
GPI - Introduction à Scrum

Pierre Bouillon - 02/10/2021

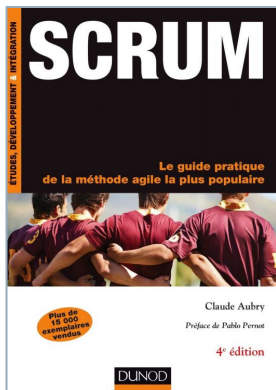
GPI - Introduction à Scrum

Pierre Bouillon

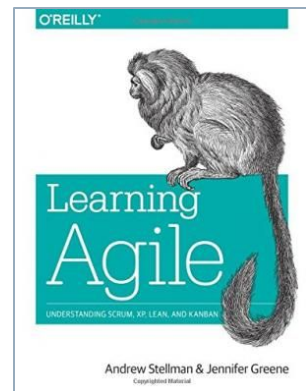
- ▷ Mail: pierre.bouillon@telecomnancy.eu
- ▷ LinkedIn: [in/pierre-bouillon](https://www.linkedin.com/in/pierre-bouillon)



Bibliographie



- ▶ Claude Aubry, *SCRUM - Le guide pratique de la méthode agile la plus populaire*, Paris, DUNOD, 2015 (et 2019)
- ▶ Andrew Stellman & Jennifer Greene, *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*, USA, O'REILLY, 2015



Meta

Avant d'aborder le cours



Vous connaître

A la fin de ce module vous devriez savoir ...

- ▷ Différencier **agilité**, **Scrum** et *Waterfall*
- ▷ **Intégrer une équipe agile**
- ▷ Connaître les **valeurs de l'agilité**
- ▷ Connaître les **rôles de Scrum** et leurs importances
- ▷ Reconnaître et éviter les **antipatterns**
- ▷ Mettre en place les **rituels de Scrum**
- ▷ Utiliser les bons **outils** (qui vous conviennent)

A la fin de ce cours vous devriez connaître ...

- ▷ L'origine des méthodes agiles
- ▷ L'origine de Scrum
- ▷ La différence entre agilité, Scrum et *Waterfall*
- ▷ La notion de sprints
- ▷ La notion de saisons
- ▷ Les rôles et valeurs d'une équipe Scrum



Votre rôle

Déroulement

- ▷ *Meta*
 - Vous connaître
 - Ce que l'on verra
 - Votre rôle
- ▷ Lab #0 | 🥚 🎨
- ▷ L'agilité
 - La mouvance agile
 - Origines
 - Principes
- ▷ Lab #1 | 🔪 🥘

Déroulement

- ▷ Scrum
 - Introduction
 - En bref
 - Place dans l'agilité
 - Les sprints
 - L'organisation
 - Les rôles
 - L'écosystème



Lab #0 |  

Les oeufs de Pâques

1.

L'agilité

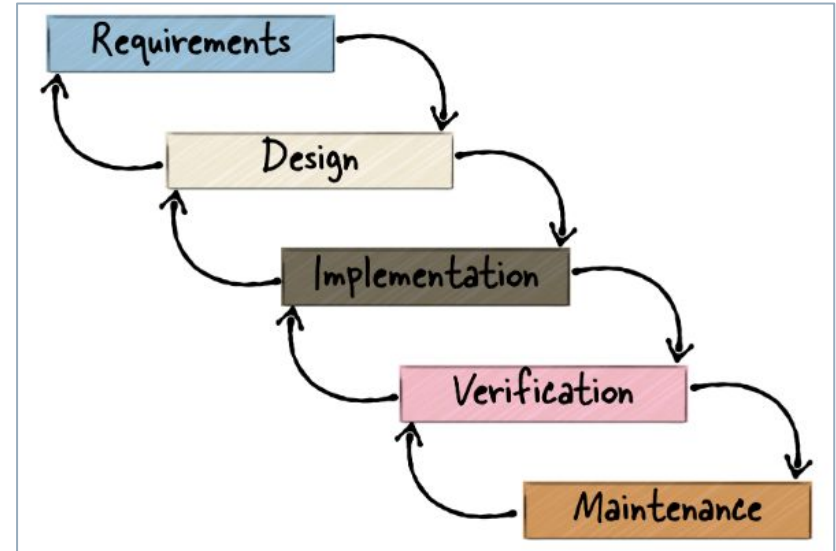
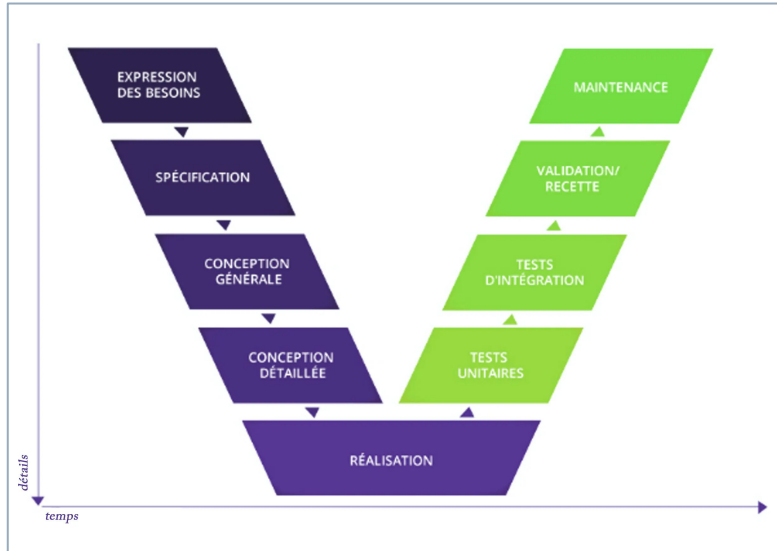
Origine, philosophie et rupture

1.1.

