

---

# Méthodes agiles

Franck Silvestre <franck.silvestre@irit.fr>

Cette oeuvre est mise à disposition selon les termes de la Licence Creative Commons Paternité - Pas d'Utilisation Commerciale 3.0 non transposé.

## Table of Contents

Préambule .....	1
Objectifs .....	1
Évaluation .....	1
Les principaux modèles de développement logiciel .....	2
Modèle en cascade .....	2
Cycle en V .....	3
Modèle en spirale .....	3
Unified Process .....	4
Les méthodes agiles .....	5
Scrum .....	6
Introduction .....	6
Définition de Scrum .....	6
The big picture .....	7
L'équipe Scrum .....	7
Les événements Scrum .....	8
Les artefacts de Scrum .....	9
Annexes .....	13
Un regard critique .....	13
Agile ! The good, the hype and the ugly .....	13

## Préambule

### Objectifs

- Connaître le contexte d'émergence des méthodes agiles
- Connaître les valeurs et principaux principes du manifeste agile
- Connaître la méthode Scrum et savoir mettre en pratique Scrum dans le contexte d'un projet
- Connaître les grands principes d'autres méthodes agiles : XP et Kanban

### Évaluation

- 1 examen écrit individuel (70%)
- 1 projet réalisé en groupe (30%)

# Les principaux modèles de développement logiciel

## Modèle en cascade

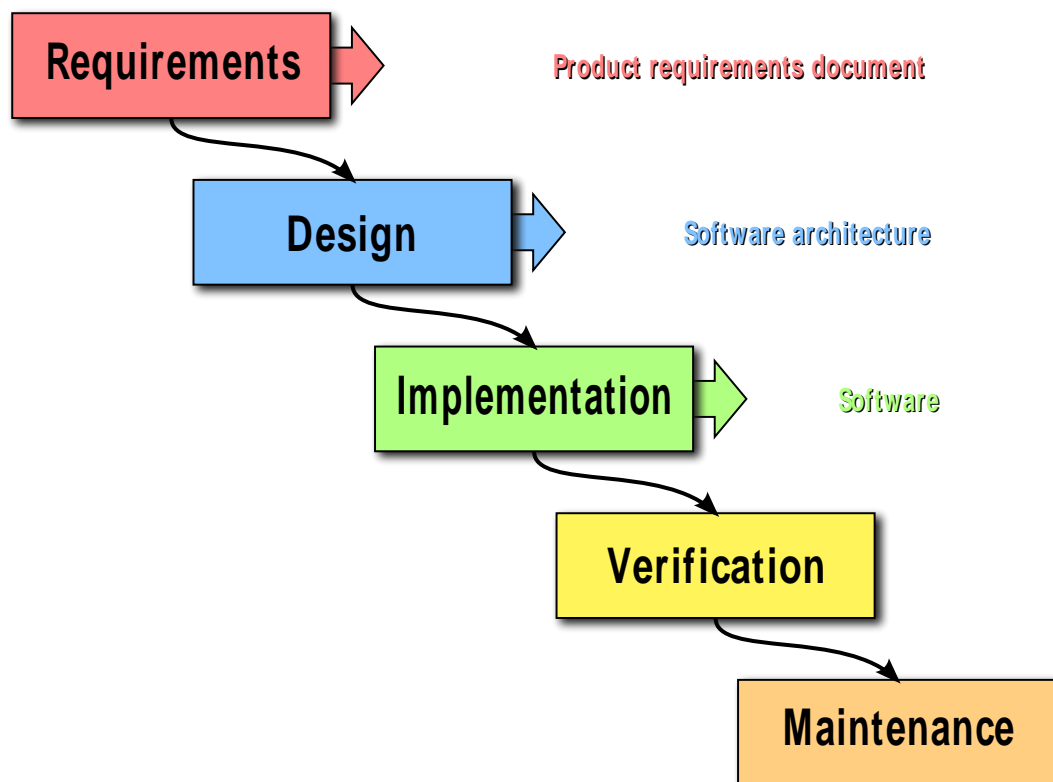
### Origine et contexte

- 1970 - Article de Winston Royce "Managing the Development of Large Software Systems"
- Projets de grande envergure et contractualisés selon les contraintes fortes du gouvernement
- Inspiré des modèles de construction de matériel

