

# Introduction aux méthodes AGILES et à SCRUM

---

## Causes d'échec en informatique

(selon le Standish Group)

- Manque de clarté ou mauvaise définition des besoins
- Évolution des spécifications
- Manque de réactivité
- Priorités non définies
- Manque de qualité du logiciel
- Conception trop ambitieuse
- Évolutions non prévues
- Rarement parce que la programmation est mauvaise.

## Pourquoi les méthodes agiles

- Une alternative aux méthodes traditionnelles.
- Des méthodes pragmatiques, partant du principe que les besoins évoluent.
- Cycles de développement itératifs, incrémentaux et adaptatifs.
- Une valeur métier privilégiée.
- Planification et gestion des risques partagées avec le client.
- Grande importance des retours utilisateurs.
- Le changement n'est plus considéré comme une perturbation, mais est intégré dans l'organisation du projet.

## Manifeste des méthodes agiles

4 valeurs

L'équipe	« Personnes et interaction plutôt que processus et outils »
L'application	« Logiciel fonctionnel plutôt que documentation complète »
La collaboration	« Collaboration avec le client plutôt que négociation de contrat »
L'acceptation du changement	« Réagir au changement plutôt que suivre un plan »

## Scrum qu'est-ce que c'est ?

- Suivi côté client par le Product Owner.
  - Fonctionnalités souhaitées collectées dans un backlog de produit.
  - Le développement est constitué de plusieurs itérations (sprints).
  - La liste des tâches est priorisée.
  - Au cours de chaque sprint, on effectue des mêlées quotidiennes.
  - A la fin du sprint, l'équipe obtient un produit partiel potentiellement livrable.
  - Après plusieurs sprints, on parle de version (release).
- 
- SCRUM est un framework de méthodologie
  - SCRUM est un framework non prescriptif

## Les rôles

### Scrum Master

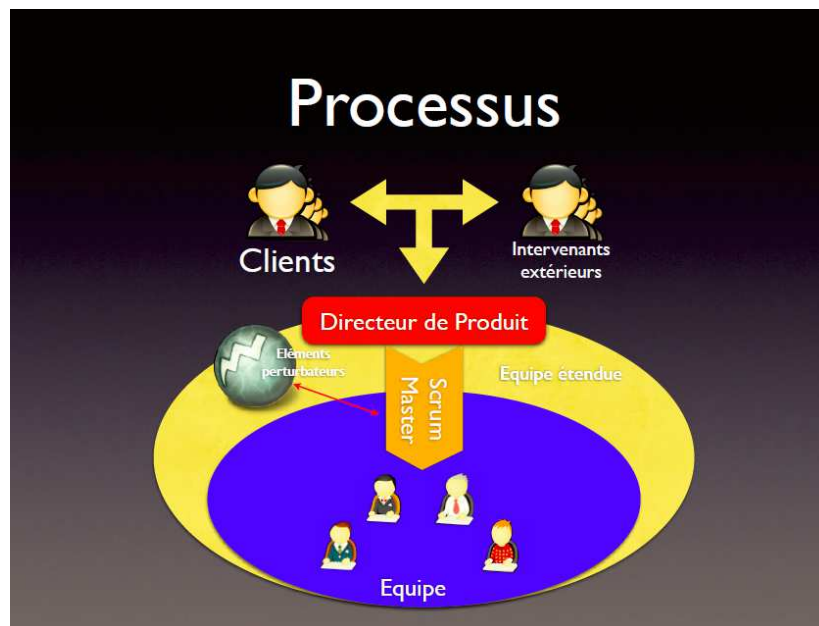
- Responsable de faire appliquer par l'équipe les valeurs et les pratiques de Scrum
- Facilite la résolution des problèmes

