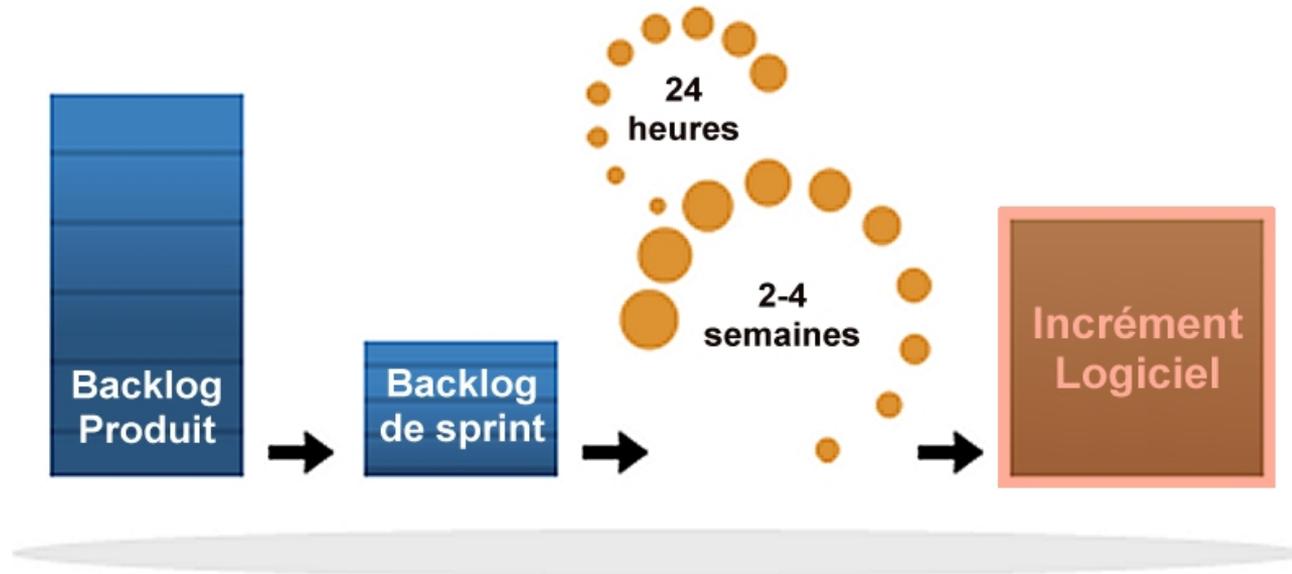


Source : www.scrumalliance.org

4. Mêlée quotidienne

- Point de contrôle quotidien de l'équipe
- Interventions limitées : 2' par personne

Scrum – Cycle (5/5)



Source : www.scrumalliance.org

5. Incrément logiciel : livré au product owner à la fin du sprint

Cérémonies de Scrum

19

- En **fin** de sprint :
 - **Revue de sprint** : on fait une démonstration du produit réalisé, devant l'équipe + parties prenantes
 - ✦ Objectifs = montrer l'avancement du produit, obtenir des retours, déceler des défauts, etc.
 - **Rétrospective** : l'équipe seule analyse ce qui a bien ou mal fonctionné pendant l'itération
 - ✦ Objectifs = amélioration continue, animée par le Scrum Master
 - ✦ Repose sur l'idée : « intelligence collective > somme des intelligences individuelles »
- En **cours** de Sprint
 - **Daily Meeting** : savoir où chacun en est, avec l'équipe seule
 - ✦ Objectif : déceler un obstacle majeur, une potentielle dérive pour réajuster la cible du sprint si besoin

19

Plan



- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage et de succès de Scrum**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Indicateurs : notion de fini (1/6)



- En Agile, on définit ce qu'est le « fini »
- Une user story (ou tâche) sera dite terminée quand
 - Elle passe les tests d'acceptation ?
 - ✦ Si oui, alors décider : Test nominal ? Test limite ? Test erreur ?
 - ✦ Sur quel jeu de données ?
 - ✦ Quel aspect de la story tester ?
 - Quand l'interface utilisateur est validée par notre graphiste ?
 - ✦ Forme, couleurs ?
 - Quand la doc utilisateur est rédigée ?
 - ✦ Avant / après correction orthographique ?
 - Quand la traduction en anglais est aussi faite ?
- Un indicateur est le % d'US avec une DoD bien claire

Indicateurs : vélocité (2/6)

