

La méthode de Scrum

Partie 2

IAM

Département Informatique- ESTS

Introduction

- La réunion de planification du sprint terminée, le Scrum master crée le backlog de sprint.
- Cela doit être fait *après la* réunion de planification du sprint, mais *avant la première mêlée* quotidienne.

Le tableau des tâches

- Un tableau de tâches sert à montrer l'avancement des travaux pendant le sprint, c'est une représentation physique du backlog de sprint. Il permet de visualiser l'état du projet
- Les tableaux de tâches sont disposés sur un mur, avec en lignes les stories et leurs tâches associées et en colonnes les états des tâches.
- Pour chaque story sélectionnée,
 - l'équipe identifie les tâches correspondantes. Sur le tableau, les histoires et les tâches sont placées avec des post-it de couleurs différentes.
 - L'état des tâches est reconnu selon la place de la tâche dans des zones représentant chaque état :
 - Les tâches **à faire**
 - Les tâches **en cours**
 - les tâches **terminées ou finies**

Le tableau des tâches

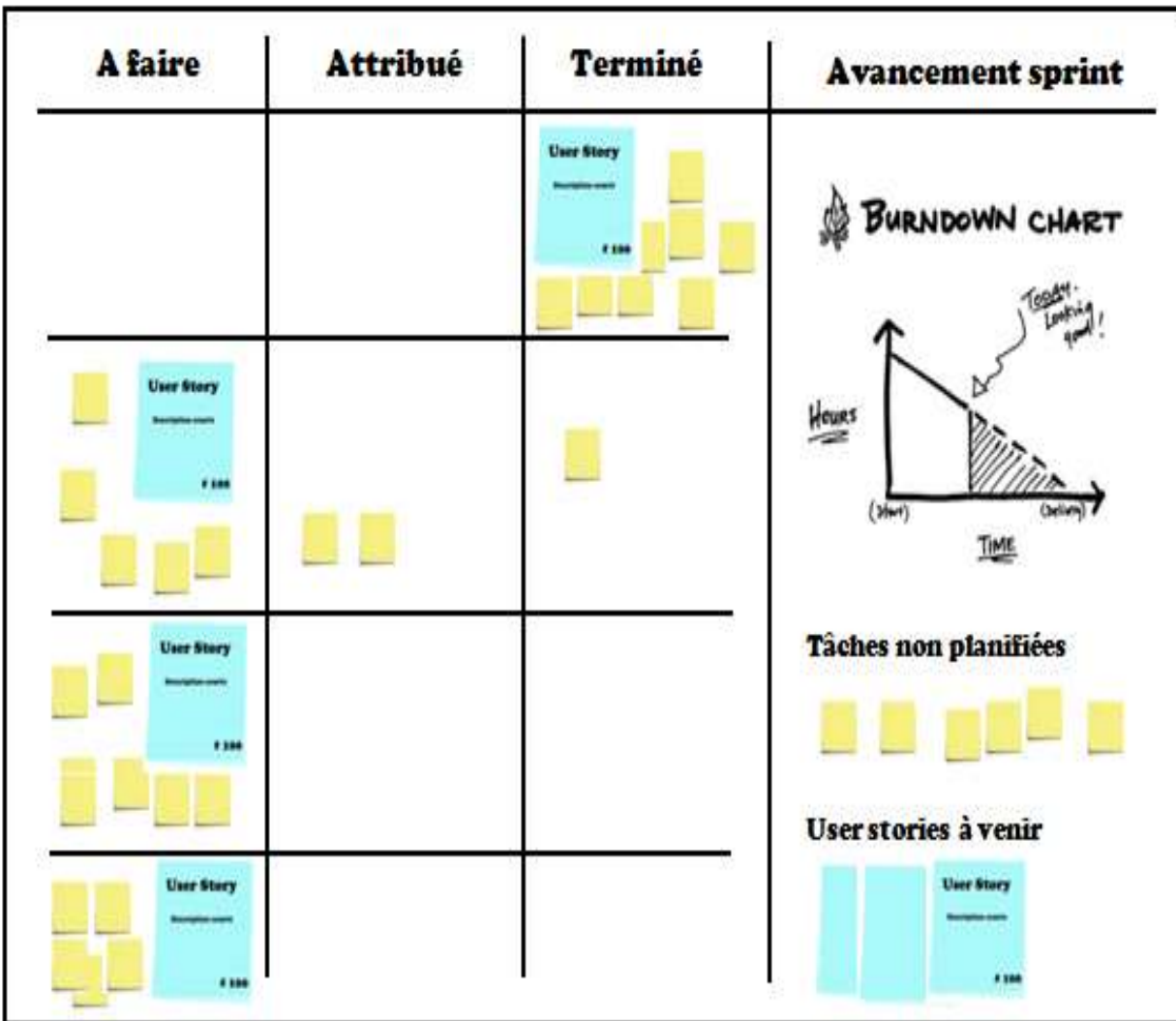
- Au niveau de l'état de l'art, il y a plusieurs exemples de tableau des tâches. Cependant, ils convergent tous vers la même finalité (des tâches à réaliser, des tâches en cours de réalisation et des tâches finies..).
- La différence réside, plus ou moins, dans l'intitulé des états des tâches. Ceci est dû au fait, que chaque équipe, a ses propres pratiques pour réaliser les tâches des histoires du sprint.

Exemple de tableau des tâches

Sprint 3 : début le 15/3, fin le 29/3				
But : lorem ipsum				
	À faire		En cours	Fini
	tâche	tâche		
story	tâche	tâche	tâche	
	tâche	tâche	tâche	
story	tâche	tâche	tâche	
	tâche			
story	tâche	tâche	tâche	
	tâche	tâche		

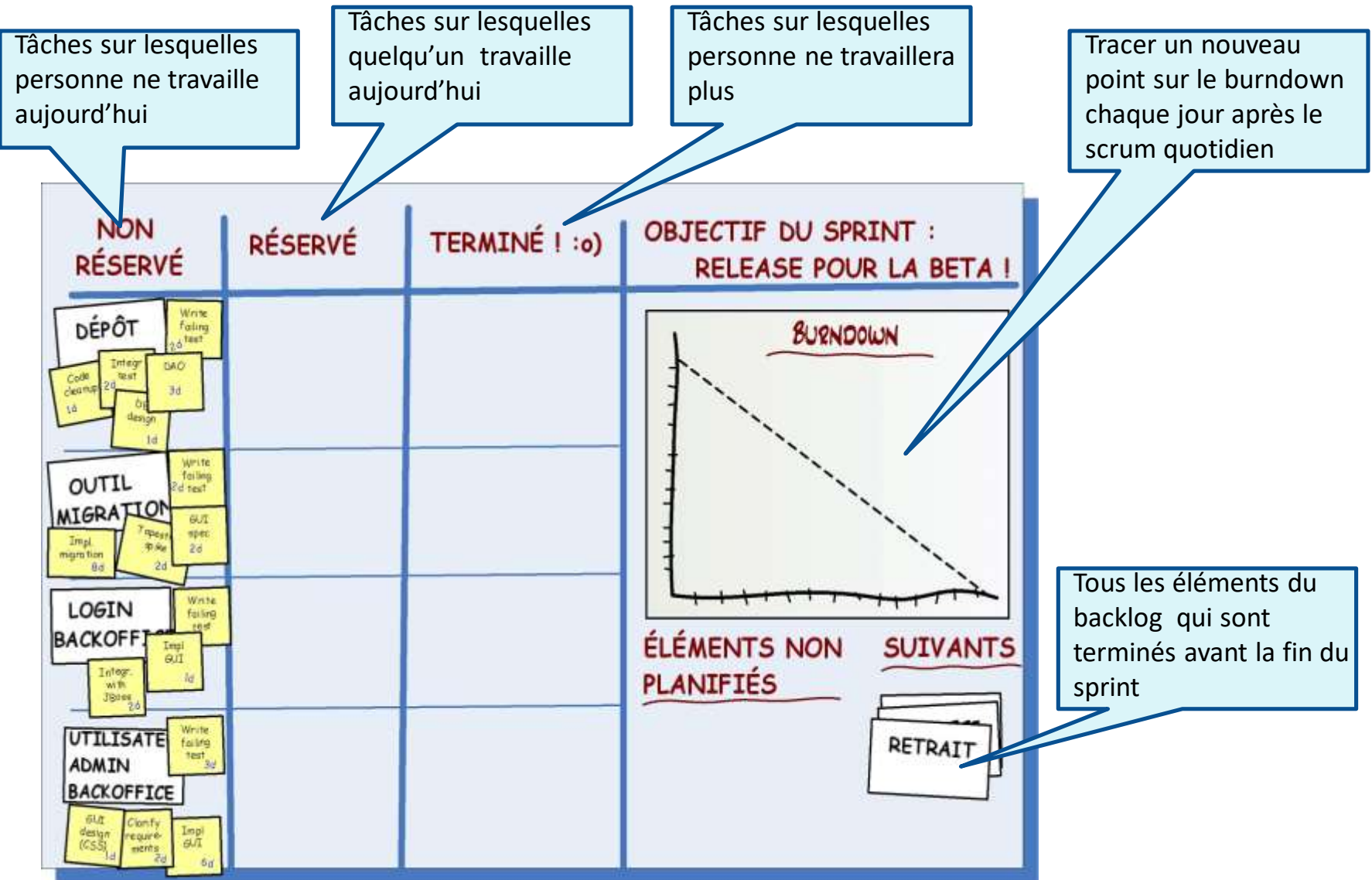
Stories en cours	Tâches		
	à faire	à finir	fini
StoryA		tâche1	tâche2
StoryB	tâche3	tâche4	tâche7
	tâche5	tâche6	tâche8
StoryC	tâche9		