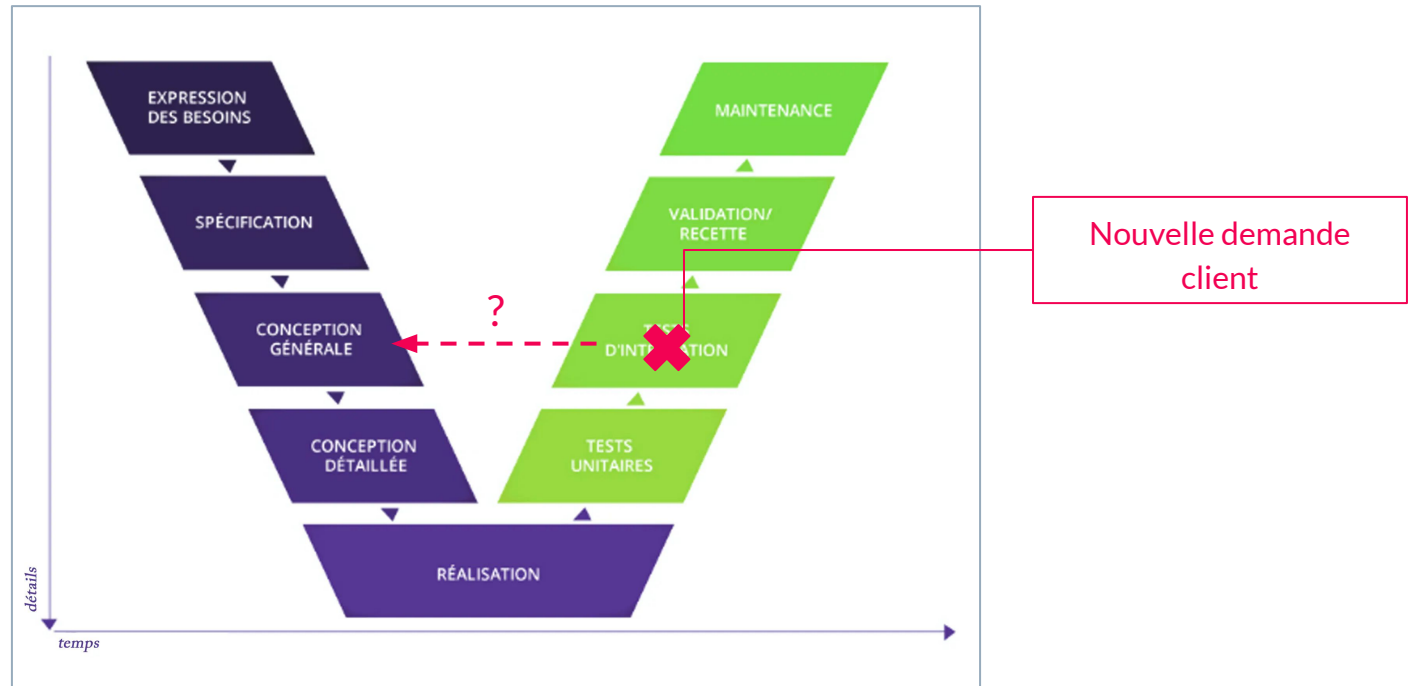
La mouvance agile

Comprendre l'origine

Le cycle en V / *Waterfall*



Le cycle en V : réponse au changement



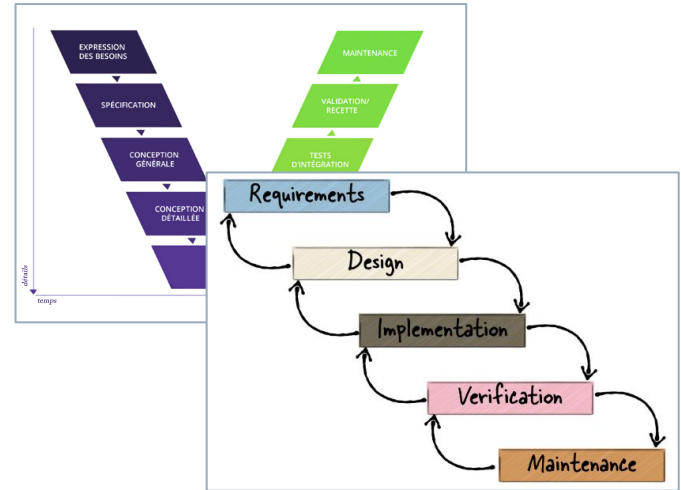
Le problème



Le cycle en V / *Waterfall*

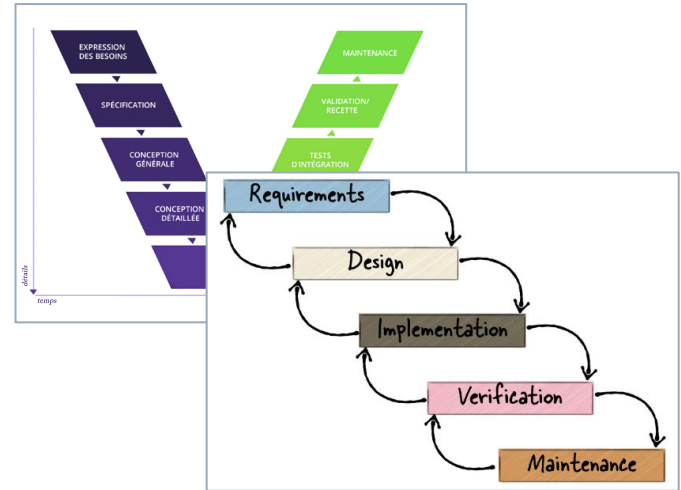
- ▷ Si rigoureusement appliquée nécessite:
 - 100% de spécification avant la conception
 - 100% de conception avant la réalisation
 - 100% de la réalisation avant les tests

... En cas de problèmes ?