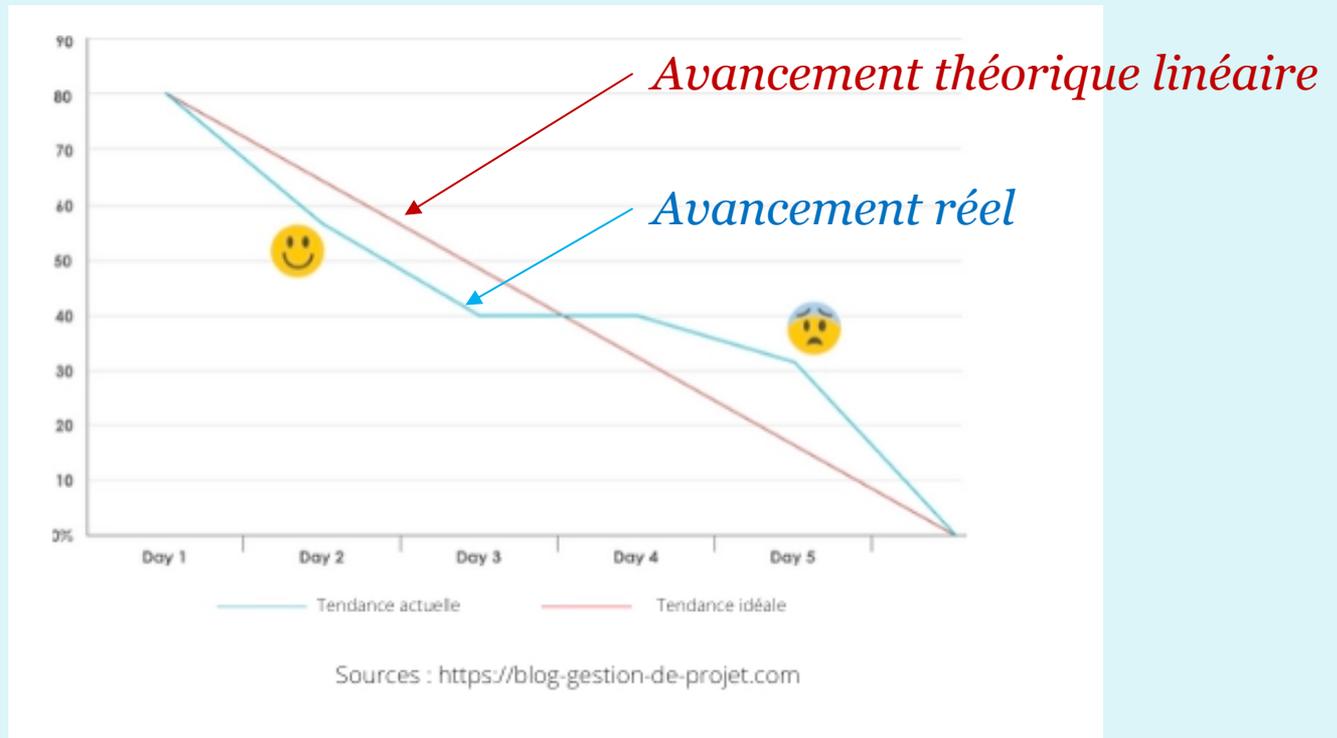


- En Agile, la vélocité de l'équipe traduit sa rapidité de production de code
 - Elle est basée le nombre de points d'effort des tâches **validées** en fin de sprint
 - Peut être prédite pour le 1^{er} sprint
 - Calculée pour les autres sprints
 - ✦ Sur la base de la réalisation de l'itération précédente (points estimés « done »)
 - DEF Capacité : ce qui a été **planifié** sur l'itération, par l'équipe
 - ✦ Après un certain temps, capacité = vélocité (moins d'erreurs d'estimation)
- Déroulement :
 - Backlog de produit : use stories, planning
 - Planification de sprint
 - ✦ Découpe en tâches
 - ✦ Estimer l'effort requis pour chaque tâche (nb de points, planning poker, ...)
 - ✦ Choisir les tâches qui vont être réalisées à la 1^{ère} itération
 - ✦ 2^{ème} itération : calcul de capacité et de vélocité

Le BurndownChart (3/6)

23

Somme des **story points** (points d'effort)

