

Panorama du Développement Classique au Développement Agile



Isabelle BLASQUEZ
@iblasquez

Septembre 2018

Une initiative autour de l'archivage du code...

Software Heritage : archive mondiale du logiciel



L'INRIA lance une **#bibliothèque #mondiale** en ligne pour les **#logiciels #libres**
ow.ly/qq62301Tb3S

Inria a annoncé le jeudi 30 juin 2016 l'ouverture au public du projet Software Heritage. Ce projet, à l'initiative d'Inria, a pour objectif de collecter, organiser, préserver, et rendre accessible, à tous, le code source de tous les logiciels disponibles. Un enjeu de portée mondiale.



preserves software source code for present and future generations

<https://www.softwareheritage.org/>

It concerns all of us. And you can help!

Software is so pervasive in our lives that its preservation concerns all of us. Our mission and the archive we are building will serve the needs of the many, from cultural institutions to scientists and industries. Everyone can help us achieving these ambitious goals.



Heritage

Software is an important part of human production. It is also a key enabler for salvaging our entire digital heritage.

We collect, preserve, and make accessible source code for the benefits of present and future generations.

[Learn more and help](#)



Science

Science relies more and more on software. To guarantee scientific reproducibility we need to preserve it.

Amassing source code at this scale will be challenging, but will also enable the next generation of software studies.

[Learn more and help](#)



Industry

Software is present in all industrial processes and products.

The universal source code archive we are building will help industry with provenance tracking, long-term archival, and software bill of materials.

[Learn more and help](#)

Extrait : <https://www.softwareheritage.org/archive/>

<https://www.inria.fr/actualite/actualites-inria/lancement-de-software-heritage>

<https://forge.softwareheritage.org>

Isabelle BLASQUEZ

Et tant qu'à parler d'archives ...

<https://archive.org/>



Ses collections comportent⁸ :

- 456 milliards de versions archivées (*snapshots*) du [World Wide Web](#) et d'[Usenet](#), soit 67 millions de sites web de 37 langues,
- 1 900 000 [films](#),
- 2 400 000 d'enregistrements [audios](#),
- 149 000 [concerts](#),
- 7 800 000 de [livres](#),
- 96 000 [logiciels](#),
- 951 000 [images](#).

Extrait : https://fr.wikipedia.org/wiki/Internet_Archive

Avec par exemple du rétro-gaming :

https://archive.org/details/softwarelibrary_msdos_games
ou <https://archive.org/details/consolelivingroom>

... Et on peut parfois même jouer :

https://archive.org/details/msdos_Donkey_Kong_1983

Internet Archive is a non-profit library of millions of free books, movies, software, music, websites, and more.



GO

[Advanced Search](#)

Software Library: MS-DOS Games

Software for MS-DOS machines that represent entertainment and games. The collection includes action, strategy, adventure and other unique genres of game and entertainment software. Through the use of the EM-DOSBOX in-browser MORE

[About](#) [Collection](#)

[SORT BY](#) [VIEWS](#) [TITLE](#) [DATE ARCHIVED](#) [CREATOR](#)

Oregon Trail, The by MECC	Prince of Persia by Jul 29, 2014	Pac-Man by Namco Limited	Wolfenstein 3D by Id Software, Inc.	Oregon Trail Deluxe, The by Distinctive Software, Inc.
View Details	View Details	View Details	View Details	View Details

Super Street Fighter II
by Capcom Co., Ltd.

SimCity
by Maxis Software Inc.

Prehistoric 2
by Titus France SA

4D Prince of Persia
by Distinctive Software, Inc.

Stunts
by Distinctive Software, Inc.

**Mais avant d'archiver un logiciel,
il faut le développer ...**

Principales phases du développement logiciel



A propos de la phase d'Analyse des exigences (Requirements phase)

La phase d'analyse des exigences est la période du cycle de vie pendant laquelle les exigences, fonctionnelles et non fonctionnelles du produit logiciel, sont définies et documentées. (**IEEE, 1990¹**).

Ce que doit faire le logiciel

Cette phase donne lieu à l'écriture d'un document de **spécifications** qui précise les missions du logiciel. Ce document est une trace des besoins utilisateurs et sera utilisé dans les autres phases du cycle de développement.

¹ IEEE Standard Glossary of Software Engineering Terminology

Abstract: IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, identifies terms currently in use in the field of Software Engineering. Standard definitions for those terms are established.
Keywords: Software engineering; glossary; terminology; definitions; dictionary

A propos de la phase de Conception (Design phase)

COMMENT
faire ce logiciel

La phase de conception est la période du cycle de vie pendant laquelle l'architecture logicielle, les composants logiciels, les données et les interfaces sont conçus et documentés afin de satisfaire aux exigences.
(IEEE, 1990).

A propos de la phase d'Implementation (Implementation phase/Coding)

Ecriture
du code

La phase d'implémentation est la période du cycle de vie pendant laquelle le logiciel est créé et débuggé à partir des spécifications de conception. (IEEE, 1990).

Les tâches de cette phase se concentrent autour du **code** où les composants sont implémentés et testés individuellement **dans un langage de programmation** donné afin de mettre en œuvre la conception

A propos de la phase de Test (Test phase)

Qualité logicielle

La phase de Test est la période du cycle de vie consacrée à l'intégration et à l'évaluation des composants et du logiciel afin de vérifier les exigences aussi bien au niveau système qu'utilisateur (**IEEE, 1990**).

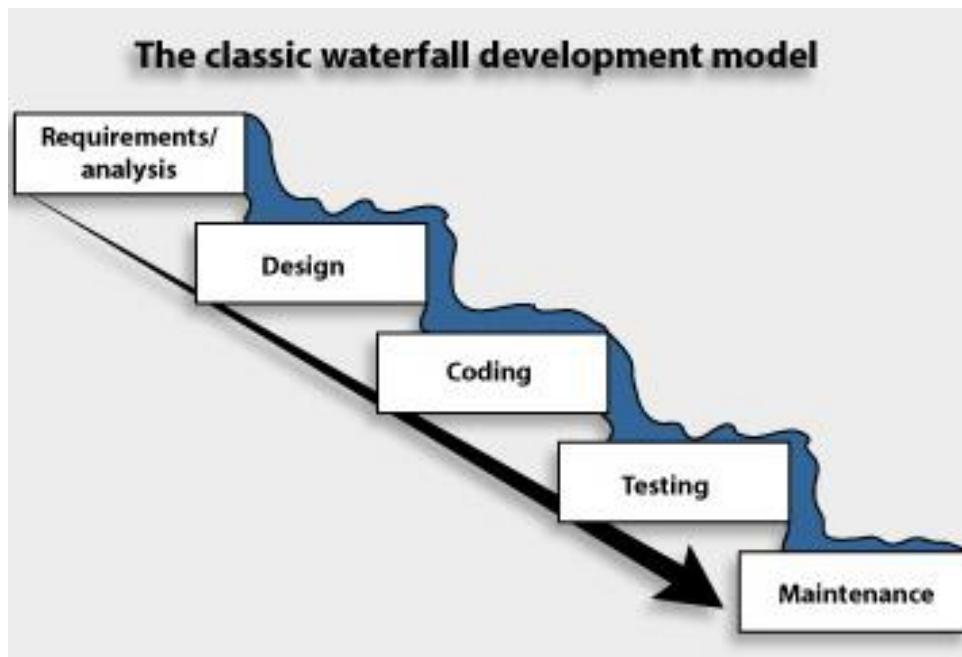
Modèle de développement dit « en cascade »

Analyse

Conception

Implémentation

Test

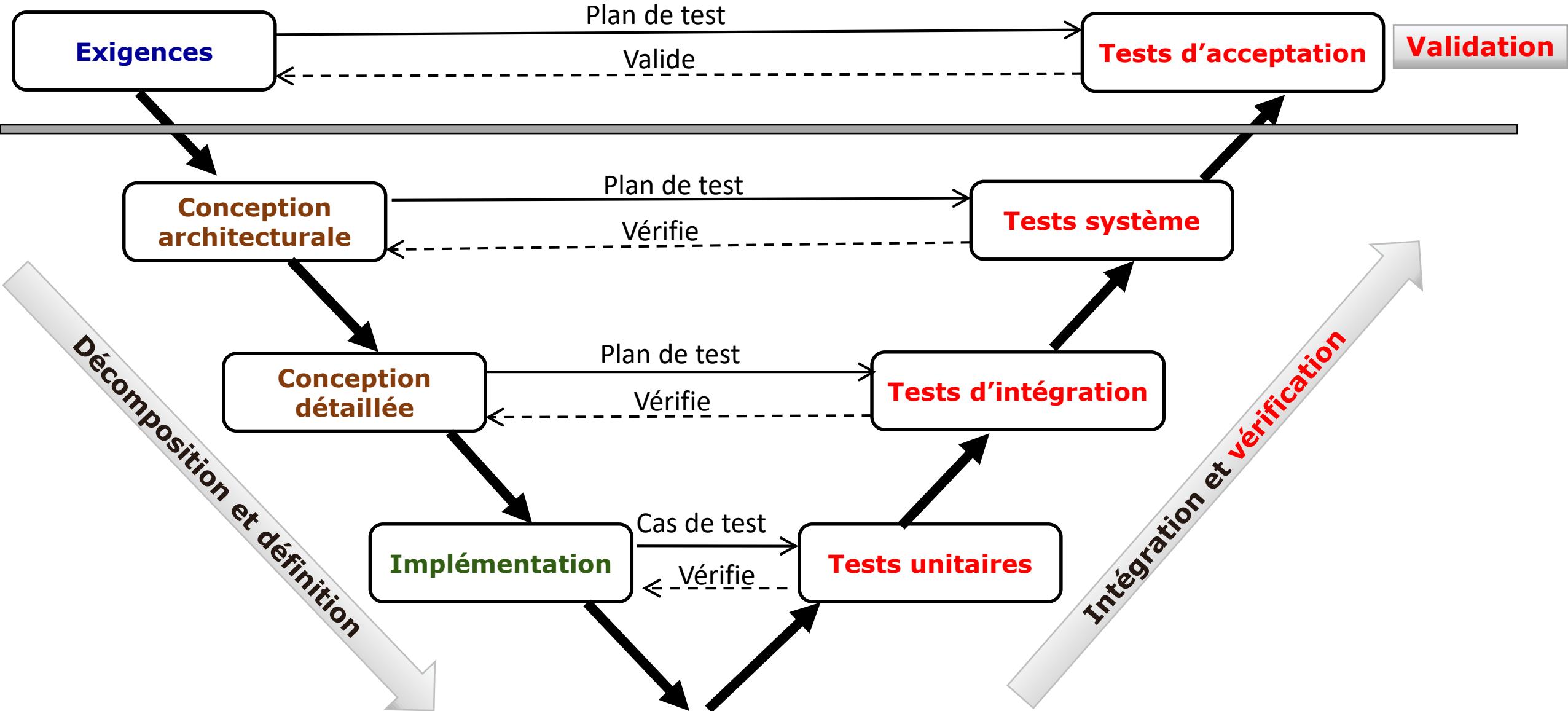


Attention !!! A l'origine, « contre-exemple » utilisé par Royce pour lui permettre d'introduire les bonnes pratiques de développement

Article original de Royce : <http://www.serena.com/docs/agile/papers/Managing-The-Development-of-Large-Software-Systems.pdf>