Exemple de tableau des tâches

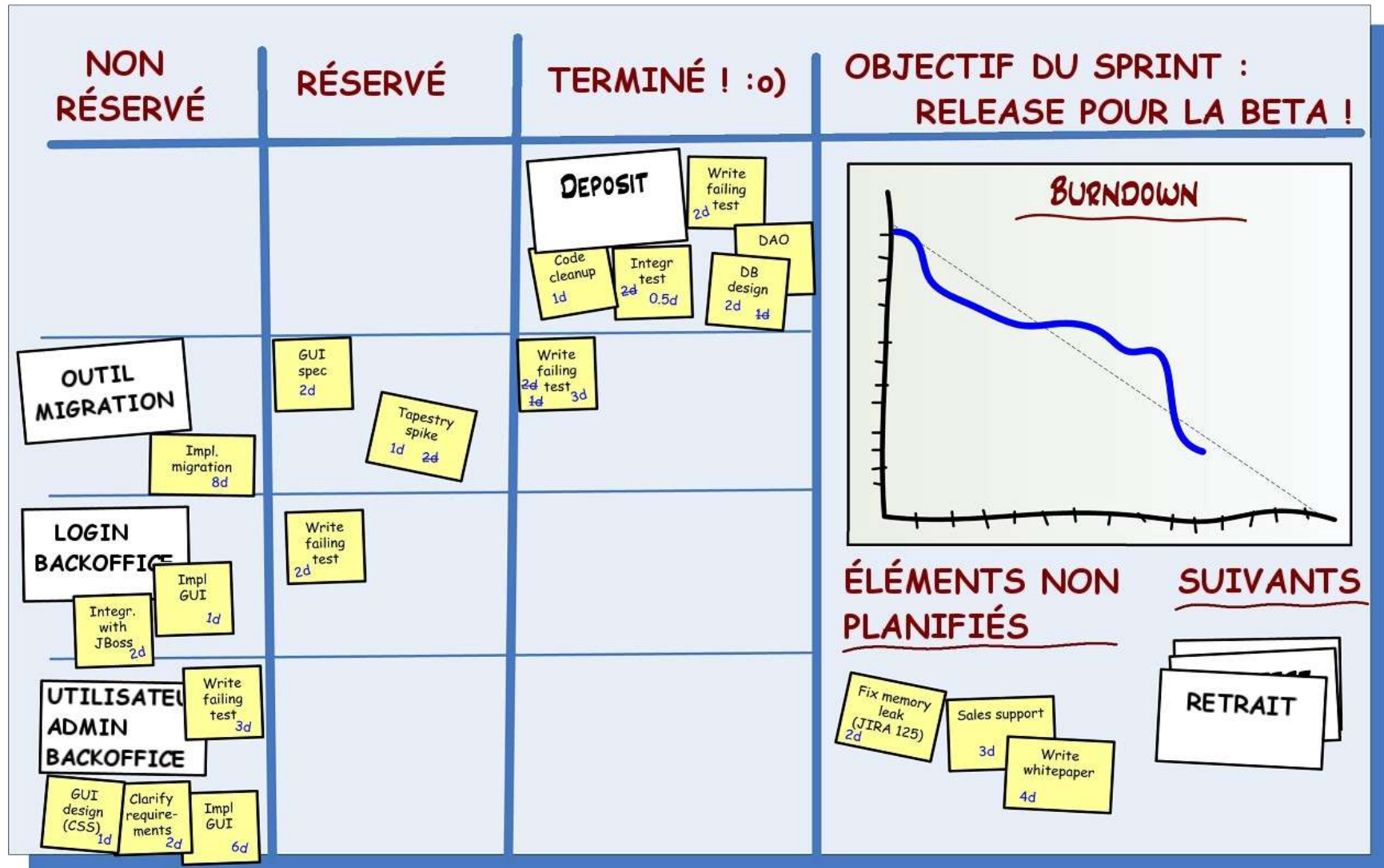


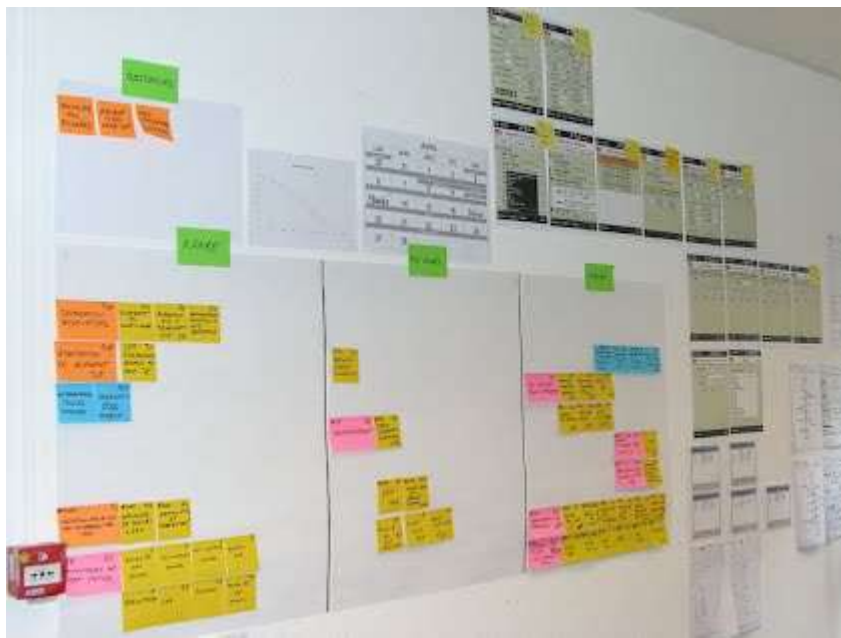
- Chaque post-it jaune représente une tâche. Lorsqu'un membre de l'équipe choisi de développer cette tâche, le post-it passe dans la colonne Attribué, puis dans Terminé lorsque la tâche est effectuée.

Exemple de tableau des tâches au début du sprint



Exemple de tableau des tâches après quelques jours du sprint





Sur le tableau de tâche, on trouve en général:

- Les histoires représentées par les gros post-it larges ,
- les sous tâches des User Stories par des post-it plus petits
- des tâches imprévues non planifiées.

Au dessus du tableau ou sur les côtés peuvent figurer :

- les obstacles courants,
- Un graphique d'avancement appelé **burdown chart**
- Un calendrier géant avec les dates clés et les congés de chacun.
- Sur la droite on trouve des maquettes d'IHM, des documents métier, etc.



Le burndown chart

- le burndown chart est un outil à très forte valeur ajoutée pour l'équipe, il est affiché sur le tableau des tâches.
- Un burndown chart de sprint montre au jour le jour le reste à faire sur les tâches du sprint.
- le burndown chart est une représentation graphique du reste à faire dans une période, actualisé aussi souvent que possible et permettant de montrer une tendance de l'avancement.