Le cycle en V / *Waterfall*: constat

- ▷ Souvent hors budget ou dans l'urgence
- ▷ Équipe tendue, déprimée
- ▷ Maintenabilité complexe
- ▷ Bugs et anomalies en nombre



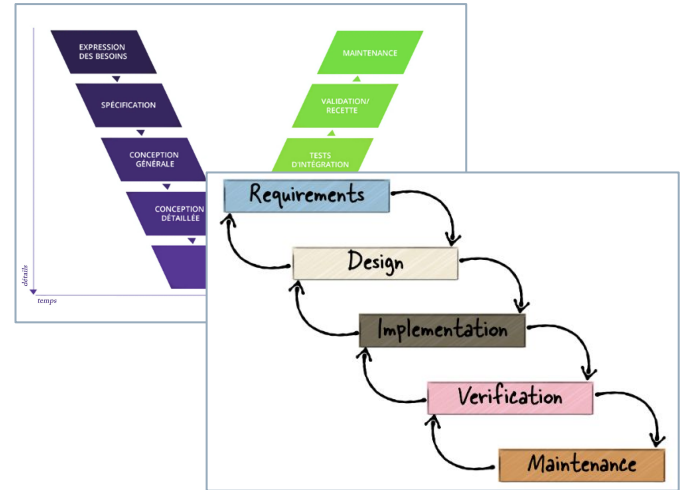


L'agilité contre l'inertie

Le cycle en V / *Waterfall* : disclaimer

- ▷ Le *Waterfall* **peut** fonctionner
 - Historiques de **certains projets réussis**:
 - Emphase de la **communication**
 - Attention particulières aux **bonnes pratiques**

⇒ Des **valeurs de l'agilité**



1.2.

Les principes

El famoso Agile Manifesto

L' Agile Manifesto

- ▷ “Agile Manifesto”, 2001 (en ligne: <https://agilemanifesto.org/>)
 - 4 valeurs
 - 12 principes



Retour sur les valeurs



Pour éviter ce genre de situations

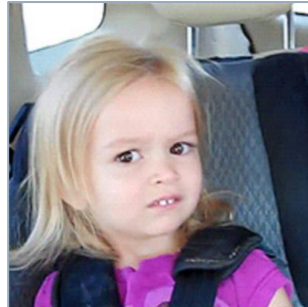
Sous-texte



Nous reconnaissons **la valeur
des seconds éléments**, mais
privilégions les premiers.

Retour sur les valeurs

- ▷ Les individus et leurs interactions plus que les processus et les outils
- ▷ Des logiciels opérationnels plus qu'une documentation exhaustive
- ▷ La collaboration avec les clients plus que la négociation contractuelle
- ▷ L'adaptation au changement plus que le suivi d'un plan



Retour sur les valeurs



Les individus et leurs
interactions plus que les
processus et les outils

Retour sur les valeurs



Des logiciels opérationnels plus
qu'une documentation
exhaustive

Retour sur les valeurs



La collaboration avec les clients
plus que la négociation
contractuelle

Retour sur les valeurs



L'adaptation au changement plus
que le suivi d'un plan

En résumé



L'agilité est la capacité à **favoriser le changement et à y répondre** en vue de s'adapter au mieux à un **environnement turbulent**

- Jim Highsmith

Différences avec les approches traditionnelles

- ▷ **Empirique** → L'adaptation au changement
- ▷ **Rythmé** → Des logiciels opérationnels
- ▷ **Priorisé** → La collaboration avec les clients
- ▷ **Autogéré** → Les individus et leurs interactions
- ▷ **Transparent** → La collaboration avec les clients
- ▷ **Collaboratif** → La collaboration avec les clients
Les individus et leurs interactions

Le début des problèmes

- ▷ “On a notre propre version de Scrum”
→ Pas Scrum
- ▷ “Bien sûr qu’on est agile”
→ Scrum est devenu *mainstream* et marketing
- ▷ “Puisqu’on est en agilité, vous devriez pouvoir rajouter ça”
→ Pas une méthode pour l’urgence
- ▷ “On applique Scrum, il faut changer X”
→ Toujours soumis à des règles



La raison

@blindXfish

Sorry for the so many jobs I didn't applied because I thought Scrum is a js framework XD omg... Actually... omg.



 devrant.com



2.

Scrum

Pratiques, méthodologies et cadres (?)

2.1.