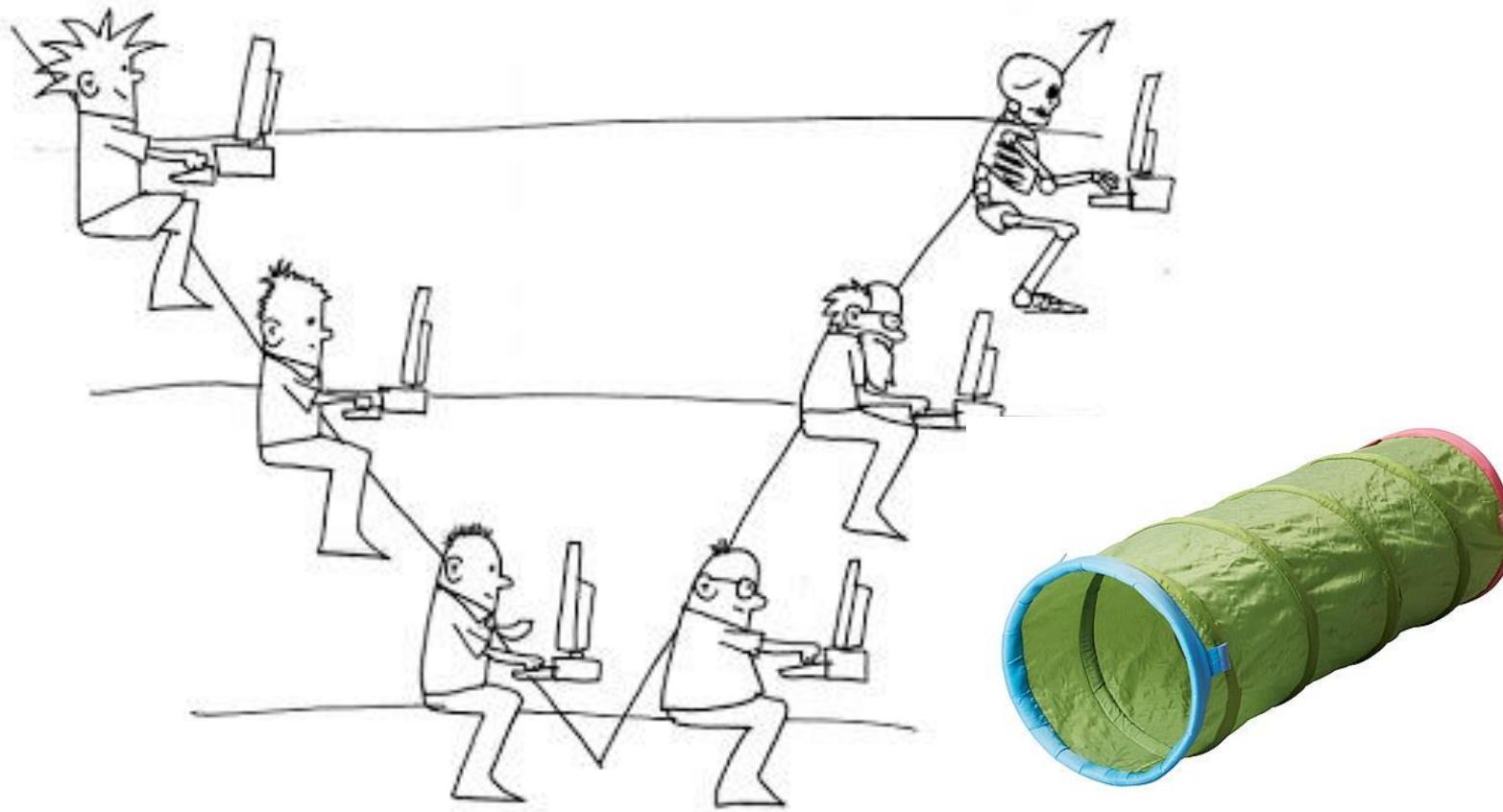
Traduction française : <http://www.fabrice-aimetti.fr/dotclear/public/traductions/waterfall-FR.pdf>

Cycle en V : standard de l'industrie logicielle

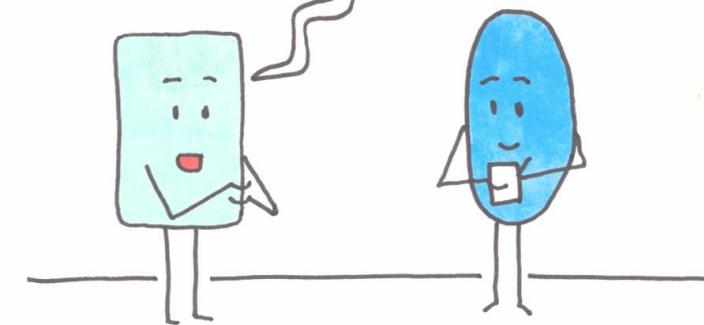


Un cycle trop long, bien souvent un effet tunnel...

PREMIER JOUR DE STAGE...



POUR ÊTRE SUR DE RÉPONDRE AUX BESOINS DE L'UTILISATEUR TU VAS FAIRE UN PLAN QUALITÉ, UN DOCUMENT DE SPÉCIFICATION FONCTIONNELLE ET TECHNIQUE, LE PLAN DE VALIDATION ET ENFIN LE DÉVELOPPEMENT. UNE FOIS L'APPLI PRÊTE, TU LE PRÉSENTES À L'UTILISATEUR.



DERNIER JOUR DE STAGE...



Et « Les joies du code » dans tout ça ?

En suivant ces cycles de vie, on peut passer plus de temps sur la façon de développer un système que sur le développement lui-même...

Quand on me dit que
je vais pouvoir coder
après 3 mois passés
sur les plans de tests



<http://lesjoiesducode.fr/post/95172054995/quand-on-me-dit-que-je-vais-pouvoir-coder-apr%C3%A8s-3-mois-pass%C3%A9s-sur-les-plans-de-tests>

Quand je peux enfin lancer
les tests unitaires
sur mon appli



<http://lesjoiesducode.fr/post/75046110363/quand-je-peux-enfin-lancer-les-tests-unitaires-sur-mon>

« Les joies du client » : *Right product ?*



<http://lesjoiesducode.fr/post/98960240314/quand-je-respecte-les-specs-du-client-a-la-lettre>

Quand je respecte les spécifications du client à la lettre...

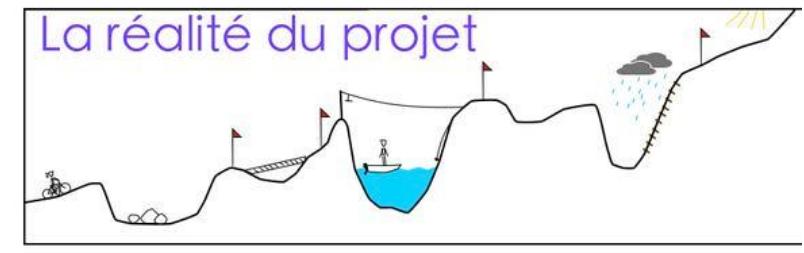
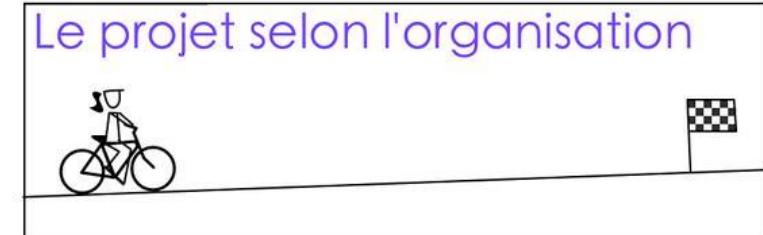


<http://www.commitstrip.com/fr/2014/10/15/when-i-thoroughly-follow-the-requirements/>

Isabelle BLASQUEZ

Prise en compte du risque ?

- ✓ **Un risque** peut être défini comme la probabilité qu'un événement, un danger, une menace ou une situation arrive, et que les conséquences indésirables qui en découlent constituent un problème potentiel.

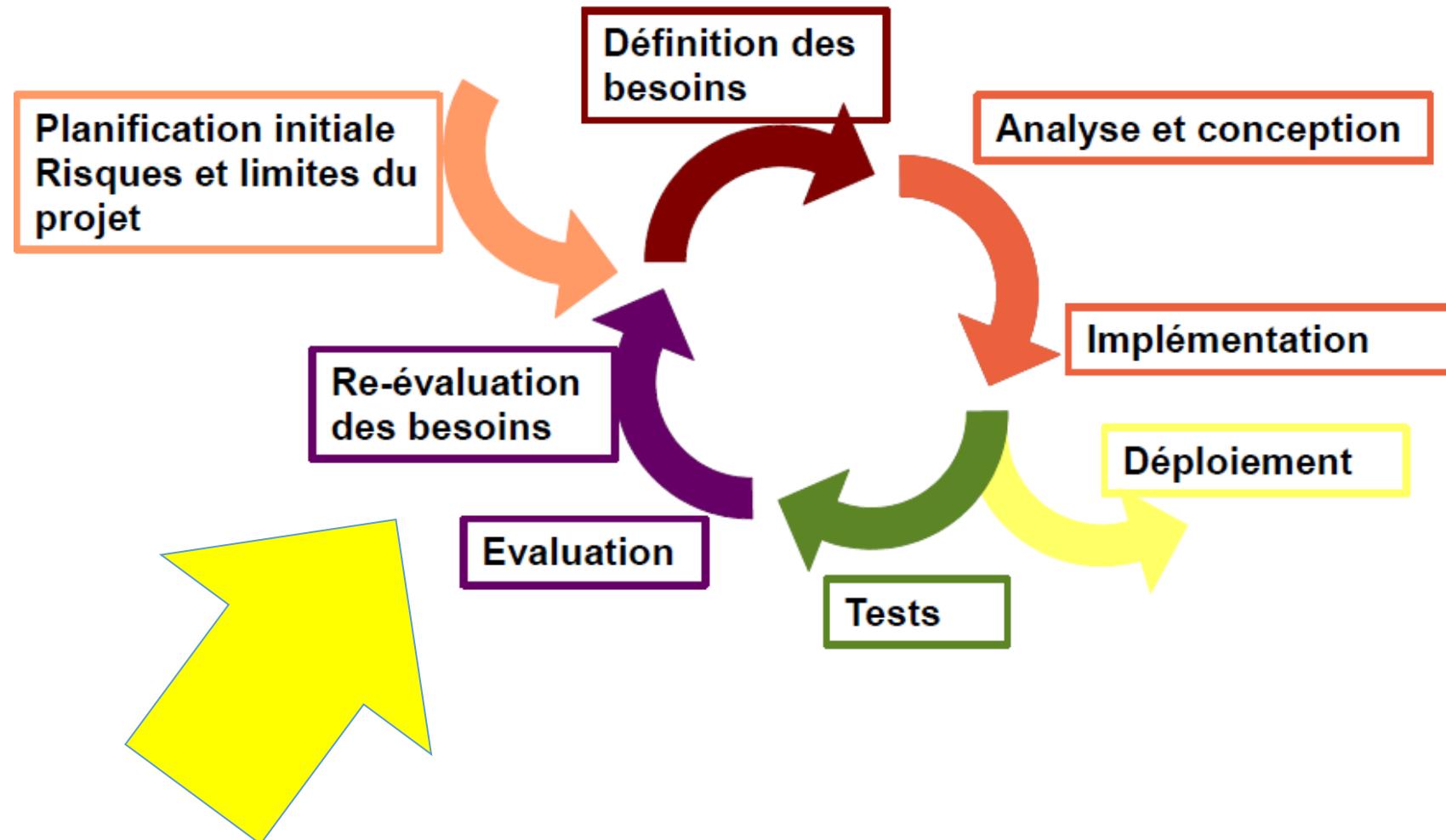


Source : <https://twitter.com/agilep/status/398831282051772416/photo/1>

- ✓ 2 types de risques :
 - **risques liés au produit** lié à la **qualité** du produit...
 - **risques liés au projet** qui menacent la capacité de ce dernier à atteindre ses objectifs:
 - les **facteurs organisationnels**
 - les **problème techniques**



Un développement **itératif** pour une meilleure évaluation et prise en compte des risques