Burndown chart

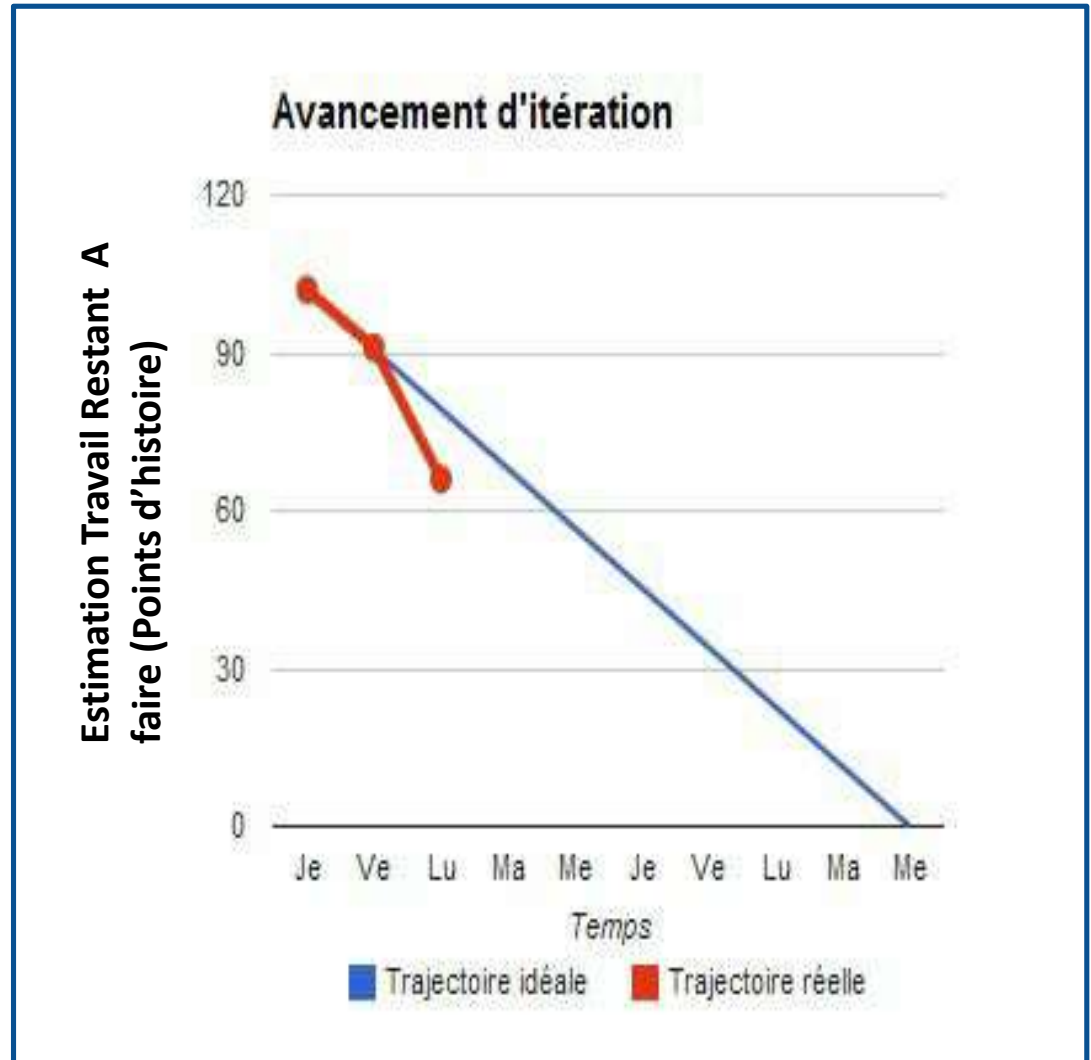
Sprint Backlog

Exigence	Sous Tâche	Reste A Faire									
Exigence X - #102	IHM	8	8	6							
	Mise à jour documentation	2	2	0							
	Services métier	10	4	0							
	Tests automatisés Perf	8	8	8							
Exigence YY - #103	IHM	8	8	8							
	Mise à jour documentation	2	2	2							
	Services métier	10	10	5							
	Tests automatisés Perf	8	8	8							
Exigence XY - #33	IHM	8	8	8							
	Mise à jour documentation	2	2	2							
	Services métier	10	10	7							
	Tests automatisés Perf	8	8	8							
Exigence Y - #365	IHM	8	8	4							
	Services métiers	10	5	0							
Jours d'itération		Je	Ve	Lu	Ma	Me	Je	Ve	Lu	Ma	Me
Trajectoire idéale		102	91	79	68	57	45	34	23	11	0
Trajectoire réelle		102	91	66							

exemple du Backlog sprint

Le burndown chart

- Le graphique est simple, il s'agit du tracé de la charge de travail Restante (points d'histoire) en fonction du temps (en jours).
- Les weekends sont exclus de l'axe des abscisses puisque le travail est rarement fait le weekend
- Pour tracer ce graphique, il suffit de mettre à jour (lors de chaque mêlée quotidienne par exemple) le **sprint backlog**



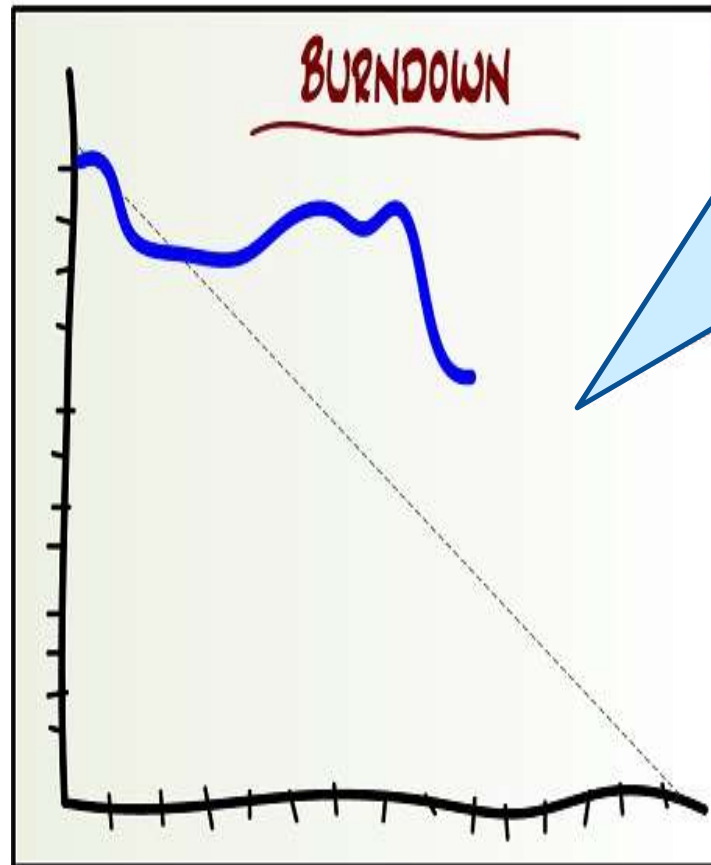
Le burndown chart

Je vous invite à télécharger un burndown chart à l'url suivante:

<http://www.burndownchart.fr/>

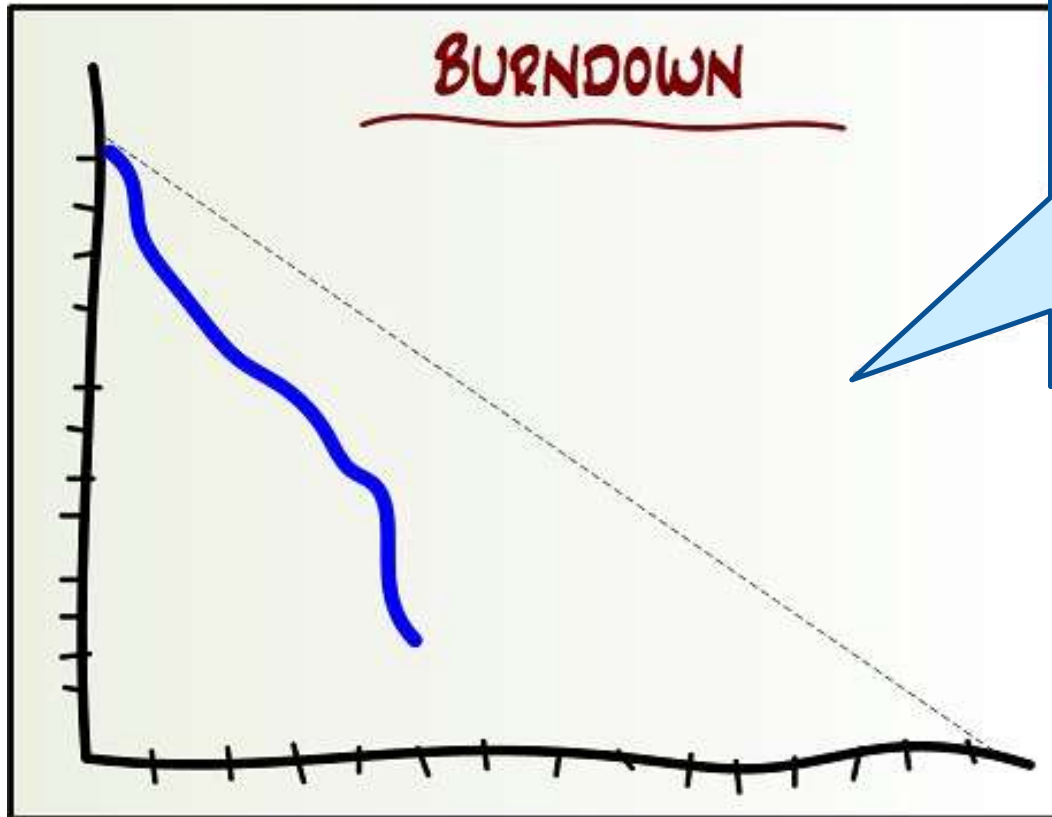
Les signaux d'avertissement du tableau des tâches