Introduction

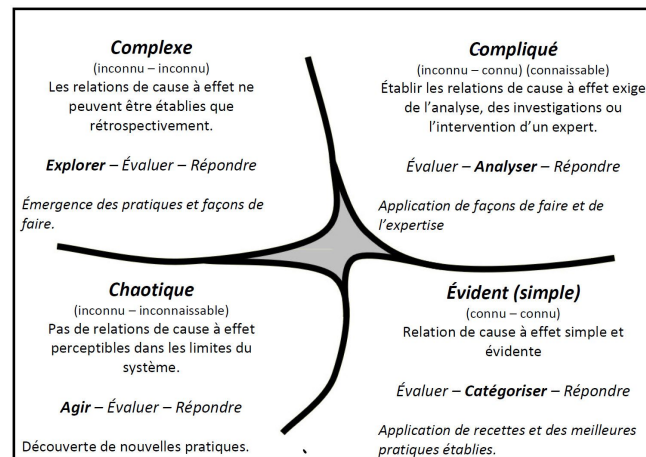
Le situer et l'appliquer

Mais Agile != Scrum ?

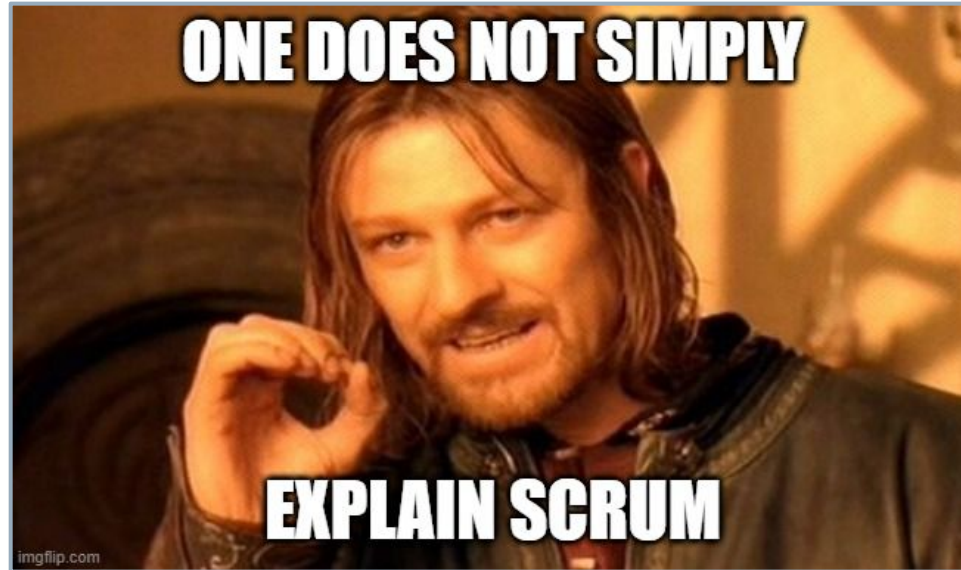
- ▷ **Pratiques** agiles != **Méthodes** agiles
- ▷ Pratique
 - Approche concrète et éprouvée
 - Résout un ou plusieurs problèmes
 - Améliore la manière de travailler
- ▷ Méthode
 - Cadre regroupant un ensemble de pratiques
 - Cohérent avec lui même

Appliquer Scrum

- ▷ Savoir quand il est pertinent de choisir Scrum
 - *Complexe* d'après le modèle Cynefin ?
 - Culture du **contrôle** ?
 - Travail dans l'**urgence** ?
 - **Participants** mobilisés ?
- ▷ Éviter la “loi du marteau”
- ▷ **Scrum n'est qu'un cadre**
⇒ Accompagner des pratiques du domaine visé



Oui mais concrètement ?

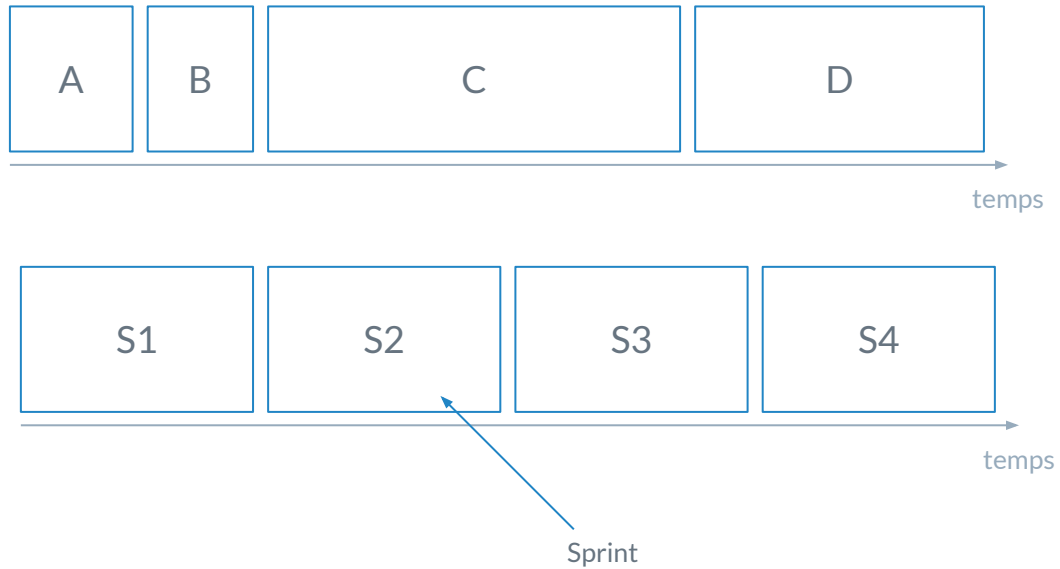


2.2.