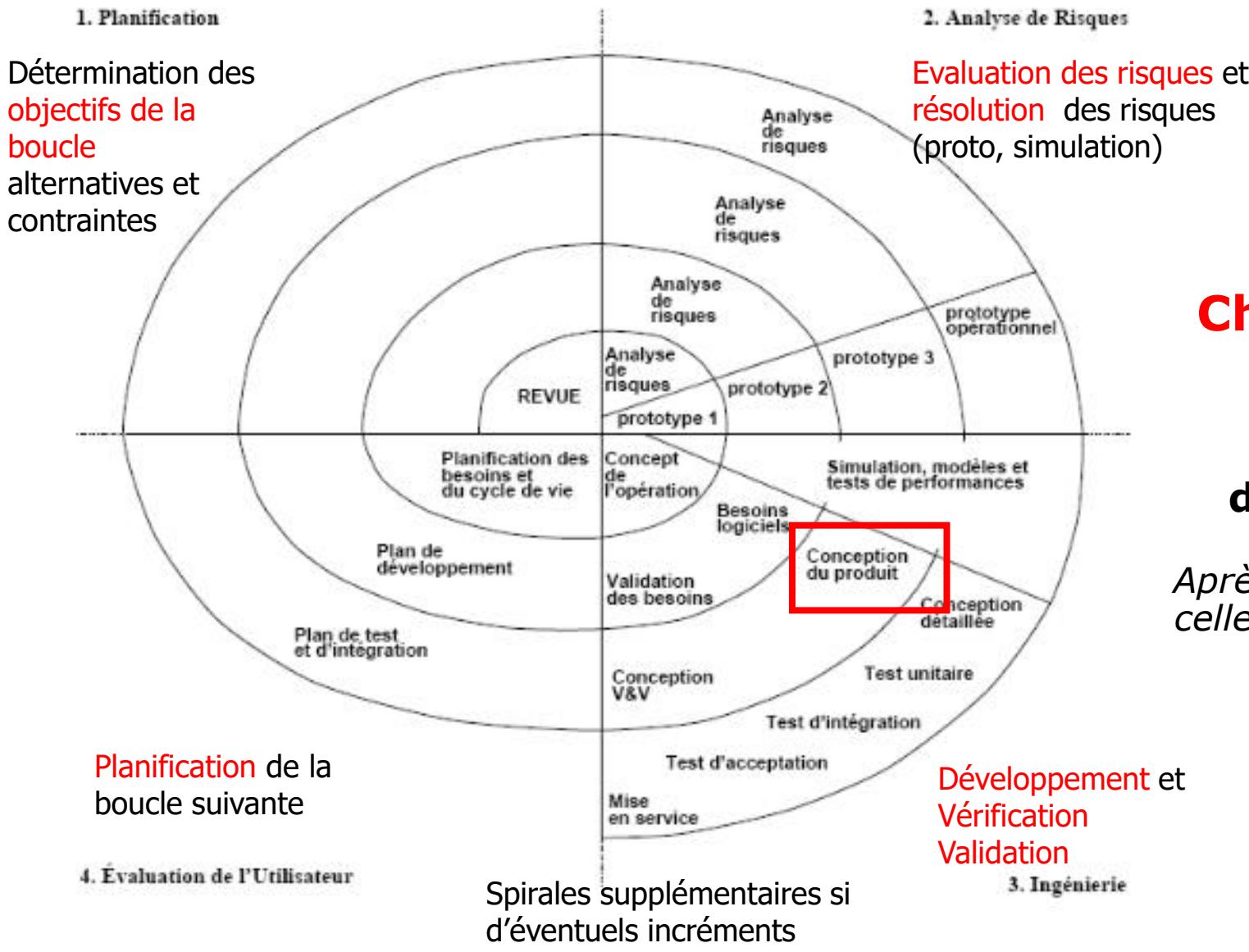


Extrait de : https://www.projet-plume.org/files/bonnespratiques_envol2010_cc_0.pdf

Isabelle BLASQUEZ

Modèle en spirale (Boehm)

Axé sur **l'analyse des risques**, principe **itératif**



réduit les risques si bien appliqué : possibilité de s'arrêter « à temps »

Chaque boucle représente une phase du développement (boucle de faisabilité, boucle de prototypage, des boucles de développement,...)

Après avoir défini les objectifs et les alternatives, celles-ci sont évaluées par différentes techniques (**prototypage, simulation**, ...), l'étape est réalisée et la suite est planifiée

Illustration des principes *itératif* et *incrémental*

✓ Développement itératif

⇒ améliorer un existant pour atteindre un résultat plus satisfaisant



⇒ Permet d'améliorer la **qualité** (affiner, **enrichir**)

✓ Développement incremental

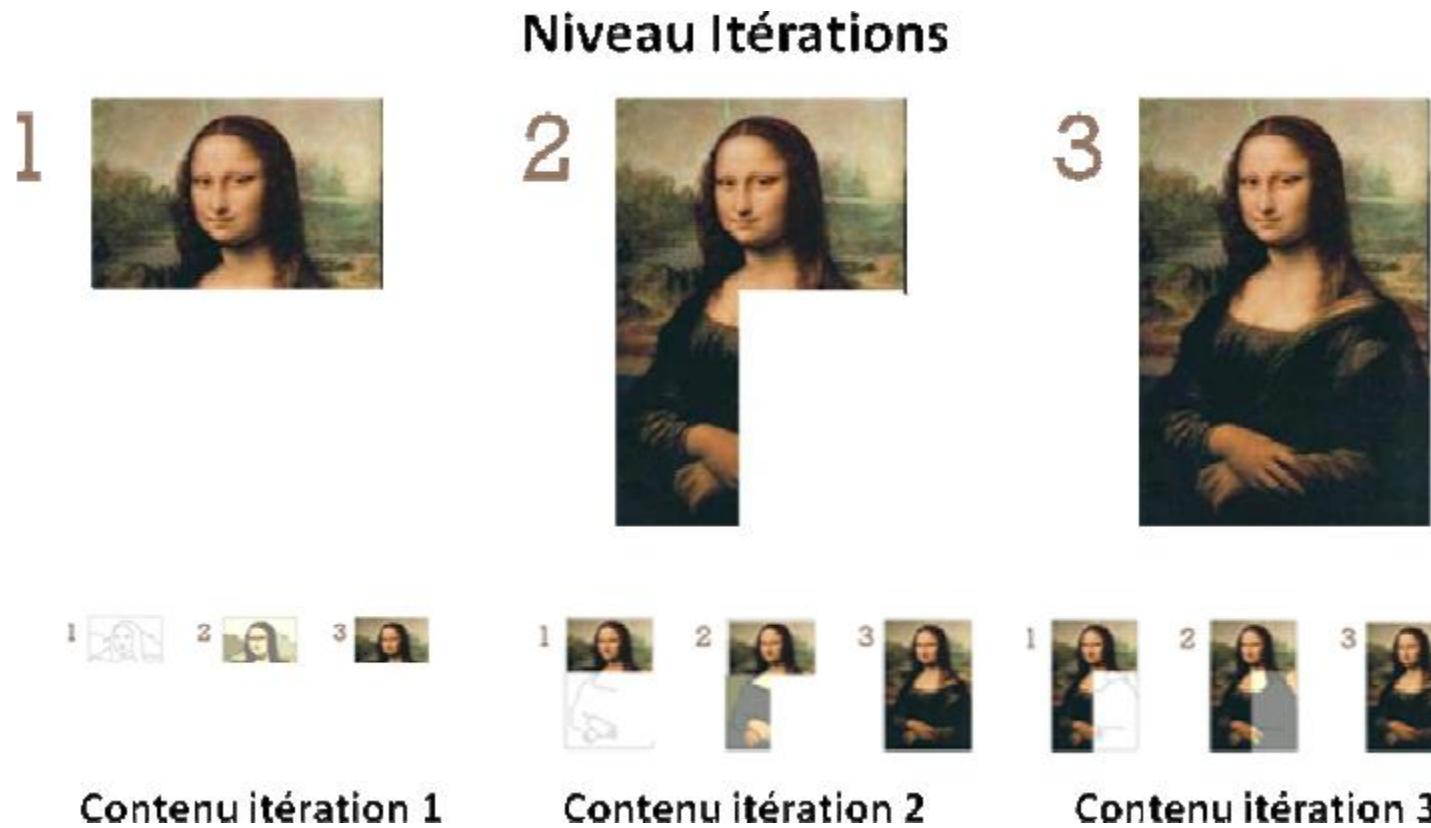
⇒ ajouter pour atteindre le résultat final



⇒ Permet d'améliorer le processus et d'**ajuster** les exigences à l'évolution de l'environnement.
Analogie : mur en brique

Démarche *itérative* et *incrémentale*

A chaque nouvelle itération on **ajuste** ce qui a déjà été réalisé et on l'**enrichit** en traitant un spectre fonctionnel plus large.



*Illustration proposée par Jean-Pierre Vickoff pour illustrer la **dimension temporelle** de l'**itération** (le pilotage du projet) en regard de la **dimension applicative** de l'**incrément**.*

Isabelle BLASQUEZ

Processus Unifié (**UP** : Unified Processus 1997)

- ✓ **Un processus UP** est un processus générique qui définit les exigences suivantes :

- **conduit par les besoins des utilisateurs**
- **itératif et incrémental**
- **centré sur l'architecture**

...mais aussi...

- **piloté par les risques** (les risques doivent être identifiés au plus tôt et levés rapidement, ce qui doit déterminer l'ordre des itérations)
- **basé le formalisme UML**

- ✓ **RUP (Rational Unified Process)** est l'une des plus célèbres implémentations de la méthode UP

En savoir plus : https://fr.wikipedia.org/wiki/Unified_process

http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf Isabelle BLASQUEZ

RUP (Rational Unified Process)

RUP est organisé autour de **4 phases** et défini en **9 disciplines**.

Contrairement au cycle en V, les disciplines ne sont pas purement séquentielle, mais se répartissent sur plusieurs itérations (itérative & incrémentale)

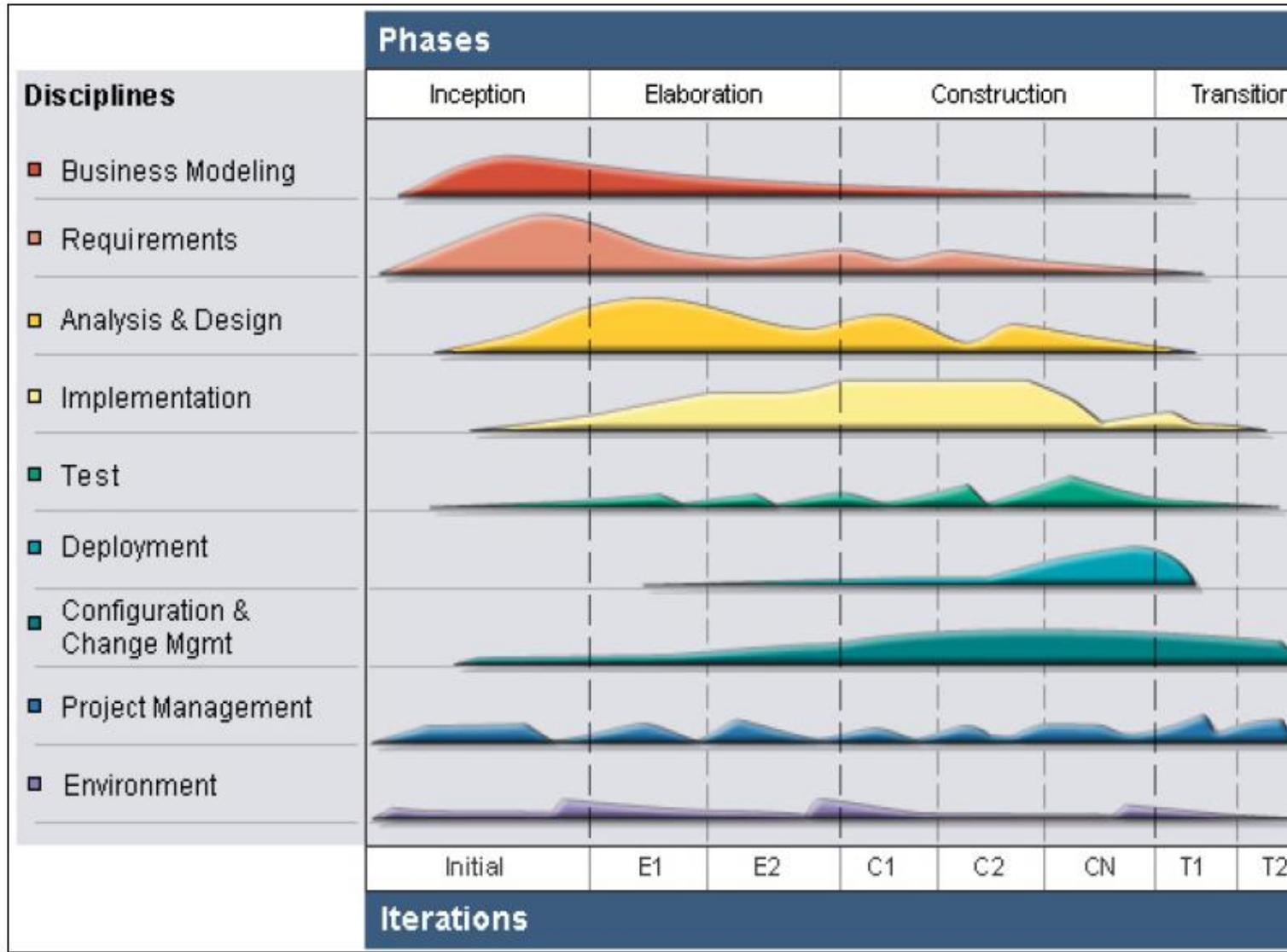


Figure 2-1 Overall Architecture of RUP

extrait de :
The IBM Rational Unified Process for System z
A consulter pour tout savoir sur le RUP disponible en ligne sur
<http://www.redbooks.ibm.com>

Inconvénient du RUP

Beaucoup de documentation



**Trop d'acteurs isolés
Et peu de communication
entre eux ...**

I.I PROJET INFORMATIQUE



**Très axé processus ...
Place tardive pour le code et
la technologie**

Just tried to explain technical debt to a customer, had to pull this out again...