### Product Owner (Directeur de produit)

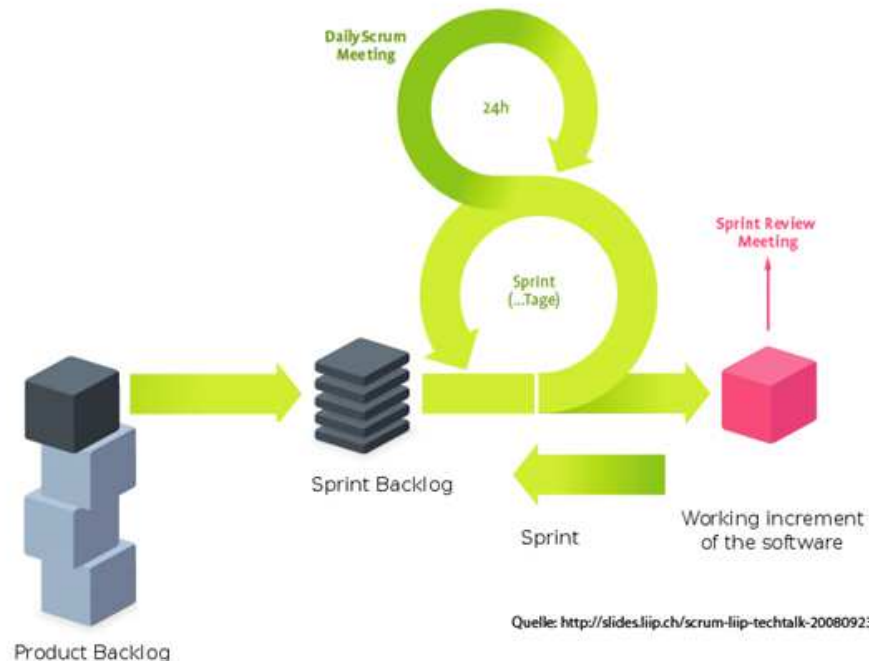
- C'est le représentant des clients et des utilisateurs
- C'est lui qui donne les fonctionnalités à traiter, et qui prend les décisions importantes concernant l'orientation du projet
- Il gère le Backlog de Produit et le Release Plan

