



1



SCRUM :
Produire au plus vite et
au plus près de vos priorités !!!

Définition ...



- Un logiciel est un ensemble d'informations relatives à des traitements effectués automatiquement par un appareil informatique. Il inclut :
 - les instructions de traitement,
 - des instructions regroupées sous forme de programmes,
 - des données
 - de la documentation.

3

Qu'est ce qu'une méthode agile?



- Une méthode agile :
 - est une approche itérative et incrémentale,
 - est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme.
 - à chaque itération, doit livrer un incrément de logiciel **TERMINÉ !!**
 - génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients
- Approche itérative et incrémentale



4

Qu'est ce qu'une méthode agile?

Avantages d'une approche itérative et incrémentale :

- **La communication est de meilleure qualité :**
 - Les malentendus, incompréhensions, incohérences sont mis en évidence tôt dans le projet ; il est donc encore possible de les corriger ;
 - 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 visualiser les travaux plus régulièrement, sans attendre la fin ;
 - À la fin de chaque itération, les fonctionnalités retenues sont développées, testées, documentées et validées, prêtes pour l'exploitation.
- **La qualité est évaluée en continu :**
 - Les tests sont effectués à chaque itération ;
 - Les anomalies détectées sont corrigées au fur et à mesure.

5

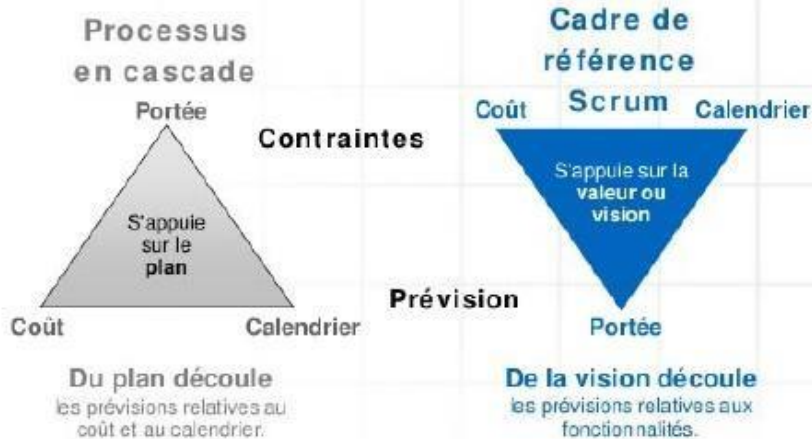
Qu'est ce qu'une méthode agile?

- **Les risques sont détectés très tôt :**
 - Les risques sont détectés tôt et résolus rapidement.
- **L'équipe prend confiance :**
 - L'itération donne une occasion d'apprendre, donc de capitaliser ou d'adapter les pratiques pour la suite du projet ;
 - Les premières itérations fiabilisent les prévisions ;
 - Le changement n'est plus une menace, mais au contraire, l'opportunité de mieux faire et de mieux satisfaire le client.
- **Les coûts sont contrôlés :**
 - Les coûts sont limités, en termes de risques, au périmètre de l'itération ;
 - S'il faut reprendre une itération, on ne perd que les efforts de cette itération et non la valeur du produit dans sa globalité ;
 - On peut aussi arrêter le projet à l'issue de quelques itérations si l'on n'a plus de budget → **produit déjà fonctionnel!!!**

6

Qu'est ce qu'une méthode agile?

Passer d'un modèle de gestion fondé sur les coûts à un modèle de gestion fondé sur la valeur d'affaires créée

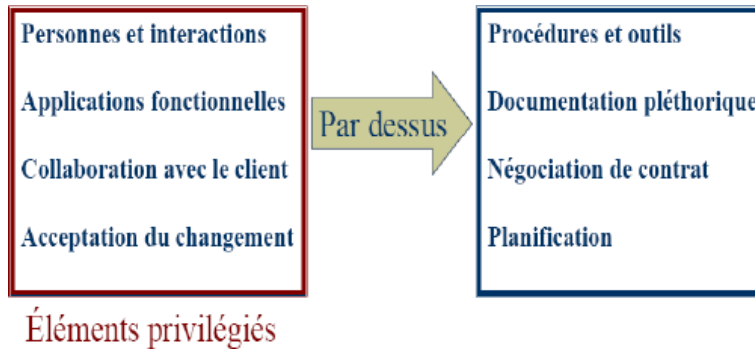


Origines et valeurs des méthodes agiles

- 2001: « **Agile Alliance** » définit les méthodesAgiles
 - But:** augmenter le niveau de satisfaction du client tout en rendant le travail de développement plus facile
- 2 caractéristiques fondamentales:
 - « **Adaptatives** » plutôt que prédictives
 - Être favorable aux changements
 - Suivre un formalisme léger → Planification plus souple
 - Orientées vers des **personnes** plutôt que vers les processus
- Adopter un **esprit collaboratif**
- Travailler avec les **spécificités de chacun**

Le manifeste (2001)

On découvre de meilleures façons de développer des logiciels en le faisant et en aidant les autres à le faire



