La méthode de Scrum
-partie1-

LP - IAM

Département Informatique- ESTS

Introduction

La **méthode Scrum**

- **« Scrum » signifie « Mêlée » en anglais**

- **le cadre méthodologique Scrum est de loin la méthode Agile la plus utilisée dans le monde.**

- **Expérimentée en** 1993, elle bénéficie aujourd'hui de nombreux retours d'expérience. Les conférences, communautés, formations, blogs, outils et ouvrages à son sujet ne manquent pas.

- La suite du chapitre va décrire le processus associé à la méthode Scrum, ses étapes, réunions, rôles, etc.

Introduction

- Scrum ne se considère pas comme une méthode mais comme un **cadre méthodologique.**
  - **Une méthode dit** généralement « comment » faire les choses.
  - Scrum ne dit pas comment réussir son logiciel, comment surmonter les obstacles, comment développer, comment spécifier, etc. Il se contente d'offrir un cadre de **gestion de projet Agile: des rôles, un rythme itératif, des réunions** précises et limitées dans le temps, des artefacts (product backlog, sprint backlog, graphique d'avancement) et des règles du jeu.

- Au sein de ce **cadre méthodologique de gestion de projet, les acteurs ajustent empiriquement, au** fil des itérations, leur propre méthode en fonction de leur contexte. On peut qualifier Scrum de simple, pragmatique, transparent et empirique.

- Scrum ne couvrant que les aspects de gestion de projet, c'est souvent la méthode **eXtreme Programming (XP) qui vient compléter le vide laissé en matière de** pratiques de développement.
  - **XP apporte ainsi les pratiques de programmation en binôme, de** développement piloté par les tests (**TDD ou Test Driven Development), intégration continue, etc.**

Utilisation de scrum
Pré requis recommandés

- Un grand mur libre et dégagé dans l'espace de travail de l'équipe.
- Blocs de post-it et marqueurs.
- Jeu de cartes ou logiciel de Planning Poker.

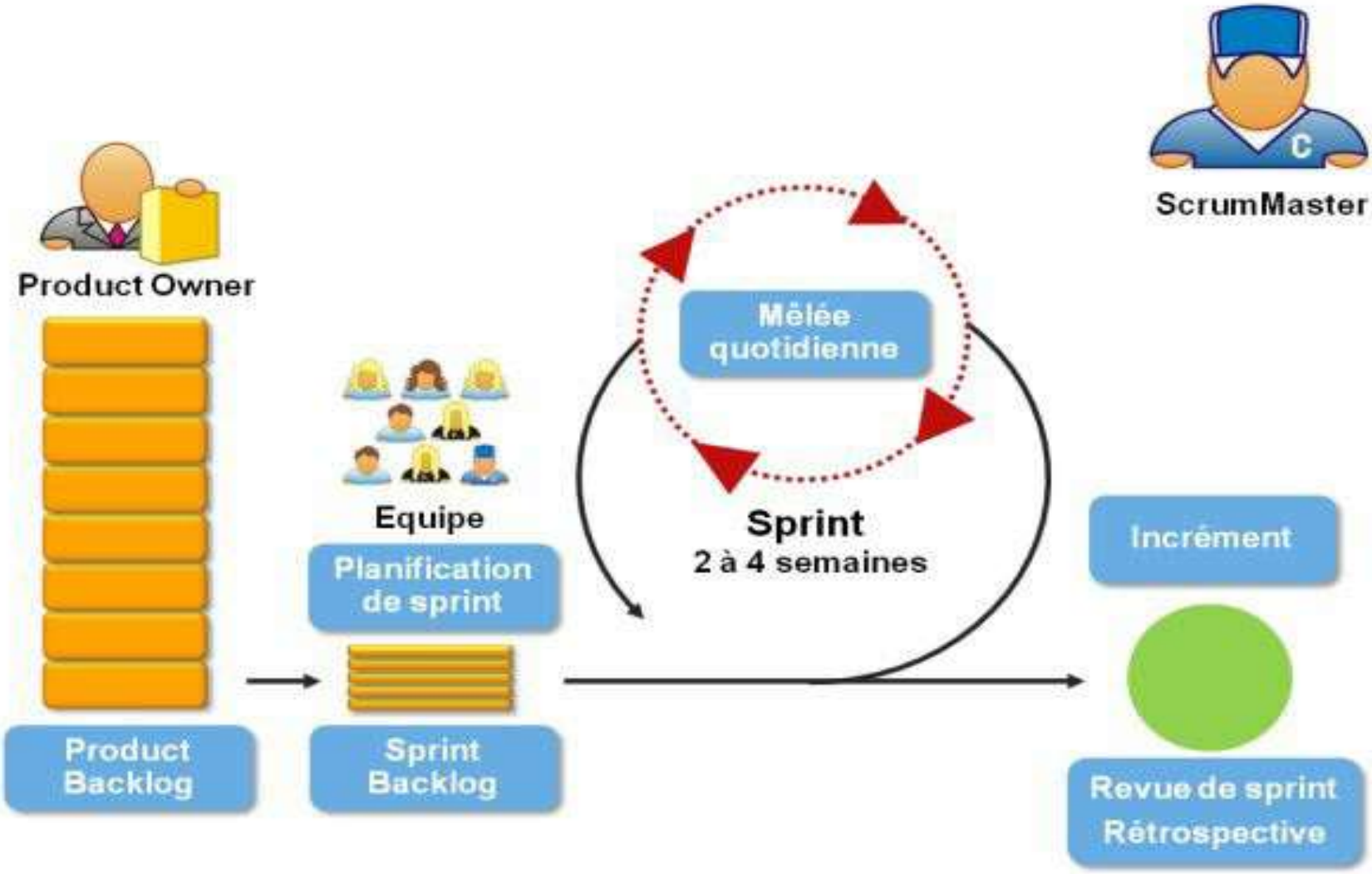
Utilisation de Scrum
Les rôles

Scrum définit seulement 3 rôles :

- Le **Product Owner** **qui** porte la **vision du produit à réaliser** et travaille en interaction avec l'équipe de développement. Il s'agit généralement d'un expert du domaine métier du projet.

- L'**Equipe de Développement** **qui est** chargée de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables. Elle est pluridisciplinaire et peut constituée de développeur, architecte logiciel, DBA, analyste fonctionnel, etc...

- Le **Scrum Master** **qui doit maîtriser Scrum et s'assurer que ce dernier est correctement appliqué. Il** a donc un rôle de **coach à la fois auprès du Product Owner et auprès de l'équipe de développement. Il** doit donc faire preuve de pédagogie. Il est également chargé de s'assurer que l'équipe de développement est pleinement productive. Généralement le candidat tout trouvé au rôle de Scrum Master est le chef de projet.



Processus Scrum (source des icônes des personnages : Mike Cohn)

Vision du produit et product backlog

- La première étape consiste à formaliser la **vision du produit c'est-à-dire le logiciel que l'on souhaite réaliser.**
  - Cette vision décrit les principaux objectifs, jalons, utilisateurs visés. Elle contribuera à guider et fédérer les acteurs du projet.
  - La suite consiste à établir la liste des **exigences fonctionnelles et non fonctionnelles du produit.**
  - **Chaque exigence est ensuite estimée par l'équipe de développement** avec la technique de **Planning Poker.**
  - **A la lueur des estimations, la liste ainsi complétée est** ordonnancée. Les exigences seront converties en fonctionnalités utilisables selon cet ordonnancement.
- Le principe étant de convertir en premier les exigences qui apportent le plus de valeur ajoutée (ou ROI: Return On Investment) au commanditaire. Il s'agit donc de faire remonter les exigences fonctionnelles de la plus haute valeur ajoutée (ou dont le ROI est le plus élevé) en haut de la liste.
- Cette liste est appelée le **Product Backlog.**

Vision du produit et product backlog

- **Le Product Backlog servira à piloter l'équipe de développement et** pourra évoluer tout au long du projet.
  - **Le changement est non seulement autorisé mais encouragé afin de pouvoir éliminer les idées de départ qui s'avéreront mauvaises et de prendre en compte les nouvelles idées qui arriveront en cours de route.**

- Cette activité de construction du **Product Backlog** est **collaborative**, elle implique le Product Owner et l'équipe de développement.

- **Et dans le cas d'un appel d'offre ? Vous répondez à un appel d'offre pour la réalisation du logiciel de** votre client ? Vous ne pouvez pas construire le **product backlog avec lui ?**
  - **Dans ce cas, vous pouvez à** partir du cahier des charges, extraire de ce dernier les exigences, les estimer (idéalement avec 3 membres de l'**équipe de développement pressentie) , proposer un ordonnancement et initialiser ainsi le Product Backlog.**

Estimations des exigences

- L'ensemble des fonctionnalités de la liste doivent être estimées par l' **équipe de développement afin** de permettre les futurs engagements de cette dernière.

- Impliquant plusieurs développeurs, le **Planning Poker permet de mettre à profit les expériences de chacun et de parvenir rapidement à une** estimation optimale et objective.

- Avant ou pendant les estimations, le Product Owner pourra être sollicité afin de répondre aux questions de l'équipe de développement.

- A ce stade, le besoin pourra être approfondi, mais sans aller trop loin (il s'agit simplement d'**estimer le coût de chaque exigence).**

- La conception détaillée se fera pendant les itérations (sprints).

Les champs des éléments du backlog

Les éléments du backlog (exigences, histoires) incluent les champs suivants :

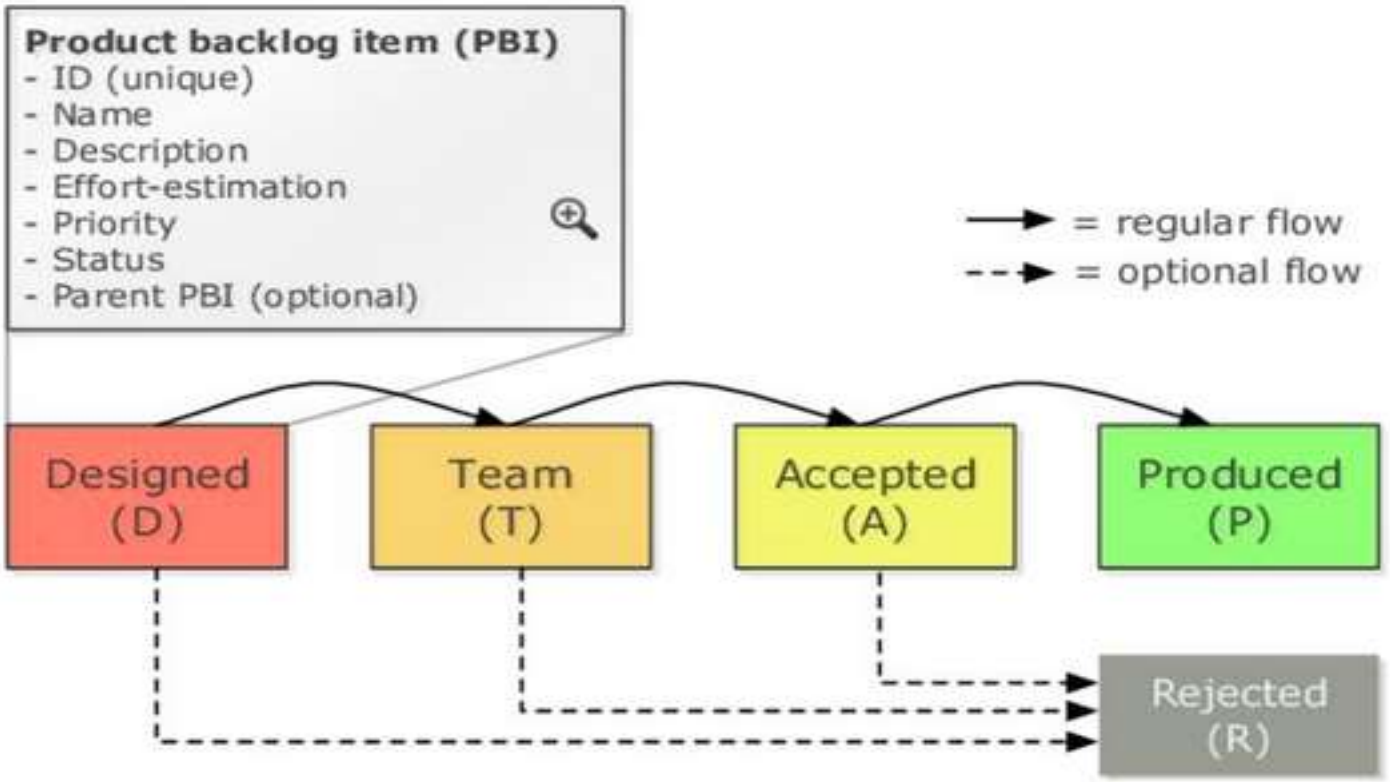
- **ID - un identifiant unique, juste un nombre auto-incrémenté. Cela** évite de perdre la trace des éléments quand on les renomme.