### Principes

- Approche prédictive et linéaire
- Séquence de phases qui s'enchainent successivement

### The big picture



### What's wrong ?

- Pas de tolérance au changement
- Effets tunnels induits

- Modèle adapté uniquement pour des projets où toutes les incertitudes sont levées durant la première phase
- Portage d'une application à l'identique
- Interfaces bas niveau

## Cycle en V

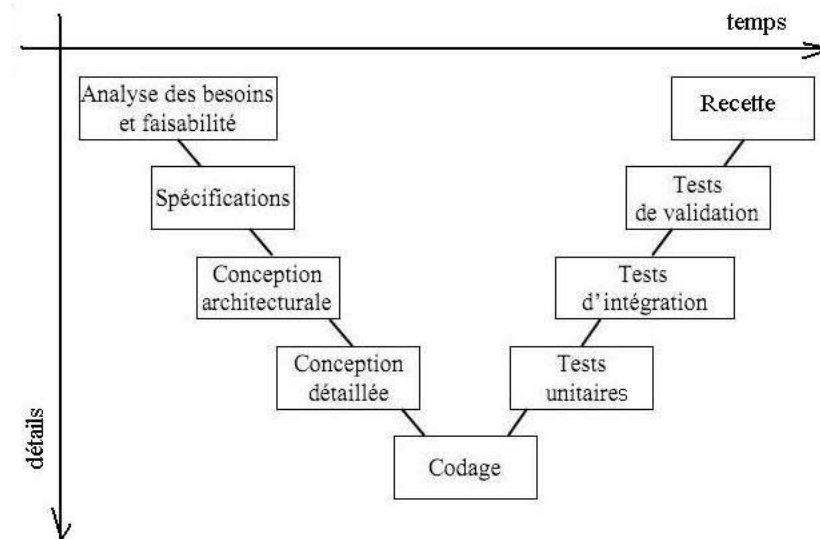
### Contexte et origine

- Amélioration du modèle en cascade
- Vise à réduire (légèrement!) l'effet tunnel
- Standard dans l'industrie (pas que logiciel) depuis les années 80

### Principes

- Approche linéaire et prédictive
- Séquence de phases qui s'exécutent successivement
- Mais les phases de vérifications permettent de réduire l'effet tunnel

### The big picture



### What's wrong ?

Mêmes choses que pour le modèle en cascade...

## Modèle en spirale

### Contexte et origine

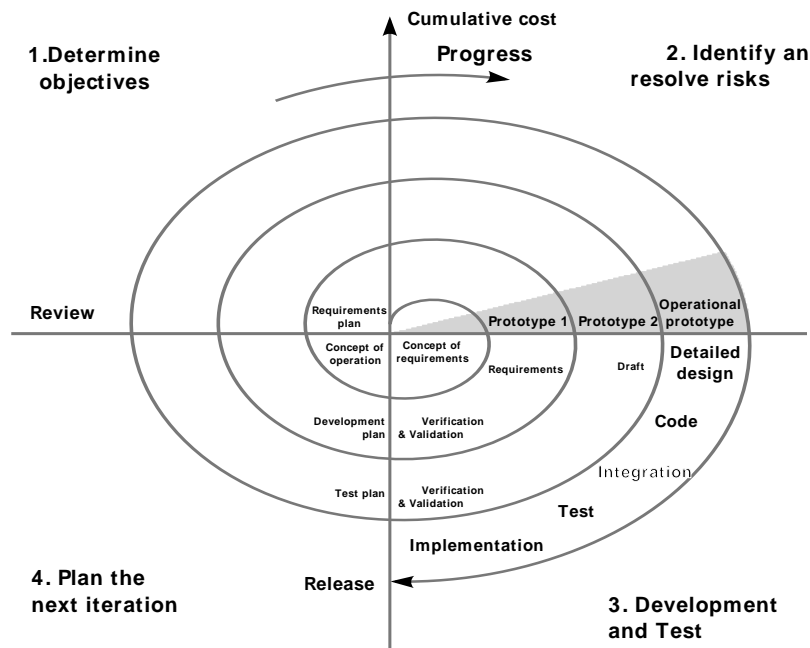
- 1986 - Article de Barry Boehm "A Spiral Model of Software Development and Enhancement"

- Constat d'échec du modèle waterfall sur les projets "guidés par les documents" dans le cas de projets où les besoins utilisateurs sont difficiles à exprimer, changent,...