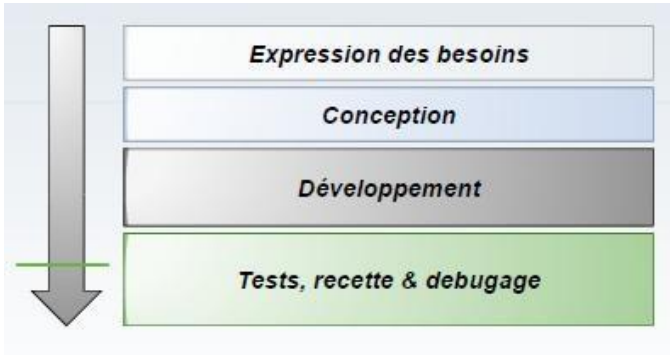
9

Les 12 principes du manifeste agile

Les 4 valeurs sont déclinées en 12 principes généraux

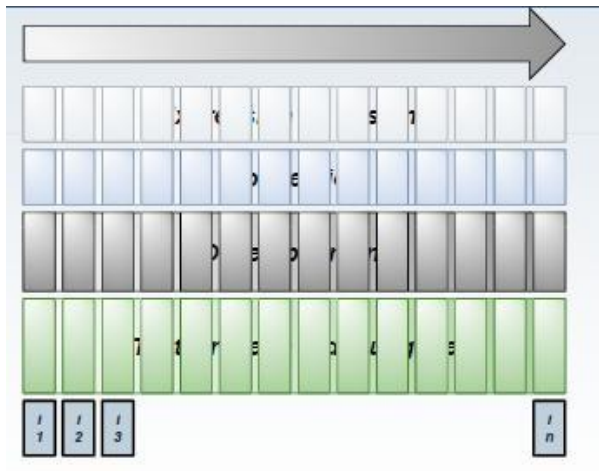
Satisfaire le client	Considérer comme naturel les changements d'exigences	Livrer fréquemment une application fonctionnelle	Fonctionnels et développeurs travaillent ensemble
Bâtir le projet autour de personnes motivées	L'échange d'information le plus efficace est en face à face	Un logiciel fonctionnel est la meilleure façon de mesurer l'avancement	Le rythme de développement doit être soutenable indéfiniment
Simplicité - l'art de maximiser la quantité de travail à ne pas faire - est essentielle	Architectures, spécifications et conceptions issues d'équipes auto-organisées	Vérifier en continue l'excellence des pratiques et techniques	Régulièrement, réflexion de l'équipe pour être plus efficace !

Les solutions classiques



11

Les solutions Agiles



12

La méthode SCRUM

- Selon le bureau d'études Gartner, en 2012 :
 - **Plus de 60% des projets informatiques deviennent agiles**
 - **80% de ces projets utilisent la méthode Scrum**
 - Ken Schwaber décrit Scrum comme un cadre (*framework*)
 - Un spécialiste des processus parlerait, pour Scrum, de *pattern* de processus, orienté gestion de projet, qui peut incorporer différentes méthodes ou pratiques d'ingénierie.
- **Scrum est une méthode agile pour la gestion de projets.**

13

La méthode SCRUM

- Le terme **Scrum** est emprunté au rugby et signifie *mêlée*.
 - une équipe soudée, qui cherche à atteindre un but.
- Scrum a été conçu pour améliorer la productivité dans les équipes auparavant paralysées par des méthodologies classiques - plus lourdes.

14

Principes de base de SCRUM



- Le principe de base de Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de 30 jours au maximum, appelées **Sprints**.
- Chaque Sprint possède un **but** à atteindre, défini par le **Product Owner / Directeur de produit**, à partir duquel sont choisies les fonctionnalités à implémenter dans ce sprint.
- Un sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel.

15

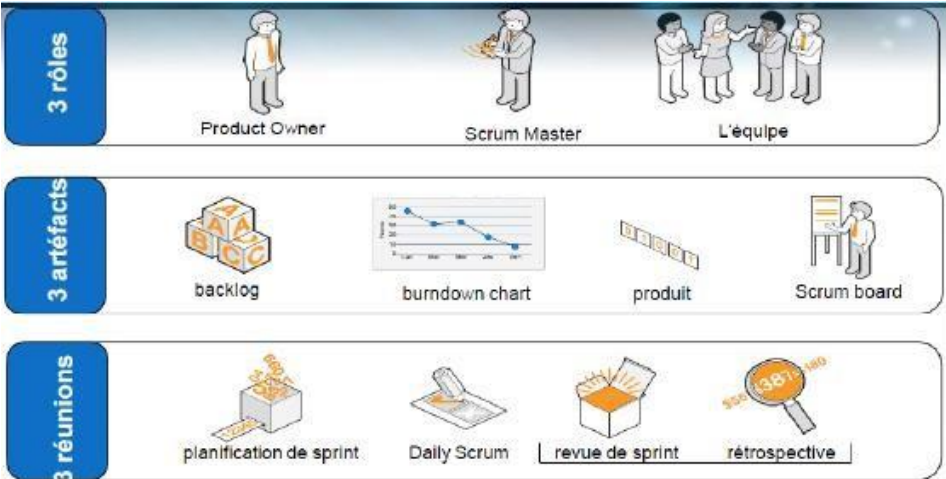
Principes de base de SCRUM



- Pendant les sprints, le **Scrum Master** a la charge de réduire au maximum les *perturbations extérieures* et de résoudre les *problèmes non techniques* de l'équipe.
- Scrum repose sur la participation active du **client** qui joue le rôle de **Product owner** pour définir les priorités dans les fonctionnalités du logiciel, et choisit lesquelles seront réalisées dans chaque sprint (itération).
- Le **Product owner** peut à tout moment compléter ou modifier la liste des fonctionnalités à réaliser, mais jamais ce qui est en cours de réalisation pendant un sprint.