- **Nom - Un nom, une description courte de l'exigence. Par exemple** « Voir l'historique de ses transactions ». Suffisamment claire pour que les développeurs et le directeur de produit comprennent approximativement de quoi ils parlent, et suffisamment claire pour la distinguer des autres exigences. Normalement 2 à 10 mots.

- **Importance - l'importance attribuée par le directeur de produit à** cette exigence. Par exemple 10. Ou 150. Élevée = plus important.

- **Estimation initiale - l'évaluation initiale de l'équipe sur la** quantité de travail qui est nécessaire pour implémenter cette exigence par rapport aux autres exigences. **L'unité est les points d'histoire habituellement des «jours*hommes idéaux ».**

Les champs des éléments du backlog



Les champs des éléments du backlog

- Demandez à l'équipe « si vous pouvez prendre le nombre optimal de personnes pour cette histoire (ni trop peu ni trop nombreux, typiquement 2), et que vous vous enfermez dans une salle avec plein de nourriture et que vous travaillez sans jamais être dérangé, après combien de jours sortiriez-vous avec une implémentation finie, démontrable, testée, et livrable ? ». Si la réponse est « avec 3 gars enfermés dans une salle ça prendrait approximativement 4 jours » alors l'estimation initiale est de 12 points d'histoire.

Les champs des éléments du backlog

- **Comment le démontrer - une description succincte de comment** cette exigence sera démontrée à la démo de fin d'itération. C'est essentiellement la spécification d'un test simple. « Fais ci, ensuite fais ça, puis ceci devrait arriver ».
  - cette description peut être utilisée comme du pseudo-code pour les tests d'acceptance dans la pratique du DDT (Développement Dirigé par les Tests)
- **Notes - n'importe quelle autre info, des clarifications, des** références aux autres sources d'info, etc. Normalement très bref.

BACKLOG DE PRODUIT (exemple)					
ID	Nom	Importance	Estimation	Démonstration	Notes
1	Dépôt	30	5	Authentification, ouvrir la page de dépôt, déposer 10 dh, aller sur la page du solde et vérifier que ça a bien augmenté de 10dh.	Nécessite un diagramme de séquences UML. Ne pas se soucier du cryptage pour l'instant.
2	Voir l'historique de ses transactions	10	8	Authentification, cliquez sur « transactions ». Faire un dépôt. Revenir aux transactions, vérifier que le nouveau dépôt apparaît.	Utiliser la pagination pour éviter des requêtes volumineuses . Conception semblable à la page des utilisateurs.

Champs supplémentaires

Parfois des champs supplémentaires sont utilisés dans le product backlog, surtout par commodité pour le directeur de produit pour l'aider à trier ses priorités.

- **Catégorie - un classement approximatif de cette histoire, par** exemple « back office » ou « optimisation ». De cette façon le directeur de produit peut facilement filtrer tous les éléments « optimisation » et leur affecter une priorité basse, etc.

- **Composants -** Habituellement représentés par des **« cases à cocher** » dans un tableau Excel, par exemple « base de données, serveur, client ». Ici l'équipe ou le directeur de produit peuvent identifier quels composants techniques vont être impliqués dans l'implémentation de cette histoire.

Champs supplémentaires

- **Demandeur -** le directeur de produit peut vouloir garder une trace de quel client ou quelle partie prenante avait demandé cet élément à l'origine, dans le but de les informer sur l'avancement.

- **ID de suivi de bug -** si vous avez un système de suivi des bogues séparé, il est utile de garder une trace de toute correspondance directe entre une fonctionnalité et un ou plusieurs bogues.

User Stories

Exemple de User Story.

- **ID** : #33
- **Titre** : Paiement par carte bleue
- **Résumé** : En tant que client du site internet FoireAuLivre.com je peux payer ma commande par carte bleue afin de me simplifier la vie.
- **Importance**: 120
- **Estimations**: 12
- **Tests d'acceptations** :
  - Tester avec une carte bleue mastercard : OK
  - Tester avec une carte sofinco : KO
  - Tester avec une carte expirée : KO
  - Tester avec un numéro de carte incorrect : KO
  - Tester avec un montant supérieur à 100 € : OK
- **Notes** :…

Le Démarrage

- L'**équipe de développement et le product owner peuvent** commencer par déterminer ensemble la **durée des itérations ou Sprints (4 semaines maximum).**

- **Cette durée devra être la même pour** l'ensemble des sprints afin de maintenir un rythme régulier propice aux automatismes et pouvoir construire des indicateurs de pilotage fiables.

- Un projet démarre généralement par ce qu'on appelle souvent le « **sprint 0 » dédié aux travaux préparatoires du projet (ex : construction du product backlog et de la vision du produit, préparation des environnements,** mise en place de l'intégration continue, définition de l'architecture générale du projet, initiation des acteurs à Scrum, etc.).

Comment préparer la réunion de planning du sprint

- **Leçon : Faites en sorte que le backlog de produit soit en ordre *avant cette* réunion de planification du sprint**. Cela signifie:
  - le backlog de produit devrait exister !
  - il devrait y avoir *un backlog de produit et un directeur de produit*
  - un niveau d'importance devrait avoir été attribué à tous les éléments importants, et ces niveaux devraient être *différents.*

- Si le directeur de produit croit qu'une histoire a une probabilité (même très faible) d'être adoptée dans la prochaine itération, alors cette histoire devrait avoir un niveau d'importance unique.

- Le niveau d'importance est utilisé uniquement pour trier les éléments par importance. Donc si l'élément A a une importance de 20 et l'élément B une importance de 100, cela signifie simplement que B est plus important que A. Cela *ne signifie pas que B est cinq fois plus important* que A. Si B avait pour niveau d'importance 21, cela signifierait exactement la même chose !

Comment préparer la réunion de planning du sprint

- Il est utile de laisser des trous dans la séquence des niveaux pour le cas où un élément C arrive et est plus important que A mais moins important que B. Bien sûr un niveau d'importance de 20,5 peut être utilisé mais cela devient déplaisant, donc il faut laisser des trous à la place !

- le directeur de produit devrait *comprendre chaque fonctionnalité* (normalement il en est l'auteur, mais il arrive que d'autres personnes ajoutent des demandes, dont le directeur de produit peut choisir la priorité). Il n'a pas besoin de savoir exactement ce qui doit être implémenté, mais il devrait comprendre pourquoi cette fonctionnalité est là.

- **Note : d'autres personnes que le directeur de produit peuvent ajouter des exigences** au backlog de produit. Mais elles n'ont pas à attribuer un niveau d'importance, seul le directeur de produit en a le droit. Elles n'ont pas à ajouter d'estimations non plus, seule l'équipe en a le droit.

Réunion de planification du sprint

- Le planning de sprint est une réunion critique, probablement l'événement le plus important dans Scrum
- Voilà l'objectif de cette réunion qui servira également à répondre à deux questions importantes :
  - qu'est-ce qui peut être fait dans ce sprint ?
  - comment transformer les éléments du product backlog que l'on a sélectionné en un incrément terminé ?

Réunion de planification du sprint

- Le but de la réunion de planning de sprint est de donner à l'équipe suffisamment d'informations pour qu'elle soit capable de travailler paisiblement, sans dérangements pendant quelques semaines, et de donner au directeur de produit suffisamment de confiance pour les laisser faire.

- Concrètement, à la fin de cette réunion on doit avoir :
  - Un but pour le sprint.
  - Une liste des membres d'équipe (et de leur niveau d'engagement, si ce n'est pas 100%).
  - Un backlog de sprint (= une liste des histoires incluses dans le sprint).
  - Une date bien définie pour la démonstration.
  - Une heure et un lieu bien définis pour la mêlée quotidienne.

Réunion de planification de sprint

- *Durée maximum : 2 heures par semaine de sprint (autrement dit : 4 heures pour des sprints de 2 semaines).*

Phase 1 : Le « Quoi »

- Une fois que le **Product Backlog est suffisamment complet et ordonnancé, on peut planifier un sprint.**

- Le **Product Owner revoit alors avec l'équipe de développement la vision du produit, la roadmap, le** plan de livraison (jalons et deadline), l'objectif du sprint et le Product Backlog.

- L'équipe de développement vérifie les estimations, confirme qu'elles sont exactes et sélectionne en haut du Product Backlog les exigences qu'elle se sent capable de convertir en fonctionnalités utilisables d'ici la fin du sprint (il s'agit d'une prévision et non pas d'un engagement « contractuel »).

- L'estimation est déterminée par l'équipe. Durant une réunion de planning de sprint, ces trois variables sont affinées continuellement grâce au dialogue face-à-face entre l'équipe et le directeur de produit.

Réunion de planification de sprint

Phase 2 : Le « Comment »

- L'**équipe de développement fait ensuite l'inventaire des tâches qui permettront de convertir les** exigences sélectionnées en fonctionnalités utilisables d'ici la fin du sprint.

- Toutes les exigences n'ont pas nécessairement besoin d'être découpées en tâches.

- En cas de manque de temps, l'équipe de développement peut se contenter de découper celles qui seront réalisées au cours des premiers jours du sprint (elle découpera en cours de sprint les autres exigences).

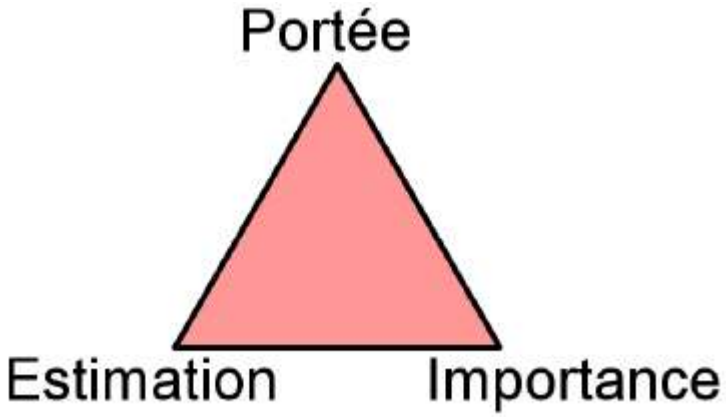
- Elle doit cependant aller suffisamment loin dans l'effort de conception pour pouvoir vérifier sa prévision. Si elle constate après analyse des exigences sélectionnées, que sa prévision est erronée, elle peut réajuster avec le Product Owner la liste des exigences sélectionnées

Réunion de planification du sprint

- L'équipe de développement au complet ainsi que le directeur de produit doivent être à cette réunion de planning de sprint car chaque exigence contient trois variables qui dépendent fortement les unes des autres.