- A tout moment, le tableau des tâches donne à tout le monde une indication sur le bon déroulement du sprint.
- Le Scrum master a la responsabilité de s'assurer que l'équipe réagit aux signaux d'avertissement comme :



Attention! La trajectoire réelle est au dessus de la trajectoire idéale. Il faut supprimer quelques éléments de backlog du sprint

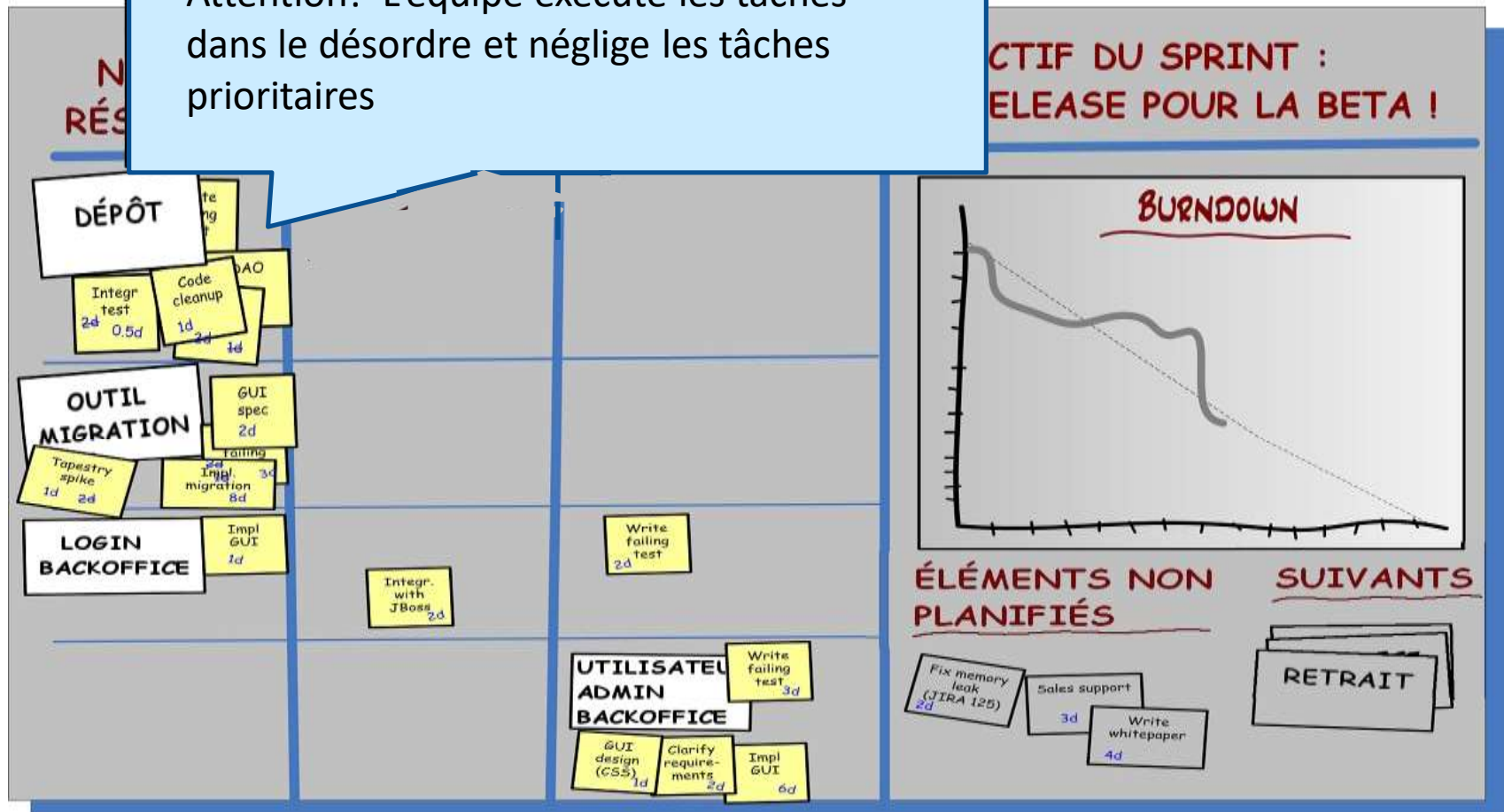
Les signaux d'avertissement du tableau des tâches



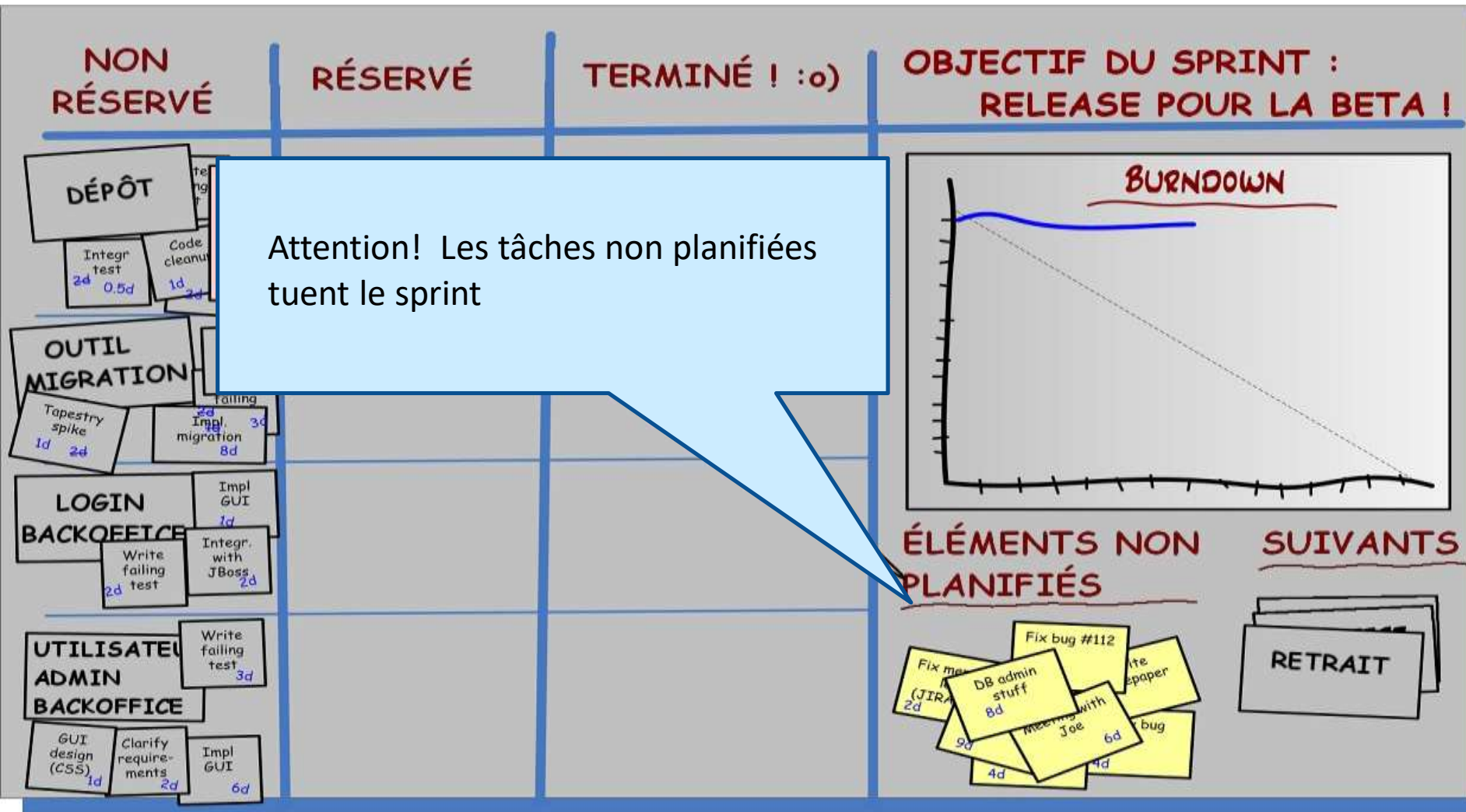
Attention! La trajectoire réelle est au dessous de la trajectoire idéale. Il faut ajouter des éléments au backlog du sprint