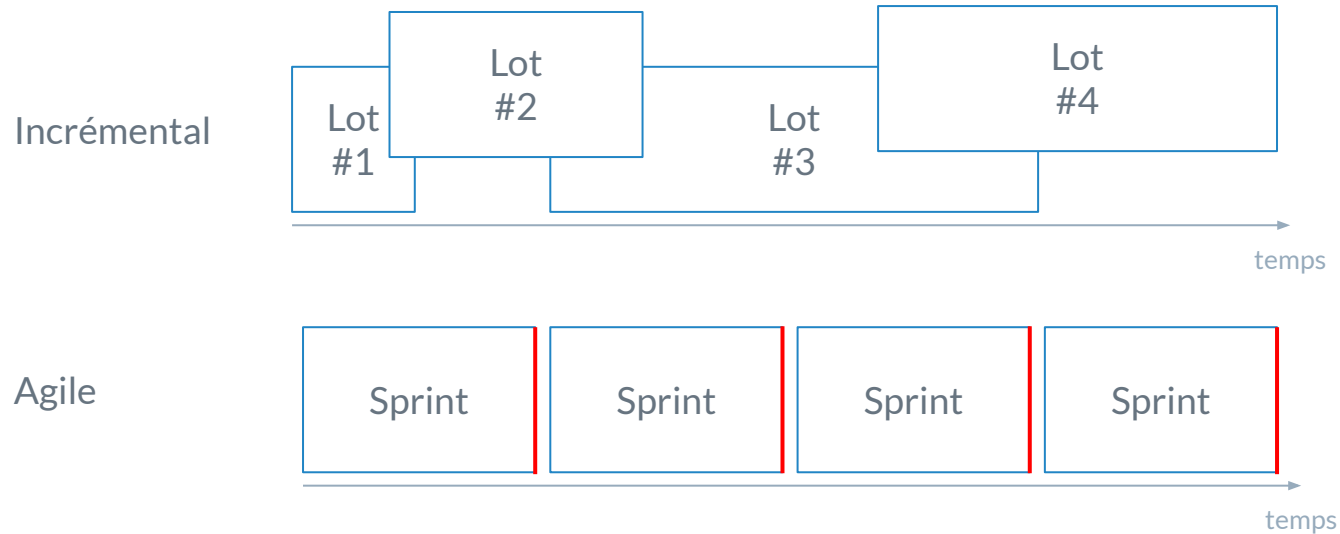
Les sprints



Le sprint

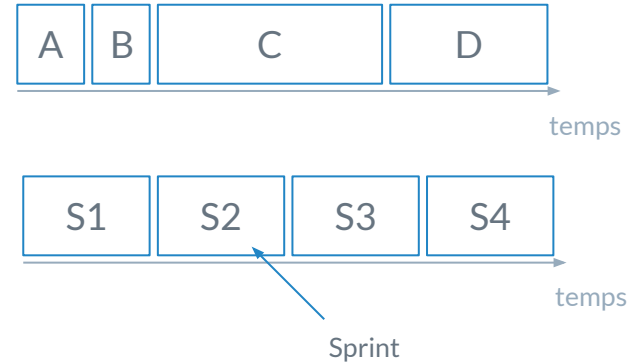


La rigidité du sprint



Le but

- ▷ Une approche **itérative** et **incrémentale**
 - **Empirisme** pour la planification
 - Encouragement et repose sur le *feedback*
- ▷ Vers un **développement durable**
(*sustainable path*)



Le contenu du sprint

- ▷ Structuré mais pas prérempli
 - Libre des activités qu'il contient
- ▷ Structure
 - **Spécifications**
 - **Conception**
 - **Programmation**
 - **Tests**





Une fin de sprint n'est pas
une release !

Les saisons



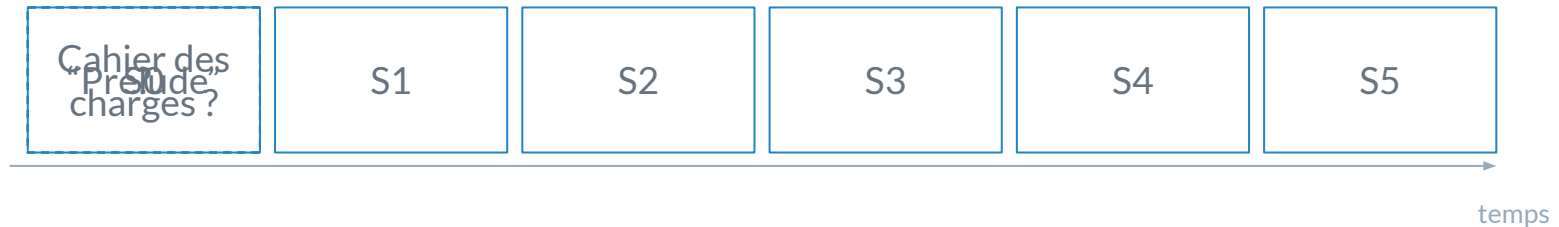


Et ce fameux sprint zéro
alors ?

