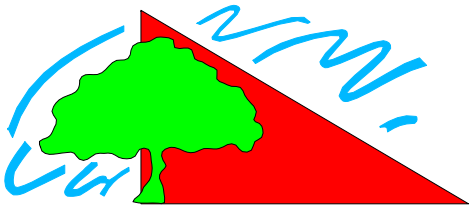


Gestion de projet Agile

STS IRIS

Module 4.2 - « Gérer et organiser un projet informatique »

Sommaire

Introduction

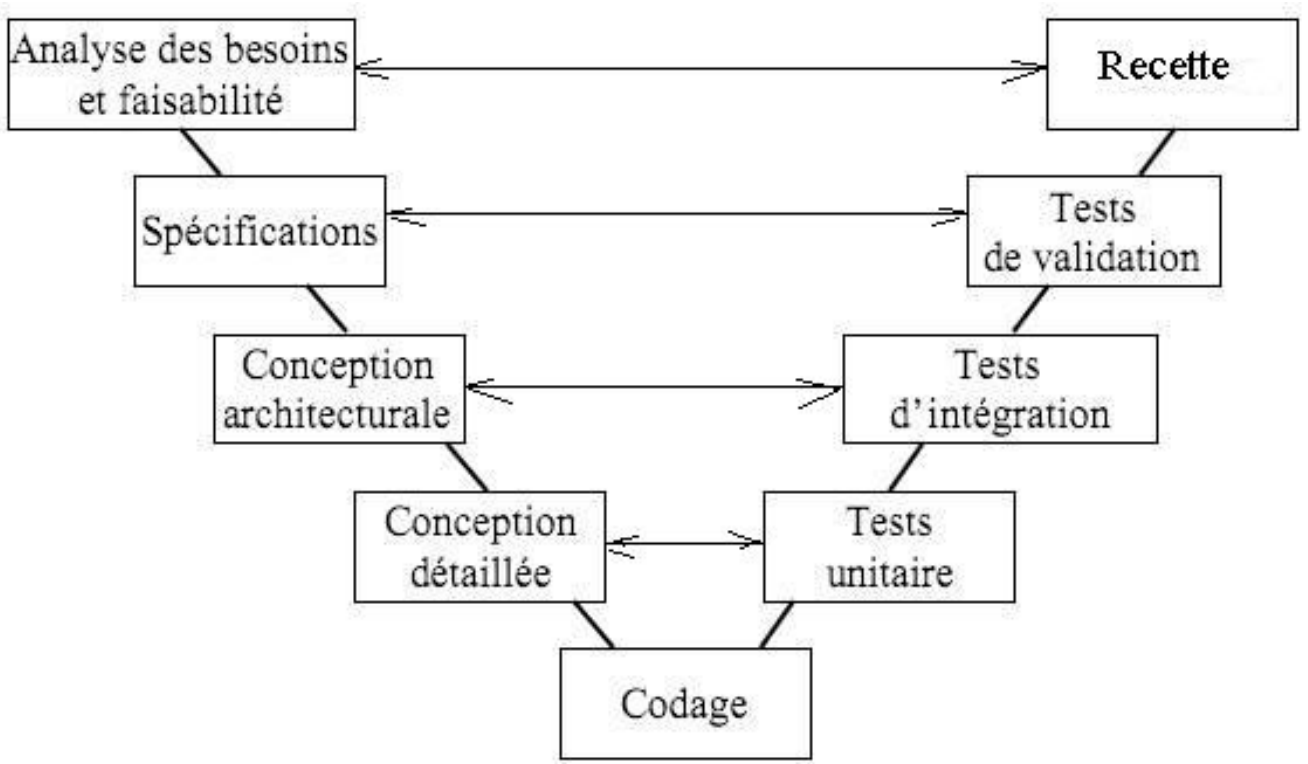
Principes et méthodes Agiles

Scrum

Introduction

- Gestion de projet : démarche structurante assurant le bon déroulement d'un projet avec :
  - Une planification
  - Une gestion des ressources humaines
  - Un suivi des enjeux financiers
- Les principales méthodes de gestion de projet :
  - Découpage en phases (voir cycle en V)
  - Découpage en activités WBS (Work Brakedown Structure)
  - Nouveau : Les méthodes Agiles