### Team Member (Équipe)

- Tous les autres



## Workflow Agile



### Définitions

**Sprint** : itération dans Scrum – 2 semaines à 1 mois

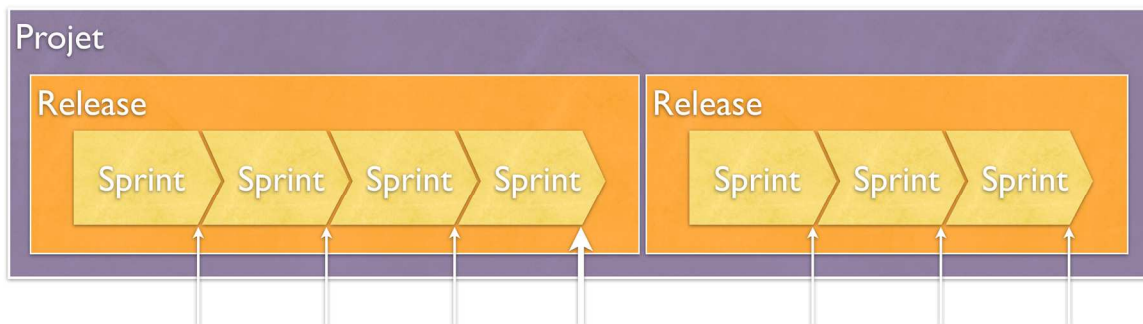
**Scrum** : mêlée quotidienne

**Product Backlog** : cahier des charges initial

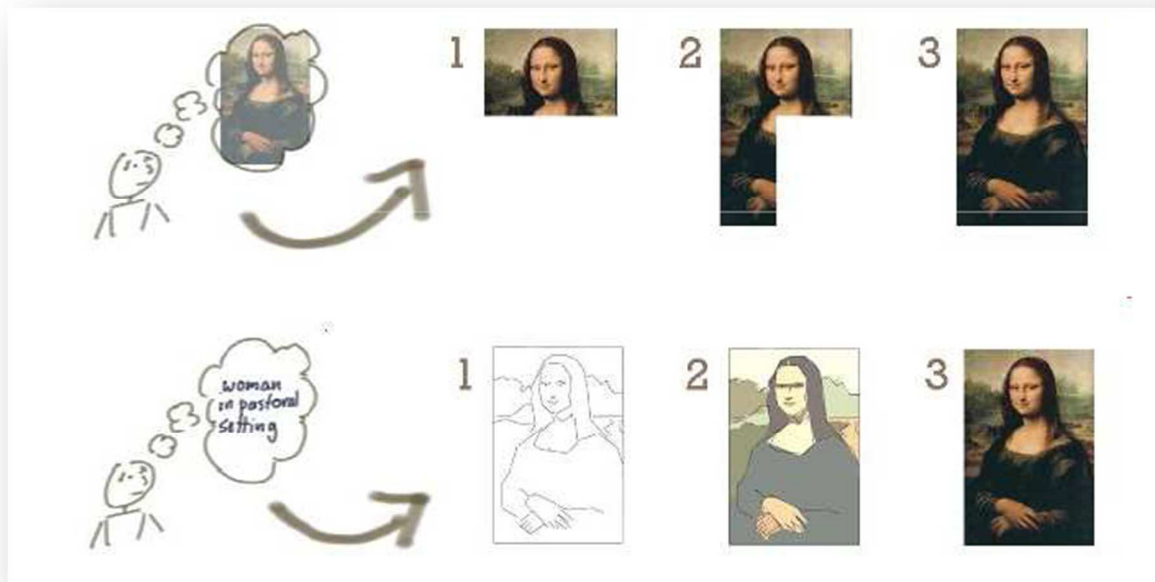
**User Story** : terme eXtreme Programming, qui définit la manière d'exprimer les attentes **utilisateur**

**Sprint Backlog** : le contenu choisi pour un sprint

**Scrum daily meeting** : réunion quotidienne de l'équipe développement.



## Itératif vs Incrémental



d'après S. Mathon (Wembla conseil)

### Les règles fondamentales

- Les itérations sont courtes : 2 semaines à 1 mois maximum
- Les itérations ne se chevauchent pas
- Les itérations ont toujours la même durée
- La date de fin du sprint n'est JAMAIS repoussée
- Les itérations s'enchaînent en général sans délai.

## Scrum : Le démarrage d'une version

- Comprendre le contexte et les objectifs du projet.
- Déterminer les utilisateurs du projet

### Objectifs du projet

- Résumer le projet en une phrase
- Quels sont les avantages « business » pour le client?
- Définir comment mesurer le succès du projet
- Le projet est-il réaliste/faisable?
- Est-ce que toutes les personnes impliquées l'approuvent?
- Quelle est l'importance du projet pour le client?

### La User Story

Les User Stories sont des « histoires »

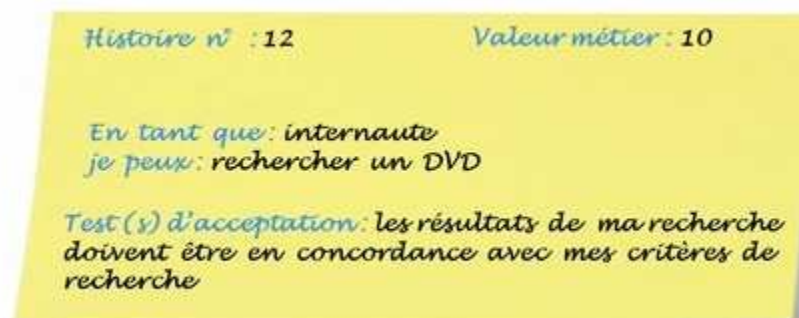
Avec :

- un **acteur**
- qui effectue une **action**
- dans un **objectif** donné.

Une User Story doit pouvoir être développée entièrement pendant une itération.  
Un Backlog contient également des Stories techniques ou méthodologiques (Ex : tests unitaires)

Exemples :

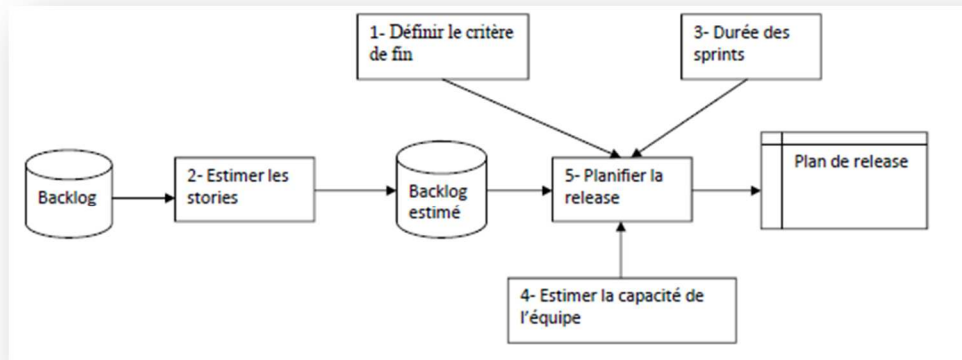
#	Backlog Item (User Story)	Story Point
1	As a Teller I want to be able to find clients by last name, so that I can find their profile faster	4
2	As a System Admin I want to be able to configure user settings so that I can control access	2
3	As a System Administrator I want to be able to add new users when required so that...	2
4	As a data entry clerk, I want the system to automatically check my spelling so that...	1