ⓘ Voir la traduction



... et voilà *How projects really works*



Ce que
le client
a expliqué



Ce que
le chef de projet a
compris



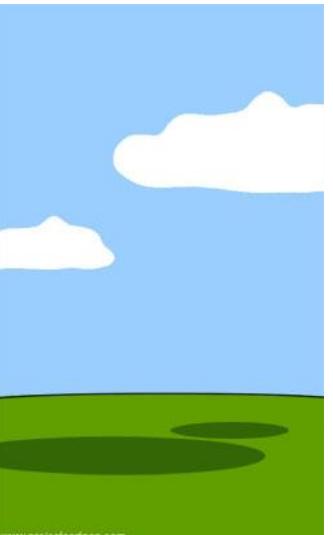
Ce que
l'analyste l'a conçu



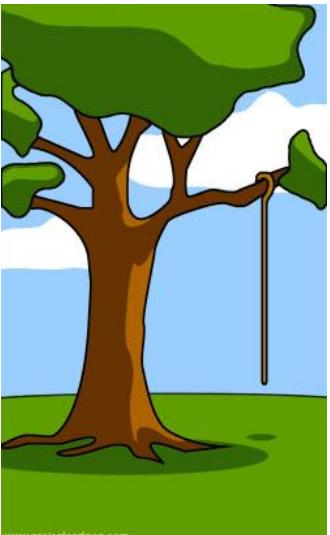
Ce que
le développeur
a fait



Comment
les commerciaux
l'ont décris



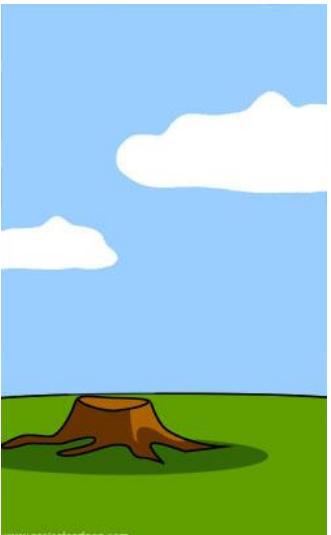
Comment
le projet a été
documenté



Ce que la production a
installé chez le client



Ce que
le client a été
facturé



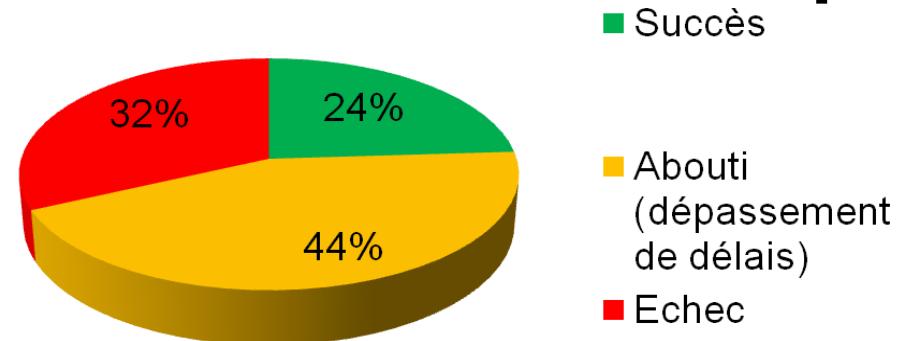
Comment le support
technique est effectué



Ce dont le client
avait réellement
besoin

Un Constat ...

✓ Un fort taux d'échec des projets ...



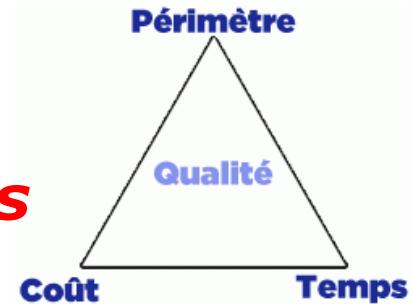
Taux de succès d'un projet
Données issues du rapport
Chaos 2009 du *Standish Group*

... et une grande partie de ce qui est fait n'est pas utilisé...

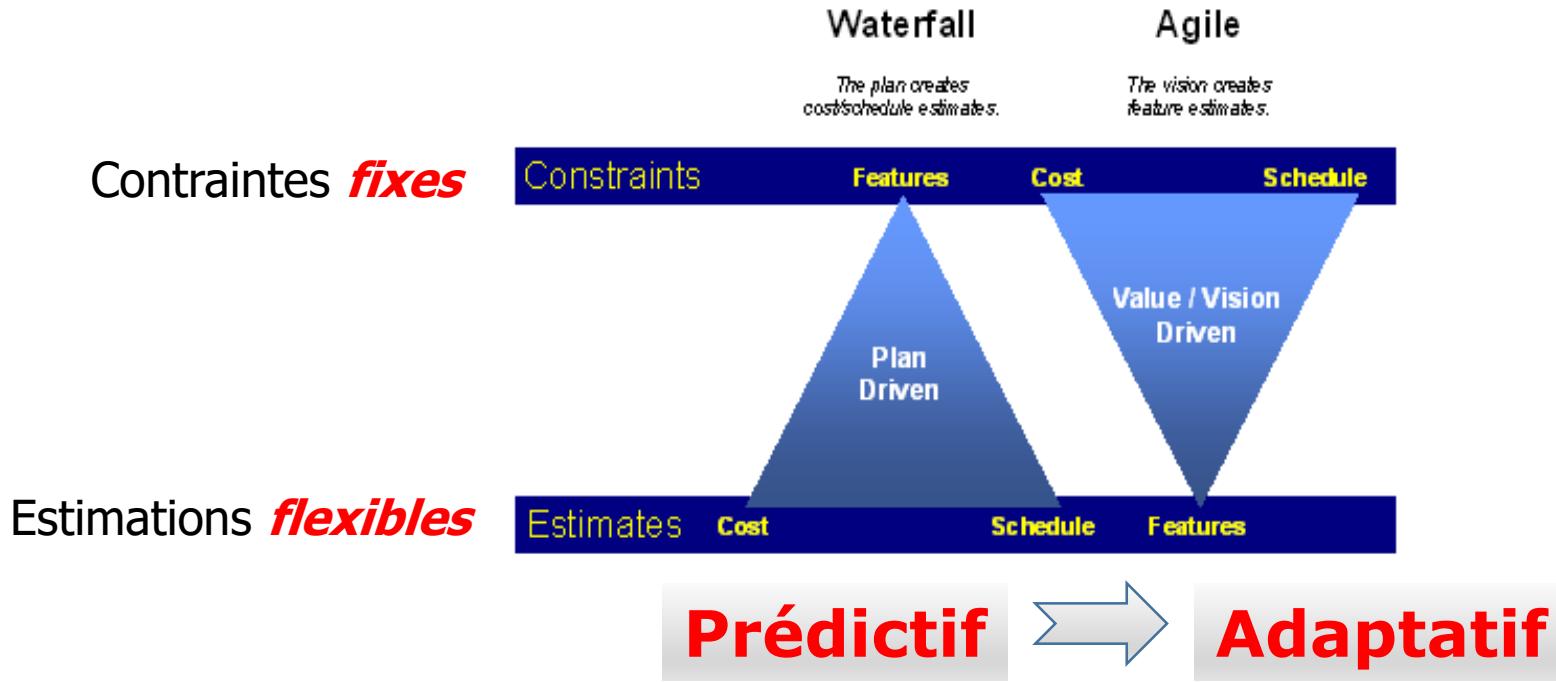
STOP !

A la recherche d'un nouveau paradigme ...

L'objectif d'un processus de développement est de produire des logiciels de **qualité qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles**...



⇒ Inverser le triangle de projet pour mieux gérer les contraintes en donnant plus d'importance à la valeur métier plutôt qu'aux prévisions initiales



En 2001, le Manifeste pour le développement Agile de Logiciels



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

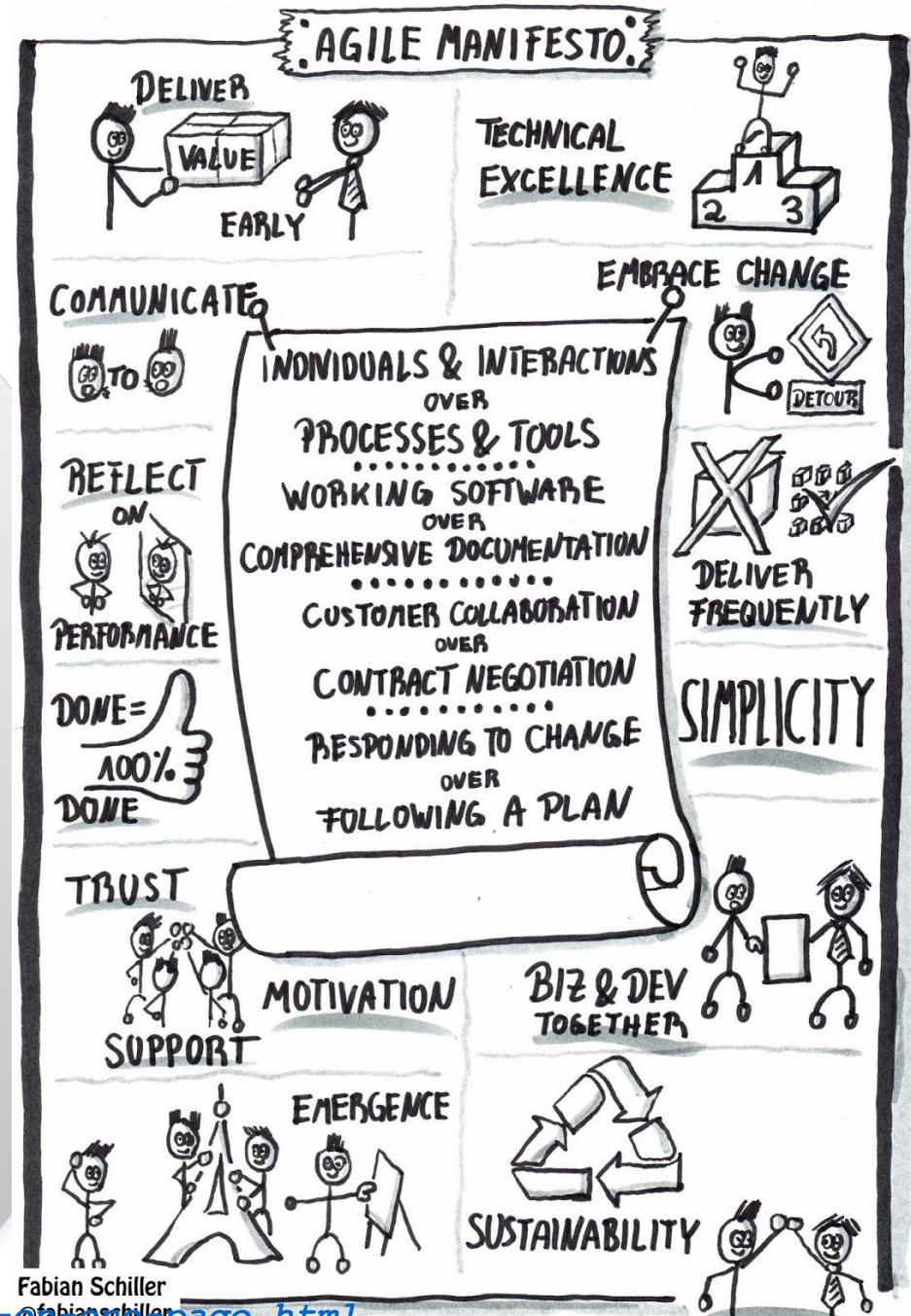
Nous reconnaissons la valeur des seconds éléments,
mais privilégions les premiers.