Approche en cascade : Cycle en V



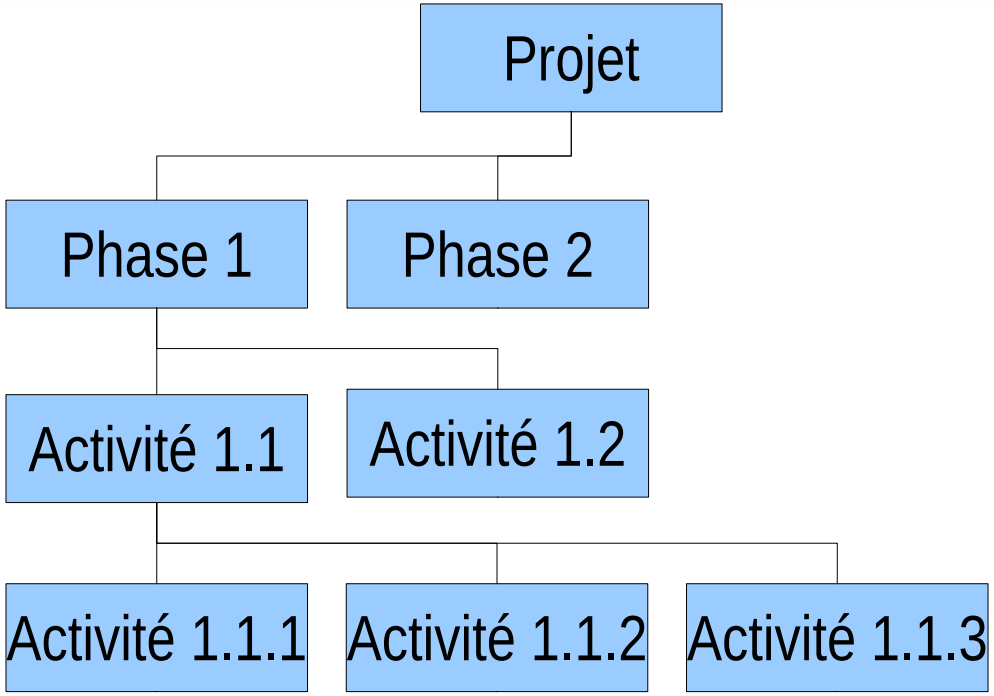
Inconvénients :

- rigidité de l'approche : on n'aime pas la nouveauté
- pour les développeurs : tests de validation tardifs
- pour les clients : recette tardive
- documentation pléthorique

Activités WBS (Work Brakedown Structure) (1)

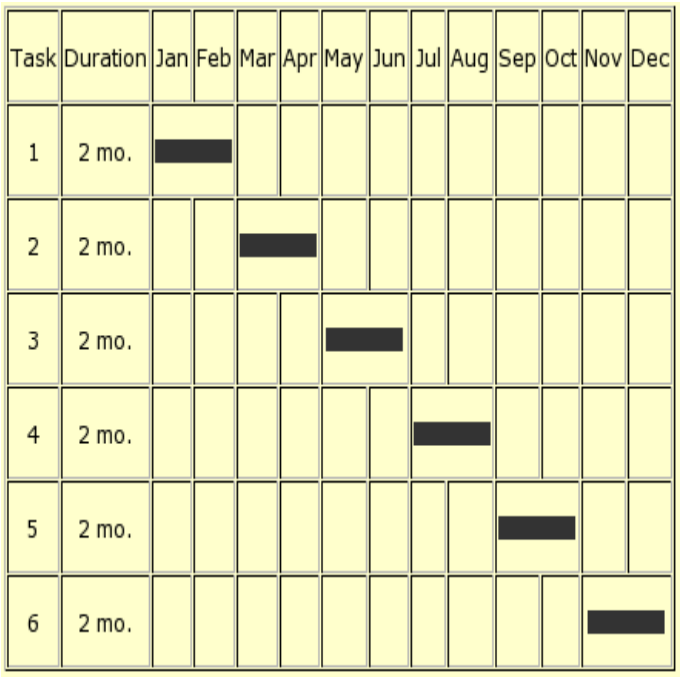
- Découpage en activités qui possèdent :
  - des entrées et des résultats
  - un responsable
- Le découpage se fait jusqu'à ce que l'on maîtrise :
  - La durée de l'activité
  - Les ressources associées
  - Le coût de l'activité
- Les tâches doivent être indépendantes les unes des autres.