## Exemple de Product Backlog

Voir exemple fourni.

## Le Plan de Release : pour garder le cap

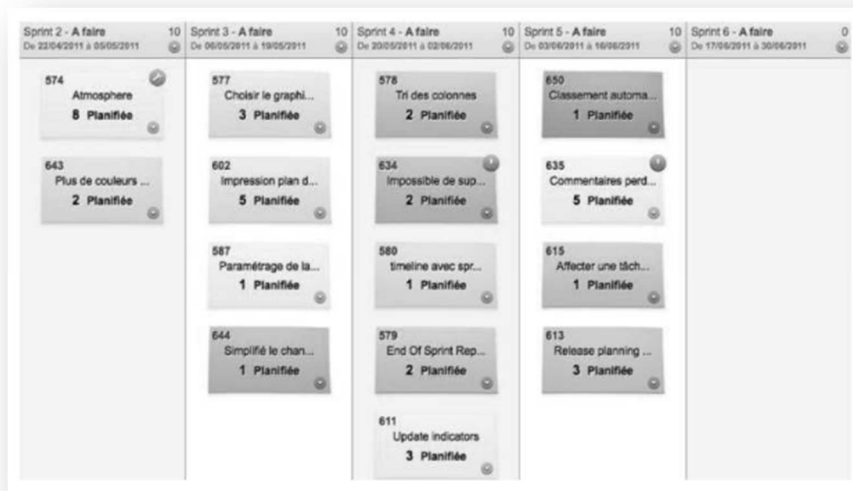


Un plan de release présente les sprints à venir et le contenu prévu de ces sprints, défini par les stories associées.

- Répartition indicative des User Stories dans les sprints
- Prise en compte des dates fatidiques
- le plan de release évolue pour tenir compte des changements
- Fait par toute l'équipe.

Le plan de release permet d'identifier les points de synchronisation nécessaires et d'anticiper en adaptant les priorités

## Exemple de plan :



## Problèmes classiques de Backlog

Les problèmes classiques de gestion des exigences :

- Stories exprimées sous forme de solution
- Stories exprimées sous forme technique
- Ambiguïté/flou
- Manques/doublons/incompatibilité

Product Backlog trop lourd

Stories trop grosses

## L'estimation dans Scrum

- Estimer la taille/difficulté plutôt que la durée
- Estimation en points = jours/hommes idéaux
- Estimer de façon relative, par rapport à une story connue
- Les estimations sont INDICATIVES

## Scrum : les sprints

Durée fixe

Consensus entre

- le besoin de feedback/la motivation
- vs le coût lié au sprint/la disponibilité du Product Owner

Au minimum 4 sprints par version

## La réunion de planification de sprint

- 1- Le Product Owner présente l'objectif du sprint et les Stories candidates
  - 2- L'équipe liste les tâches nécessaires (<1 jour) et affine l'estimation
  - 3- Accord sur le périmètre du sprint
- Compromis entre la capacité, la faisabilité et l'importance

## A préparer avant

Le Product Backlog existe :

Les exigences/User stories sont listées

Le Product Owner a mis l'importance des stories les plus importantes et sélectionné ses candidates

Le Scrum Master a calculé la capacité du sprint (quelle quantité de stories peut être traitée)