- La portée et l'importance sont déterminées par le directeur de produit.

- L'estimation est déterminée par l'équipe. Durant une réunion de planning de sprint, ces trois variables sont affinées continuellement grâce au dialogue face-à-face entre l'équipe et le directeur de produit.



Réunion de planification du sprint

- Normalement le directeur de produit commence la réunion en résumant son but pour le sprint et les histoires les plus importantes. Ensuite l'équipe parcourt chaque histoire et estime le temps qu'il faudra pour la réaliser, en commençant par la plus importante. Pendant qu'ils font ça, ils auront des questions importantes sur la portée – « est-ce que cette histoire de suppression d'utilisateur comprend le fait de parcourir chaque transaction en attente pour cet utilisateur, et de l'annuler? » Dans certains cas les réponses surprendront l'équipe et la conduiront à changer son estimation.

- Dans d'autres cas l'estimation du temps pour une histoire ne sera pas ce que le directeur de produit attendait. Cela peut le conduire à changer l'importance de l'histoire. Ou à changer la portée de l'histoire, ce qui conduira l'équipe à ré-estimer, etc, etc.

- Ce type de collaboration directe est fondamental dans Scrum et en fait dans tout développement logiciel agile.

La qualité n’est pas négociable

Dans le triangle ci-dessus, une quatrième variable ***qualité*** n’apparaît pas

- La *qualité externe est ce qui est perçu par les utilisateurs du* système. Une interface utilisateur lente et pas intuitive est un exemple de mauvaise qualité externe.

- La qualité interne fait référence à des points qui habituellement ne sont pas visibles par l'utilisateur, mais qui ont un profond effet sur la maintenabilité du système. Des choses comme la cohérence de la conception, la couverture de test, la lisibilité du code, les remaniements (*refactoring).*

- En général, un système avec une qualité interne élevée peut tout de même avoir une qualité externe faible. Mais un système avec une faible qualité interne aura rarement une qualité externe élevée. Il est difficile de bâtir quelque chose de bien sur des fondations pourries.

- La qualité externe est traitée comme faisant partie de la portée. Toutefois la qualité interne n'est pas sujette à discussion. C'est la responsabilité de l'équipe de maintenir la qualité du système dans toutes les circonstances et ceci est simplement non négociable. Jamais.

L’ordre du jour de la réunion de planification du sprint

- Comment éviter que, les réunions de planning de sprint , s’éternisent?
  - Avoir un ordre du jour préliminaire pour cette réunion réduira le risque de ne pas respecter la limite de temps.

Voici un exemple d’un plan typique:

Réunion de planning de sprint : 13:00 – 17:00 (10 minutes de pause toutes les heures)

13:00 – 13:30. Le directeur de produit décrit le but du sprint et résume le backlog de produit. Le lieu, la date et l’heure de la démonstration sont fixés.

13:30 – 15:00. L’équipe fait les estimations en temps, et décompose les éléments selon les besoins. Le directeur de produit met à jour les niveaux d’importance selon les besoins. Les éléments sont clarifiés. « Comment démontrer » est explicité pour chacun des éléments les plus importants.

L’ordre du jour de la réunion de planification du sprint

15:00 – 16:00. L’équipe sélectionne les histoires à inclure dans le sprint. Elle fait les calculs de vélocité pour se confronter à la réalité.

16:00 – 17:00. On choisit le lieu et l’heure pour la mêlée quotidienne (s’ils sont différents du dernier sprint). On poursuit la décomposition des histoires en tâches.

- Ce plan n’est en aucune façon respecté trop strictement. Le Scrum master peut allonger ou raccourcir ces limites de temps comme nécessaire au fur et à mesure que la réunion progresse.

Le choix de la durée du sprint

- L'un des produits de cette réunion de planning de sprint est une date de démonstration bien définie.
  - Cela signifie que vous devez décider d'une durée de sprint.

Donc quelle la bonne durée d'un sprint ?

- Les sprints courts sont bénéfiques. Ils autorisent l'entreprise à être « agile », c'est-à-dire à changer souvent de direction. Les sprints courts = cycle de feedback court = des livraisons plus fréquentes = plus de feedback des clients = moins de temps passé à courir dans la mauvaise direction = apprendre et améliorer plus rapidement, etc.

- Cependant, les sprints longs sont bien aussi. L'équipe a plus temps pour monter en puissance, ils ont plus de temps pour récupérer des problèmes et parvenir tout de même à atteindre le but du sprint, il y a moins de surcharge en terme de réunions de planning de sprint, de démonstrations, etc.

Le choix de la durée du sprint

- En général les directeurs de produit aiment les sprints courts et les développeurs aiment les sprints longs. Donc la durée du sprint est un compromis.

- Une durée moyenne de 3 semaines est correcte. des sprints de 3 semaines, c'est assez court pour donner suffisamment d'agilité d'entreprise, et suffisamment long pour permettre à l'équipe d'avoir du débit et de récupérer des problèmes qui surgissent au cours du sprint.

- **Conclusion** : **faites des expériences avec la durée des sprints au début**. Ne perdez pas trop de temps *à analyser,* ***choisissez*** **simplement une durée de sprint correcte, essayez là pour un sprint ou deux, puis changez cette durée si elle ne convient pas.**

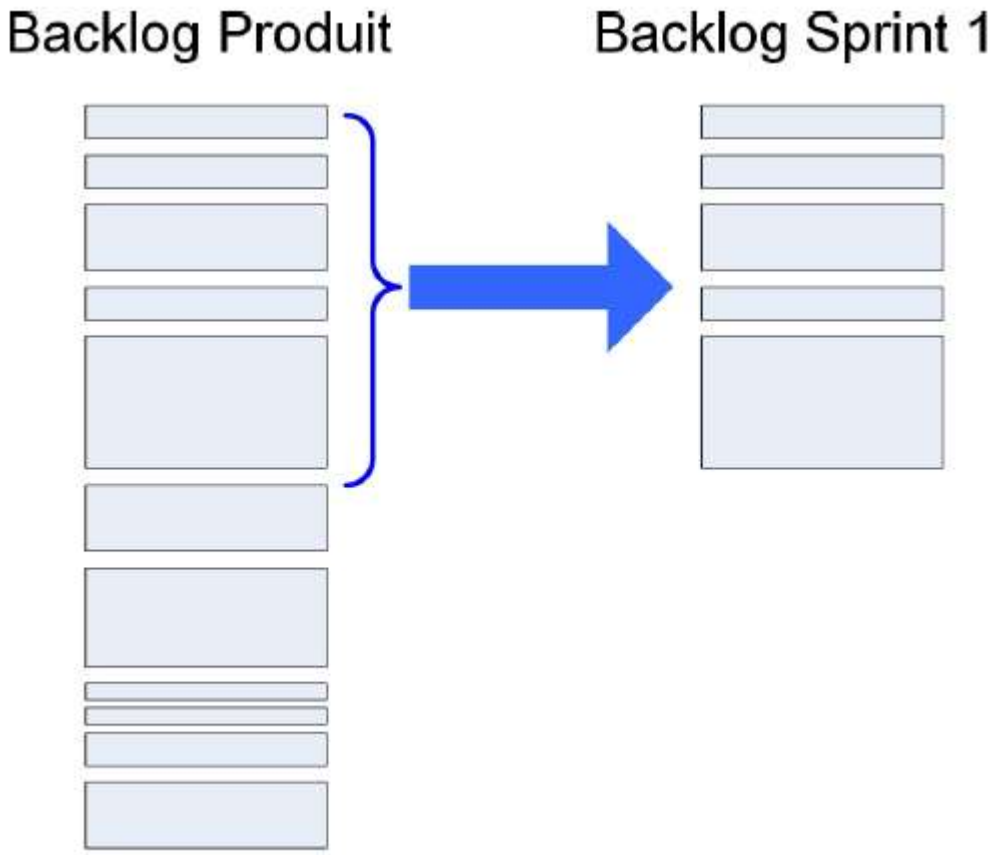
La définition du but du sprint

- Il se trouve qu'il est *difficile de déterminer un but de sprint.*
- Le but du sprint devrait répondre à la question fondamentale **«Pourquoi faisons-nous ce sprint?».**
- Le point important est que le but devrait être défini en termes métier, pas en termes techniques.

NB: Si vous avez plusieurs équipes Scrum qui travaillent sur différents produits, il est très utile de lister les buts de sprint de toutes les équipes sur une seule page et de les mettre à un endroit bien visible, de telle sorte que tout le monde dans l'entreprise sache ce que l'entreprise est en train de faire – et pourquoi !

Le choix des exigences à inclure dans le sprint

- L'une des activités principales de la réunion de planning de sprint est de décider quelles exigences du backlog de produit faudra-t- il inclure dans le backlog du sprint.



Le choix des exigences à inclure dans le sprint

- Chaque rectangle représente une exigence, triée par importance.
  - L'histoire la plus importante est au sommet de la liste.
  - La taille de chaque rectangle représente la taille de cette histoire (c'est-à-dire l'estimation de temps en points d'histoire).
  - La hauteur de l'accolade bleue représente la **vélocité estimée de l'équipe**, c'est-à-dire combien de points d'histoire, l'équipe estime qu'elle pourra terminer durant le prochain sprint.
- Le backlog de sprint à droite est un ensemble d'histoires du backlog de produit. Il représente la liste des histoires sur lesquelles l'équipe va s'engager durant ce sprint.

Le choix des exigences à inclure dans le sprint

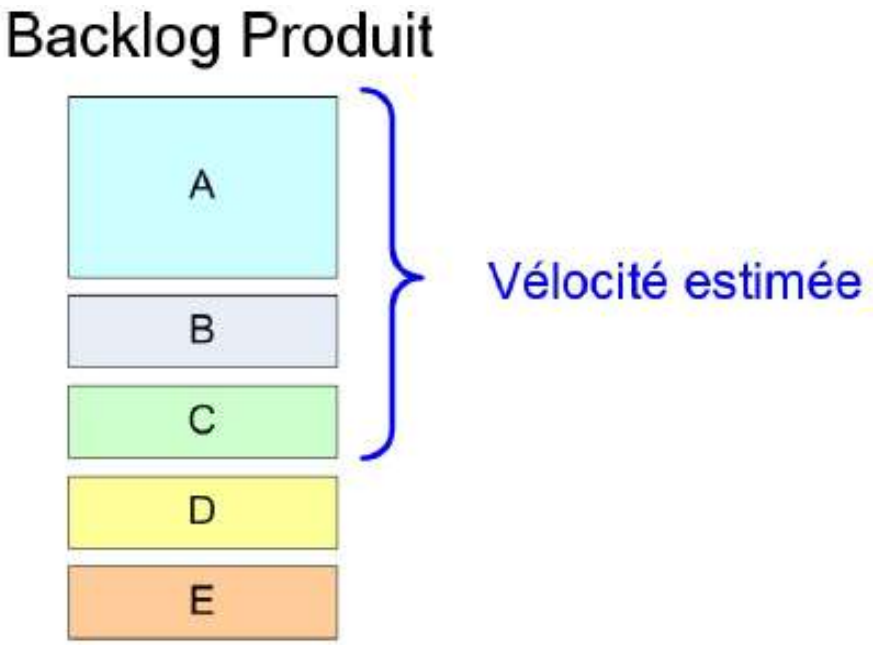
- C’est l’équipe qui décide combien d’histoires vont être prises en compte dans le sprint. Ce n’est ni le directeur de produit ni qui que ce soit d’autre.