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

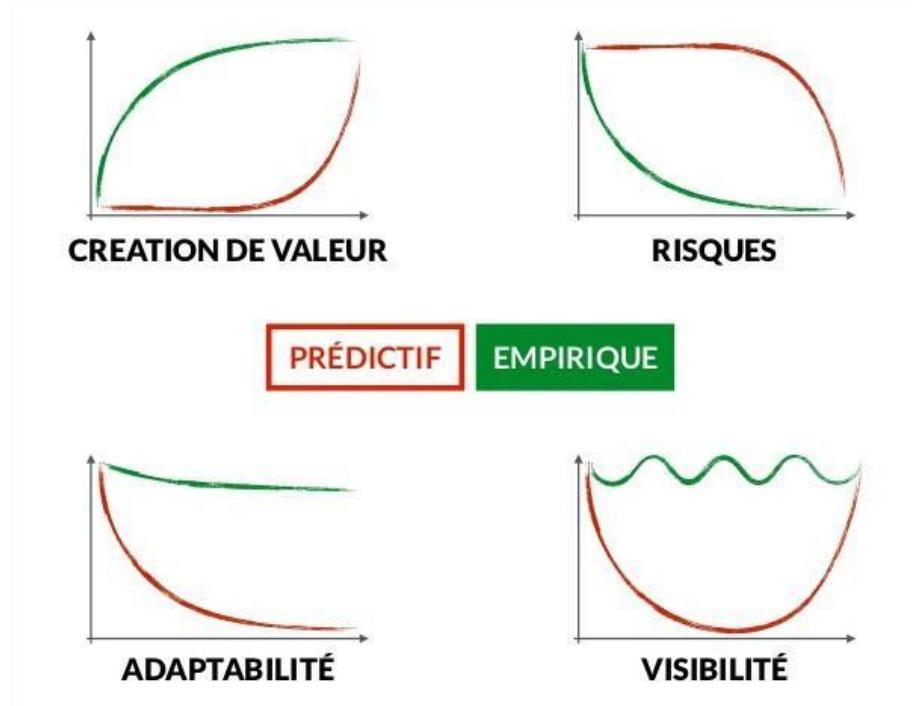
Les Principes sous-jacents au manifeste

12 principes découlent de ce manifeste.
Ils indiquent comment implémenter l'Agilité :

- ✓ Livrer de manière **continue** des **fonctionnalités complètes**, utiles et utilisables
- ✓ Développer par **itérations courtes** (quelques semaines)
- ✓ Etre adepte du **changement**, privilégier les solutions **simples** et **adaptables**
- ✓ **Rapprocher** utilisateurs et développeurs
- ✓ **S'améliorer** continuellement



Prédicatif vs Adaptatif (empirique)



Extrait original: <http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/>
Repris en français : <http://fr.slideshare.net/rvignes/lentreprise-agile-29135486>



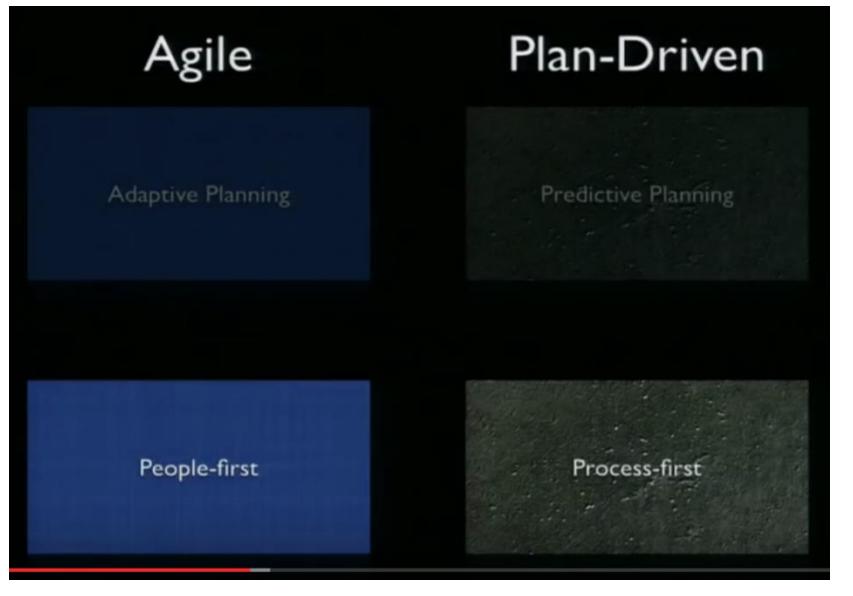
Agile Versus Traditional Waterfall

Metric	Waterfall	Agile
Planning scale	Long-term	Short-term
Distance between customer and developer	Long	Short
Time between specification and implementation	Long	Short
Time to discover problems	Long	Short
Project schedule risk	High	Low
Ability to respond quickly to change	Low	High

Extrait : <http://www.venveo.com/blog/agile-vs-waterfall-project-management>

Empirique : lorsque le processus doit être expérimenté puis adapté à l'environnement de travail

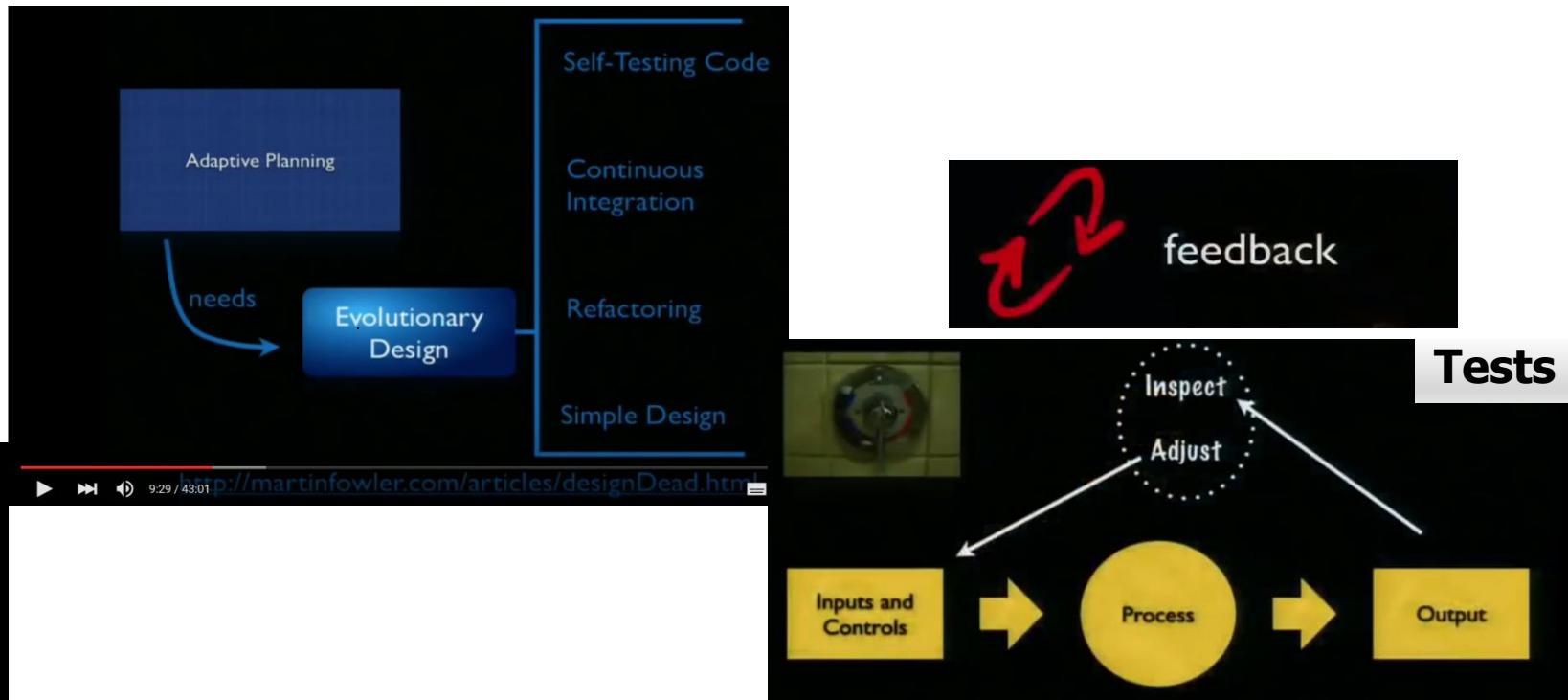
Explaining Agile - Martin Fowler and Neil Ford at USI



communication > technology

A lire également : <http://www.martinfowler.com/articles/newMethodology.html>
<http://martinfowler.com/articles/designDead.html>

Extraits de la vidéo :
<https://www.youtube.com/watch?v=GE6lbPLEAzc>



Pair programming

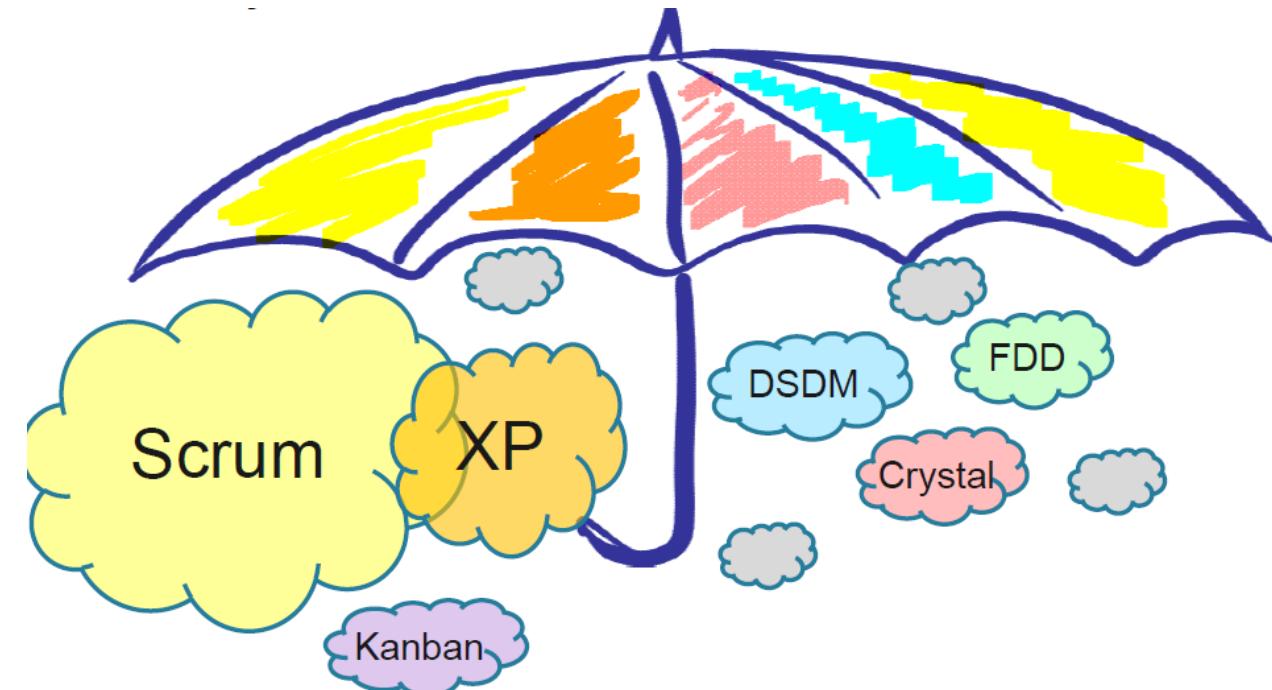
Retrospective :
how to change
regularly process