## Principes

- Approche non linéaire, itérative
- Guidée par les risques
- Succession de construction de prototypes pour arriver au prototype opérationnel

## The big picture



## Unified Process

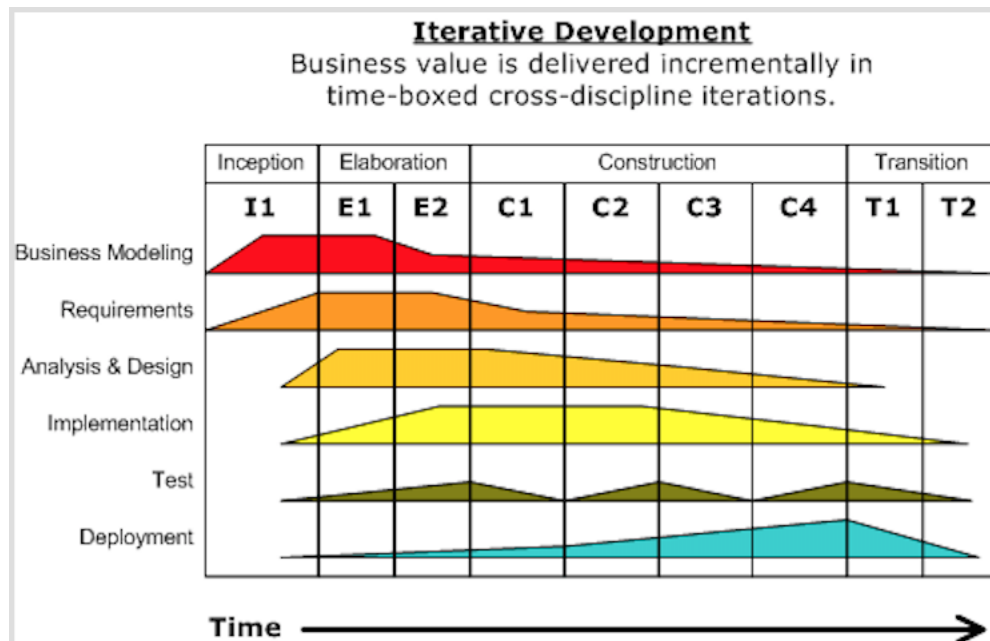
### Contexte et origine

- 1999 - Ivar Jacobson, Grady Booch and James Rumbaugh - The Unified Software Development Process (ISBN 0-201-57169-2)
- Large adoption du standard UML et du développement en Programmation Orientée Objets

## Principes

- UP décrit un processus extensible, itératif et incrémental
- Des phases et des activités
- Des itérations dans chaque phase
- Chaque itération reprend l'ensemble des activités à des degrés différents

## The big picture



## Les méthodes agiles

### Contexte et origine

- Développements d'une multitude de méthodes de développement itératives et incrémentales (DII)
- 2001 - 17 experts représentant DSDM, XP, Scrum, FDD et autres méthodes de DII - Agile Alliance ([www.agilealliance.org](http://www.agilealliance.org))
- 2001 - <http://agilemanifesto.org>

### Valeurs

Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire. Ces expériences nous ont amenés à valoriser :

- . 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.

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

Extrait du manifeste agile

### Quelques principes parmi les 12

- . Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.

- . Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- . Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
- . Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
- . Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées.
- . À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

# Scrum

## Introduction

- Scrum a été défini et développé par Ken Schwaber et Jeff Sutherland
- Le guide Scrum est la référence officielle
- La suite du contenu est principalement extraite du guide Scrum
- Scrum est un cadre de processus qui a été utilisé pour gérer le développement de produits complexes depuis le début des années 1990.