Cela soulève deux questions :

1. Comment l’équipe décide-t-elle quelles histoires inclure dans le sprint ?
2. Comment le directeur de produit peut-il influencer leur décision ?

Comment le directeur de produit peut-il influencer leur décision ?

- Supposons qu'on se trouve devant la situation suivante durant une réunion de planning de sprint.

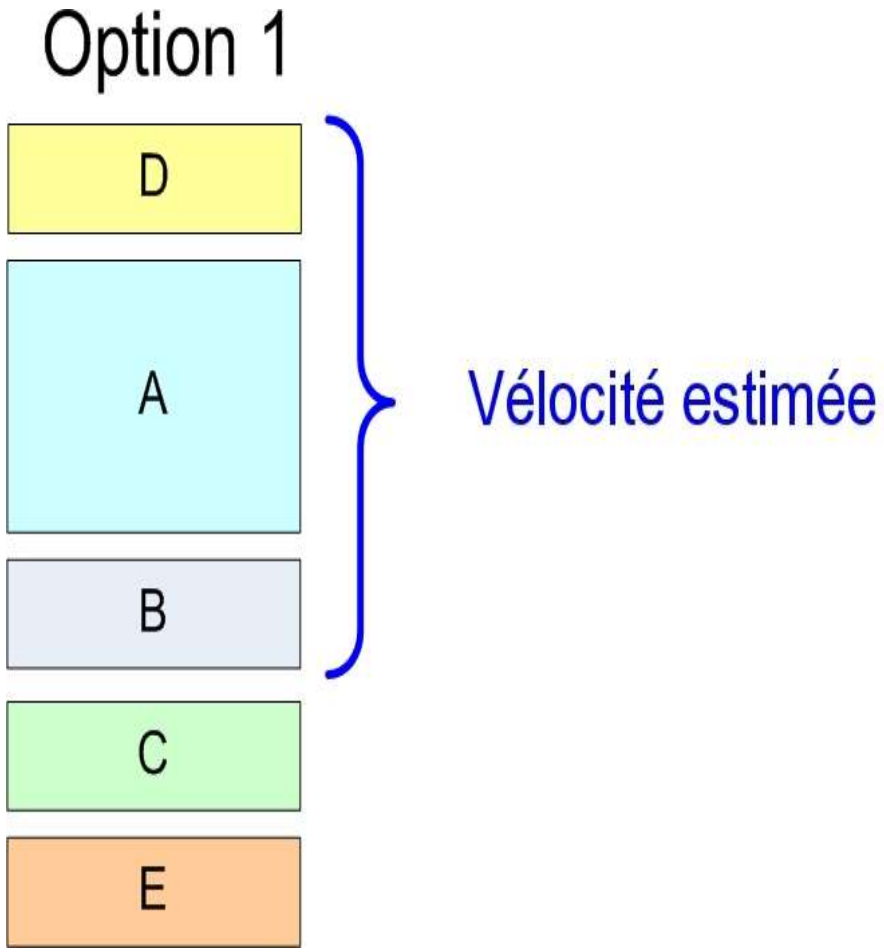


- Le directeur de produit aimerait que l'exigence D soit incluse dans le sprint, alors quelles sont les choix à faire pour inclure cette exigence durant la réunion?

Comment le directeur de produit peut-il influencer leur décision ?

Option1: Redéfinir les priorités.

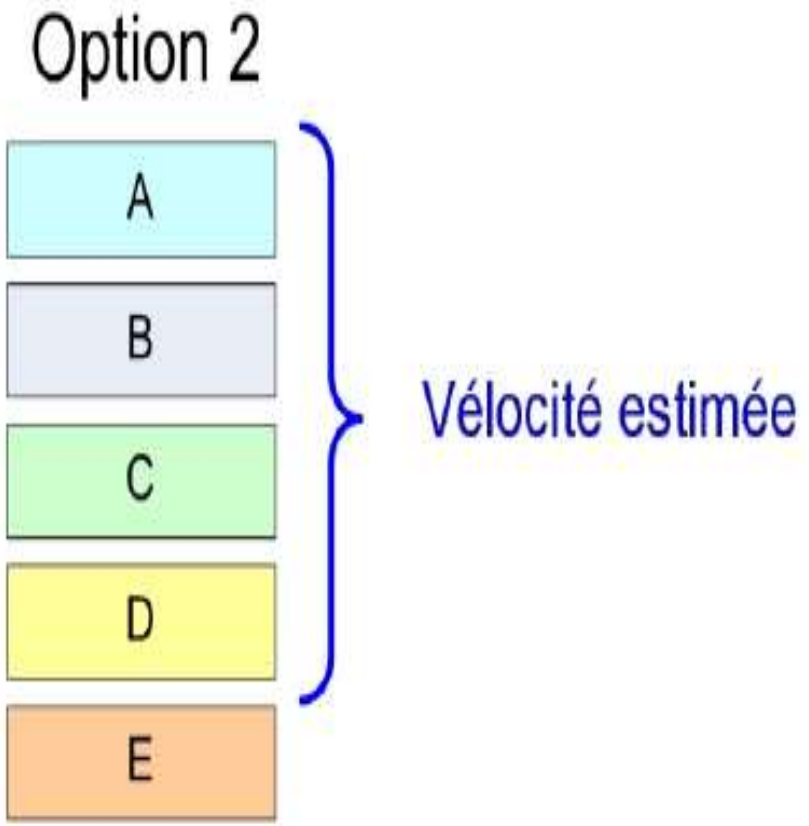
Si le directeur de produit donne à l'élément D le plus haut niveau d'importance, l'équipe sera obligée de l'ajouter en premier dans le sprint (et dans ce cas éjecter l'histoire C).



Comment le directeur de produit peut-il influencer leur décision ?

Option2: Changer la portée

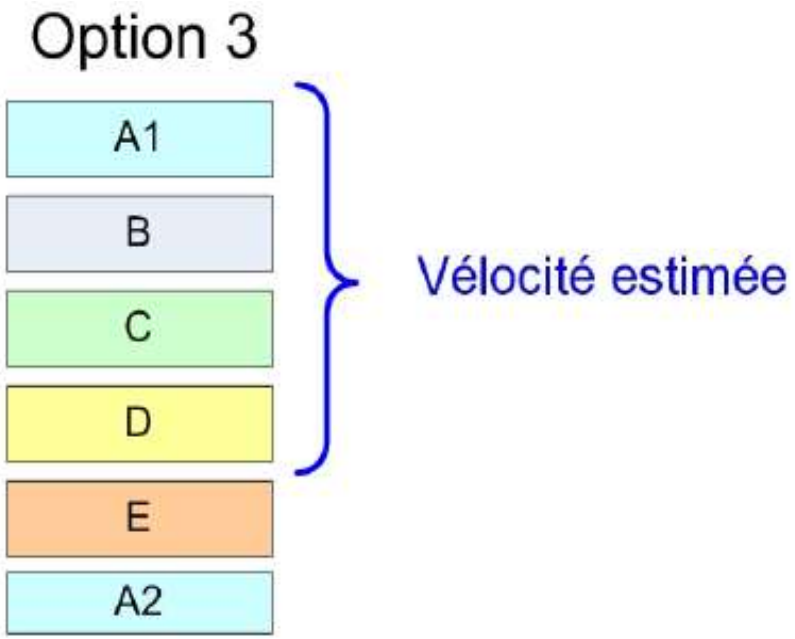
Le directeur de produit décide de réduire la portée de l'exigence A jusqu'à ce que l'équipe estime que l'histoire D va pouvoir tenir dans le sprint.



Comment le directeur de produit peut-il influencer leur décision ?

Option3: Partager l'exigence en de sous exigences

Le directeur de produit pourrait décider qu'il y a certains aspects de l'histoire A qui ne sont pas finalement si importants, et donc il partage A en deux sous exigences A1 et A2 avec des niveaux d'importance différents.



- Comme vous le voyez, même si le directeur de produit ne puisse normalement pas contrôler la vélocité estimée, il y a plusieurs moyens par lesquels il peut exercer une influence sur les histoires qui seront prises dans le sprint.

Comment l'équipe décide-t-elle quelles histoires inclure dans le sprint ?

L'équipe utilise deux techniques d'estimation afin de décider quelles exigences doit être incluses dans le sprint:

1. L'intuition
2. Le calcul de vélocité basé sur:
    1. La météo de la veille
    2. Le nombre de jours*hommes disponible et l'estimation du facteur de focalisation

Estimation par l'intuition

Le scrum master commence par demander aux membres de l'équipe s'ils pensent finir l'exigence de plus haute importance du Backlog durant le sprint? Si c'est le cas, il demande encore si l'histoire de niveau d'importance suivante peut être incluse dans le sprint et ainsi de suite jusqu'à ce que les membres de l'équipe lui signifient qu'ils sont arrivés au plafond pour ce sprint et que les prochaines fonctionnalités devront être incluses dans le prochain sprint?

L'intuition marche plutôt bien pour des petites équipes et des sprints courts.

Exemple d'estimation par l'intuition

- **Scrum master: «**Peut-on finir l'histoire A dans ce sprint?» (montrant l'élément le plus important du backlog de produit)
- **Hamid: «**Bien sûr que nous pouvons. Nous avons 3 semaines**,** et c'est une fonctionnalité plutôt facile.»
- **Scrum master: «**OK, et nous rajoutons aussi l'histoire B **?»** (montrons le deuxième élément le plus important)
- **Hamid & Ali ensemble: «**Pas de problème» .
- **Scrum master: «**OK, et avec les histories A, B et C maintenant ? »
- **Anisse (à l'adresse du Directeur de produit): «**Est-ce que l'histoire C comporte une gestion des erreurs avancée?»
- **Directeur de produit: «**non, vous pouvez la laisser de coté pour le moment, implémentez juste une gestion basique des erreurs. «
- **Anisse:** «alors C devrait être incluse aussi**.»**

Exemple d'estimation par l'intuition

- **Scrum master: «**OK, et si nous ajoutons l'histoire D ?»
- **Hamid: «**Hum**....»**
- **Ali: «**Je pense que nous pouvons le faire**.»**
- **Scrum master: «**Confiant à 90% ? à 50% ?»
- **Hamid & Ali: «**A peu près 90%».
- **Scrum master: «**OK, prenons D. Et si nous ajoutons l'histoire E? »
- **Anisse: «**Peut-être»**.**
- **Scrum master: «**90%? 50%?»
- **Anisse: «**Je dirais plus proche de 50%».
- **Hamid: «**Je ne suis pas sûr.»
- **Scrum master: «**OK, alors ne la prenons pas. Nous nous engagerons pour A, B, C, et D. Nous finirons E bien sûr si nous le pouvons, mais personne ne devrait compter dessus, du coup nous la laisserons en dehors du plan du sprint. Qu'en dîtes-vous? »
- **Tout le monde: «**OK**!»**