Les signaux d'avertissement du tableau des tâches

Attention! L'équipe exécute les tâches dans le désordre et néglige les tâches prioritaires



Les signaux d'avertissement du tableau des tâches



Estimations en jours vs.heures

- Dans la plupart des livres et articles sur Scrum , les tâches sont estimées en heures, pas en jours.
- Une estimation peut être faite selon la formule suivante :

1 jour-homme effectif = 6 heures-homme effectives.

- Les estimations en heures-homme sont trop fines, cela engendre trop de petites tâches de 1-2 heures et entraîne par conséquent le micro-management.
- En général, tout le monde pense en jours-homme , et on multiplie juste par 6 avant d'écrire en heures-homme. « Hmmmm, cette tâche devrait prendre à peu près un jour. Oh je dois écrire en heures, j'écrirai 6 heures alors ».
- Deux unités différentes sèment la confusion. « Cette estimation était-elle en heures-homme ou en jours-homme ? ».

Récapitulatif

- Au cours du Sprint, l'équipe se concentre sur l'accomplissement des tâches du **Sprint Backlog**.
- **En cas de retard (indiqué par le Burndown Chart), des exigences ou tâches seront retirées du Sprint Backlog** en cours de route en essayant de préserver l'objectif du sprint (pour cela, il est conseillé d'ordonnancer les exigences au sein du sprint). Et inversement, si l'équipe avance plus vite que prévu, des exigences ou tâches y seront ajoutées. En accord avec le **Product Owner dans les deux cas**.
- **Le but est de** développer les fonctionnalités de bout en bout (de la conception aux tests) au fur et à mesure au cours du sprint. Autrement dit d'éviter **de se retrouver avec une** surcharge d'effort de test en fin de sprint.
- Les développeurs doivent donc éviter de trop paralléliser les exigences et encore moins les tâches de développement. Pour cela, le **pair programming peut se** révéler utile ainsi que la définition d'une limite maximum d'éléments au sein d'une colonne du tableau des tâches

Récapitulatif

- Le tableau des tâches physique rempli de post-it est pratiquement indispensable. Il permet d'avoir une vision claire du travail à accomplir, en cours et terminé.
- Il peut également s'avérer précieux lors des mêlées quotidiennes, surtout si on calcule à la main le « reste à faire » du Sprint afin de tracer le graphique d'avancement.
- Le tableau facilite également l'affectation des tâches par l'équipe en ayant une vision d'ensemble du sprint en un coup d'œil. Ce sont les développeurs qui « tirent » les tâches et non pas le Scrum Master qui les affecte.

Les mêlées quotidiennes

Les mêlées quotidiennes

- L'équipe se **réunit une fois par jour, à heure fixe**, pour mettre en commun les apports de chacun au produit, et signaler les obstacles rencontrés.
- Les mêlées quotidiennes sont structurées autour des trois questions suivantes:
 - **Qu'as-tu terminé depuis la précédente mêlée?**
 - **Que penses-tu pouvoir terminer d'ici la prochaine mêlée?**
 - **Quels obstacles rencontres-tu en ce moment?**
- Le Scrum Master est ainsi immédiatement au courant des obstacles rencontrés, il doit impérativement les prioriser, les suivre et bien sûr s'efforcer de les lever au plus tôt afin de garder l'équipe pleinement concentrée et productive.
- Cette **réunion** a généralement **une durée de 15 minutes**. Tout sujet risquant de déborder est noté et abordé en dehors de la réunion en plus petit comité.

Les mêlées quotidiennes

- Il est préférable que les mêlées quotidiennes se tiennent dans le bureau de l'équipe en face du tableau des tâches.
- Il est préconisé de faire les mêlées debout, afin de diminuer le risque de dépasser les 15 minutes.



La mise à jour du tableau des tâches

- le tableau des tâches est mis à jour pendant la mêlée quotidienne.
 - Pendant que chaque personne décrit ce qu'elle a fait hier et ce qu'elle va faire aujourd'hui, elle déplace les post-its sur le tableau des tâches.
 - Quand elle décrit une tâche non planifiée, elle colle un post-it pour ça.
 - Quand la personne met à jour son estimation de temps, elle écrit la nouvelle estimation sur le post-it et raye l'ancienne valeur.
- Parfois le Scrum Master s'occupe des post-its pendant que les personnes parlent.

Ecrire le test d'échec 2j	Ecrire le test d'échec 2j 3j	Ecrire le test d'échec 2j 1j 3j
------------------------------------	--	---

- Il faut essayer d'impliquer toute l'équipe pour garder le backlog de sprint à jour.
- Immédiatement après la mêlée quotidienne, quelqu'un additionne toutes les estimations de temps (en ignorant celles de la colonne «terminé») et marque un nouveau point sur la courbe du reste à faire du sprint

Quels bénéfices en attendre

- la mêlée quotidienne favorise la circulation d'informations importantes; elle apporte une occasion explicite permettant à l'ensemble de l'équipe de se synchroniser
- le partage d'informations à l'occasion d'une réunion courte et énergique contribue également à la cohésion de l'équipe

La revue ou démo du sprint

La démo de sprint: la revue de sprint

- La revue de sprint permet un feedback concret sur le produit.
 - Le but de la démo est de montrer ce qui a été réalisé pendant le sprint afin d'en tirer les conséquences pour la suite du projet.

Participants

- L'équipe Scrum étendue, avec le ScrumMaster et le ProductOwner, est présente à la réunion. Toutes les personnes qui sont partie-prenantes du projet y sont invitées et leur présence encouragée.

Produit en entrée

- le produit partiel potentiellement livrable. C'est la version opérationnelle obtenue comme résultat du sprint

Les étapes de la revue de sprint

Préparer la démonstration

- La préparation de la réunion consiste à imaginer un scénario pour présenter les différentes histoires d'utilisateur qui seront présentées et à vérifier que le matériel utilisé pour la démonstration fonctionne bien. La revue de sprint ne nécessite pas une présentation formelle.

Rappeler les objectifs du sprint

- Le ScrumMaster rappelle le but du sprint défini lors de la réunion de planification. Il donne la liste des histoires d'utilisateur sélectionnées.

Les étapes de la revue de sprint

Effectuer la démonstration

- L'équipe présente le produit partiel résultat de ses travaux en faisant une démonstration des éléments du backlog de produit (histoires d'utilisateur) réalisés. Seules les histoires d'utilisateur complètement finies sont présentées. Cela permet d'avoir une mesure objective de l'avancement. La démonstration est faite par un ou plusieurs membres de l'équipe, en principe les plus impliqués dans les choses montrées. Si possible, les personnes présentes à la réunion sont aussi invitées à manipuler le produit.

Evaluer les résultats du sprint

- Le Product Owner et les intervenants présents posent des questions à l'équipe et donnent leur impression sur le produit montré. Leur feedback se concrétise en propositions et demandes de changement. Le backlog de produit est enrichi avec les éventuels bugs découverts et les demandes d'évolution suggérées.

Les étapes de la revue de sprint

Calculer la vélocité réelle

- La liste des éléments du backlog considérés comme finis est établie. **La vélocité du sprint** est obtenue en faisant la somme de tous les points attribués à ces éléments. Elle est comparée à celle des sprints précédents, pour obtenir une vélocité moyenne.