Activités WBS (Work Brakedown Structure) (2)



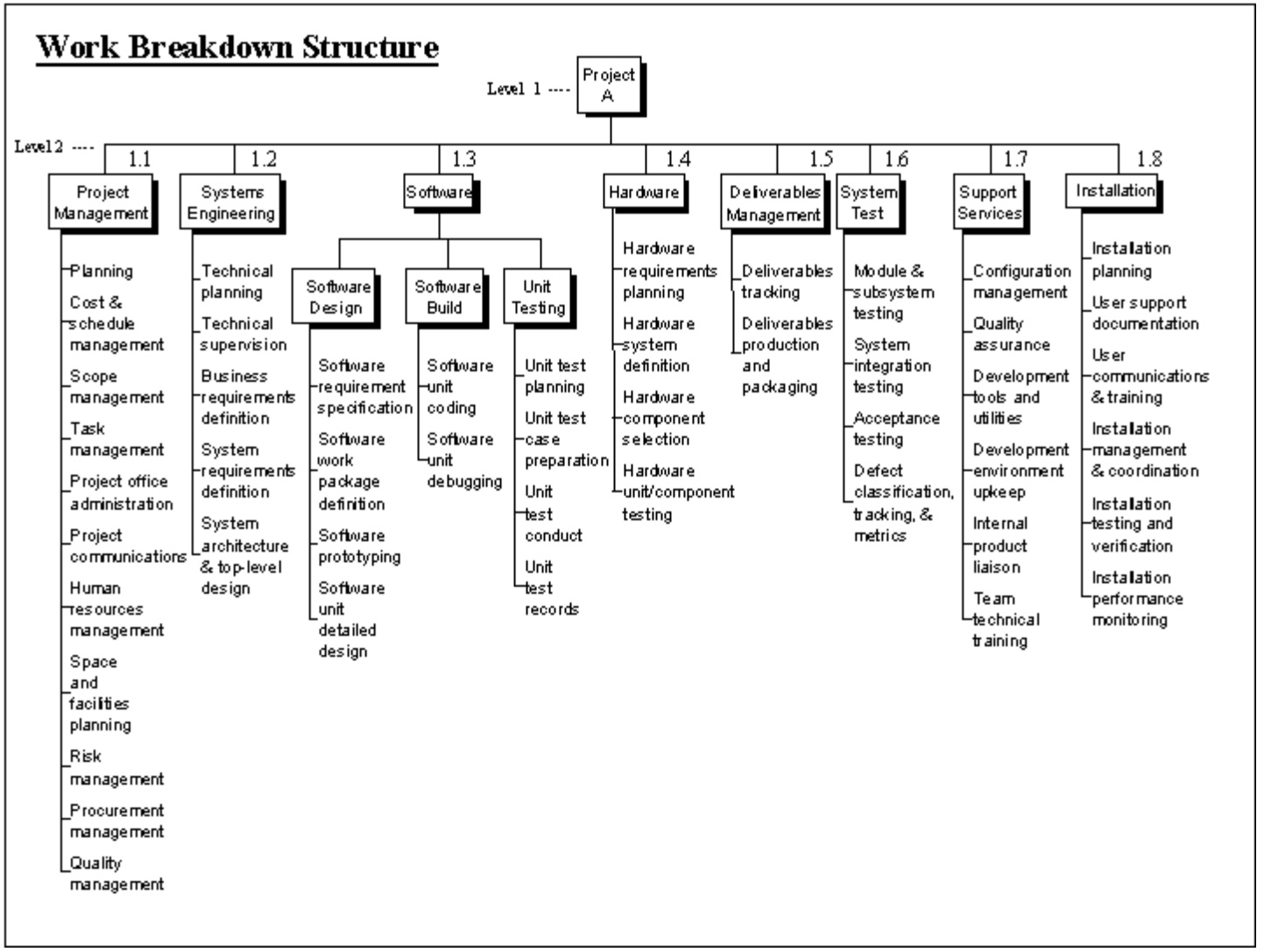
- Les phases sont réalisées en groupe
- Les activités sont réalisées individuellement
- Une activité doit durer entre quelques jours et quelques mois maximum.