~~Le sprint zéro~~

- ▷ Alimenter la charge de travail
- ▷ Identifier les acteurs
- ▷ Préparer le fonctionnement de l'équipe

⇒ Pas un sprint



Les événements du sprint (preview)

- ▷ Quatre principaux événements
 - Planifier → Sprint planning
 - Suivre → Daily meeting
 - Présenter → Review de sprint
 - Prendre du recul → Rétrospective de sprint

- ▷ Loin du peu de planification que l'on entend

Vue d'ensemble

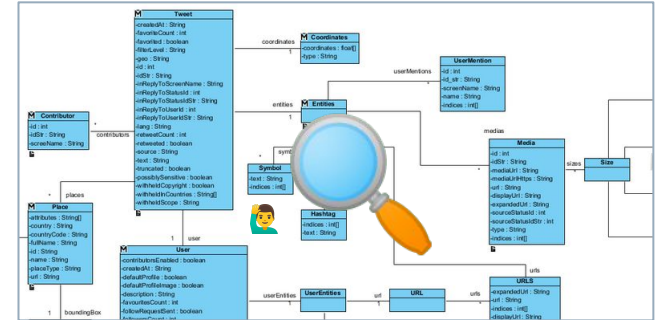


2.3. L'organisation

L'équipe et l'humain au coeur de l'agilité

“Les individus et leurs interactions plus que les processus et les outils”

- ▷ Principale rupture avec les autres outils
 - UML, cycle en V, BPMN, etc.

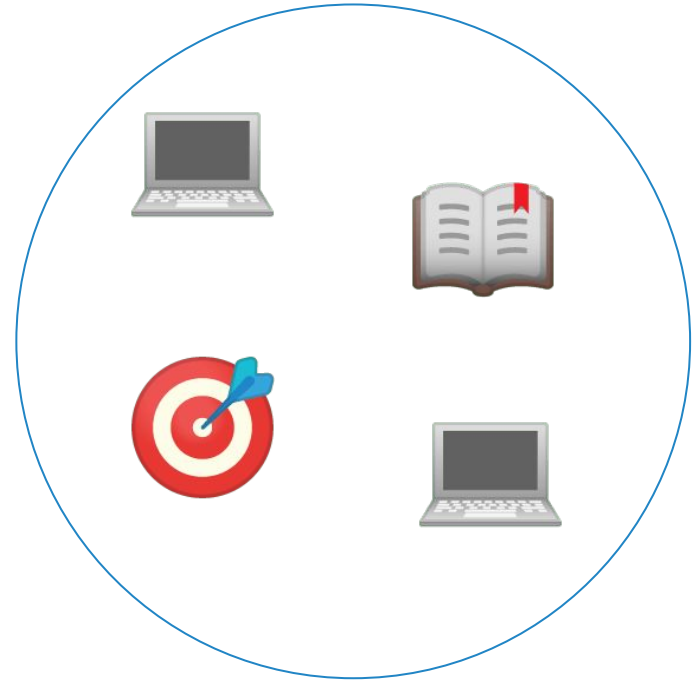
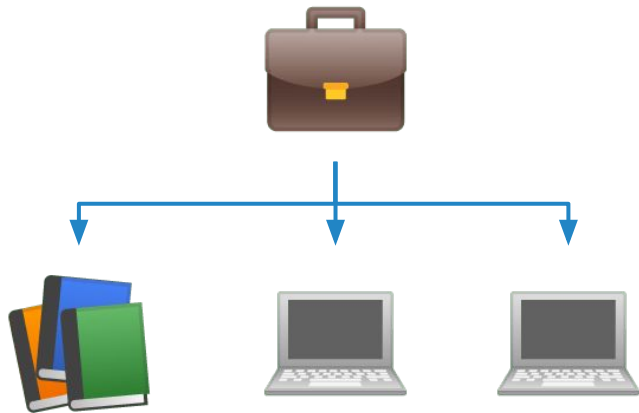


⇒ Rétablir l'équipe au centre de la création de valeur

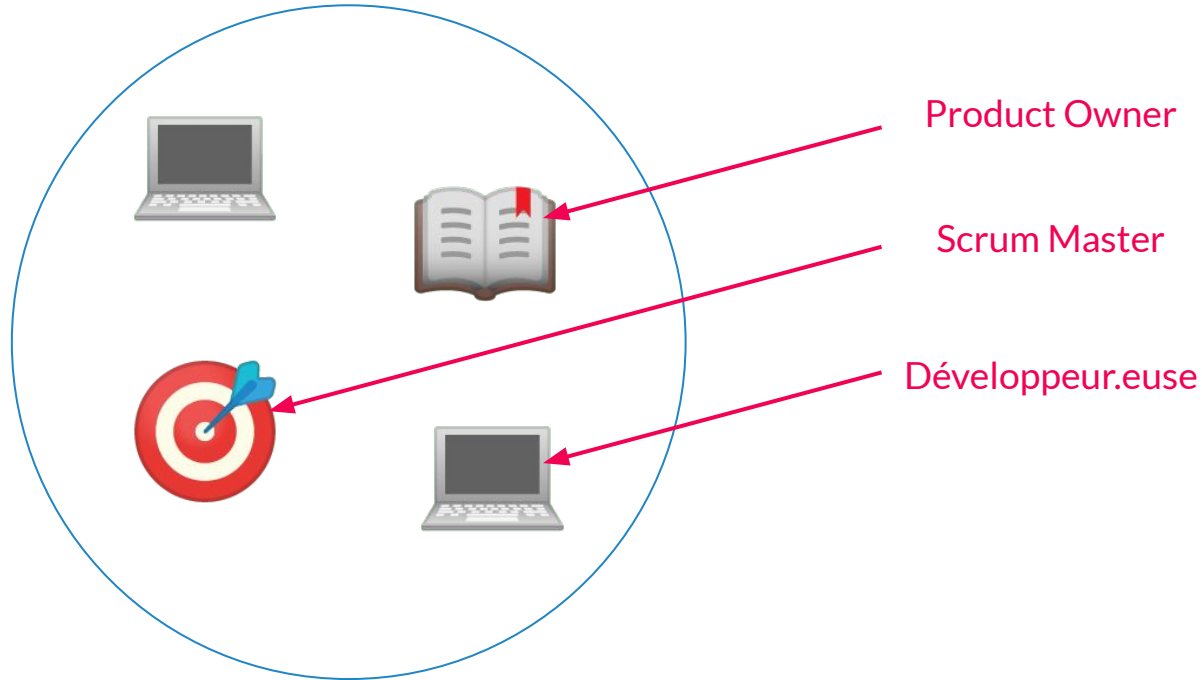


C'est bien joli, mais
comment ?