In software development, “perfect” is
a verb, not an adjective

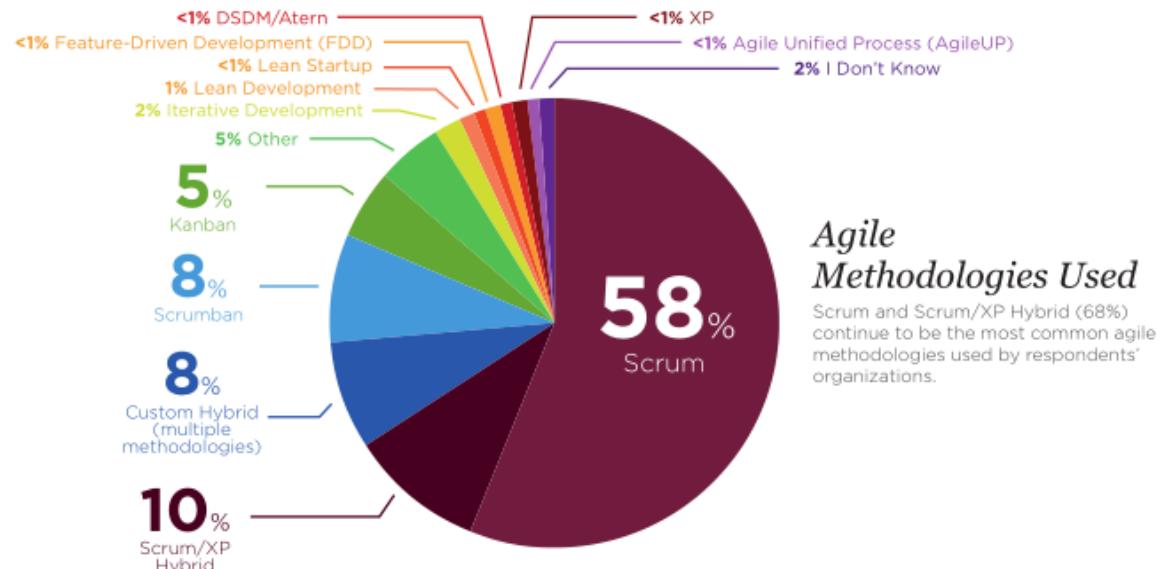
Citation de Kent Beck dans Extreme Programming Explained

labeled BLASQUEZ

Des méthodes agiles ...



AGILE METHODS AND PRACTICES



Agile Methodologies Used

Scrum and Scrum/XP Hybrid (68%) continue to be the most common agile methodologies used by respondents' organizations.

Méthodes agiles les plus utilisées en 2017 (USA et Europe) :

Extrait : <http://stateofagile.versionone.com/>

<https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>

MAIS ATTENTION ...



David J Bland
@davidjbland

Scrum - XP = Agile Theater

Et un jargon agile à s'approprier ...

Scrum en 100 mots

SCRUM est une méthodologie AGILE.

Elle vise à satisfaire au mieux le besoin du PRODUCT OWNER en l'impliquant aux différentes phases du projet. Celui-ci est responsable du PRODUCT BACKLOG, liste priorisée des USERS STORIES.

Lors du SPRINT PLANNING, le PRODUCT OWNER, le SCRUM MASTER et la TEAM définissent le SPRINT BACKLOG que la TEAM s'engage à réaliser au cours du SPRINT.

Au quotidien, le SCRUM MASTER se réunit avec la TEAM pour le DAILY SCRUM et la BURN DOWN CHART est mise à jour.

La TEAM conclut le SPRINT par une DEMO du travail réalisé, puis une SPRINT REVIEW.

Extrait de : <http://blog.soat.fr/2010/05/un-apercu-de-scrum-pour-neophyte-%E2%80%A6/>

Quelques liens pour s'y retrouver dans le jargon agile

<http://guide.agilealliance.org>

<http://referentiel.institut-agile.fr/>

<http://www.mountaingoatsoftware.com/agile>

<https://www.scrum.org/Resources/Scrum-Glossary>

<http://www.aubryconseil.com/post/2007/07/17/262-glossaire-scrum>

<http://thierry-leriche-dessirier.developpez.com/tutoriels/general/memento-scrum-destination-equipe>



Des traductions en français d'articles, des (mini) livres, de jeux sérieux :

<http://www.les-traducteurs-agiles.org/>

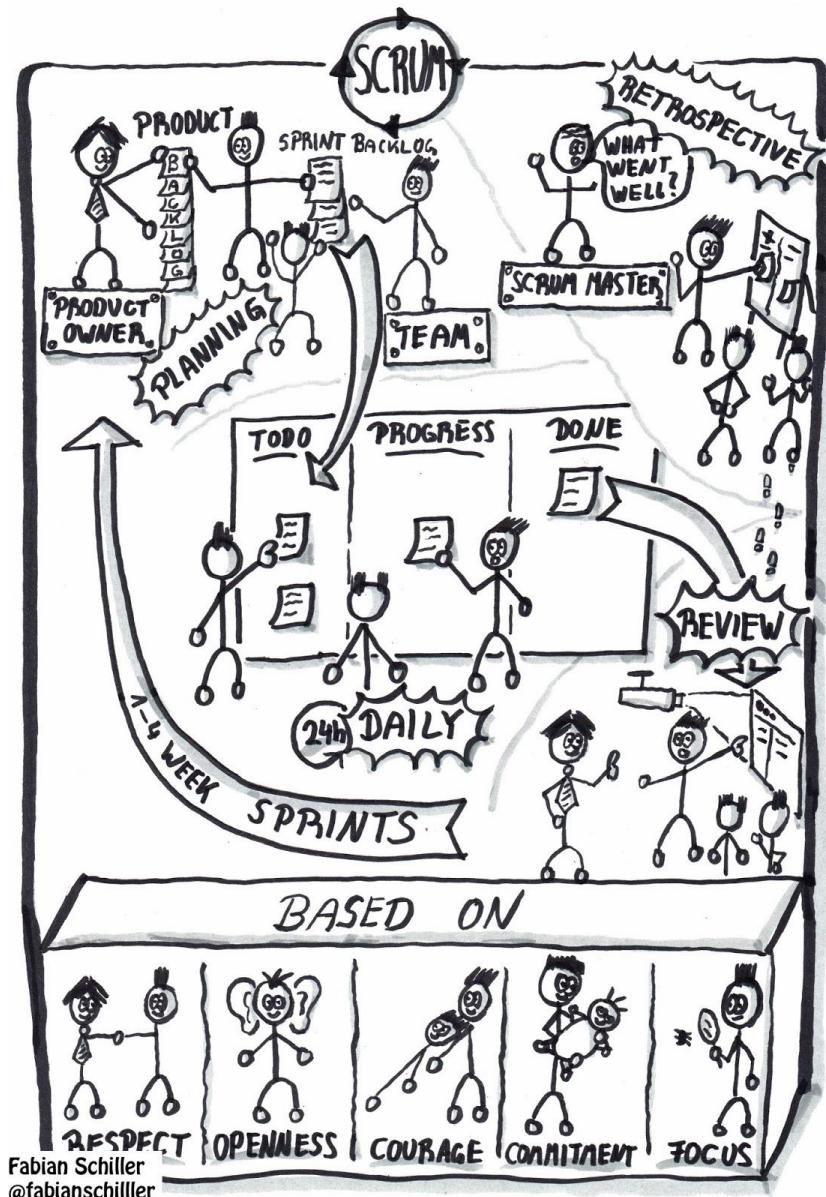
<http://wiki.ayeba.fr/> (librement payant)

proposant aussi des traductions librement mises à disposition :

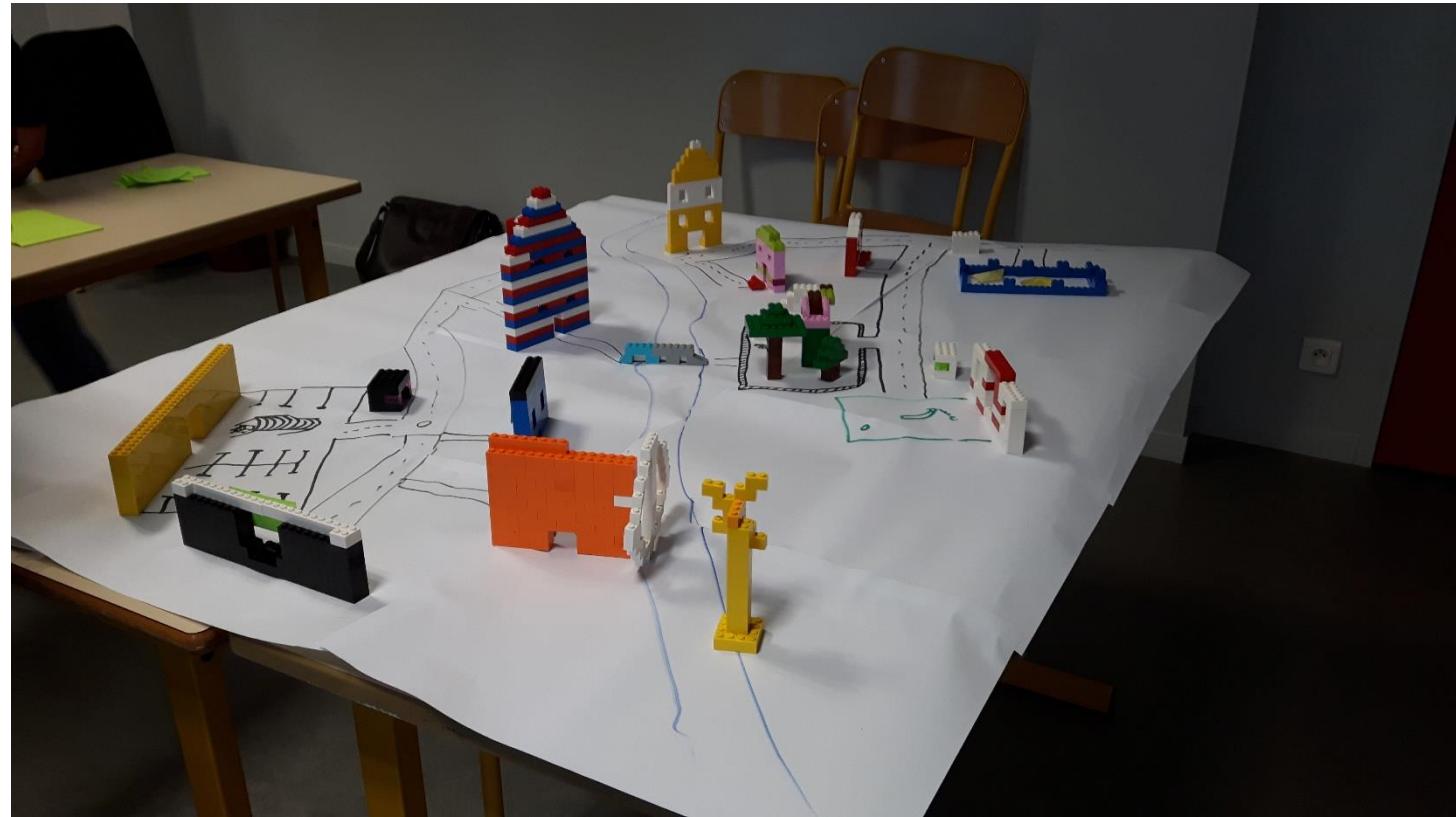
<http://wiki.ayeba.fr/Traducteurs+Agiles> (reprise traducteurs agiles)

<http://wiki.ayeba.fr/Portail+Traductions+librement+mises+%C3%A0+disposition>

La semaine dernière ...