Rôle des diagrammes de Gantt



- Utile pour afficher le statut des activités en parallèles
- Peut aider à identifier les activités dans une démarche WBS
- Pour les projets plus complexe on pourra utiliser le modèle de gestion de projet PERT

Exemple de structure WBS



Les risques engendrés par les méthodes classiques

- Mauvaise interprétation des souhaits du client
- Changement des besoins fonctionnels
- Dépassements des délais et des budgets
- Bugs
- Abandon du projet
- Au final, seuls ¼ des projets sont considérés comme réussis

Sommaire

Introduction
Principes et méthodes Agiles
Scrum

Agile : les variables d'ajustement d'un projet

Coût

Qualité

Durée

Périmètre fonctionnel

Règle du jeu :

- Le client a le droit de fixer 3 variables
- L'équipe de développement ajuste la dernière.

Le périmètre fonctionnel est la variable qui fournit la maîtrise la plus efficace.

Les méthodes Agiles

- Barry W. Boehm a introduit en 1986 un nouveau modèle de développement itératif et incrémental, précurseur des méthodes Extreme programming (XP), Scrum ou Crystal clear…
- En 2001, un manifeste écrit par 17 experts introduit 4 valeurs fondamentales déclinées en 13 principes permettant de définir une nouvelle façon de développer des logiciels.
- http://www.agilemanifesto.org/

Gang of 17... Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Les 4 valeurs de l'Agilité

- L'équipe :
  - Les individus et leurs interactions avant les processus et les outils.
- L’application :
  - Des fonctionnalités opérationnelles avant la documentation.
- La collaboration :
  - Collaboration avec le client plutôt que contractualisation des relations.
- L’acceptation du changement :
  - Adaptation au changement plutôt que conformité aux plans

Les principes de l'Agilité (1)

- « Notre priorité est de satisfaire le client par des livraisons rapides et continues de logiciel utile.
- Accepter le changement dans les exigences, même tard dans le cycle de vie, pour garantir la compétitivité du client.
- Livrer fréquemment du logiciel opérationnel, de quelques semaines à quelques mois en visant les délais courts.
- Client et développeurs doivent coopérer quotidiennement tout au long du projet
- Élaborer des projets autour d'individus motivés. Leur procurer l'environnement et le support nécessaire et leur faire confiance pour réaliser le travail.

Les principes de l'Agilité (2)

- La méthode la plus efficace de communiquer des informations à une équipe et entre ses membres reste la conversation en face à face.
- Le fonctionnement de l'application est le premier indicateur d'avancement du projet
- Agile favorise le développement à rythme "normal" ou soutenable.
- Les gestionnaires, développeurs et utilisateurs devraient être en mesure de maintenir un rythme constant et ce, indéfiniment.
- Porter une attention continue à l'excellence technique et à la conception améliore l'agilité.

Les principes de l'Agilité (3)

- La simplicité garantit l'évolutivité du système
- Les meilleures architectures, exigences et designs prennent naissance dans des équipes qui se gèrent elles-mêmes.
- Régulièrement, l'équipe fait une réflexion sur les façons de devenir plus efficace, s'ajuste et modifie son comportement en conséquence. »

Responsabilisation de l'équipe de développement Agile

- Les méthodes Agiles responsabilise l'équipe :
  - l'équipe connaît les besoins et les priorités,
  - elle fait les estimations,
  - elle décide de son organisation,
  - elle produit un travail de qualité,
  - elle remonte les problèmes.

Les différentes méthodes Agiles

- Adaptative Software Development (ADS)
- Crystal
- Scrum
- Extreme Programming (XP)