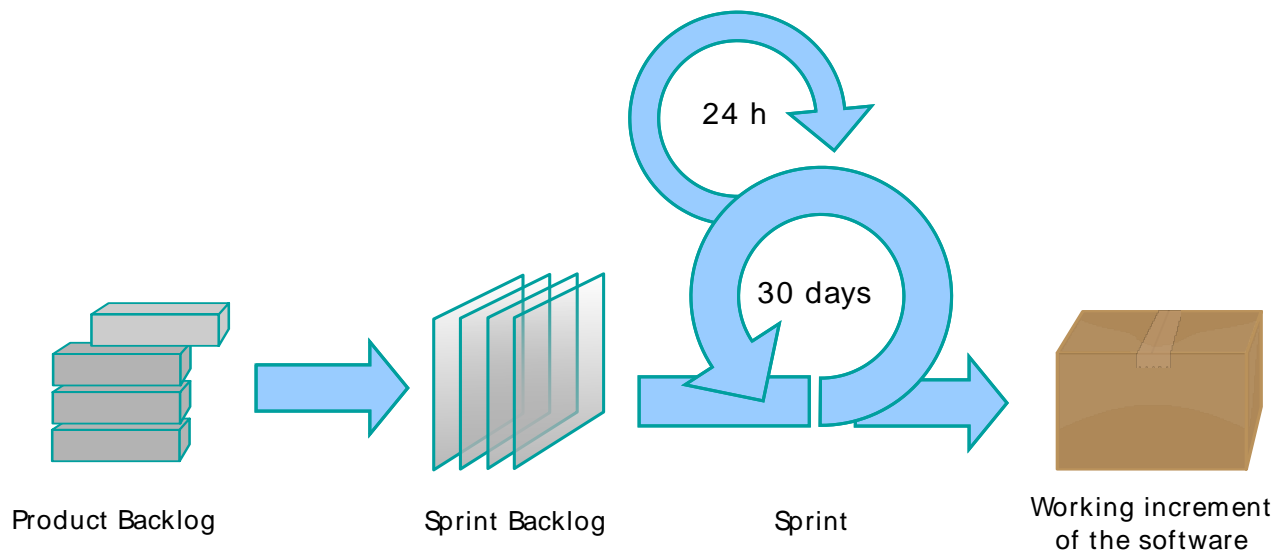
## Définition de Scrum

Scrum: Un cadre adaptable dans lequel des personnes peuvent résoudre les problèmes complexes, tout en délivrant de manière productive et créative des produits de la plus haute valeur possible.

Scrum est:

- Léger
- Simple à comprendre
- Difficile à maîtriser

## The big picture



## L'équipe Scrum

### L'équipe Scrum

- Le Product Owner
- L'équipe de développement
- Le Scrum Master

Les équipes Scrum délivrent des produits de manière itérative et incrémentale, maximisant les opportunités d'obtenir du feedback.

Les Livraisons incrémentales du produit "fini" ("done") assurent qu'une version potentiellement utilisable du produit est toujours disponible.

### Le Product Owner

- Le Product Owner est chargé de maximiser la valeur du produit et le travail de l'équipe de développement.
- Le Product Owner est le représentant du client et des parties prenantes au sein de l'équipe
- Il gère le "Backlog" de produit qui contient toutes les exigences du produit

### L'équipe de développement

- L'équipe de développement fournit un incrément potentiellement livrable du produit "fini" à la fin de chaque Sprint (itération).

- L'équipe de développement est autoorganisée

## Le Scrum Master

- Il est le garant de la bonne mise en oeuvre du process Scrum
- Process Owner
- Doit bien maîtriser le processus pour accompagner le reste de l'équipe dans sa bonne mise en oeuvre

## Les évènements Scrum

### Les évènements Scrum

- Le Sprint
- La réunion de planification de Sprint
- La daily Scrum
- La revue de Sprint
- La rétrospective de Sprint
- La revue de backlog

## Le Sprint

- Un Sprint correspond à une itération "time-boxée" : une période d'un mois ou moins au cours de laquelle un incrément de produit "fini", utilisable, et potentiellement livrable est créé.
- Un Sprint est caractérisé par un Objectif qui donne le cap du Sprint en cours
- Aucune modification n'est apportée qui pourrait mettre en danger l'atteinte de l'objectif du Sprint
- Un Sprint commence avec la réunion de planification et se termine avec sa rétrospective