Ajuster le plan de la release

- Les conditions ont changé depuis la dernière planification de la release : des éléments ont pu être ajoutés ou supprimés, les estimations modifiées, l'ordre dans lesquels les éléments sont classés (la priorité) a pu être modifié. De plus la vélocité moyenne a pu évoluer.
Le Planning de la release est ajusté en tenant compte de ces nouveaux paramètres. Le burndown chart de la release est produit. Avec cette mesure objective de l'avancement, il est possible de prendre une décision sur la date de la mise en production de la release.

La revue de sprint

Produit en sortie

- Le backlog de produit actualisé, avec le plan de release mis à jour et son burndown chart

Considérations de timing

- La préparation de la démonstration ne devrait pas dépasser une heure.
- La revue elle-même est limitée à 2 heures et dure en moyenne une heure.
- La revue a lieu à la fin du sprint et elle est suivie de la **rétrospective**.

Les erreurs à éviter dans la revue de sprint

Avant la réunion

- l'équipe modifie le code au dernier moment et présente une version mal testée

Pendant la démonstration

- des histoires d'utilisateur inachevées sont montrées
- la démonstration est trop rapide et ne permet pas à toutes les personnes de suivre
- la démonstration ne met pas en évidence toutes les histoires d'utilisateur réalisées pendant le sprint

Après la démonstration

- le feedback n'est pas sollicité
- le backlog de produit et le plan de release ne sont pas mis à jour

La rétrospective de sprint

La rétrospective du sprint

- Le but de la réunion est d'améliorer le processus pour le prochain sprint.

Participants

- Toute l'équipe Scrum participe à la réunion.

La rétrospective a lieu juste après la revue de sprint et les intervenants qui sont venus y assister peuvent rester pour la rétrospective, à titre d'observateurs. En général, il y a le directeur de produit, l'équipe complète et le scrum master

- La réunion peut être animée par le Scrum Master. En effet, Le Scrum master montre le sprint backlog et, avec l'aide de l'équipe, résume le sprint. Les événements et décisions importantes, etc.

Produit en entrée

- le plan d'actions de la rétrospective précédente
- le backlog de problèmes s'il existe

Les étapes de la rétrospective du sprint

Créer un environnement propice à l'expression

- l'objectif de la rétrospective est de s'améliorer et non pas de juger, le principe de base d'une rétrospective est rappelé donc aux participants :
 - « Indépendamment de ce que nous allons découvrir aujourd'hui, nous comprenons et nous croyons vraiment que chacun a fait de son mieux, en fonction de ses connaissances, de ses compétences et de ses capacités, des ressources disponibles et de la situation courante ».
- S'il y a des doutes sur la confiance, il est préférable de demander à chacun, à bulletin secret, de dire s'il se sent capable de s'exprimer librement.

Les étapes de la rétrospective du sprint

Collecter les informations relatives au processus

- Avant de chercher à améliorer les pratiques, il convient de collecter le ressenti des participants sur ce qui s'est passé pendant le sprint qui s'achève.
 - L'approche basique est de demander à chaque participant
 - **Qu'est-ce qui a bien marché pendant le sprint?**
 - **Qu'est-ce qui a mal marché et qui pourrait être amélioré pour le prochain sprint?**
- En pratique , des tours de table sont effectués.
 - Chaque personne a une chance de dire, sans être interrompu, **ce qu'elle pense être bien, ce qu'elle pense qui peut être mieux, et ce qu'elle pense qui devrait être différent pour le prochain sprint.**
 - la vélocité estimée et la vélocité effective sont comparées. S'il y a une grosse différence, il faut essayer d'analyser le pourquoi.
- l'objectif est d'aboutir à lister les événements qui sont intervenus en les classant selon qu'ils ont eu un impact positif ou négatif sur le projet afin d'améliorer le prochain sprint

Les étapes de la rétrospective du sprint

Consolider

- Les événements sont regroupés en catégories d'amélioration, de façon à en obtenir entre 5 et 10.
- Ce travail sera fait de préférence de façon collective, ce qui est facilité par l'utilisation de notes collantes (post-it) pour l'identification des événements à l'étape précédente.

Définir les priorités

- Il s'agit de définir sur quelle catégorie va porter l'effort d'amélioration. Il est préférable de n'en sélectionner qu'une seule. La technique de sélection doit permettre la participation de toute l'équipe, avec par exemple un système de votes.

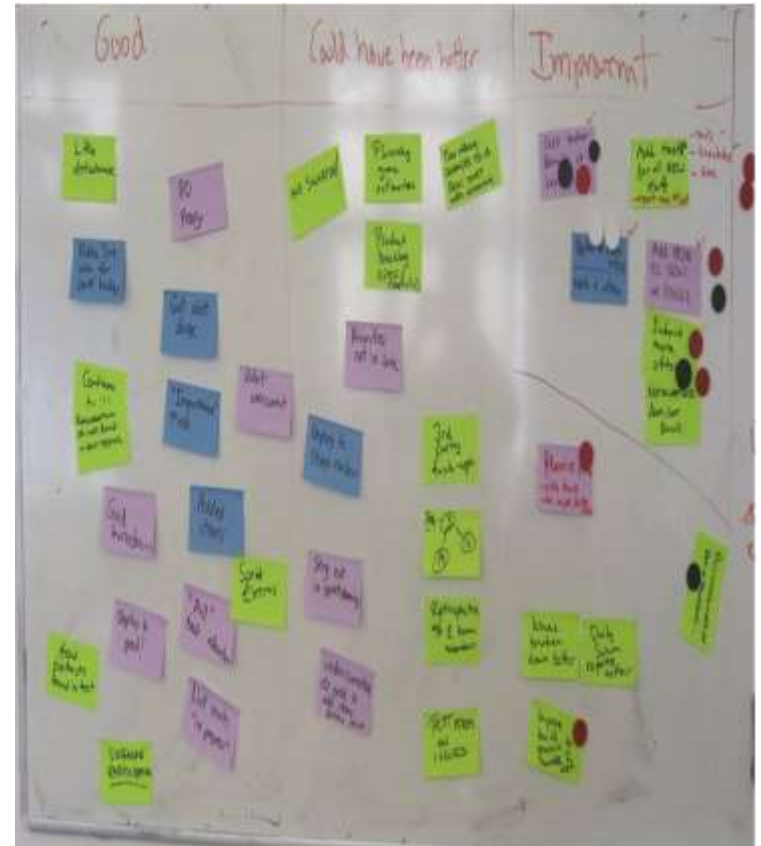
Planifier des actions d'amélioration

- Pour la catégorie choisie pour l'amélioration, il s'agit de définir les actions concrètes qui vont être menées en ayant un responsable de l'application de chacune.

Les étapes de la rétrospective du sprint

Trois colonnes :

- **Bien** : Si nous pouvions refaire le même sprint, nous ferions ces choses exactement pareil.
- **Peut mieux faire** : Si nous pouvions refaire le même sprint, nous ferions ces choses différemment.
- **Améliorations** : Idées concrètes pour s'améliorer dans le futur.
- Les colonnes 1 et 2 concernent le passé, tandis que la colonne 3 concerne le futur.



exemple de tableau blanc lors d'une réunion de rétrospective

Les étapes de la rétrospective du sprint

- Après que les membres de l'équipe aient déballé tous ces post-its, ils utilisent « le vote par points » pour déterminer quelles améliorations prendre en compte au prochain sprint.
- Chaque membre de l'équipe dispose de 3 aimants et sont invités à voter pour les améliorations qu'ils aimeraient voir pris en compte en priorité pendant le prochain sprint.
- Chaque membre peut distribuer ces aimants comme il veut, il peut même placer les 3 sur le même post-it.
- En se basant là-dessus ils sélectionnent les 5 améliorations à faire en priorité, et les suivront à la prochaine rétrospective.
- C'est important de ne pas être trop ambitieux ici. Il faut se concentrer sur un petit nombre d'améliorations par sprint.

La rétrospective du sprint

Produit en sortie

- le plan d'actions pour le sprint qui va démarrer.
- Des actions peuvent être consignées dans le backlog de sprint.

Considérations importantes

- La réunion est limitée dans le temps.
 - La première faite avec une équipe est toujours un peu plus longue. Ensuite la durée peut être limitée à une heure.