Sommaire

Introduction
Principes et méthodes Agiles
Scrum

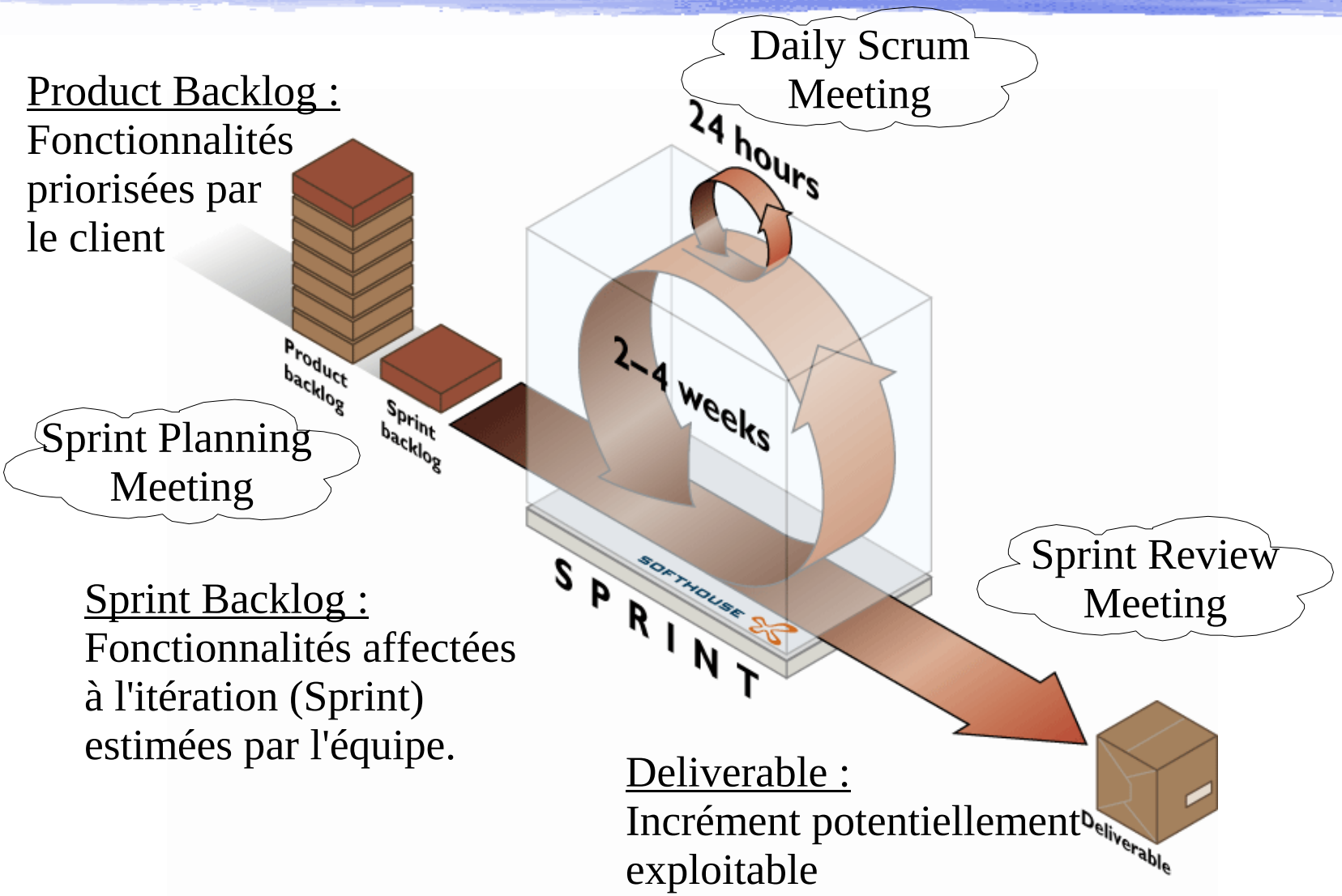
Scrum

Introduction à Scrum

- Scrum est une méthode Agile qui permet de produire la plus grande valeur métier dans la durée la plus courte.
- Du logiciel qui fonctionne est produit à chaque sprint, c'est à dire toutes les 3 / 4 semaines.
- Le métier définit les priorités, l'équipe s'organise elle-même pour déterminer la meilleure façon de produire les exigences les plus prioritaires.
- A chaque fin de sprint, tout le monde peut voir fonctionner le produit courant et décider soit de le livrer dans l'état, soit de continuer à l'améliorer pendant un sprint supplémentaire.

Cycle de vie de Scrum

Product Backlog :
Fonctionnalités priorisées par le client



Sprint Backlog :
Fonctionnalités affectées à l'itération (Sprint) estimées par l'équipe.