16

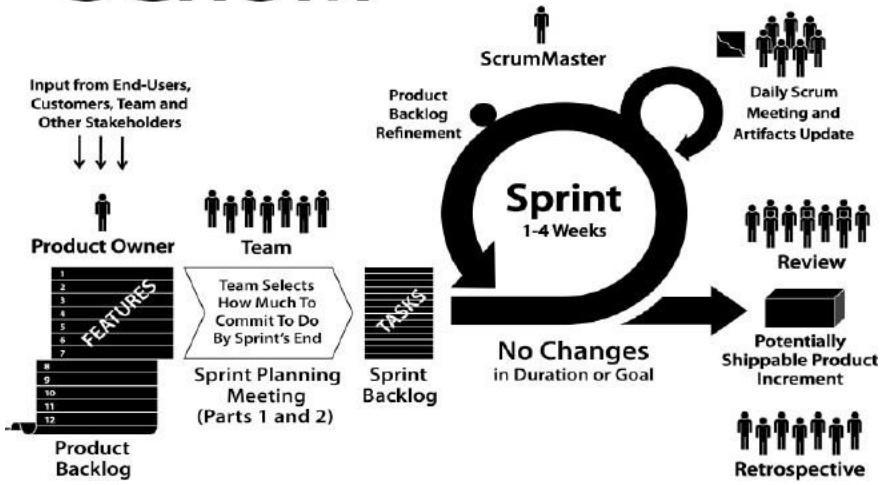
SCRUM : Vue d'ensemble



17

Le processus Scrum

SCRUM



Scrum est ...

Transparence !
Inspection !
Adaptation !

19

Scrum est ...

Il y a **trois points d'inspection et d'adaptation** dans Scrum :

- Le **scrum quotidien** permet d'inspecter la progression par rapport au but du sprint et de faire des adaptations qui optimisent la valeur du travail du jour suivant.
- La **planification et la revue de sprint** sont utilisées pour inspecter l'avancement du développement par rapport au but de la release et faire des adaptations sur le contenu du produit pour le prochain sprint.
- La **rétrospective** inspecte la façon de travailler dans le sprint pour déterminer quelles améliorations du processus peuvent être faites dans le prochain sprint.

20



Des sprints pour une release

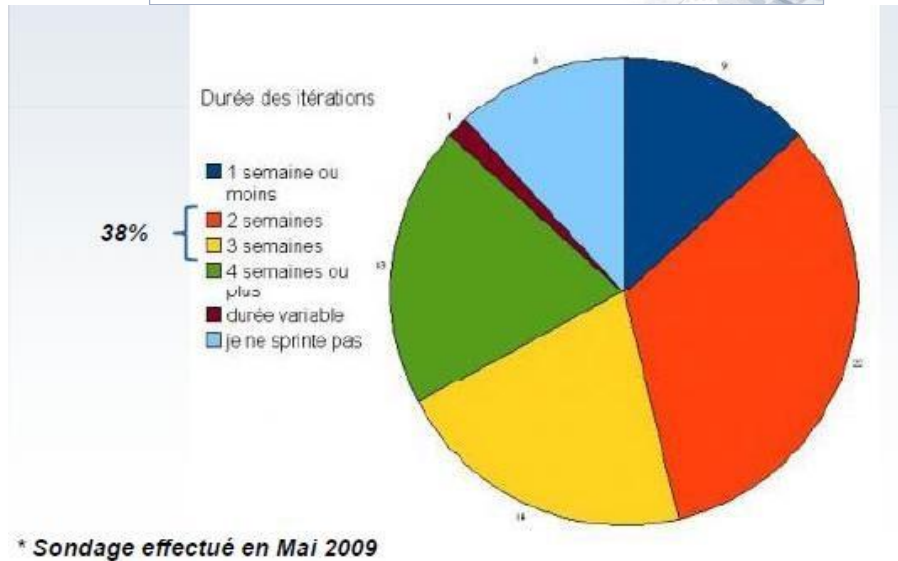
21



- **Sprint (ou itération)** : un bloc de temps fixé aboutissant à créer un incrément du produit potentiellement livrable.
- **Release peut avoir plusieurs sens :**
 - Release comme version – Le dictionnaire du jargon informatique français définit une release comme suit : « version d'un logiciel effectivement diffusée, donc « Mise sur le marché ».
 - Release comme période de temps – Par extension, on appelle également release la **période de temps** qui permet de la produire.

22

Durée des sprints

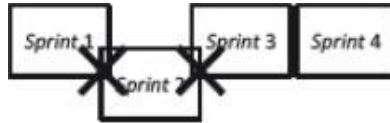


Cycle de développement SCRUM

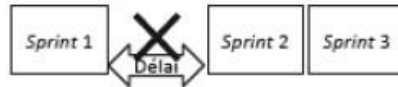
- Une **release** est une série de sprints qui se termine quand les incréments successifs constituent un produit qui présente suffisamment de valeur à ses utilisateurs.
- La durée des releases est définie par l'équipe et le Product Owner.
- La tendance est à raccourcir ces durées : une release dure environ trois mois, avec des sprints de 2 ou 3 semaines.
→ idéalement, dérouler de 4 à 6 sprints dans une release.

Cycle de développement SCRUM

- Les sprints sont **séquentiels** : Il n'y a pas de chevauchements



- Les sprints s'enchaînent **sans délai** : le nouveau démarre immédiatement après le précédent.



25

Le résultat d'un sprint

- Le résultat attendu = un incrément du produit final, qui est **potentiellement livrable**.
 - Le produit ne doit pas être seulement « potentiellement » utilisable à la fin d'un sprint, il doit simplement être utilisable.
 - Certes, le produit n'est pas complet par rapport à ce qui est prévu dans la release, mais il est livrable, au moins à des utilisateurs privilégiés.
- à la fin du sprint, déployer le produit potentiellement livrable, avec si nécessaire, la documentation permettant de l'utiliser.

Le résultat d'une release



- Le résultat de la release est le produit **livrable**, fourni à ses utilisateurs.
- Le déploiement dépend de la nature du projet :
 - Très fréquent pour des applications web hébergées :
 - Exemple : eBay re-déploie ses applications tous les 15 jours
 - Moins fréquent pour des applications internes ajoutées à un gros système d'information, ou pour des systèmes embarqués.
- Souvent, la release correspond à une annonce marketing : à l'occasion de la « sortie » du produit, les équipes marketing préparent un matériel pour sa promotion.