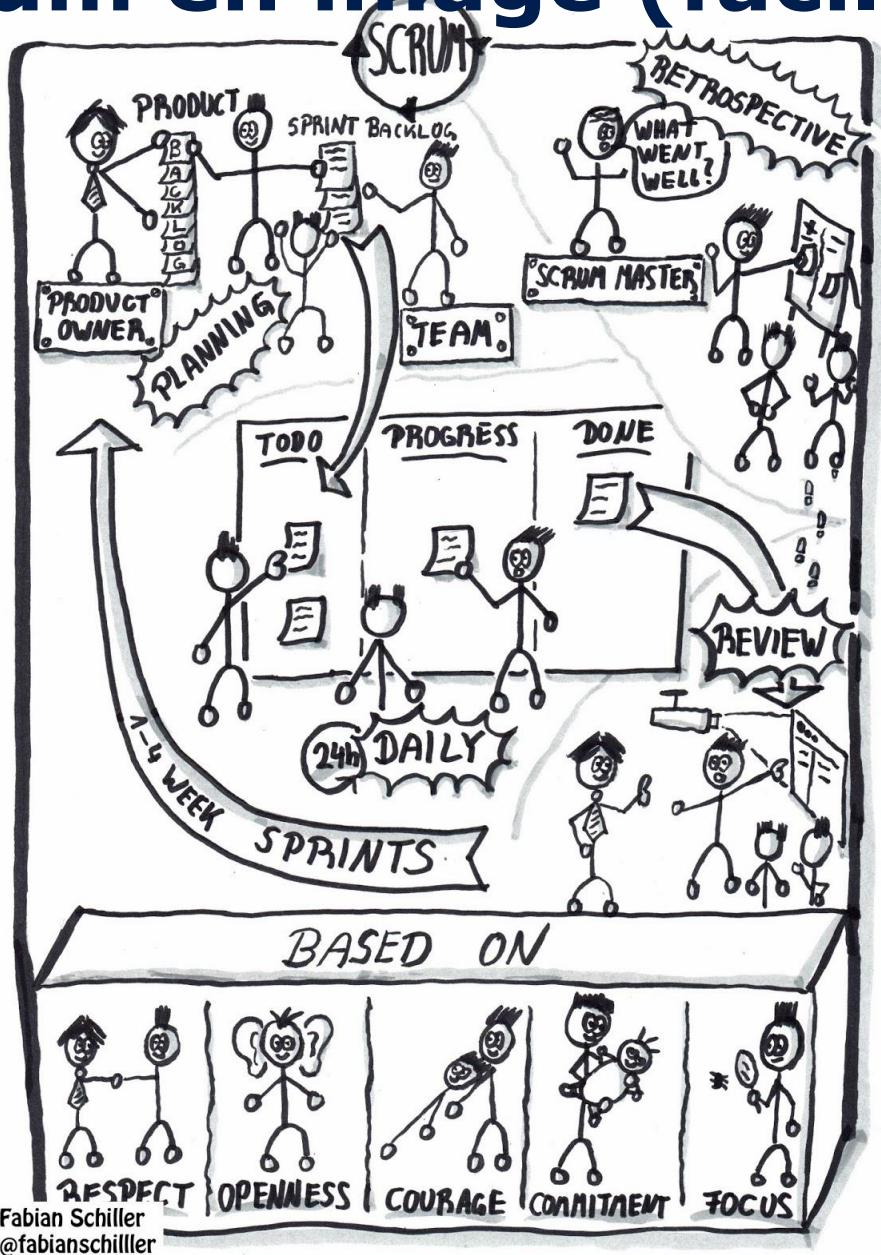


SCRUM SIMULATION WITH LEGO



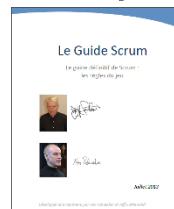
En savoir plus sur : <http://www.lego4scrum.com/>

Scrum en image (facilitation graphique)

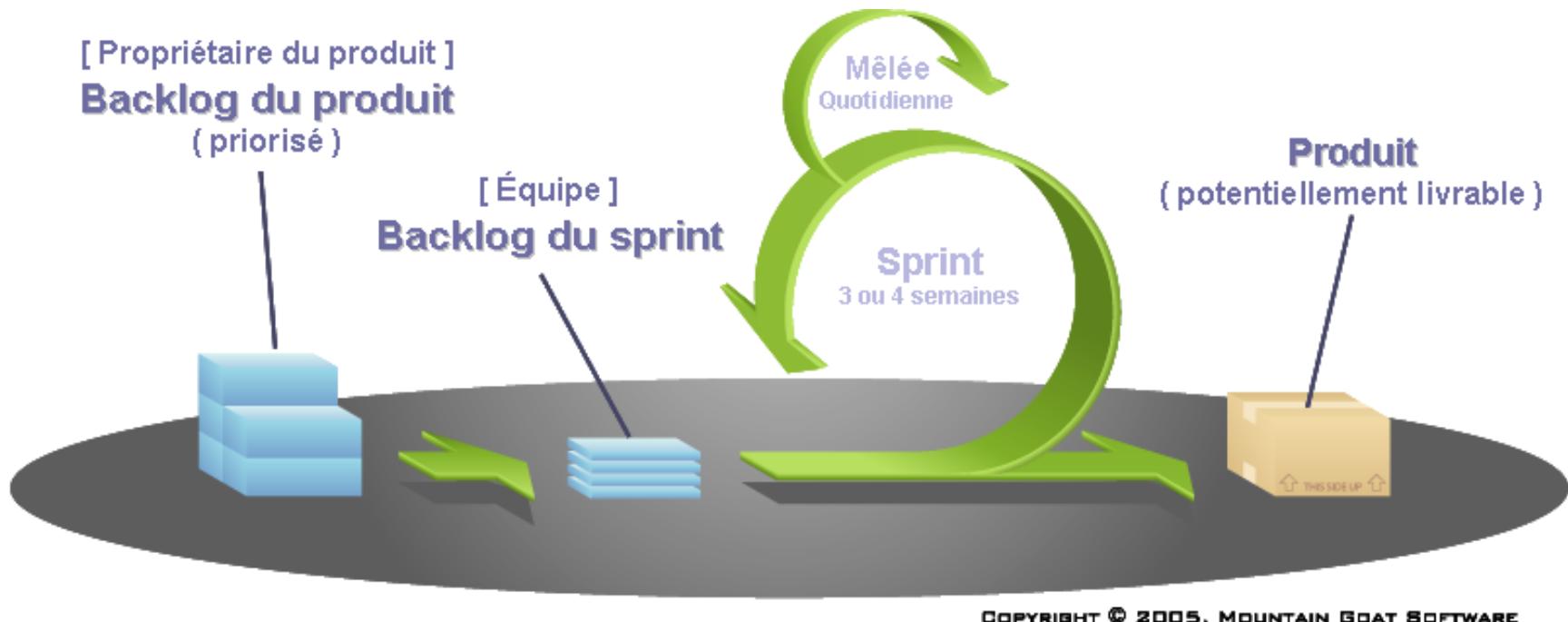


Scrum:

Un cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible
(Définition extraite du Guide Scrum)



Scrum : une méthodologie orientée Processus



3

Rôles

Product Owner

Scrum Master

Equipe de développement

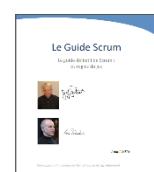
3

Cérémonies

Sprint Planning

Daily Scrum (Mélée quotidienne)

Sprint Review et retrospective



3

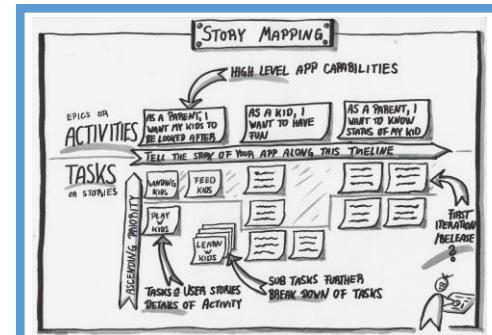
Artefacts

Product Backlog

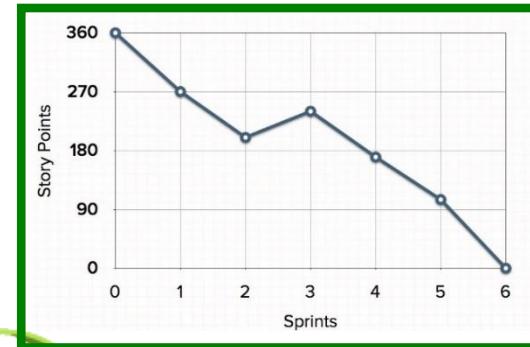
Burndown Chart

Sprint BackLog

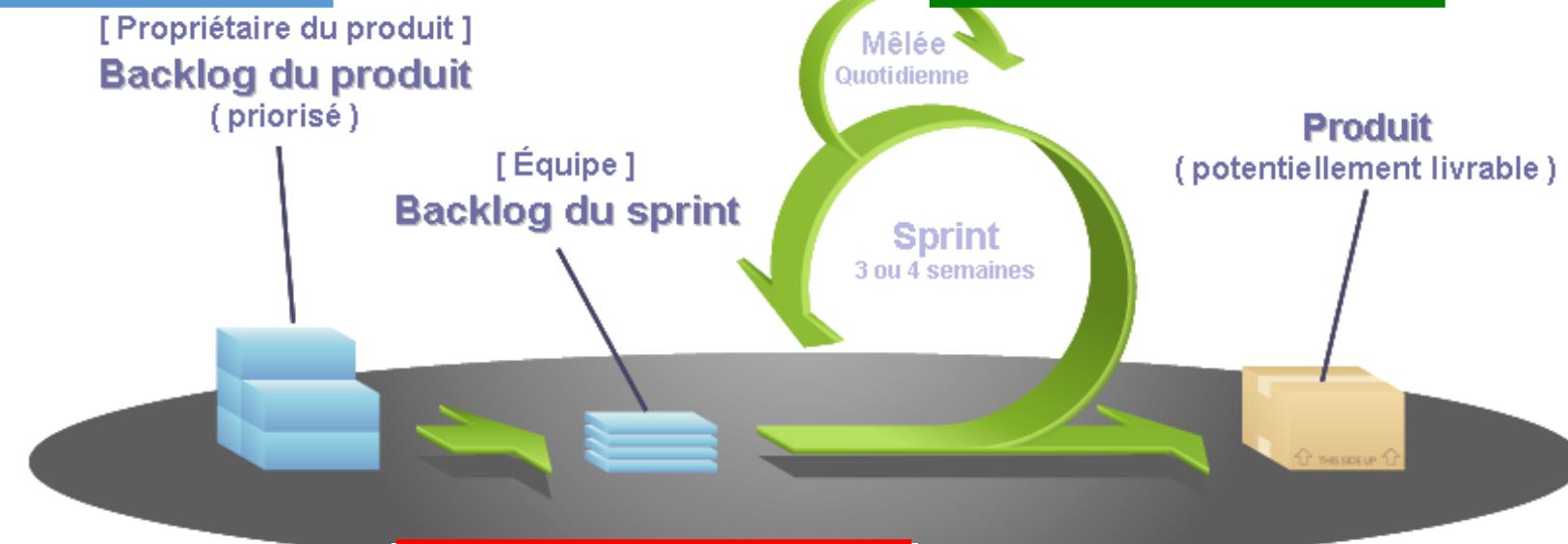
Un management Visuel pour une transparence des artefacts



Product Backlog dans un Story Mapping par exemple



Burndown Chart



Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Test the... 8 Code the... 2 Test the... 8 Test the... 8 Test the... 4	Code the... 4 Test the... 8	Test the... SC 6	Code the... 13 Test the... 4 Test the... SC 4 Test the... SC 6 Test the... SC 6
As a user, I... 5 points	Code the... 8 Code the... 4 Test the... 8 Code the... 6	Code the... 8		Test the... SC 6 Test the... SC 6 Test the... SC 6

Sprint Backlog & Task Board

COPYRIGHT © 2005. MOUNTAIN GOAT SOFTWARE

Transparence des artefacts

Scrum repose sur la **transparence**.

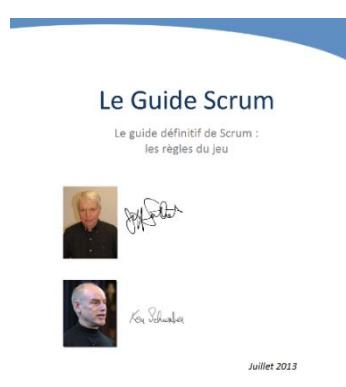
Les décisions pour **optimiser la valeur** et **contrôler le risque** sont prises en se basant sur l'état perçu des artéfacts.



Dans la mesure où la transparence est complète, ces décisions ont une base solide.

Dans la mesure où les artéfacts ne sont pas totalement transparents, ces décisions peuvent être faussées, la valeur moindre et le risque accru.