Deliverable :
Incrément potentiellement exploitable

Les rôles dans une équipe Scrum (1)

- Un directeur de produit (product owner) qui est soit le client, soit une personne représentant le client, il:
  - définit les fonctionnalités du produit
  - choisit la date et le contenu de la release
  - responsable du retour sur investissement
  - définit les priorités dans le backlog en fonction de la valeur métier
  - ajuste les fonctionnalités et les priorités à chaque sprint si nécessaire
  - accepte et rejette les résultats

Les rôles dans une équipe Scrum (2)

- Un Scrum Master qui:
  - représente le management de projet
  - est responsable de faire appliquer les valeurs et les pratiques de Scrum par l'équipe
  - résout les problèmes
  - s'assure que l'équipe est complètement fonctionnelle et productive
  - facilite une coopération poussée entre tous les rôles et fonctions
  - protège l'équipe des interférences extérieures

Les rôles dans une équipe Scrum (3)

- Les équipiers qui:
  - se composent de 5 à 10 personnes
  - regroupent tous les rôles: architecte, concepteur, analyste, développeur, testeur, …
  - sont à plein temps sur le projet
  - s'organisent eux-mêmes
  - ne changent pas de composition pendant un sprint
  - se concentrent sur un sprint à la fois (sprint courant)

Scrum : les réunions (1)

- Planification du Sprint (2 à 4h)
  - Définir le but du sprint
  - Définition du périmètre du sprint
  - Identification les tâches à partir des éléments sélectionnés
  - Estimation des tâches
  - Attribution des tâches
  - Obtenir l'engagement de l'équipe

Scrum : les réunions (2)

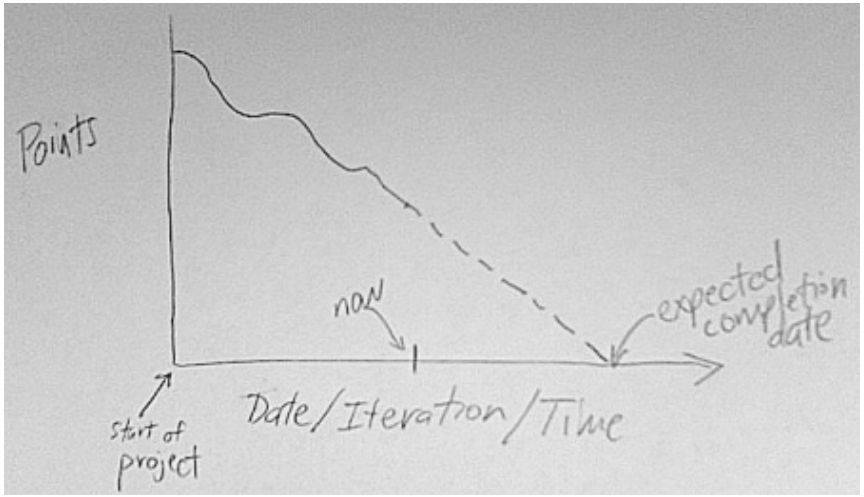
- Scrum quotidien (15mn debout)
  - Qu'as-tu fait depuis la dernière fois ?
  - Que prévois-tu de faire jusqu'à la prochaine réunion ?
  - Qu'est-ce qui te gêne pour réaliser ton travail aussi efficacement que possible ?
- Revue de sprint (2 à 4h)
  - Préparer la démonstration
  - Rappeler les objectifs du sprint
  - Effectuer la démonstration
  - Évaluer les résultats du sprint
  - Calculer la vélocité réelle et ajuster le plan de release

Estimation et planification (1)

- Une release se compose de plusieurs itérations, chaque itération contient 2 à 3 histoires utilisateurs (user story ~ cas d'utilisation UML).
- Chaque histoire utilisateur se découpe en tâches.
- Chaque tâche possède un nombre de points qui correspond à la taille de son exigence en terme de travail et de complexité.
- Les différents acteurs participent sur l'attributions des points de toutes les tâches et se mettent d'accord.
- Une fonctionnalité avec un point précis sert de référence pour l'estimation des points des autres tâches. Les points suivent la suite de Fibonacci: 1 2 3 5 8 13.

Estimation et planification (2)

- Dans le cadre du suivi, seul le reste à faire est pris en compte.
- Le reste à faire se compte en heures.
- La vélocité de l'équipe correspond au nombre de points faits pendant une itération.
- Le suivi peut se faire à l'aide d'un graphe d'activité de l'équipe (burndown).



Scrum au quotidien…