27



Les rôles dans Scrum

28

Le client



- **Sa fonction :**

- Il demande le produit
- Il contracte l'organisation pour le développement de son produit
- Typiquement, il s'agit d'un responsable qui achète un développement de produit par un sous-traitant.
- Dans les projets internes, il s'agit principalement du sponsor au projet, c'est à dire la personne validant le projet et le budget.

- **Sa mission :**

- Il commande le produit
- Il paie le développement du produit
- Il donne des **feed-back et des révisions**

29

Le manager



- **Sa fonction :**

- Le management, la gestion, est primordial dans tout projet Scrum. Il permet à l'Équipe de constituer un environnement optimal pour le déroulement du projet Scrum.
- Le manager donne de la structure et de la stabilité.
- Il travaille de concert avec le ScrumMaster pour réorganiser l'organigramme de la structure et donner de la guidance si nécessaire.

- **Sa mission :**

- Il s'assure que l'organisation puisse survivre en cas de défaillance.
- Il crée des règles et des lignes directrices.

30

L'utilisateur final

- **Sa fonction :**

- Ce rôle peut être joué par un grand nombre de personnes.
- L'utilisateur final est celui qui connaît les besoins et avec cette connaissance, il définit le produit en disant à l'équipe ce dont il a besoin comme fonctionnalités.

- **Sa mission :**

- Il connaît ses besoins et ses exigences
- Il donne son **feed-back lors des revues**
- Il participe, aussi, au **Sprint Planning 1**

31

Les rôles de l'équipe Scrum



32



Le Product Owner - PO

33

Les compétences d'un P.O

- La personne idéale pour jouer ce rôle devrait posséder les compétences suivantes :



- Quelqu'un qui a été Analyste Métier (Business Analyst) est un bon candidat pour ce rôle.

34



Le Scrum Master et l'équipe

35

Les responsabilités d'un S.M

- **Pas de chef de projet dans Scrum ! Le rôle est éliminé.**
- Le travail et les responsabilités d'un chef de projet ne disparaissent pas pour autant dans les projets Scrum.
→ une grande partie est dévolue au Product Owner, une autre partie est laissée à l'équipe.
- Un des principes de Scrum est l'auto-organisation
→ Pas besoin d'un chef qui assigne le travail à faire à l'équipe.
- **ScrumMaster n'est donc pas un nouveau nom pour chef de projet !!!**

36

Les responsabilités d'un S.M

- Le ScrumMaster a pour responsabilité essentielle d'aider l'équipe à appliquer Scrum et à l'adapter au contexte.
- Il a une grande influence sur la façon de travailler, sur le processus, comme le **Product Owner** en a une sur le produit.
Le **ScrumMaster** pourrait être qualifié de **Process Owner** par équivalence.

37

Les responsabilités de l'équipe

- Le rôle de **l'équipe de réalisation (E.R.) est essentiel** : c'est elle qui va réaliser le produit, en développant un incrément à chaque *sprint*.
- Dans Scrum, l'équipe s'organise elle-même et doit avoir toutes les compétences nécessaires au développement du produit.
→ une équipe Scrum est multifonctionnelle.
- C'est l'équipe qui définit elle-même la façon dont elle organise ses travaux, ce n'est pas le ScrumMaster (ni le Product Owner).
- Chaque membre de l'équipe apporte son expertise, la synergie améliorant l'efficacité globale.

Equipe Scrum



- De 5 à 9 personnes
- Regroupant tous les rôles
 - Architecte, concepteur, développeur, spécialiste IHM, testeur, etc.
- A plein temps sur le projet, de préférence
 - Exceptions possibles (administrateur, ...)
- L'équipe s'organise par elle-même
- La composition de l'équipe ne doit pas changer pendant un ³⁹Sprint



Les artéfacts Scrum



Le Backlog de produit - BP

41



- Au départ, la difficulté fondamentale est de transformer l'idée de départ en quelque chose d'utilisable par l'équipe de réalisation (ER).
- Dans les projets traditionnels, cette transformation se fait entièrement au début du projet et se concrétise dans un document, qui décrit :
 - ce que va faire le produit,
 - quelles sont ses fonctions
 - quel est son comportement.

deviner et imaginer les détails du comportement du produit avant d'entreprendre aucune action

42

Le Backlog la liste unique des stories

- Un backlog = un référentiel des exigences
- Un backlog = Liste ordonnée de toutes les choses à faire.
- Les éléments du Backlog sont appelés des stories



- On distingue :
 - le backlog de produit qui énumère les exigences avec le point de vue du client
 - le backlog de sprint qui contient les tâches de l'équipe.

43

Les stories

- Une User story est une exigence du système à développer, formulée en une ou deux phrases dans le langage de l'utilisateur.
- Les User Stories émergent au cours d'ateliers de travail menés avec le Product Owner ou les Utilisateurs.

- Quelques Exemples :
 - "En tant que recruteur, je peux déposer des offres d'emploi";
 - "En tant que jeune diplômé, je peux créer un CV";
 - "En tant que jeune diplômé je peux modifier un CV";
 - "En tant que jeune diplômé je peux sélectionner des offres"...