## La réunion de planification de Sprint

- La réunion de planification de sprint répond aux questions suivantes :
  - Qu'est-ce qui sera livré dans l'incrément résultant du prochain sprint ?  
**Définition de l'objectif du Sprint et sélection des stories**
  - Comment le travail nécessaire pour réaliser l'incrément sera-t-il accompli ?  
**Découpage des items du backlog en tâches**
- La réunion est "time-boxée" - maximum 8h pour Sprint d'un mois
- Le Product Owner a autorité sur le "quoi"
- L'équipe de développement a autorité sur le "comment"



## Le daily Scrum

- Réunion quotidienne de l'équipe Scrum
- "Time boxée" - 15 minutes maximum
- Pendant la réunion chaque membre de l'équipe de développement répond aux questions suivantes:
  - Qu'ai je fait depuis hier qui a aidé l'équipe de développement à réaliser l'Objectif du Sprint ?
  - Que vais je faire aujourd'hui pour aider l'équipe de développement à réaliser l'Objectif du Sprint ?
  - Ai je identifié des obstacles qui empêchent l'équipe de développement de réaliser l'Objectif du Sprint ?

**Liste des obstacles à gérer et à faire diminuer**

## La revue de Sprint

- Une réunion de revue de Sprint est tenue à la fin du sprint pour inspecter l'incrément du produit et adapter le Backlog de produit si nécessaire.
- Pendant la revue de Sprint, l'équipe Scrum et les parties prenantes échangent sur ce qui a été fait durant le sprint.
- Cette réunion se veut informelle, la présentation de l'incrément est destinée à susciter des réactions (« feedbacks ») et à favoriser la collaboration.
- "Time-boxée" - 4 heures maximum pour un Sprint d'un mois

## La rétrospective de Sprint

- La rétrospective de sprint est une occasion pour l'équipe Scrum de s'inspecter et de créer un plan d'améliorations qui sera mis en place au cours du sprint suivant.
- La rétrospective de sprint survient après la revue de sprint et avant la prochaine réunion de planification de sprint.
- Pour un sprint d'un mois, cette rencontre est limitée à une durée de trois heures.

## La revue de backlog

- Permet à l'équipe de développement d'évaluer les items du backlog
- Se fait pendant le sprint en cours en préparation du sprint suivant
- Planning poker
- <http://www.aubryconseil.com/post/Les-reunions-d-un-sprint>

## Les artefacts de Scrum

### Les artefacts de Scrum

- Le Backlog de produit

- Le Backlog de Sprint
- Les éléments de suivi
- L'incrément

## Le backlog de produit

- Le Backlog de produit est une liste ordonnée de tout ce qui pourrait être requis dans le produit.
- Il est la source unique des exigences pour tous les changements à effectuer sur le produit.
- Le Product Owner est responsable de ce Backlog, de son contenu, de sa publication ainsi que de l'ordonnancement des items qu'il contient.
- En pratique
  - Items d'exigences fonctionnelles : "User Stories"
  - Items d'exigences techniques : "Technical Stories"
  - Autres exigences : qualité, "omniprésentes", référencées par les items du backlog

## User Story et INVEST

Indépendante

Négotiable

Apporter de la Valeur

Estimable

Small enough

Testable

## User Story format type

As a ...

I want to...

So that...

Exemple extrait de Tsaap-Notes :

As a owner of a scope

I want to duplicate a scope

So that I can replay questions to another group of users

## Tests d'acceptation

- Test fonctionnel

- Un ou plusieurs tests décrits pour chaque story
- Un format privilégié

Given...

When...

Then...

- Un des critères dans la définition de "fini" : les tests d'acceptance sont clairement définis et passent
- <http://guide.agilealliance.org/guide/gwt.html>

## Exercice

### **Exercice** *Tests d'acceptance*

Écrire un jeu de tests d'acceptance pour la story

"As a owner of a scope

I want to duplicate a scope

So that I can replay questions to another group of users"

## Feature

- Une feature est une fonctionnalité de plus haut niveau qu'une story
- Nécessite un raffinement, découpage en stories
- Niveau intermédiaire entre la vision produit et la story

## Epic (épopée)

- Dans XP : une grosse story
- Dans d'autres contextes (Scrum à grand échelle), un regroupement de features