Estimation par calcul de vélocité

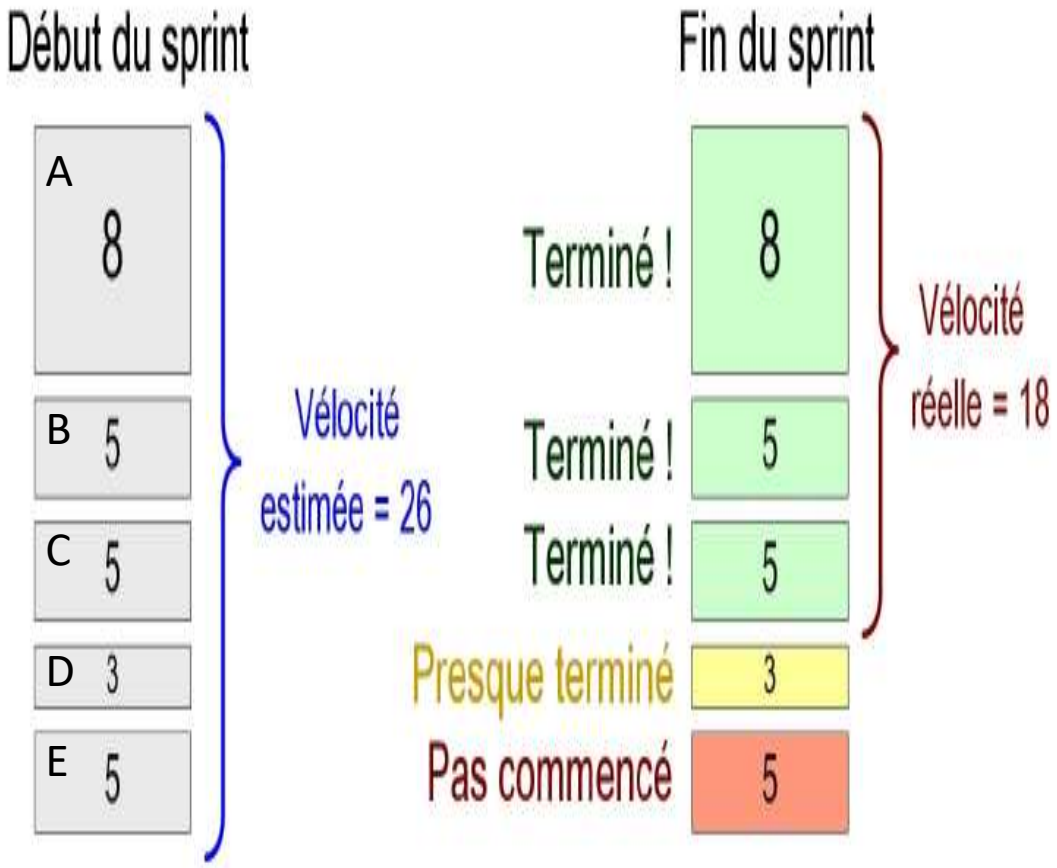
- A la fin d'un sprint, l'équipe additionne les estimations associées aux histoires qui ont été terminées au cours de ce sprint. Ce total est appelé **vélocité**.

- La vélocité mesure la «quantité de travail fini», où chaque élément est pondéré selon son estimation initiale.

Estimation par calcul de vélocité

- L'exemple suivant montre une **vélocité estimée au démarrage** d'un sprint et **la vélocité effective à la fin** du sprint.

- Chaque rectangle est une user story, et le numéro à l'intérieur est l'estimation initiale de cette histoire.



Estimation par calcul de vélocité

- Dans l'exemple précédent les histoires A,B,C sont terminées à 100% et que l'histoire D est presque fini à peu prés à 80% et l'histoire E n'est pas encore commencé.

- Que dire alors d'une histoire qui est *presque finie durant un sprint ? Pourquoi* ne pas compter les points partiellement accomplis pour celle-ci dans notre vélocité effective ?

- Scrum est entièrement focalisé sur l'obtention de quelque chose de complètement terminé, pouvant être livré ! **La valeur d'une chose à moitié fini est zéro**. Par conséquent D n'est pas comptabilisée, et la vélocité de l'itération est de 18 points.

Estimation par calcul de vélocité

- Supposons que la totalité des stories du projet représente 180 points. En émettant l'hypothèse que la vélocité sera à peu près la même au prochaines itérations , on peut prévoit alors une durée totale du projet équivalente à 10 sprint.

- Cette technique est connue sous le nom de **la météo de la veille**.

- Elle est faisable seulement avec des équipes qui ont déjà fait quelques sprints (des statistiques sont alors disponibles) et qui feront le prochain sprint de la même façon, avec la même taille d'équipe et dans les mêmes conditions de travail, etc. Bien sûr ceci n'est pas toujours le cas.

Estimation par calcul de vélocité

Une alternative consiste à faire un simple calcul de ressource.

supposons que nous planifions un sprint de 3 semaines (15 jours de travail) avec une équipe de 4 personnes.

- Hamid est en congés durant 2 jours.
- Jamal est disponible seulement à 50% et il sera en congés durant 1 jour.

Membre de l'équipe	Jours disponibles
Hamid	13
Ali	15
Anisse	15
Jamal	7
Total (jours hommes disponibles)	50

Cela donne 50 jours-homme pour ce sprint

Estimation par calcul de vélocité

- Le tableau précédent ne permet pas à lui seul d'estimer vélocité estimée.
- **L'unité d'estimation pour la vélocité est le point d'histoire qui, dans notre cas, correspond à peu près à autant de «jours-homme idéaux».**
- Un **jour-homme idéal est un jour de travail, parfaitement efficace, sans perturbation**, ce qui est rare.
- De plus, si on doit tenir compte de choses comme le travail inattendu qui va s'ajouter au sprint, des personnes malades, etc.
  - notre vélocité estimée sera alors, certainement inférieure à 50.
  - Mais de combien sera t- elle inférieure ?
    - Nous utilisons à cette fin le terme **«facteur de focalisation».**

Estimation par calcul de vélocité – Facteur de focalisation

Pour le sprint, le calcul de la vélocité estimée se fait selon la formule suivante:

Vélocité estimée pour le sprint en cours = (jours*hommes disponibles pour le sprint en cours) x (facteur de focalisation du sprint précédent)

- Le facteur de focalisation est une variable qui indique à quel point l'équipe est concentrée.
- Un faible facteur de focalisation devrait signifier que l'équipe s'attend à avoir de nombreuses perturbations ou s'attend à ce que ses estimations soient trop optimistes.

Estimation par calcul de vélocité – Facteur de focalisation

La meilleure façon de déterminer un facteur de focalisation raisonnable est de regarder le dernier sprint (ou même mieux, la moyenne des quelques derniers sprints).

$$\textbf{Facteur de focalisation} = \frac{\textbf{(Vélocité réelle)}}{\textbf{(jours*hommes disponibles)}}$$

- La vélocité réelle est la somme des estimations initiales de toutes les histories qui ont été terminées durant le dernier sprint.

Estimation par calcul de vélocité – Facteur de focalisation

Exemple de cas:

- Le dernier sprint a fait 18 points d'histoire avec une équipe de 3 personnes constituée de Hamid, Ali et Anisse travaillant durant 3 semaines pour un total de 45 jours-homme.

- Maintenant notre objectif est d'essayer d'avoir une idée sur la vélocité estimée pour le sprint suivant.

- un nouveau membre Jamal a rejoint l'équipe pour ce sprint. Et en tenant compte des congés et du reste nous obtenons 50 jours*homme pour le prochain sprint.

Estimation par calcul de vélocité – Facteur de focalisation

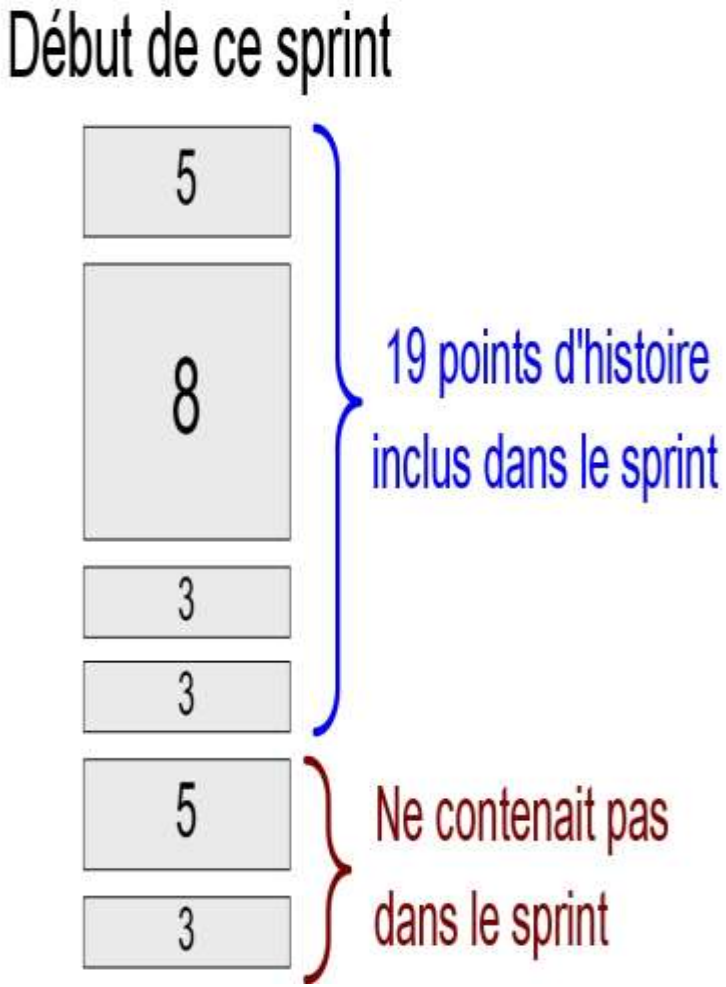
- Facteur de focalisation du dernier sprint= (18 points d'histoires) / (45 jours*hommes)

Facteur de focalisation du dernier sprint= 40%

Vélocité estimée de ce sprint = (50 jours*hommes) x 40% = 20 points d'histoire

- La vélocité estimée pour le prochain sprint est donc de 20 points d'histoire. Cela veut dire que l'équipe devrait rajouter des histoires au sprint jusqu'à cela atteigne environ 20.

Estimation par calcul de vélocité – facteur de focalisation



- Dans ce cas l'équipe peut choisir soit:
  - les 4 histoires en tête pour un total de 19 points d'histoire,
  - ou les 5 histoires en tête pour un total de 24 points.
- Disons que l'équipe choisit 4 histoires, car cela est le plus proche de 20 points.
- **En cas de doute, choisissez moins d'histoires.**
- Comme ces 4 histoires font au total 19 points d'histoires, leur vélocité estimée finale pour ce sprint est 19.

Estimation par calcul de vélocité – facteur de focalisation

- Chaque fois que cela est possible, regardez en arrière de plusieurs sprints et faites la moyenne des chiffres afin d'obtenir des estimations plus sûres.

- **Que faire si l'équipe est complètement nouvelle et que vous n'avez aucune statistique?**
  - Observez le facteur de focalisation d'autres équipes sous des conditions similaires.