## Conditions

Ailleurs que dans le bureau

Tableau blanc

Fixer la durée maximum à respecter : en général, 2\* heures

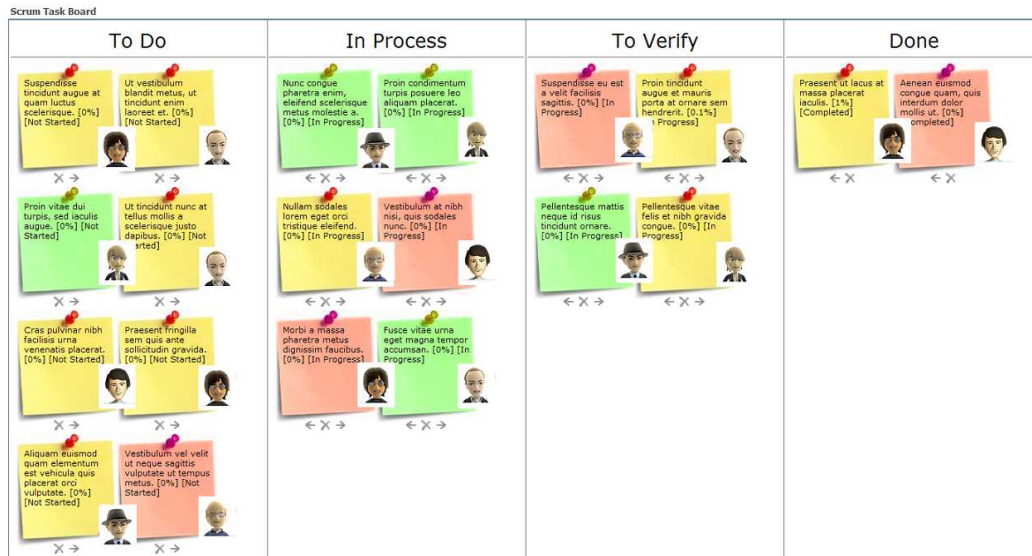
Ne pas commencer à résoudre les problèmes techniques, mais faire de la conception

Poser les bonnes questions au Product Owner

Garder une marge de manœuvre.



## Le tableau blanc (tableau des tâches)



## Backlog de Sprint

Requirement	Task	Who's working	Status	Work Left (Hours)				
				Day 1	Day 2	Day 3	Day 4	Day 5
Member Sign In	Database Coding	Anjali	Done	1	0	0		
	Unit Testing Scripts	Anjali	Done	2	0	0		
	Business Logic	Anjali	Done	1	0	0		
	UI Test Scripts	Anu	Done	2	2	0		
	Front End Screens	Anjali	Done	2	2	0		
	Load Testing	Anu/ Anjali	Done	2	2	1		
Reset Password	Unit Testing Scripts	Thomas	Done	1	0	0		
	Business Logic	Thomas	Done	1	0	0		
	UI Test Scripts	Anu	Done	1	1	0		
	Front End Screens	Thomas	Done	1	1	1		
	Integration Testing	Anu	Pending	1	1	1		
Change Password	Unit Testing Scripts	Thomas	Done	0.5	0	0		
	Business Logic	Thomas	Done	0.5	0	0		
	UI Test Scripts	Anu	Done	0.5	0	0		
	Front End Screens	Thomas	Done	0.5	0	0		
	Integration Testing	Anu	Pending	0.5	0.5	1		
Change Email	Unit Testing Scripts	Anjali	Done	0.5	0	0		
	Business Logic	Anjali	Done	0.5	0	0		
	UI Test Scripts	Anu	Done	0.5	0	0		
	Front End Screens	Anjali	Done	0.5	0	0		
	Integration Testing	Anu	Pending	0.5	0.5	0.5		
Help Link	Front End Screens	Anjali	Pending	0.5	0.5	0.5		
	Manual Testing	Anu	Pending	0.5	0.5	0.5		
	Integration Testing	Anu	Pending			0.5		
Work Remaining				21	11	6		



## Les tests

Prévoir les tests d'acceptation dès la planification du sprint

Les scénarios de tests permettent :

- De comprendre les Stories
- De préparer la recette de sprint

Les tests sont effectués en cours de sprint, pas à la fin.

## Déroulement du sprint

Chaque développeur s'approprie des tâches des User Stories de l'itération

- Premières tâches attribuées à la réunion de sprint
- Ensuite, au cours des réunions quotidiennes

Tous les matins, réunion café/standup meeting/scrum meeting pour débloquer les problèmes et mesurer l'avancement

Au bout de l'itération, seules les User Stories complètes sont livrées

### *Le Scrum Daily Meeting*

Faire le point sur les tâches depuis le dernier Scrum meeting

S'attribuer de nouvelles tâches

Organiser le travail de la journée en cas d'obstacle (besoin d'expertise, de travailler à 2, problèmes de serveurs...)