44

Story	Story Points	Release #
Web Portal Site		
1 - Security		
1.1 As a Customer I want to login so I can access my account.	2	1
1.2 As a Customer I want to register for new online account so I can access my information	3	1
1.3 As a Customer I want to retrieve my forgotten password so I can login again	3	1
1.4 As a Customer I want to add a password hint so I can remember my password	2	2
2 - Profile Management		
2.1 As a Customer I want to add another user to my account so they have access to it	5	1
2.2 As a Customer I want to update my existing profile information so my information is accurate	2	1
3 - Place Order		
3.1 As a Customer I want to search the product list so I can find what I want	3	1
3.2 As a Customer I want to browse all products so I can see what is available	2	1
3.3 As a Customer I want to add a product to my cart so I can check out	5	1
3.4 As a Customer I want to complete my order so I can receive my product	8	1
* 3.5 As a Customer I want to view product reviews so I can make an informed decision	5	1
4 - Order Maintenance		
4.1 As a Customer I want to check the status of my order so I know when I will receive it	2	1
4.2 As a Customer I want to cancel my recent order so that I don't receive it or get charged	2	1
4.3 As a Customer I want to search order history so I can view previous orders	3	1
100 - Non Functional Stories		
100.1 Migrate customer data from legacy system to new oracle db	8	1
100.2 Develop High Level Architectural diagram for new system	2	1
100.3 Develop high level ERD and Logical database diagrams.	3	1
100.4 Develop High Level business process models	2	1
100.5 Design look and feel for site and high level site navigation prototype	3	1

La notion de priorité dans le Backlog

- Le backlog est la liste unique de tout ce qui est à faire, ce qui donne beaucoup d'importance à la notion de **priorité**.
- Cette **priorité** permet de constituer le flux de stories qui va alimenter l'équipe. L'ordre peut changer tant que l'équipe n'a pas commencé à développer la Story
- Exemple :
 - Dire que la story A est plus prioritaire que la story B signifie que A sera réalisée avant B.
 - Les priorités sont utilisées pour définir l'ordre de réalisation

Les éléments du Backlog

- Les principaux attributs

Story
<ul style="list-style-type: none">• Nom• Identifiant• Description• Type (user, technique, défaut)• État• Taille

- Les types d'éléments :

- **User story** : Décrit un comportement du produit visible pour les utilisateurs et qui leur apporte de la valeur ou de l'utilité.
- **Story technique** : Elle est invisible pour les utilisateurs mais visible par l'équipe de développement. Elle est nécessaire pour pouvoir développer certaines user stories, *son utilité est donc indirecte*.
- **Défaut** : Il est relatif à un comportement visible des utilisateurs mais qui enlève de la valeur au produit. En développement de logiciel, on parle couramment de **bug**.

47

User stories

- User Stories :
 - As a <actor>, I would like to <action>, so that <value>.
- Attributs :
 - Taille (points , journées idéales), Business Value (\$, H / M);
 - Conditions de satisfaction.
- La taille d'une User Story
 - Influencéepar :
 - La difficulté de la Story
 - La charge de travail
 - Une valeur relative
 - Les points n'ont pas d'unités :
 - Suite de Fibonacci (0, 1, 2, 3, 5, 8, 13, 21....)

Cycle de vie d'un élément du B.P



- **Créé** : par n'importe qui
- **Accepté** : par le Product Owner
- **Estimé** : par l'équipe dans une séance de planning poker
- **Planifié** : associé à un sprint futur lors de la planification de release
- **En cours** : développé dans le sprint courant
- **Fini** : terminé, [selon la signification de fini](#)

49

Définition de « Terminé – Fini – Done »

Pour l'équipe

- Le Code est conforme aux normes. Il doit être :
 - Propre
 - Refactoré
 - Testé unitairement
 - Validé (checked in)
 - Intégré (Built)
 - Dispose d'une suite de tests unitaires qui lui est appliquée.
- Pour arriver à cela, l'environnement de développement est constitué :
 - D'une bibliothèque de code source
 - De codes standards,
 - Build automatisé,
 - D'un environnement pour les tests unitaires.



50

Définition de « Terminé – Fini – Done »



Pour Scrum

- Une Story/Item est « Finie / Done » lorsque l'équipe a déclaré qu'elle est « Terminée »
- Le Sprint/Iteration est “done” lorsque
 - tous les items sont “done”
 - et que le Sprint atteint son objectif
 - et que les critères d'acceptation sont adressés.
- La Release est “done”
 - “done” pour l'intégration
 - “done” pour la production

51

Les réunions Scrum



52

Sizing: triangulation

5 8 13 21

1. On demande à 5 pourquoi c'est si simple
En tant que SM, il doit faire une étude poussée
2. On demande à 21
3. Le SM demande aux autres membres de trouver une autre manière de faire ça
4. Le PO doit choisir entre plusieurs options:
 - Op1: 21: tout est ok mais c'est dur
 - Op2: 5 : tout est ok mais c'est assez simple
 - Op3: 8 : ~

53

Sizing: triangulation

5 8 13 21

Le PO doit faire **une analyse business**:

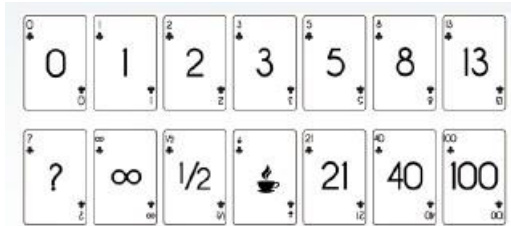
- 21: 500dt
- 5: 300dt
- 8: 450dt

Le PO doit calculer le ROI en terme financier: $450/8$

54

Le poker planning

- Le « poker planning » est une pratique Scrum qui permet d'évaluer la complexité d'un élément du Backlog Produit.
- Il s'agit de réunir l'équipe de réalisation, le Product Owner et le Scrum Master dans une même salle et de leur donner à chacun un jeu de cartes :