- **Que faire si vous n'avez pas s'autre équipe à observer?**
  - Utiliser pour le premier sprint l'estimation par intuition. Après cela il y aura des statistiques, et vous pourrez mesurer de façon continue et améliorer votre facteur de focalisation et la vélocité estimée.

Comment l'équipe décide-t-elle quelles histoires inclure dans le sprint ?

- l'objectif est de décider des histoires à inclure dans le sprint. Les techniques d'estimation comme le facteur de focalisation, la disponibilité des ressources, et la vélocité estimée sont juste des moyens pour atteindre ce but.

- Toutes ces techniques peuvent être combinées à un certain degré.

- Par exemple, nous regardons le facteur de focalisation et la vélocité effective du dernier sprint. Ensuite, Nous regardons le total de nos ressources disponibles pour ce sprint et estimons le facteur de focalisation. Nous discutons de la moindre différence entre ces deux facteurs de concentration et procédons à des ajustements si nécessaire.

- Une fois que nous avons une liste préliminaire des histoires à inclure dans le sprint, on peut faire une vérification à «l'intuition». Le scrum master demande à l'équipe d'ignorer les chiffres pour un moment et de sentir tout simplement si les histoires incluses sont possible pour le sprint. Si cela semble trop gros, il faudra enlever une histoire ou deux. Et vice-versa.

La pratique de l'utilisation des fiches

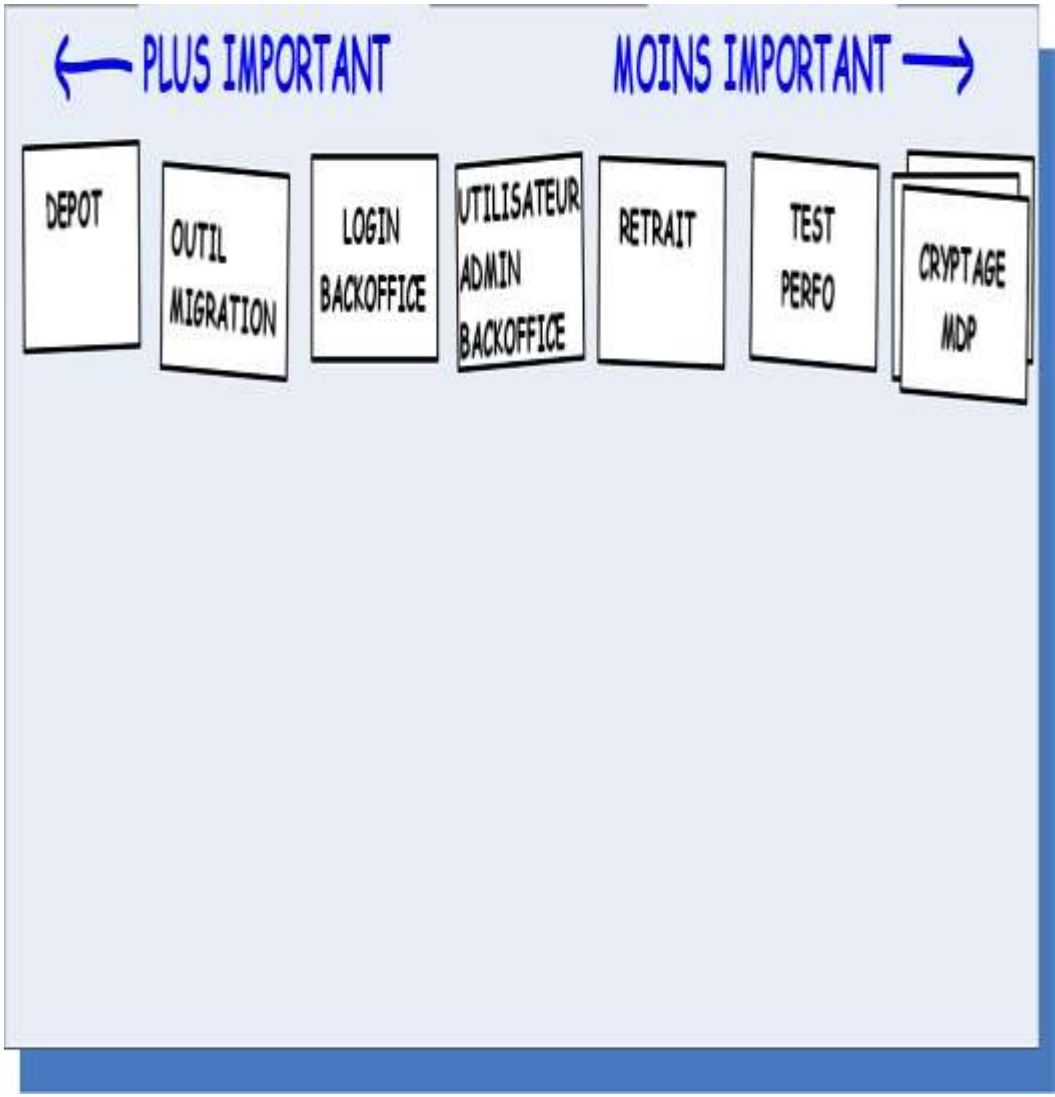
- La plupart des réunions de planification de sprint se déroulent en se basant sur les histoires du backlog de produit. On les estime, on redéfinit leur priorités, on les clarifie, on les découpe, etc..

Alors comment ça se passe en pratique ?

- Une solution qui est efficace consiste à **créer des fiches et de les mettre au mur** (ou sur une grande table).

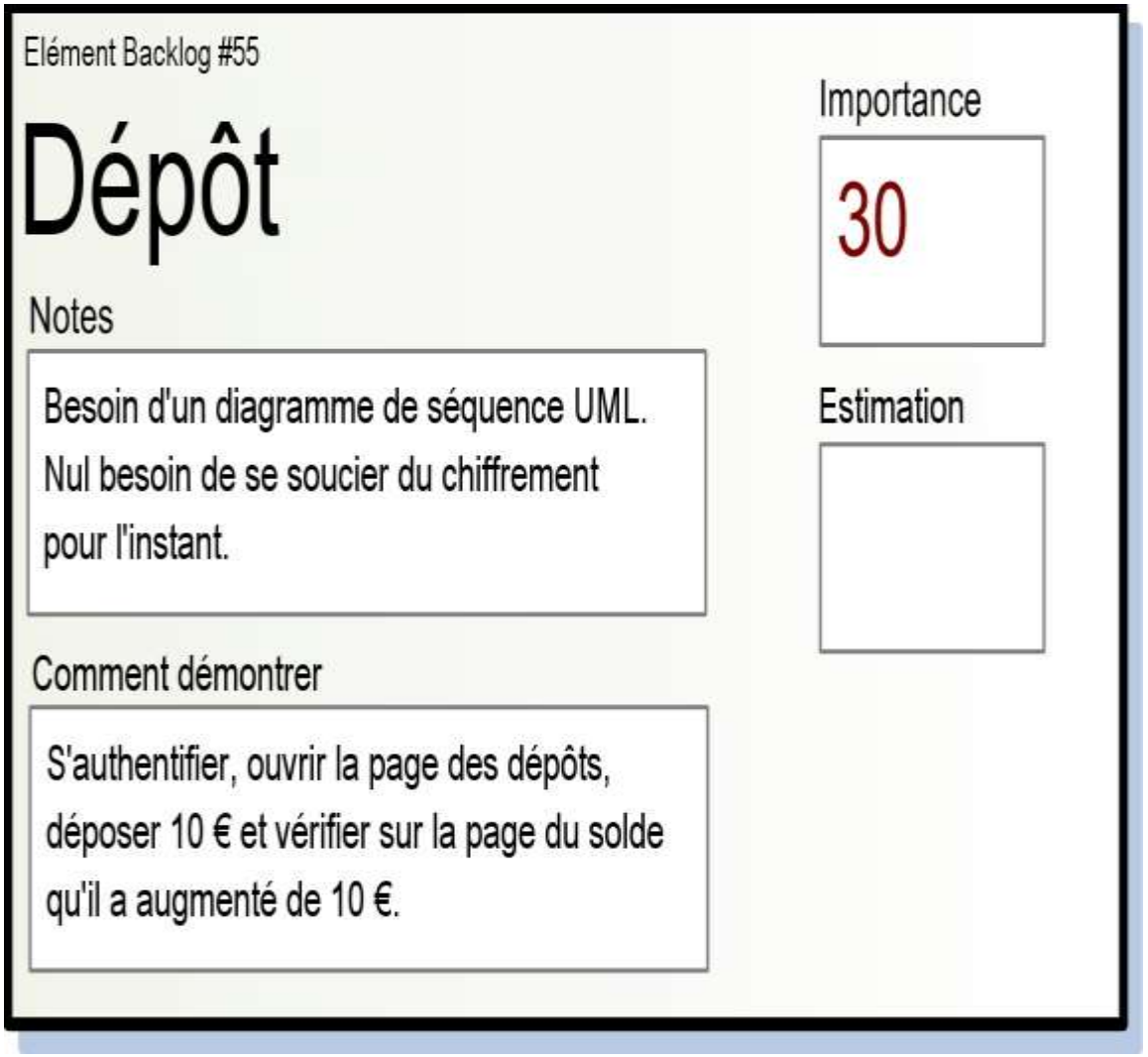
La pratique de l'utilisation des fiches

- Tout le monde se sent plus impliqué personnellement
- Les gens sont debout et marchent autour . Ils restent éveillés, plus en alerte
- Plusieurs histoires peuvent être modifiées en même temps
- Changer la priorité est facile, il faut juste déplacer les fiches
- Après la réunion, les fiches peuvent être transférées directement vers la salle de l'équipe et être utilisées comme panneau mural des activités



La pratique de l'utilisation des fiches

- Les fiches peuvent être:
  - écrites soit à la main
  - ou utiliser des fiches générée, imprimée directement à partir du backlog de produit.



La pratique de l'utilisation des fiches

- Le fait d'afficher le champ «Importance» spécifiée dans le backlog du produit sur la fiche, rend facile leur tri par ordre d'importance (en général, les éléments les plus importants sont mis à gauche, et les moins importants à droite).