Les erreurs à éviter dans la rétrospective du sprint

Sur le plan d'actions de la rétrospective précédente :

- Les actions n'ont pas été faites

Pendant la réunion :

- Peu d'informations sont collectées
- Des personnes ne s'expriment pas du tout
- Une seule personne parle tout le temps

A la fin de la réunion :

- Le plan d'actions n'existe pas ou ne précise pas concrètement ce qui va être fait
- Le plan d'actions porte sur de trop nombreux axes d'amélioration
- Il n'y a pas de responsable pour les actions définies
- L'erreur la plus grave est d'arrêter de faire des rétrospectives. Le Manifeste Agile dit :
 - A intervalles réguliers, l'équipe réfléchit à comment devenir plus efficace, puis adapte et ajuste son comportement en conséquence.
- Ne pas utiliser cette possibilité de feedback sur la façon de travailler, ce serait passer à côté d'un des bénéfices majeurs des méthodes agiles.

La diffusion des enseignements tirés de la rétrospectives entre les équipes

- Les informations tirées d'une rétrospective de sprint ont généralement beaucoup de valeur.
- Une alternative serait que chaque équipe Scrum publie un rapport de rétrospective de sprint. Cependant, peu de gens lisent les rapports, et encore moins s'y conforment.
- Une approche simple qui marche assez bien. Une personne assiste à toutes les rétrospectives de sprint et agit comme un pont de connaissance.
- Les règles importantes pour la personne qui fait le « pont de connaissances » :
 - elle doit savoir écouter.
 - Si la rétrospective est trop silencieuse, elle doit être prête à poser des questions simples mais ciblées afin de stimuler les discussions au sein du groupe. Par exemple « si vous pouviez remonter le temps et refaire le même sprint depuis le 1er jour, qu'est-ce que vous changeriez ? ».
 - elle devrait prendre le temps d'aller aux rétrospectives de toutes les équipes.
 - elle devrait avoir une certaine autorité, de façon à mettre en œuvre les suggestions d'amélioration qui ne sont pas sous le contrôle de l'équipe.
- Il ne faut surtout pas oublier d'accrocher la rétrospective de sprint sur le mur du bureau de l'équipe

Entre les sprints

Relâcher le rythme entre les sprints

- Dans la vraie vie, on ne peut pas sprinter tout le temps. On a besoin de se reposer entre deux sprints. Si on sprinte tout le temps, on gardera une vitesse constante et on manquera d'énergie pour redoubler de vitesse pour les prochains sprints.
- C'est la même chose avec Scrum et le développement logiciel en général. En effet, les sprints sont plutôt intenses et durant le sprint en tant que développeur, vous ne vous relâchez jamais.
- La raison pour laquelle il est nécessaire d'avoir un relâchement entre les sprints est que après la démo du sprint et la rétrospective, à la fois l'équipe et le directeur de produit seront saturés d'informations et d'idées à assimiler. S'ils se remettent immédiatement à courir et commencer à planifier le sprint suivant, il est peu probable que quiconque ait l'occasion de digérer une information ou des leçons apprises, et que le directeur de produit n'aura pas le temps d'ajuster ses priorités après la démo de sprint, etc.

Temps entre les sprints

- L'objectif d'insérer du temps de détente après la rétrospective de sprint et avant la prochaine réunion de planification de sprint suivant

Mauvais

Lundi

09 – 10 : Démon de sprint 1
10 – 11: Rétrospective sprint 1

13 – 16: Planning sprint 2

Mieux

Lundi

09 – 10 : Démon de sprint 1
10 – 11: Rétrospective sprint 1

Mardi

09 – 13 : Planning sprint 2

Encore Mieux

Vendredi

09 – 10 : Démon de sprint 1
10 – 11: Rétrospective sprint 1

Samedi

Dimanche

Lundi

09 – 13 : Planning sprint 2

Temps entre les sprints

- La manière la plus appropriée est d'organiser une journée « **freework** » durant laquelle les développeurs seront libre pour s'adonner à une activité de leur choix (étudier à fond les derniers outils et APIs, suivre une formation, discuter de trucs d'informatique avec les collègues, coder un projet perso, etc.)

Meilleur

Jeudi	Vendredi	Samedi	Dimanche	Lundi
09 – 10 : Démonstrations de sprint 1 10 – 11: Rétrospective sprint 1	Journée Freework			09 – 13 : Planning sprint 2

Planification de release

La planification de release

- Des fois, on a besoin de planifier à l'avance plus d'un sprint à la fois.
- Spécifiquement dans un contexte de contrat au forfait où on *doit planifier en avance*, ou sinon on risque de signer pour quelque chose qu'on ne peut délivrer à temps.
- La planification de release est une tentative de répondre à la question « *quand, au plus tard, serons-nous en mesure de délivrer la version 1.0 de ce nouveau système ?* ».

Définition des seuils d'acceptation

- En plus du backlog de produit habituel, le directeur de produit définit une liste des seuils d'acceptation qui est une classification simple de ce que signifient les niveaux d'importance utilisés dans le backlog de produit par rapport aux aspects contractuels.

Exemple de règles de seuils d'acceptation :

- Tous les éléments avec une importance ≥ 100 *doivent être inclus* dans la version 1.0, ou sinon nous serions condamnés à payer des pénalités de retard.
- Tous les éléments avec une importance de 50-99 devraient être inclus dans la version 1.0, mais nous devrions être en mesure de nous en sortir en les incluant dans une release disponible à court terme.
- Les éléments avec une importance de 25-49 sont requis mais peuvent être faits dans une version future 1.1.
- Les éléments avec une importance < 25 sont spéculations et pourraient ne jamais être requis.

Définition des seuils d'acceptation

- L'objectif est de sélectionner un sous-ensemble du backlog pour la release.
- Comme les éléments sont rangés par priorité, il suffit de fixer une limite en disant : « on s'arrêtera là pour la release courante, les stories suivantes seront faites dans la release suivante ». C'est ce qu'on appelle une release à périmètre fixé.
- **La planification de la release a pour objectif d'estimer la date de fin de release.**

Attention : même restreint de cette façon, le périmètre peut toujours évoluer, il serait stupide de figer le backlog en refusant un changement qui apporte de la valeur.

Fixer la date à l'avance

- Une autre façon de procéder est de définir une date de fin de release et de s'y tenir absolument.
- La planification d'une release à date fixée a pour objectif d'estimer quel contenu sera fourni à la date prévue.
- La release à date fixée présente de nombreux avantages :
 - elle donne un objectif pas trop lointain, ce qui motive plus l'équipe,
 - elle demande obligatoirement une réflexion poussée sur les priorités des éléments du backlog par le Product Owner,
 - des éléments du backlog apportant le moins de valeur ne seront pas développés
 - on passe généralement moins de temps à estimer et planifier, puisque la date de livraison est connue.

Fixer la date à l'avance

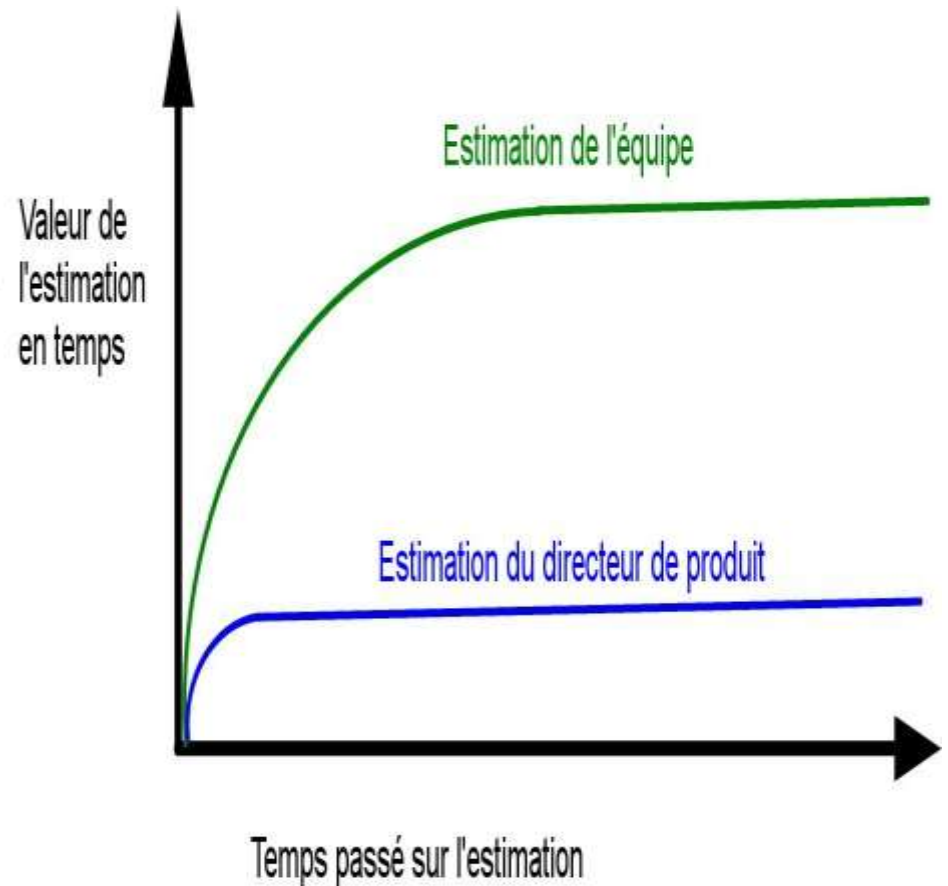
- Un autre avantage est le rythme donné par des releases régulières :
 - une organisation avec des releases régulières s'habituerà à cette fréquence, qui cadence le travail de l'équipe mais aussi celui des utilisateurs et de leurs représentants.
- La release à date fixée et à durée uniforme, par exemple une release tous les 3 mois, est la formule la plus facile à mettre en œuvre.