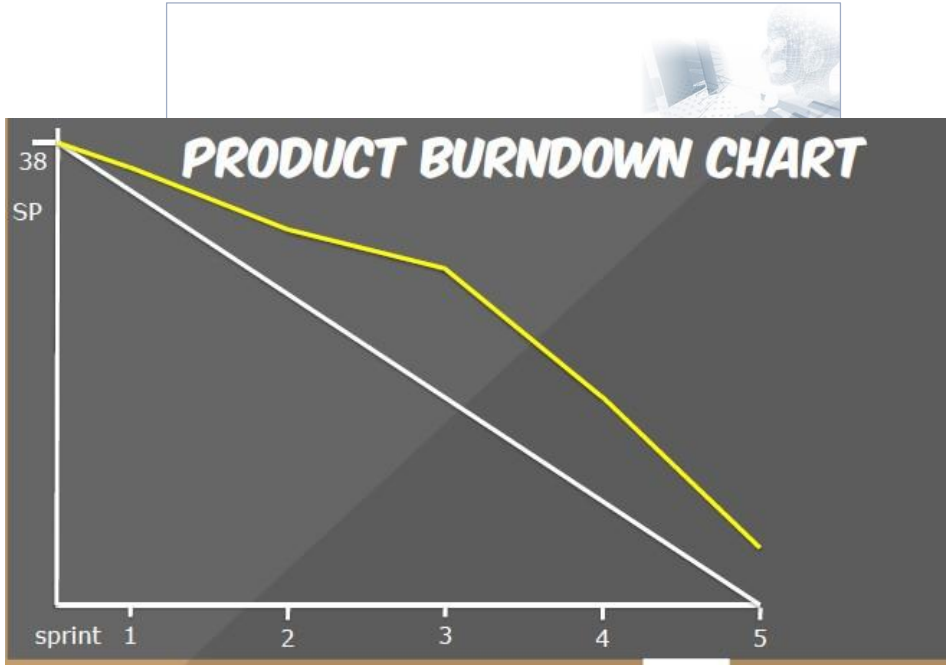
toute l'équipe s'exprime sur l'effort requis, en minimisant l'influence des autres membres de l'équipe.

55

Burndown chart de release

- Un **burndown** de release est un indicateur graphique basé sur la mesure de ce qui reste à faire.
- Un point dans le graphe est ajouté pour chaque sprint.
- Pour l'obtenir il faut donc une liste de ce qui reste à faire et une mesure de la taille de chaque élément, ce qu'on trouve dans le backlog de produit.
- À quoi sert le burndown chart de release ?
 - à montrer l'avancement réel, en tout cas le meilleur qu'on ait, puisqu'il est basé sur la distinction entre ce qui est complètement fini et ce qui reste.
 - à montrer la tendance et se poser des questions sur la façon de continuer.

56



La réunion de planification de sprint

Le planning de sprint

- Éviter que certains chefs de projet font la planification tout seuls, ils identifient des grandes tâches, les « chiffrent » et les affectent aux personnes de leur équipe.

Si les tâches n'avancent pas comme prévu, le chef de projet aura beau râler, les équipiers diront que les estimations n'étaient pas réalistes.

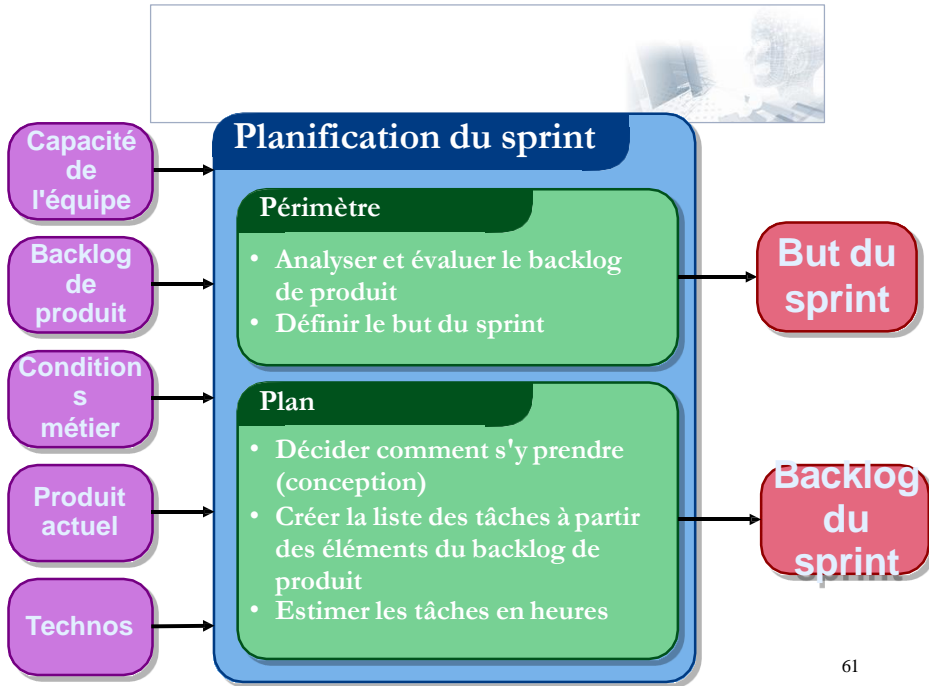
- La planification de sprint est basée sur l'idée qu'on ne peut pas prévoir de façon précise au-delà d'un certain horizon
- L'horizon pour la planification détaillée correspond au sprint (2 - 4 semaines).