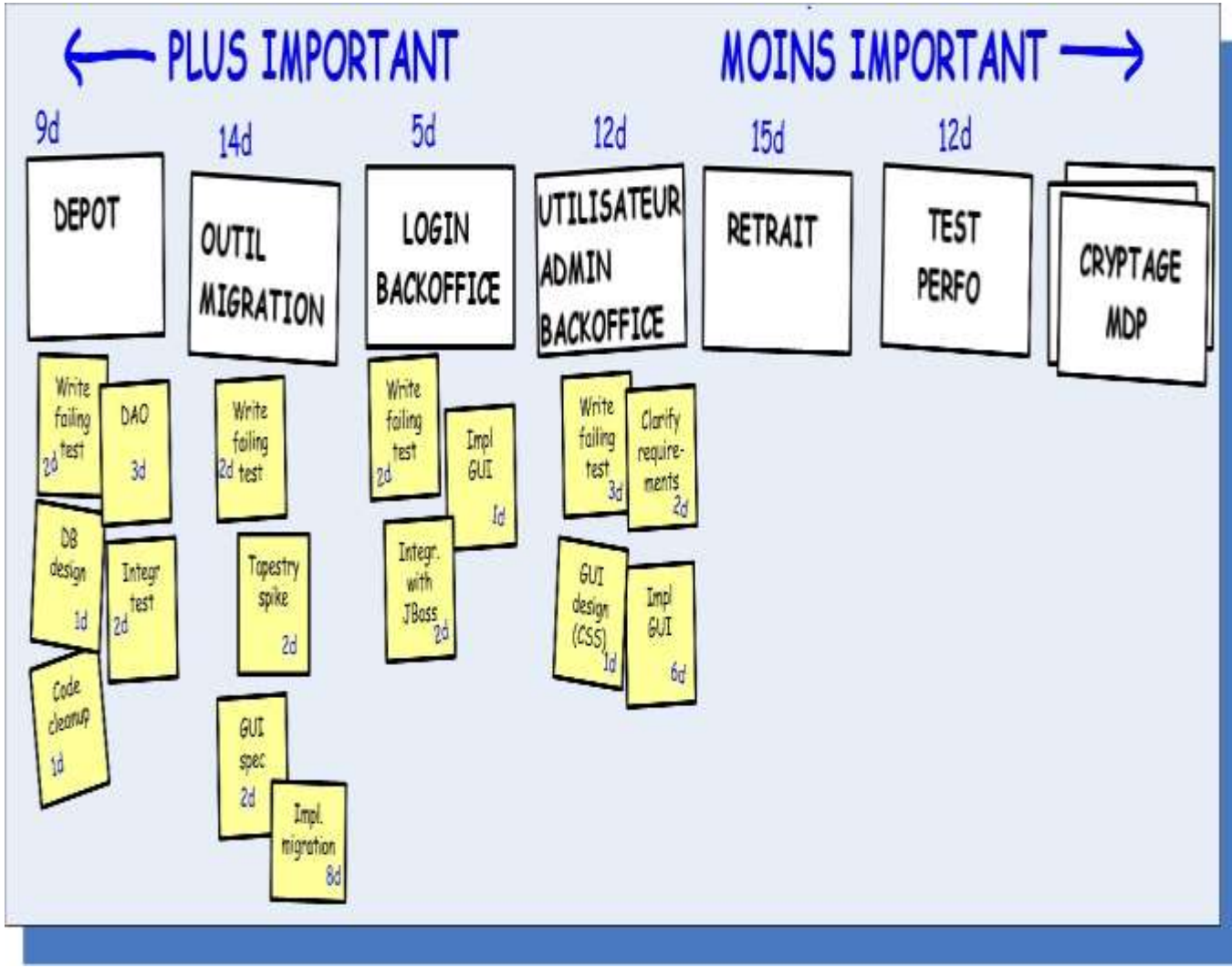
- Cependant, une fois que les fiches sont au mur le niveau d'importance peut être ignoré et utiliser à la place l'ordre dans lequel les fiches sont physiquement placées.

- Si le directeur de produit permute deux éléments, il ne faut pas perdre de temps à mettre à jour le niveau d'importance sur le papier.

- Cependant, après la réunion de planification de sprint, le Scrum master met à jour le backlog de produit sous Excel, en respectant le moindre changement apporté physiquement sur les fiches.

La pratique de l'utilisation des fiches

- Pour rendre plus facile et plus précises les estimations en temps, on préconise de découper les histoires en tâches ou activités.
- Pratiquement, des post-its sont ajoutés sous chaque histoire. Chaque post-it correspond à une tâche de l'historie




Tech Backlog

Définition du « terminé »

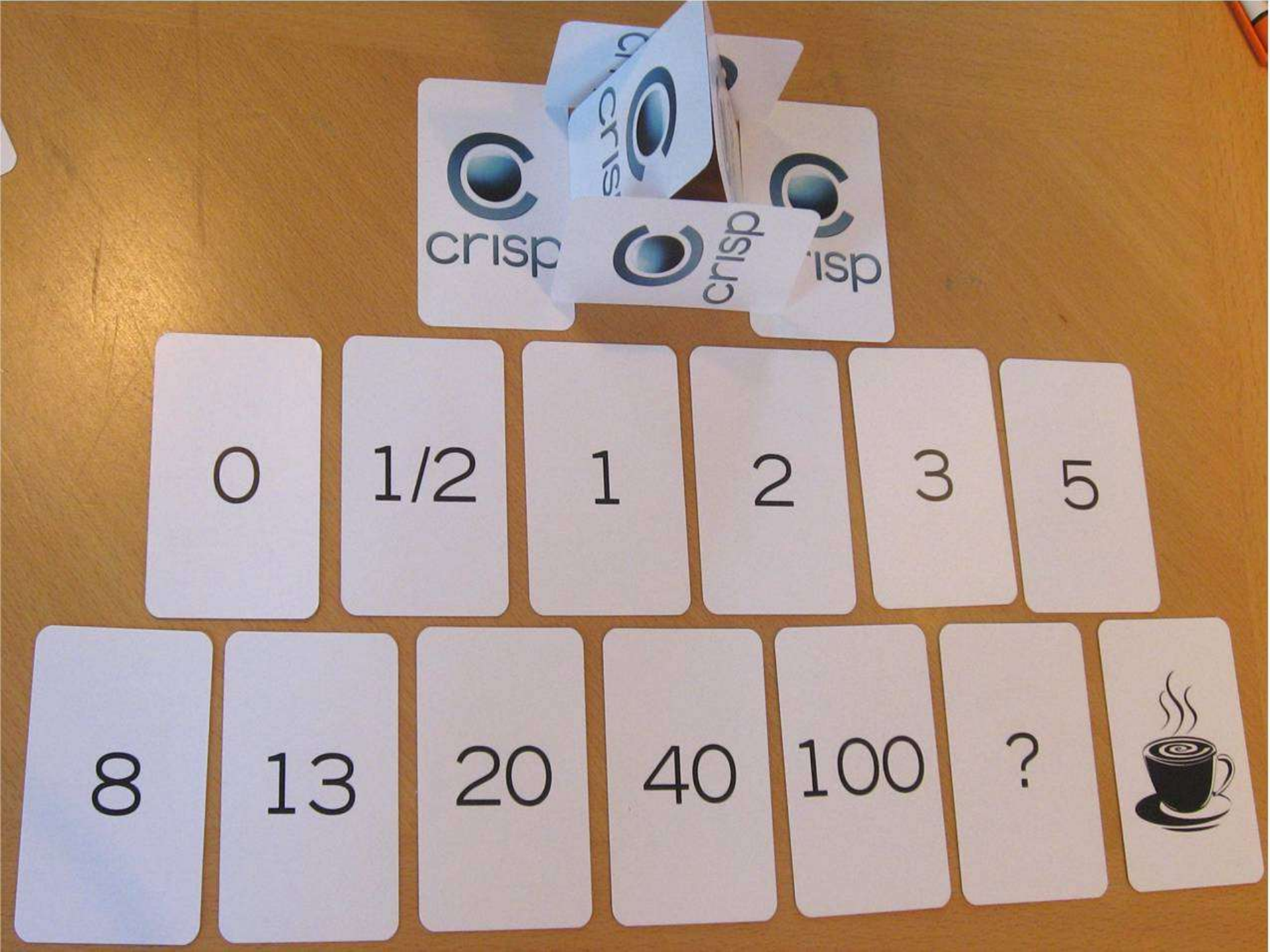
- Il est important que le Directeur de produit et l'équipe s'accorde sur une définition claire de «**terminé**».
  - Une histoire est-elle terminée lorsque tout le code a été archivé ?
  - Ou est-elle terminée seulement quand elle a été déployée sur un environnement de test et vérifiée par une équipe de test d'intégration ?
- Chaque fois que cela est possible, la définition de terminé «prêt à déployer en production» est utilisée mais parfois il faut se rabattre sur la définition de terminé «déployé sur serveur de test et prêt pour le test d'acceptation».
- Si la définition de terminé vous semble confuse, **il faut ajouter un champ «définition de terminé» pour chaque histoire individuelle**. En général, le bon sens est de bon conseil

L’estimation de temps avec le poker planning

- L’estimation est une activité d’équipe – chaque membre de l’équipe est normalement impliqué dans l’estimation de chaque histoire. Pourquoi ?

- Si on demande à l’équipe de fournir une estimation, normalement la personne qui comprend le mieux l’histoire va se lancer en premier. Malheureusement cela influence fortement les estimations de tous les autres.

- pour éviter cela, la technique de poker planning est utilisée


crisp
crisp
crisp
0
1/2
1
2
3
5
8
13
20
40
100
?

L’estimation de temps avec le poker planning

- Chaque membre de l’équipe reçoit un jeu de 13 cartes dont quelques unes sont particulières:
  - 0 = « cette histoire est déjà faite » or « cette histoire c’est pratiquement rien, juste quelques minutes de travail ».
  - ? = « Je n’ai aucune idée. Vraiment aucune. »
  - Tasse de café = « Je suis trop fatigué pour penser. Faisons une courte pause. »

- Chaque fois qu’une histoire doit être estimée,
  - chaque membre de l’équipe sélectionne une carte qui représente son estimation de temps (en points d’histoire) et la place sur la table face cachée.
  - Quand tous les membres de l’équipe ont terminé, toutes les cartes sur la table sont révélées simultanément.

- De cette manière chaque membre de l’équipe est forcé à réfléchir par lui-même au lieu de s’appuyer sur l’estimation de quelqu’un d’autre.

L’estimation de temps avec le poker planning

- S’il y a un gros écart entre deux estimations, l’équipe discute les différences et tente d’élaborer une vision commune du travail impliqué par l’histoire.

- Ils peuvent même faire une sorte de décomposition en tâches. Après, l’équipe estime de nouveau. Cette boucle est répétée jusqu’à ce que les estimations convergent, c’est-à-dire que toutes les estimations soient *approximativement les mêmes pour cette histoire.*

- Il est important de rappeler aux membres de l’équipe qu’ils sont là pour estimer la quantité de travail total pour cette histoire

La clarification des histoires

- Le pire c'est quand un membre de l'équipe démontre fièrement une nouvelle fonctionnalité à la démonstration du sprint, et que le directeur de produit fronce les sourcils et dit « eh bien c'est sympa, mais ce n'est *pas ce que j'ai demandé ! »*
  - Comment peut-on s'assurez que la compréhension d'une histoire par le directeur de produit corresponde à la compréhension de l'équipe ?
  - Ou que chaque membre de l'équipe a la même compréhension de chaque histoire ?
- Eh bien on ne peut pas. Mais il y a quelques techniques simples pour identifier les incompréhensions les plus flagrantes. La plus simple des techniques est de s'assurer que tous les champs sont bien remplis pour chaque histoire.

La clarification des histoires

- Le « Comment démontrer » d'une histoire peut (et devrait) être *très court.* Sinon le planning du sprint ne sera pas fini à temps.

- Essentiellement c'est une description de haut niveau en français de comment exécuter manuellement le scénario de test le plus typique. « Faire ceci, puis cela, et alors vérifier ceci ».

- En général, cette description simple fait souvent apparaître des incompréhensions importantes sur la portée d'une histoire.