Changer les règles



Changer les règles



La bonne équipe

▷ 7 personnes + - 2 (idéalement)

▷ 5 Piliers

- Auto-gérée
- Pluridisciplinaire
- Volontaire
- Stable
- Identité

⇒ Synergie, **une équipe et non un groupe**

Les valeurs de l'équipe

- ▷ 5 valeurs
 - Focus
 - Ouverture
 - Respect
 - Courage
 - Engagement
- ▷ Optionnellement
 - Responsabilité
 - Humour
 - Celle(s) qui correspond(ent) à votre équipe

2.3.1

Les rôles

Le développeur

- ▷ Responsable de ce qu'il produit devant l'équipe
- ▷ Décision commune de ses tâches
 - Pas de directives
 - Auto-organisation
- ▷ Possibilité d'un lead technique (mais optionnel)
 - Le *leadership* est un comportement, pas un rôle

Le développeur - Aptitudes

- ▷ Motivé
 - S'impliquer dans l'auto-organisation
 - Tenir aux autres et au produit
 - “Jouer le jeu”
- ▷ Pluridisciplinaire
 - Permettre à l'équipe d'avancer
- ▷ Volontaire
 - La fin de la hiérarchie n'est pas faite pour ne plus s'impliquer

Le Product Owner

- ▷ Responsable du produit
- ▷ En charge des décisions tactiques **du produit**
 - Formalise les besoins
 - Les priorise, débat, etc.
- ▷ Aide l'équipe à comprendre les besoins

Le Product Owner - Aptitudes

- ▷ Connaît le métier / le client
 - A même d'identifier les besoins
 - Capable de concrétiser une pensée
 - Peut jauger la valeur ajoutée d'une idée
- ▷ Motivé
- ▷ Disponible
 - Doit être en mesure de décider
- ▷ Impliqué
 - Pas / peu de fonctionnalités
 - Produit inutile ou “bancal”

Le Scrum Master

- ▷ **Garant** de Scrum (et pas *maître*)
 - Veille à son suivi
 - Accompagne, ne juge pas
- ▷ Facilitateur, **pas manager !**
 - Un problème se comprend avant de se régler
 - Pas d'autorité pour une action (renvoi, ou autre)
- ▷ Ne nécessite **pas une mobilisation à 100%**
 - Accompagnement seulement, support
 - Dépend de la maturité de l'équipe Scrum
 - Souvent un *lead*