59

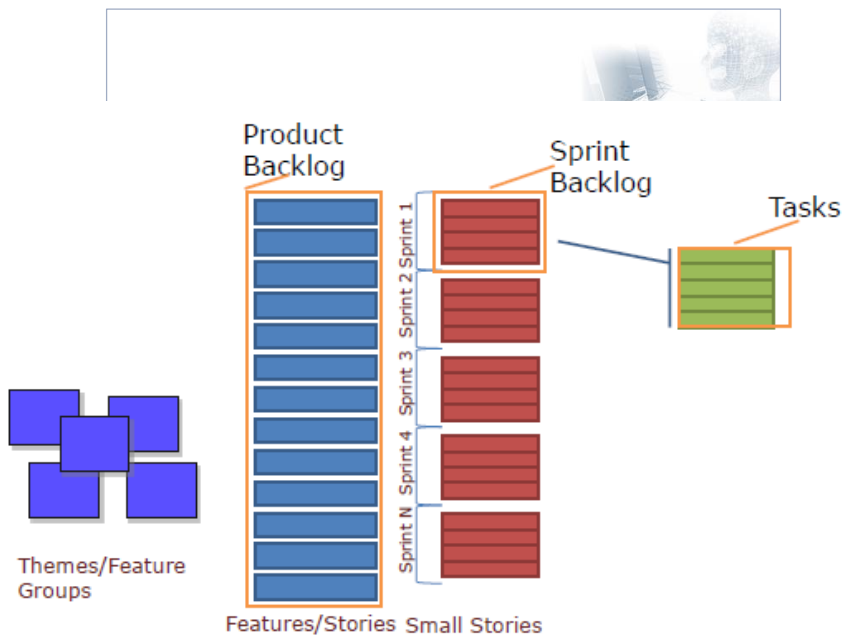
Le planning de sprint

- Cette réunion met en évidence, peut-être encore plus que pour la planification de release, le rôle essentiel de l'équipe dans l'élaboration des plans.
- Le travail du sprint appartient à l'équipe : ce n'est pas un chef qui définit ce qu'il y a à faire, c'est l'équipe qui s'organise elle-même.
- Au-delà de sa fonction première de planification, la réunion est un rituel qui prépare l'équipe à travailler de façon collective pendant le sprint.

60



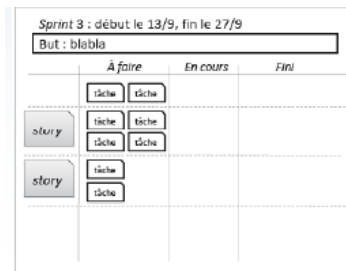
61



62

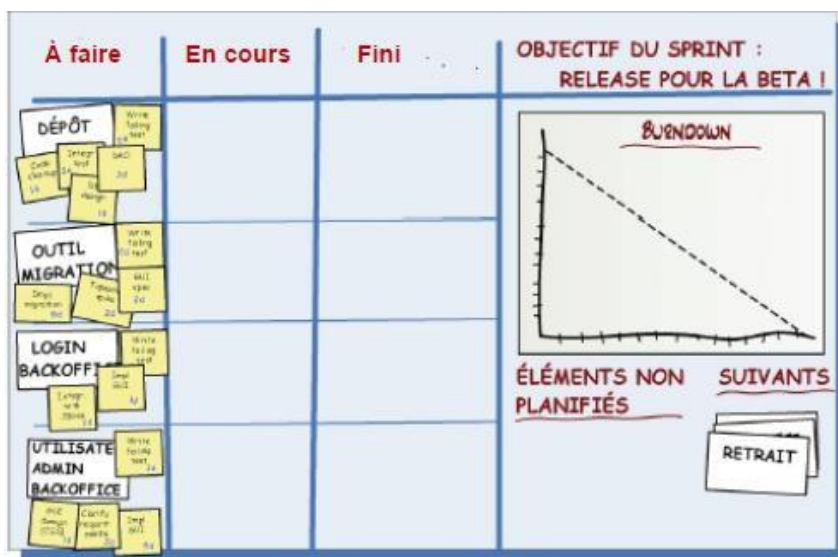
Tableau mural des tâches

- Il est élaboré lors de la réunion de planification du sprint.
- Pour chaque story sélectionnée, l'équipe identifie les tâches correspondantes.
- Sur le tableau, les stories et les tâches sont placées avec des **Post-it**.
- L'état des tâches est reconnu selon la place de la tâche dans des zones représentant chaque état : **à faire, en cours et finie**.



63

Plan de sprint




Faire de la planification de sprint



- Préparer le backlog de produit en anticipation
 - Laisser l'équipe décider du périmètre
 - Laisser l'équipe identifier les tâches
 - Décomposer en tâches courtes
 - Prendre un engagement raisonnable
 - Garder du mou dans le plan de sprint
 - Pour les incertitudes dans les estimations sur les tâches.
 - Pour le travail en anticipation sur le *sprint suivant*.
 - Pour les impondérables susceptibles de se produire.
- prévoir entre 10 et 30% de mou (temps non affecté)
- Faire de la conception tout au long des sprints

65



Le Scrum quotidien ou La Mêlée quotidienne

66

Une réunion quotidienne

- Le scrum quotidien est un point de rencontre où tous les membres de l'équipe répondent à trois questions simples et actualisent le plan de sprint :
 - **Présenter ce qui a été fait** : **Qu'as-tu fait depuis le dernier scrum?**
 - **Prévoir ce qui va être fait** : **Que prévois-tu de faire jusqu'au prochain scrum?**
 - **Identifier les obstacles** : **Quels sont les obstacles qui te freinent dans ton travail ?**
- Son but principal est d'optimiser la probabilité que l'équipe atteigne les objectifs du sprint.
- Les moyens pour atteindre ce but consistent en :
 - Éliminer les obstacles nuisant à la progression de l'équipe.
 - Garder l'équipe concentrée sur l'objectif du sprint.
 - Évaluer l'avancement du travail pour le sprint en cours.
 - Communiquer objectivement sur l'avancement.

67

Les obstacles potentiels

- Un obstacle empêche une tâche (ou plusieurs) de se dérouler normalement.
- La personne doit penser à ce qui l'a gêné dans les tâches qu'elle a faites.
- C'est au ScrumMaster de rechercher les vrais obstacles derrière les différentes formulations potentielles :
 - « *difficulté à communiquer avec le Product Owner !* ».
 - « *j'ai proposé deux façons de procéder pour l'arrangement des boutons sur la fenêtre de validation de la story 22 et j'attends toujours la réponse du Product Owner !* ».