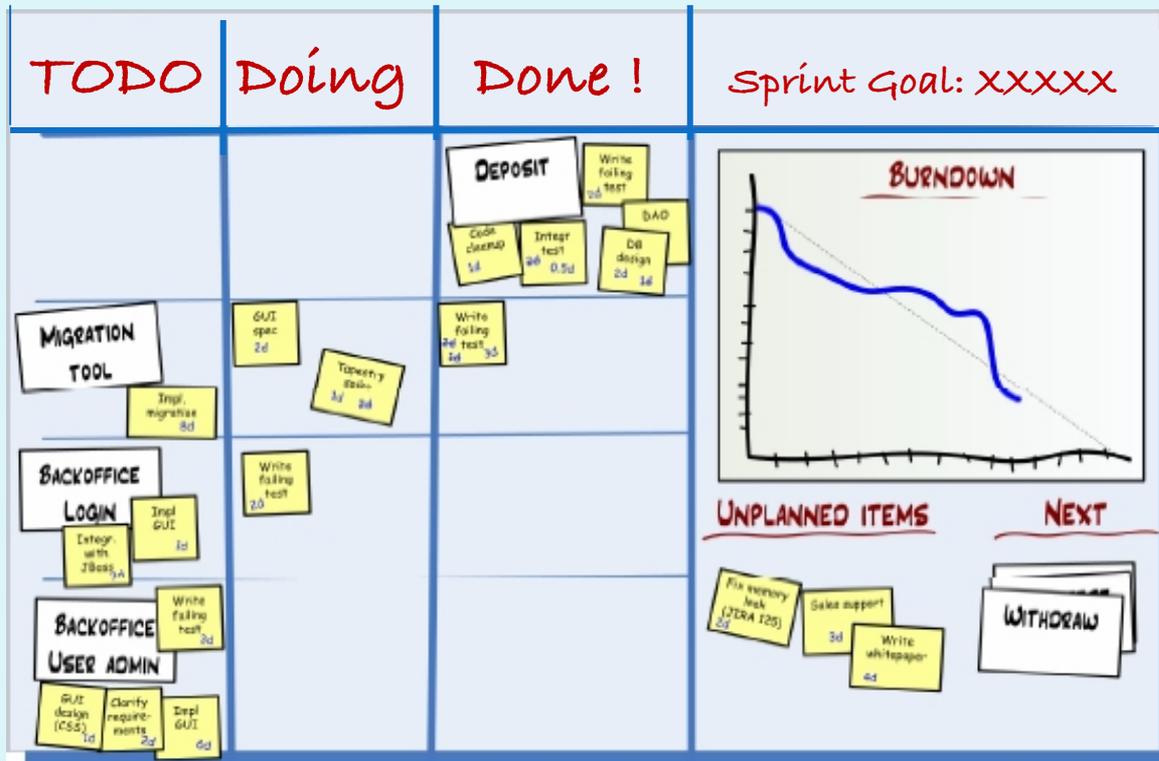


Le Burndown chart indique la **quantité de travail restant** à effectuer avant la fin d'un sprint

Task Board SCRUM



Le tableau des tâches (*colonnes KANBAN*)



Source : « Scrum and XP from the trenches » de H. Kniberg, 2007

Indicateurs : **code coverage** (4/6)

25

- La qualité des livrables est importante en Agile
- Couverture de *code exercé* par des Tests Unitaires
 - Le code a été exécuté par un test (on ne sait pas si ça a réussi)
 - Par ex.: 100 lignes de code, des tests pour 10 lignes : CC de 10%

Différent de la part de *code testé* : code pour lequel on a évalué la qualité des Tests Unitaires (par des outils de mutation de code, tel Jumble en Java : on introduit des erreurs volontairement)
- Mesure de qualité des Pull Request
 - Règle : une PR ne doit pas faire baisser le Code Coverage
- Objectif managérial : % à atteindre par l'équipe

Indicateurs : **satisfaction Client** (5/6)

26

- en Agile, on cherche d'abord à *fidéliser* les Clients avant d'en découvrir de nouveaux
- Mesure la satisfaction client en demandant aux utilisateurs de **noter le produit**
 - sur une échelle de 1 à 10
- Ou encore : méthode NPS (Net Promoter Score)
 - Définit le pourcentage de personnes qui font la promotion de votre produit, voire le déconseilleraient

Indicateurs : **satisfaction équipe** (6/6)

27

- Le succès SCRUM s'atteint de façon collaborative
- Une équipe « en bonne santé » sera plus performante qu'une équipe épuisée, peu motivée, où l'engagement sera faible
- *Par exemple, démarrer chaque rétrospective en demandant à l'équipe de noter **l'ambiance du sprint** sur une échelle de 1 à 5.*
 - Suivre ces chiffres de sprint en sprint pour analyser les tendances.