### *Daily Meeting : les principes*

Tous les matins

Pas plus d'1/4 heure

Personne ne dirige la réunion, même si le Scrum Master peut l'animer

Tout le monde participe

- Équipe (y compris Scrum Master)
- Product Owner au moins quelques fois par semaine

Utilisation et mise à jour du tableau blanc

L'équipe peut faire appel à des experts

D'autres personnes peuvent y assister mais n'interviennent pas

## *La notion de « fini » ou « done »*

Définir dès le départ ce que veut dire « fini » :

*Les stories :*

- Est-ce que ça inclut la documentation?
- Est-ce que ça inclut des tests unitaires?
- Est-ce que ça inclut des tests d'intégration/croisés?

*La version*

*Une story en particulier :* chaque story ne nécessite pas le même travail

- Permet d'aborder la notion de « portée »

## **La revue de sprint**

### **Les principes**

A lieu le dernier jour du sprint

Durée maximum : de 2 à 4 heures

Prend en général la forme d'une démonstration :

- Build avec les stories terminées
- Idéalement faite par le Product Owner ou un membre de l'équipe.

### **La préparation**

Au minimum, toute l'équipe y compris Product Owner et Scrum Master

Parfois les autres personnes intéressées

- Marketing/commercial
- Support
- Éventuellement clients ou partenaires

### **Le contenu**

Le Product Owner émet des demandes de modification et recueille les feedbacks des participants

Il mettra ensuite à jour le Product Backlog et le Plan de Release, qui serviront à la planification du sprint suivant

Les demandes de changement et les bugs sont priorisés et pas forcément pris en compte dans le sprint suivant

### **La rétrospective**

Revenir sur le déroulement du sprint pour optimiser l'organisation

Réunion suite à la réunion de fin de sprint pour faire un bilan intermédiaire

- Qu'est-ce qui s'est bien passé?
- Qu'est-ce qui s'est mal passé?
- Comment nous améliorer?

Idéalement, brainstorming

Choisir une amélioration pour le sprint à venir.

## Résumé des rôles

### Actions du **Scrum Master**

- Veiller à ce que les pratiques Scrum soient appliquées
- Encourager l'équipe et le Product Owner et les inciter à devenir autonomes
- Protéger l'équipe des obstacles/interférences en cours de sprint
- Organiser et animer les réunions « Le Scrum Master est au service de l'équipe »

#### **Qualités :**

- Bonne connaissance de Scrum
- Comprend les aspects techniques
- Communication Bon guide, fait confiance
- Médiateur
- Tenace
- Transparent
- Au service de l'équipe
- Sait prendre des risques

### Actions du **Product Owner**

- Participe aux réunions
- De début de sprint
- Quotidiennes, parfois
- De fin de sprint
- À la rétrospective
- Est responsable du Backlog de Produit
- Répond aux questions sur le produit
- Définit les tests d'acceptation
- Passe ou fait passer ces tests

#### **Qualités :**

- Bonne connaissance du domaine métier
- Maîtrise des techniques de définition de produit
- Capacité à prendre des décisions rapidement
- Capacité à détailler quand il le faut
- Ouvert au changement...mais sans changer d'avis tout le temps
- Aptitude à la négociation
- Disponible pour le rôle

### L'équipe

- Multi-disciplinaire
- Esprit d'équipe
- Pas d'élément perturbateur
- Mieux vaut un correct niveau moyen que des stars individuelles

## Références

- « Scrum », par Claude Aubry, DUNOD
- <http://www.agiliste.fr/fiches/guide-demarrage-scrum/>
- <http://www.agiletour.org/>
- Cas pratique : <http://henrik-kniberg.developpez.com/livre/scrum-xp/>  
<http://www.scrum.org/>
- <http://www.infoq.com/minibooks/kanban-scrum-minibook>  
<http://blog.octo.com/index.php/2008/01/25/69-pourquoi-les-methodesagiles-peinent-elles-a-penetrer-lentreprise>
- Jeux agiles: <http://tastycupcakes.org>