## Le Backlog de Sprint

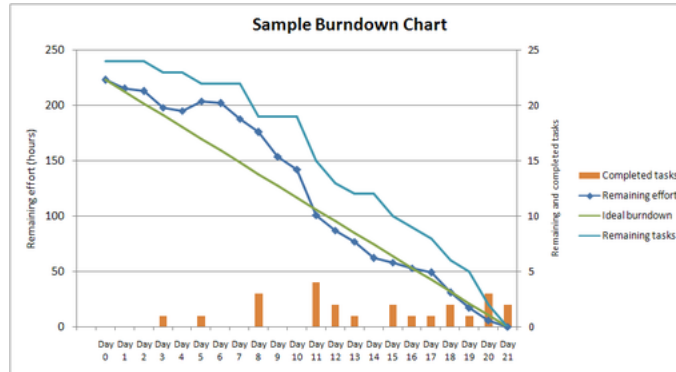
Le backlog de Sprint est l'ensemble des items choisis pour le Sprint accompagné du plan pour les réaliser dans le cadre d'un incrément de produit qui concrétisera l'objectif du Sprint.

## Les éléments de suivis

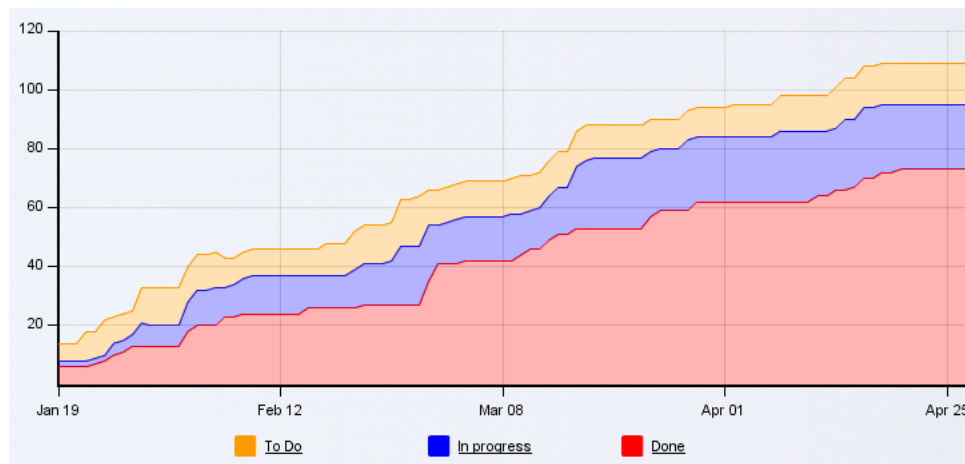
- Suivi de l'avancement calculable à tout moment
- Utilisation de diagrammes burn-downs, burn-ups, flux cumulés
- [http://en.wikipedia.org/wiki/Burn\\_down\\_chart](http://en.wikipedia.org/wiki/Burn_down_chart)

- La vélocité : nombre moyen de points réalisés au cours d'un sprint

## Exemple burn-down chart de sprint (final)



## Exemple de diagramme de flux cumulé



## L'incrément

- L'incrément est la somme de tous les items du Backlog de produit terminés pendant un sprint, additionné de tous les items terminés des sprints précédents.
- Il doit être dans un état utilisable, que le Product Owner décide de le livrer ou non.

## La définition de "fini"

- Décrit à quelle condition un travail est considéré comme terminé
- Partagée par toute l'équipe Scrum
- Portée : item de backlog ou incrément
- Évaluation systématique du caractère terminé d'un travail
- La définition de "fini" peut évoluer dans le temps

## Exemples de définition de fini

- <http://www.scrum-breakfast.com/2012/11/sample-definition-of-done.html>
- <https://www.mitchlacey.com/intro-to-agile/scrum/definition-of-done>

## Illustrations

- ENE Atrium
- Tsaap-Notes
- Elastic

## Annexes

### Le Sprint 0

- Seul Sprint dont la durée n'est pas fixée
- Définir la vision du produit
- Initialiser le backlog
- Initialiser la plateforme de production

### Le guide des pratiques agiles

<http://guide.agilealliance.org/>

<http://guide.agilealliance.org/guide/definition-of-done.html>

<http://guide.agilealliance.org/guide/acceptance.html>

## Un regard critique

### Agile ! The good, the hype and the ugly

- Bertrand Meyer
  - <http://www.springer.com/us/book/9783319051543>
- <https://www.youtube.com/embed/nchmlSwqGBc>