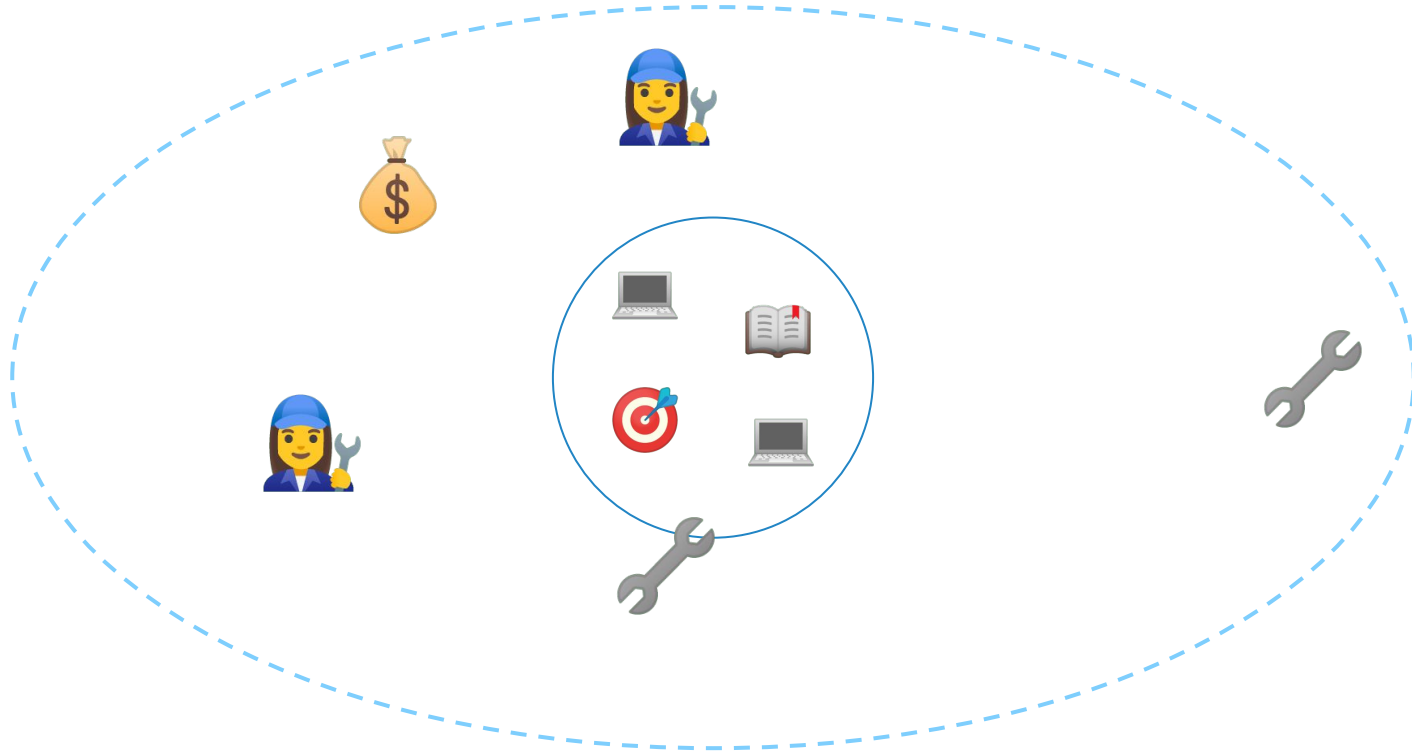
Le Scrum Master - Aptitudes

- ▷ Bonne **connaissance de Scrum**
 - Ou volontaire pour
- ▷ **Compréhension** du technique et du fonctionnel
- ▷ Bon **communiquant**, guide, **médiateur**
 - Aide à l'application, n'impose pas
 - Négocie, élimine les obstacles
- ▷ Transparent et dévoué à l'équipe
- ▷ Tenace

2.3.2

L'écosystème

L'écosystème



Les parties prenantes

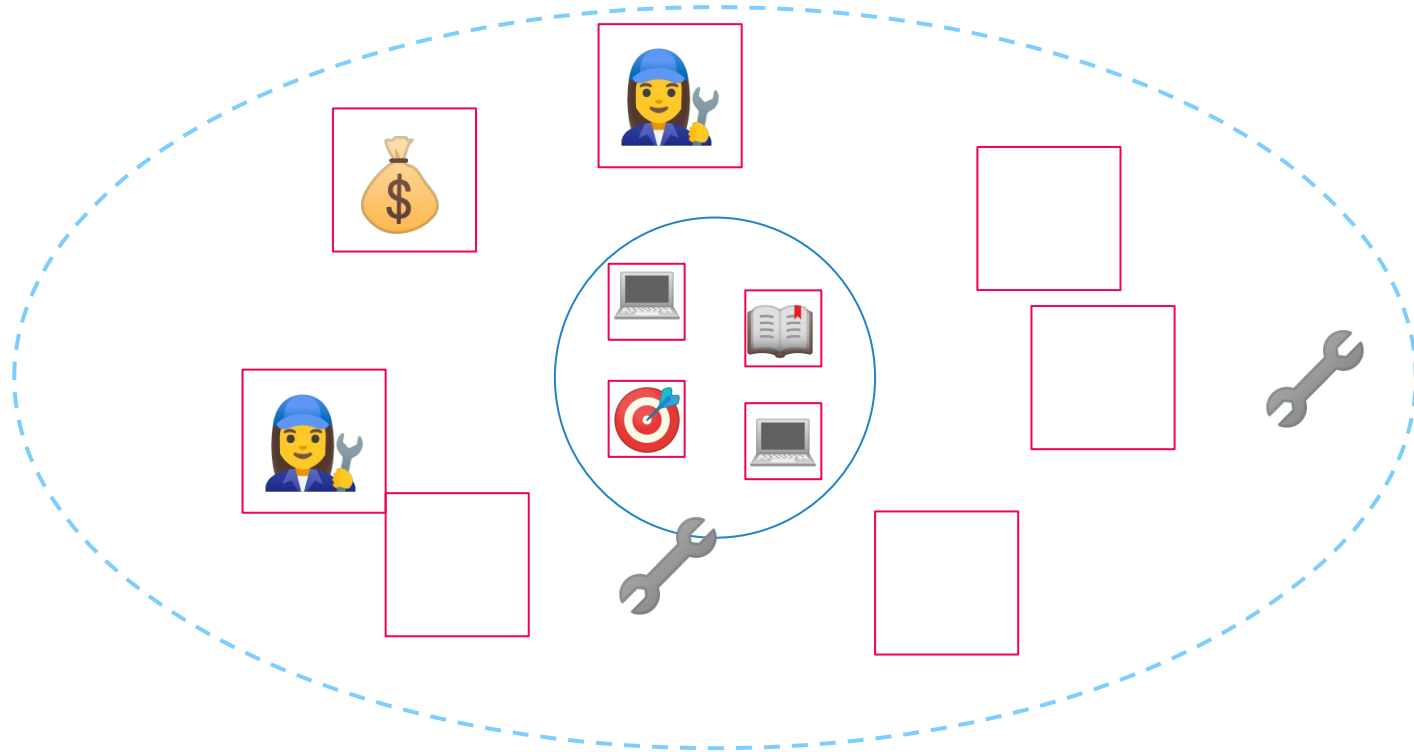


Une partie prenante est une
personne intéressée par les
résultats de l'équipe

L'intérêt des parties prenantes

- ▷ Amélioration du produit
 - ▷ Découverte de fonctionnalités, de besoins
 - ▷ Optimisation de la valeur ajoutée
- ⇒ Feedback des utilisateurs réels (et plus)

Identifier les parties prenantes



Les experts

- ▷ Parties prenantes à part
 - Expertise localisée, externalisée
 - Appui ponctuel
- ▷ Privilégier la pluridisciplinarité à une ressource tierce



... Et du coup ?



Lab #1 |  

Identifier les gens