Estimation en temps des éléments les plus importants

- Afin de faire la planification de releases, le directeur de produit a besoin d'estimations, au moins pour toutes les histoires qui sont incluses dans le contrat.
- Tout comme la planification de sprint, il s'agit d'un effort mutuel entre le directeur de produit et l'équipe – l'équipe estime, le directeur de produit décrit les éléments et répond aux questions.

Estimation en temps

- Habituellement le directeur de produit rassemble toute l'équipe dans une pièce, et explique que le but de cette réunion est d'estimer en temps les 10 premières (ou autre) histoires du backlog de produit.
- Il aborde chaque histoire une fois, puis il laisse l'équipe faire ses estimations. Le directeur de produit reste dans la pièce pour répondre aux questions et clarifier le périmètre de chaque histoire si nécessaire. Juste comme pendant la planification du sprint.
- le champ « comment démontrer » est un moyen très utile pour réduire le risque de malentendus.
- Cette réunion doit être strictement bornée en temps,



Exemple d'estimations en temps

Imp	Nom	Estimation
130	banane	12
120	pomme	9
115	orange	20
110	goyave	8
100	poire	20
95	raisin	12
80	cacahuète	10
70	beignet	8
60	oignon	10
40	pamplemousse	14
35	papaye	4
10	myrtille	
10	pêche	

Estimation de la vélocité

- Après la définition des estimations brutes pour les histoires les plus importantes. La prochaine étape serait d'estimer la vélocité moyenne par sprint.
 - le besoin de définir le facteur de focalisation
- Supposons que le sprint dure 3 semaines (15 jours) et que la taille de l'équipe est 6 et que le facteur de focalisation déterminé est de 50%
- Pour rappel, le calcul de vélocité de sprint se calcule comme suit:

$$\text{Vélocité estimée pour le sprint en cours} = \frac{(\text{jours} * \text{hommes disponibles pour le sprint en cours})}{(\text{facteur de focalisation du sprint précédent})} \times$$

- Chaque sprint fait alors 90 jours*homme, mais produira seulement 45 jours*homme d'histoires (à cause du facteur de focalisation de 50%).
- La vélocité estimée est donc de 45 points d'histoire.
- Si chaque histoire avait une estimation en temps de 5 jours (ce qui n'est pas le cas) alors cette équipe effectuerait approximativement 9 histoires par sprint.

Le plan de release

- Maintenant que nous avons des estimations en temps et une vélocité (45)
- nous pouvons facilement découper le backlog de produit en plusieurs sprints :
- Chaque sprint contient autant d'histoires que possible dans la limite de la vélocité estimée à 45.
- Maintenant on peut voir que nous aurons probablement besoin de 3 sprints pour finir tous les « *doit être inclus* » et les « *devrait être inclus* ».

Imp	Nom	Estimation
Sprint 1		
130	banane	12
120	pomme	9
115	orange	20
Sprint 2		
110	goyave	8
100	poire	20
95	raisin	12
Sprint 3		
80	cacahuète	10
70	beignet	8
60	oignon	10
40	pamplemousse	14
Sprint 4		
35	papaye	4
10	myrtille	
10	pêche	

Le plan de release

- 3 sprints = 9 semaines calendaires = 2 mois calendaires.
- Maintenant est-ce vraiment la date de livraison promise au client ? Cela dépend entièrement de la nature du contrat ; à quel point le périmètre est-il fixé, etc.
- Généralement, il faut ajouter une réserve significative pour se protéger contre les mauvaises estimations en temps, les problèmes imprévus, les fonctionnalités inattendues, etc.
- Donc dans ce cas, on peut se mettre d'accord sur une livraison dans 3 mois, ce qui donne 1 mois de « réserve ».
- L'important, est que un livrable utilisable peut être démontré au client toutes les 3 semaines et l'inviter à modifier ses exigences au fur à mesure de l'avancement (cela dépend bien sûr du type de contrat).

La revue du plan de release

- Après chaque sprint, la vélocité effective est calculée pour ce sprint. Si la vélocité effective était très différente de la vélocité estimée, la vélocité estimée est revue pour les futurs sprints et le plan de release est mis à jour.
- Si cela devait mettre l'équipe en situation difficile, le scrum master pourrait soit:
 - négocier avec le client,
 - voir comment il peut réduire le périmètre sans rompre le contrat.
- Ou peut-être que lui et l'équipe trouvent-ils le moyen d'augmenter la vélocité ou le facteur de focalisation en supprimant des obstacles sérieux qui ont été identifiés durant le sprint.

Récapitulatif

- Une release est une série de sprints successifs. C'est **une période de temps**.
- Comment et quand déterminer la fin de la série et donc la fin de la release ?
- Il y a la **façon adaptative** : la fin de la release n'est pas fixée, elle est envisagée à la fin de chaque sprint. On parle **de release ajustable**.
- Il y a la **façon prédictive** : la date de fin de la release est décidée, c'est ce qu'on **appelle une release à date fixée**. Cela comporte des avantages.
- Une release ajustable peut devenir à date fixée pendant son déroulement. A la fin d'un sprint, on décide avec le product owner que le produit est dans un état qui permet de définir à quelle date on pourra finir la release.
- On peut évoquer un troisième type de release, même s'il n'est pas agile, c'est **la release à périmètre fixé** (dans la pratique, le périmètre n'est jamais fixe). Le périmètre est défini par les éléments du backlog à faire.

Récapitulatif

- A quoi sert de connaître le type de release ? Cela est déjà une information publiée, importante pour l'équipe et à l'extérieur. Cela permet également de faire une planification :
 - pour une release à périmètre fixé, l'objectif de la planification est de calculer la date de fin.
 - pour une release à date fixée, l'objectif de la planification est de calculer les fonctionnalités (stories) qui seront finies à cette date.

Références

1. *Scrum et XP depuis les Tranchées* par Henrik Kniberg
2. Aubry conseil
<http://www.aubryconseil.com/post/2008/03/31/395-disposition-pour-le-tableau-des-taches>
<http://www.aubryconseil.com/post/2007/11/25/336-la-revue-de-sprint>
<http://www.aubryconseil.com/post/2007/11/13/328-la-retrospective-de-sprint>
<http://www.aubryconseil.com/post/2007/02/09/174-retrospective-du-sprint-5-d-icescrum2>
<http://www.aubryconseil.com/post/2008/02/04/372-la-release-a-points>
<http://www.aubryconseil.com/post/La-fin-d-une-release>
<FR.pdf#zoom=100http://www.aubryconseil.com/post/2008/12/18/513-burndown-chart-en-couleurs>
3. Guide de démarrage Scrum - Florent Lothon, mis à jour en juin 2013
[agiliste.fr /fiches/guide-demarrage-scrum/](http://agiliste.fr/fiches/guide-demarrage-scrum/)
4. Le Guide Scrum - Le guide définitif de Scrum :
<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide->
5. <http://www.burndownchart.fr/>
6. Institut Agile
<http://referentiel.institut-agile.fr/daily.html>
7. <http://blog.pascal-martin.fr/post/integration-continue-jenkins-projet-php>
8. <http://www.gazelle-du-web.com/strategies/scrum/>