68

Les obstacles potentiels

- Si un obstacle est identifié, trois situations sont envisageables:
 - Si un des membres de l'équipe connaît déjà la solution pour l'éliminer, il est bien évident qu'il la donne immédiatement.
un gain de temps considérable
 - si des personnes ont seulement des pistes de solution, une discussion peut s'engager, mais ce n'est pas le lieu adéquat pour la prolonger.
Le ScrumMaster arrête la discussion et propose aux personnes intéressées de la repousser à plus tard, en dehors du scrum.
 - Si aucun membres de l'équipe n'a une idée sur la solution du problème posé, la tâche est mise en attente et c'est au Scrum Master de rechercher les solutions
Le ScrumMaster peut faire intervenir des personnes extérieures

69

Réussir un Scrum quotidien

- S'en tenir à un **quart d'heure**
- Tout le monde est debout
- Ne s'intéresser qu'au reste à faire, pas au temps passé
- Faire le suivi des tâches avec les états plutôt que les heures
- Veiller à finir les stories
- Organiser des variations dans le déroulement du scrum:
 - la prise de parole se fait un coup de gauche à droite, un coup de droite à gauche ;
 - la réponse aux trois questions se fait une fois à la suite pour chaque personne, une autre fois par trois tours avec une question à chaque fois.
 - de la musique pour annoncer le scrum, qui change à chaque *sprint*
 - un ballon de rugby passé de main en main pour montrer qui a la parole.

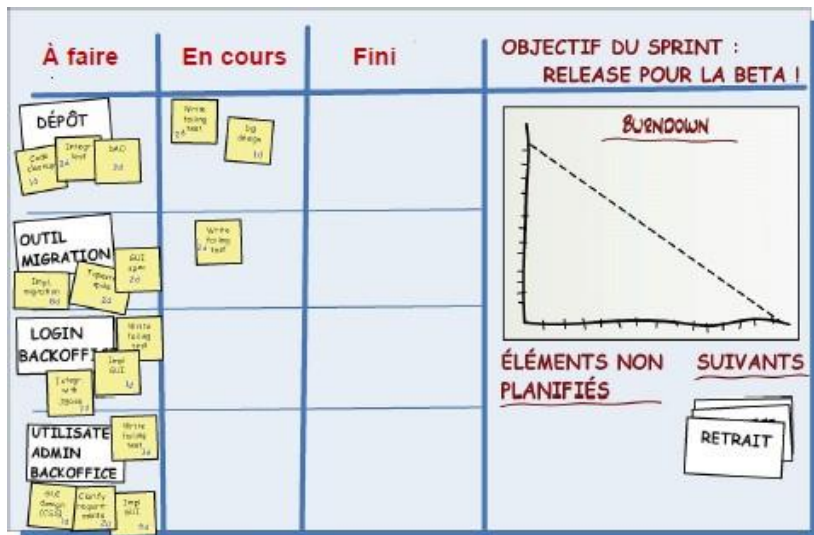
70

Réussir un Scrum quotidien



71

Après le premier Scrum quotidien





La revue de Sprint

73



- L'un des principes des méthodes agiles :
« *pour connaître l'avancement d'un développement, il vaut mieux se baser sur du logiciel fonctionnel plutôt que sur de la documentation* ».
 - Dans un développement de logiciel, on va montrer du code qui marche → démonstration de l'incrément de produit réalisé pendant le sprint.
- favoriser les notions de visibilité, inspection et adaptation.

74

La revue est basée sur une démo

Le but de la revue est de montrer ce qui a été réalisé pendant le *sprint afin d'en tirer* des enseignements pour la suite du projet.

Parmi les caractéristiques de cette réunion :

- **La revue accueille de nombreux invités**
 - Toute l'équipe Scrum participe à la réunion: l'équipe étendue, avec le ScrumMaster et le Product Owner.
 - Toutes les personnes qui sont parties prenantes (*stakeholders*) du projet y sont invitées et leur présence vivement encouragée.
 - C'est la réunion à laquelle assistent le plus grand nombre de personnes.
- C'est l'occasion de faire collaborer tous les intervenants sur l'avenir du produit.

La revue est basée sur une démo

- **Durée de la réunion**
 - La revue de sprint a lieu le dernier jour du sprint, elle sera suivie de la **rétrospective**.
 - La durée de la réunion de planification de sprint est à ajuster en fonction de la durée du sprint en question → limiter à **1*n heures**, où, **n étant le nombre de semaines dans le sprint**.
- Par exemple** : Pour un sprint de deux semaines, la revue a une limitation à 2 heures.
- La revue nécessite une préparation, au moins pour accueillir le public et être en capacité de leur faire la démonstration.
 - Le temps de préparation global ne devrait pas excéder une heure, car il n'y a pas de présentation formelle à faire : pas de slides à préparer.

76

La revue est basée sur une démo

- **La revue montre le produit**
 - La revue de sprint porte sur le résultat du travail de l'équipe pendant le sprint : le produit partiel potentiellement livrable.
C'est cette version opérationnelle qui est présentée.
 - Dans le cas d'un développement de logiciel, il a la forme d'un build incluant les stories finies, accompagné de ce qui est nécessaire pour le faire fonctionner (tests, documentation, scripts...) et déployé sur un environnement de test.

77

La rétrospective de Sprint

78

La rétrospective de Sprint

- L'un des principes des méthodes agiles :
«À intervalles réguliers, l'équipe réfléchit à comment devenir plus efficace, puis adapte et ajuste son comportement en conséquence».
- Cette réunion permet à l'équipe d'optimiser son auto-évaluation en effectuant une rétrospective du sprint
 - faire ressortir les éléments qui ont bien fonctionné et ceux qu'il reste à améliorer.
 - des actions concrètes sont alors proposées pour le sprint suivant