Exercice Indicateurs Classiques vs. Agiles

28

Pour chaque indicateur proposé, dire s'il relève plutôt de l'approche **classique** ou **agile**

Analyser la boucle de rétroaction : les feedbacks utilisateur ont-ils permis de réaliser les ajustements rapidement ?

Mesurer le nb de changements par rapport au plan ou à la documentation initiale

Respect des exigences du CDC ?

Objectif atteint pour le Client ?

Mesurer la valeur ajoutée du produit

Mesurer la réussite du projet

Analyser le respect de la roadmap

Mesurer la qualité de la livraison

Analyser le niveau de collaboration des équipes et des membres au sein des équipes

Analyser les contributions individuelles pour réduire les goulots d'étranglement

Plan



- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Scrum – Ingénierie logicielle



- **Scrum** est une méthode de gestion de projet
- Doit être complétée par des techniques d'ingénierie logicielle
 - **Tests unitaires** automatisés
 - **Refactoring**
 - Intégration continue : **CI/CD** - *Continuous Iteration / Continuous Deployment*
- Complémentaire avec Extreme Programming :
 - **TDD** - *Test Driven Development*

Plan



- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Scrum – Passage à l'échelle



- Si projet avec de grandes équipes
 - Ex.: 9 équipes de 7, réparties géographiquement
- Principes :
 1. Commencer par une équipe Scrum standard
 2. Création de plusieurs équipes – essaimage
- Adaptation de la méthode : **Scaled Agile Framework (SAFe)**
 - Scrum des scrums
 - Rôle de team lead
- Problèmes à traiter :
 - Dispersion géographique
 - Développement off-shore

Les outils



- **Outils traditionnels**

- Tableau blanc et post-its
- Excel – Backlog produit et backlog de sprint

- **Outils dédiés**

- Outils répandus outre JIRA : miro <https://miro.com/> ,taïga <https://www.taiga.io/>, trello <https://trello.com/>
 - ✦ [Liste d'autres outils](#) gratuits (2021) : Flying Donut, ClickUp, Wrike
- Gèrent tout ou partie des éléments de Scrum
- Gestion par tickets / couplage Scrum : **JIRA**
- Absence de PERT / Gantt
- Intégration avec :
 - ✦ IDE, contrôle de sources, gestion des tests, bug tracking, intégration continue



Plan



- **Introduction à Scrum**
- **Les méthodes agiles**
- **Scrum**
 - **Principes clés**
 - **Les rôles**
 - **Planifier un projet**
 - **Cycle Scrum et cérémonies**
 - **Indicateurs de pilotage**
 - **Techniques d'ingénierie logicielle**
 - **Passage à l'échelle**
- **Perspectives**
- **Conclusion**

Perspectives



- Mise en place variable selon l'organisation
 - Des réussites, des échecs
 - Equipes pilotes fréquentes
- Limites
 - Passage à l'échelle : **SAFe** ne fait pas l'unanimité
 - Culture d'entreprise inappropriée
 - ✦ Ce qui engendre un risque élevé d'échec
 - ✦ La **maturité technologique** (ex.: CI/CD) est une clef de succès
 - Responsabilité / Maturité des programmeurs
 - ✦ Polyvalence
- Contrats Clients : **spécifiques**
- Stratégie d'introduction de Scrum en entreprise
 - Soutien de la hiérarchie
 - Accepter plus de souplesse managériale

Conclusion



- Méthode de gestion de projet – développement logiciel
 - Peut transformer les valeurs de l'organisation
- A compléter avec des techniques d'ingénierie logicielle
- Conditions propices nécessaires, fin de l'effet de mode
 - Plus de 20 ans d'existence
- Efficacité prouvée : **3x** plus de réussites de projet
- Principal bénéfice : des équipes motivées, une meilleure gestion du processus de développement, des livraisons + rapides

Références



- En Anglais
 - www.mountangoatsoftware.com/scrum
 - www.scrumalliance.org
- Blog **My Agile Partner** (en Français)
 - <https://blog.myagilepartner.fr/index.php/category/blog-scrum/page/18/>
- Chaine YouTube **Scrum Life**
 - <https://www.youtube.com/channel/UCMCnZGIOeLVO65-LBxkkHyQ>
- **Collective Genius** : blog Scrum & formations
 - <https://www.collectivegenius.fr/blog/>