La décomposition des histoires en plus petites histoires

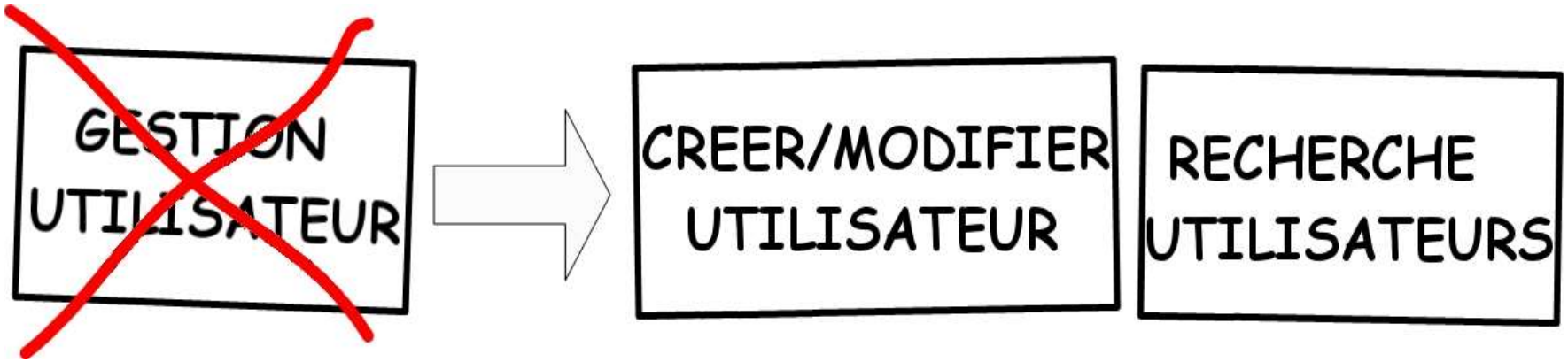
- Les histoires ne devraient pas être trop petites ou trop grosses (en termes d'estimations).
  - Si on a un lot d'histoires à 0,5 points , on va probablement être une victime du micro-management.
  - Une histoire de 40 points implique un risque élevé de finir *partiellement* terminée, **ce qui ne produit aucune valeur** pour l'entreprise et augmente simplement le travail administratif.
  - Si la vélocité estimée est de 70 et que les deux histoires prioritaires sont à 40 points chacune, le planning devient quelque peu difficile.

- Il faut choisir entre **le sous-engagement** (c'est-à-dire prendre juste un élément) et **le surengagement (**c'est-à-dire prendre les deux éléments).

- On remarque qu'il est presque toujours possible de couper une grosse histoire en plusieurs histoires plus petites. Cependant, **il faut faire attention que les petites histoires continuent à représenter des livrables avec une valeur pour le client.**

La décomposition des histoires en tâches

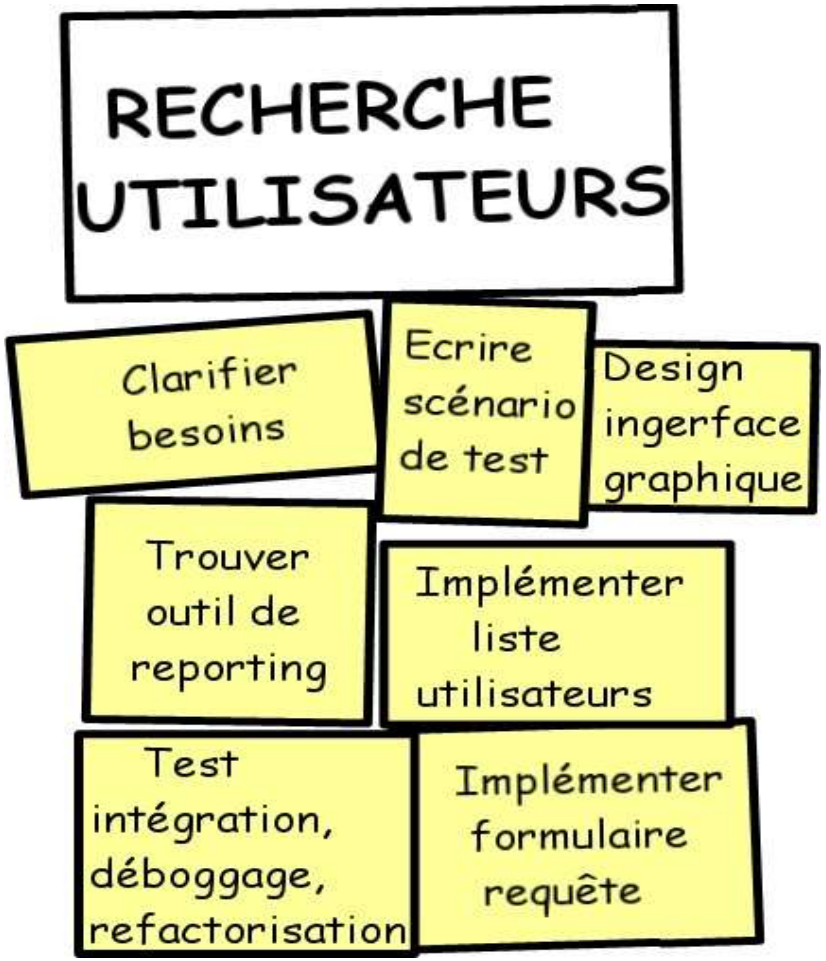
- quelle est la différence entre « tâches » et « histoire » ?
- La distinction est assez simple. Les histoires sont des choses livrables qui intéressent le directeur de produit.
- Les tâches sont des choses non livrables, ou des choses auxquelles le directeur de produit ne s'intéresse pas.
- **Exemple de décomposition d'une histoire en histoires plus petites** :



La décomposition des histoires en tâches

- Il faut essayer d'avoir une limite de temps pour la réunion de planning de sprint suffisamment importante pour faire tenir tout cela.

- Si vous pratiquez le DDT (développement dirigé par les tests) ce qui en pratique signifie que la première tâche pour presque toutes les histoires est « écrire un test qui échoue » et la dernière tâche est « remaniement (*refactor) » (= améliorer la lisibilité du code et supprimer les* duplications).



Exemple de décomposition d'une histoire en tâches

Le choix du lieu et l'heure pour la mêlée quotidienne

- Un des objectifs de la réunion de planning du sprint est « **un lieu et une heure bien définis pour la mêlée quotidienne »**. Sans cet élément votre sprint va faire un mauvais départ.

- La première mêlée quotidienne est essentiellement le point où tout le monde décide où commencer à travailler.
  - **Inconvénient des mêlées de l'après-midi : quand vous venez travailler le** matin, vous devez essayer de vous rappeler ce que vous avez dit hier à vos collègues au sujet de ce que vous feriez aujourd'hui.

  - **Inconvénient des mêlées du matin : quand vous venez travailler le** matin, vous devez essayer de vous rappeler ce que vous avez fait hier pour pouvoir le rapporter.

- Le premier inconvénient est le pire, puisque la chose la plus importante est ce que vous *allez faire, pas ce que vous avez fait.*

- par défaut, choisir le moment le plus tôt où personne dans l'équipe ne se plaigne. Habituellement 9:00, 9:30, ou 10:00. La chose la plus importante est une heure que tout le monde dans l'équipe accepte de bon cœur.

Où tracer la limite

- OK, le temps commence à manquer. Parmi toutes les choses que nous voulons faire durant le planning de sprint, qu'est-ce nous laissons de côté si le temps manque ?

- Pour réaliser cela, on utilise la liste de priorités suivante :

  - **Priorité 1 : Un but de sprint et une date de démonstration. C'est vraiment** le minimum dont vous avez besoin pour démarrer un sprint. L'équipe a un but, une date de fin et ils peuvent travailler directement à partir du backlog du produit. Ca craint, bien sûr, et vous devriez considérer sérieusement de prévoir une nouvelle réunion de planning demain, mais si vous devez vraiment démarrer le sprint alors cela ira probablement.

Où tracer la limite

- **Priorité 2 :** Liste des histoires que l'équipe a acceptées pour ce sprint**.**
- **Priorité 3 :** Estimations remplies pour chaque histoire du sprint.
- **Priorité 4 : «** Comment démontrer » rempli pour chaque histoire du sprint.
- **Priorité 5 : Les calculs de vélocité et de ressources, pour vérifier la** vraisemblance de votre plan de sprint. Cela inclue la liste des membres de l'équipe et leurs engagements (sinon vous ne pouvez pas calculer la vélocité).
- **Priorité 6 : une heure et un lieu bien définis pour la mêlée quotidienne.** Cela prend juste un moment pour décider, mais si vous manquez de temps le Scrum master peut simplement le décider après la réunion et envoyer un mail à tout le monde.
- **Priorité 7 : Les histoires découpées en tâches. Ce découpage peut sinon** être fait quotidiennement en conjonction avec les mêlées quotidiennes, mais cela va légèrement perturber le déroulement du sprint.

Les histoires techniques

- Les histoires techniques **font référence aux choses qui doivent être réalisées mais qui ne sont pas livrables, et qui n'ont pas de valeur directe pour le directeur de produit.**

Exemple

- **Installer un serveur de construction (*build) en continu***
  - Un des principes de l'Intégration Continue est de vérifier, à chaque modification du code-source d'une application, que le résultat de cette modification ne produit pas de régression sur l'application en question

- **Rédiger une vue d'ensemble de la conception du système**
  - Un document qui montre une vue d'ensemble pour garder tout le monde synchronisé sur la conception et éviter d'écrire du code incohérent

Les histoires techniques

- **Remanier (*refactor) la couche d'accès aux données (DAO)***
  - Nettoyer le code va faire économiser du temps à tout le monde et va améliorer la robustesse du système.

- **Mettre à jour le système de suivi de bugs**
  - Le principe de cet outil consiste à enregistrer la déclaration d'un bogue informatique, puis pour les techniciens de maintenance informatique concernés, à mettre à jour l'avancement de sa résolution, jusqu'à sa clôture. Le déclarant de l'anomalie peut s'informer à tout moment de l'avancement du traitement de son problème.

- Alors comment peut-on gérer ce type d'histoires particulières? Qui va fixer leurs priorités? Est-ce que le directeur de produit devrait-t-il être impliqué dans ces choses ?

Les histoires techniques

Une façon de faire consiste à:

1. Essayer d'éviter les histoires techniques. Essayer de transformer une histoire technique en histoire normale avec une valeur métier mesurable. De cette manière le directeur de produit a de meilleures chances de faire des compromis corrects.

2. S'il n'est pas possible de transformer une histoire technique en histoire normale, voir si le travail peut être fait en tant que tâche dans une histoire normale.

3. Si les deux méthodes ci-dessus échouent, définir une histoire technique, et maintenir une liste séparée de telles histoires.
   - Permettre au directeur de produit de la voir mais pas de l'éditer.
   - Utiliser les paramètres « facteur de focalisation » et « vélocité estimée » pour négocier avec le directeur de produit et obtenir un peu de temps dans le sprint pour implémenter les histoires techniques.

- Dans tous les cas, il faut essayer de trouver un consensus

Conclusion sur la réunion du planning du sprint

- la réunion du planning de sprint est la chose la plus importante qui est faite dans Scrum.

- Il faut faire beaucoup d'efforts pour la réaliser correctement, et le reste sera tellement plus facile

Références

1. ***Scrum et XP depuis les Tranchées*** *par* Henrik Kniberg

2. **Guide de démarrage Scrum - Florent Lothon, mis à jour en juin 2013**
   agiliste.fr /fiches/guide-demarrage-scrum/

3. **Le Guide Scrum - Le guide définitif de Scrum** :
   http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-FR.pdf#zoom=100

4. **Planning Pocker**
   http://fr.wikipedia.org/wiki/Planning_poker

5. **Institut Agile**
   http://referentiel.institut-agile.fr/velocity.html

6. http://blog.pascal-martin.fr/post/integration-continue-jenkins-projet-php

7. Mantis bug Tracker
   http://fr.wikipedia.org/wiki/Mantis_Bug_Tracker