Extrait de : <http://www.scrumguides.org>



5

Pratiques

Visualiser le flux des travaux

Limiter le nombre de travaux en cours

Gérer le flux

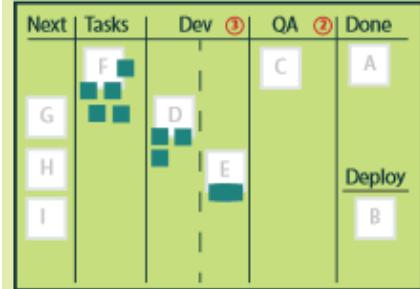
Rendre le processus explicite

Identifier des opportunités d'améliorations

Developed at Toyota in the 1940's, designed as a pull-based, self-stocking system for automobile production. Adopted as a software development management technique in 2003, focusing on "work in progress" limitation to achieve short cycle times (= time from start to finish). The 5 steps:

Visualize the Workflow

1



Identify the stages in your project. Capture them on a board. Each stage is a column. Cards represent the tasks that have to go through in process.

Limit Work-In-Progress

2

Limit the maximum workload on specific stages. Because tasks are pulled from column to column not pushed, each stage can only work on as many tasks as the next stages allow. This way you won't overload your team members and can identify bottlenecks.

Manage Flow

3

Is cycle time going up or down? Where are bottlenecks? Is something stuck somewhere, or should tasks be merged/split up?

Explicit Policies

Identify and define policies for standard tasks, tasks with fixed delivery dates, intangible tasks, etc. Make sure that even exceptional tasks are integrated well into the process.

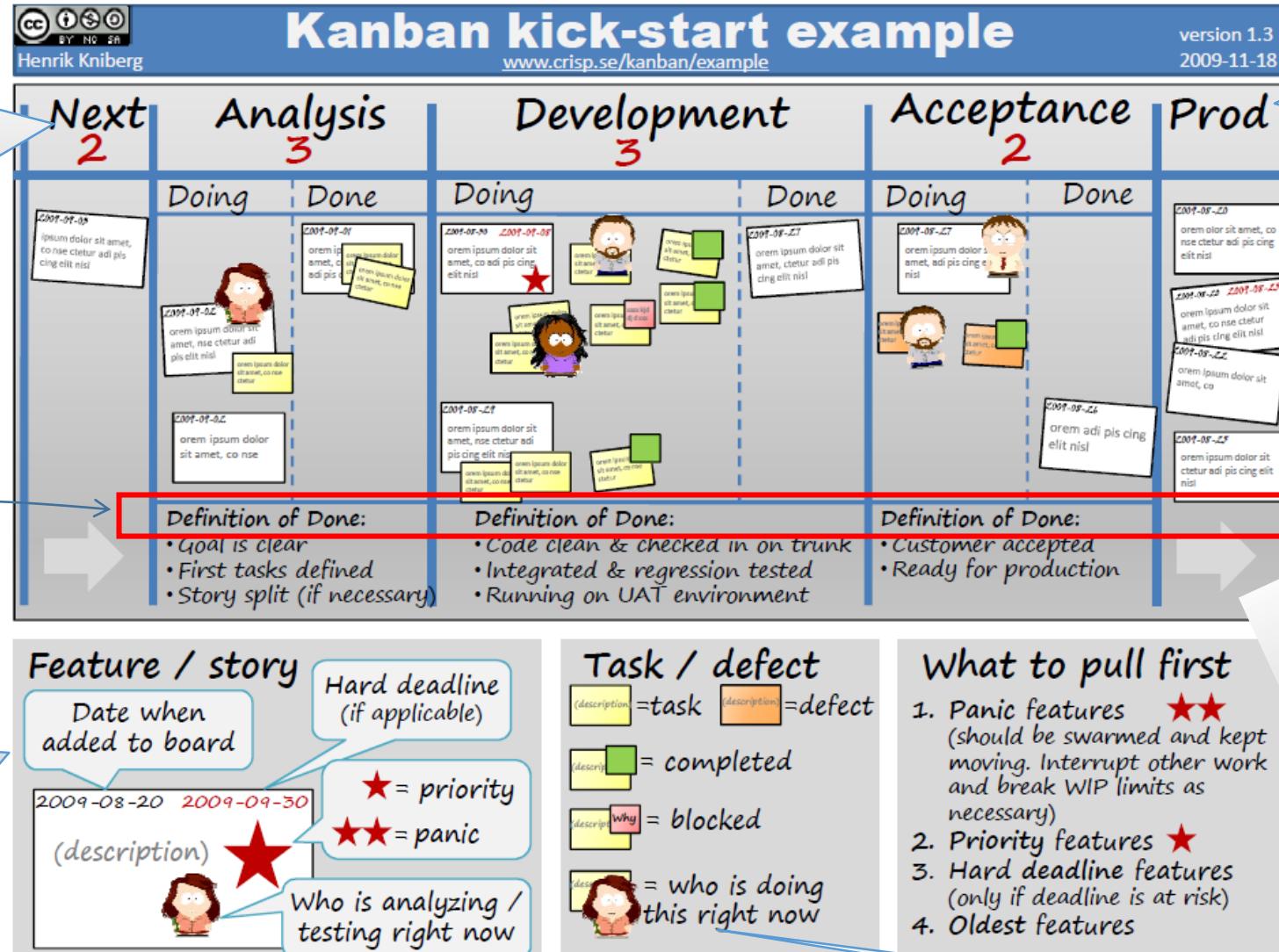


Feedback Loops & Continuous Improvement

5

Talk about what worked and what didn't. Identify bottlenecks and fix them. Kanban is evolutionary and lives of the experiments you do with WIPs, policies, etc. Run the experiments using a scientific method.

« Kanbaniser » et personnaliser votre radiateur d'informations



Extrait : <http://www.crisp.se/file-uploads/kanban-example.pdf>

A voir aussi : <http://toolsforagile.com/blog/archives/1045/infothographic-ingredrients-of-kanban>

<http://www.infoq.com/minibooks/kanban-scrum-minibook>

Isabelle BLASQUEZ

Et dans la vraie vie ...



Olivier Marquet @omarquet · 19 mai

Un beau scrum board c'est un board qui attire et qui donne envie aux personnes de lire son contenu et de l'utiliser



Extrait : <https://twitter.com/omarquet/status/733321369233043457>

XP : une méthodologie proposant aussi des pratiques techniques

5

Valeurs

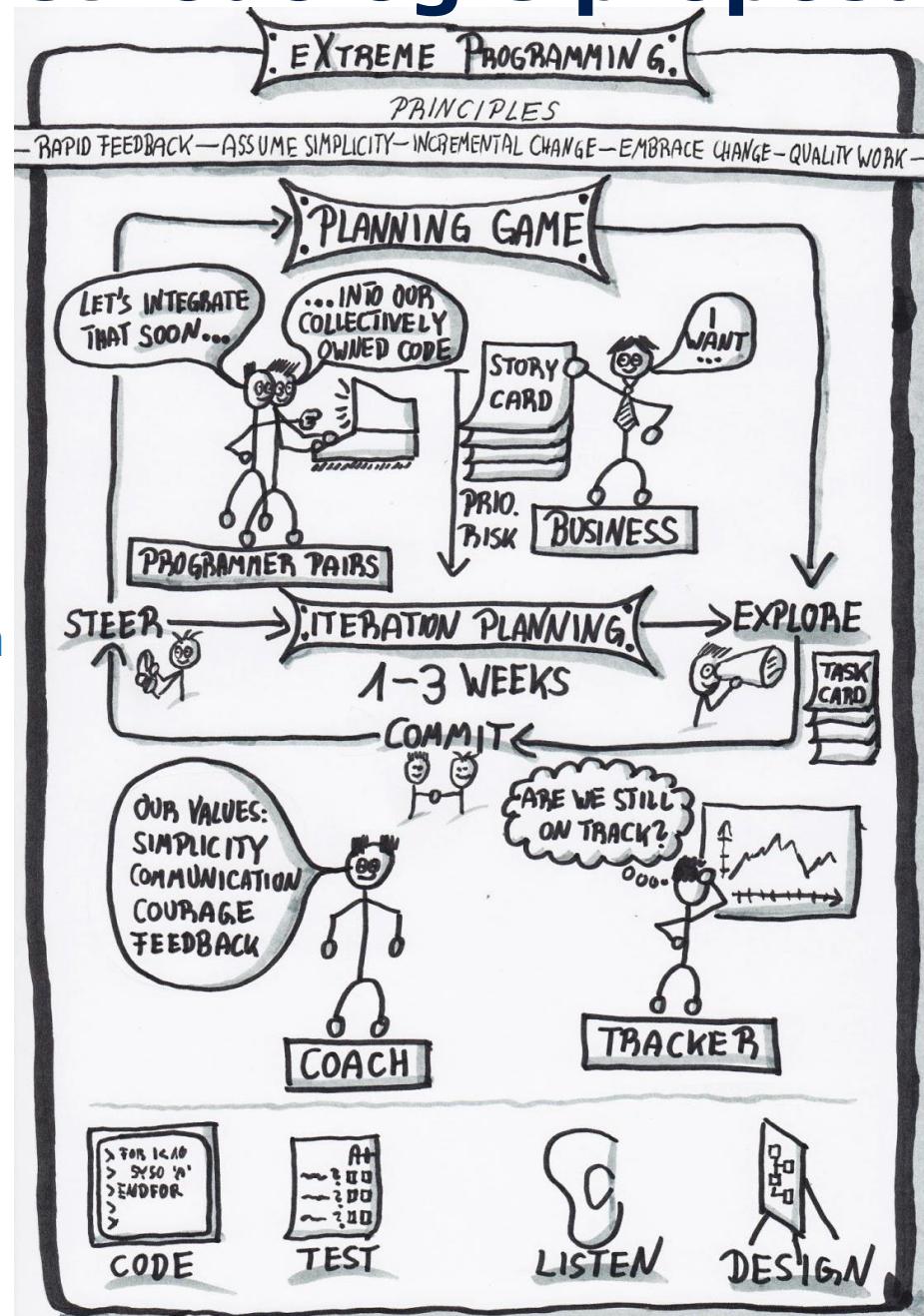
Communication

Simplicité

Feedback

Courage

Respect



13

Pratiques

Client sur site

Jeu du Planning ou Planning poker

Intégration continue

Petites livraisons

Rythme soutenable

Tests de recette (ou tests fonctionnels)

Tests unitaires

Conception simple

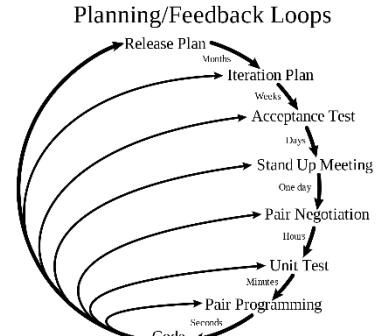
Utilisation de métaphores

Refactoring (ou remaniement du code)

Appropriation collective du code

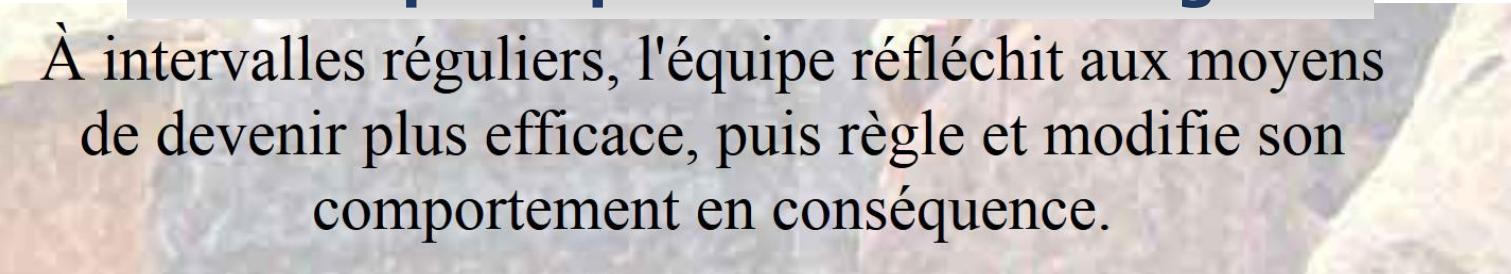
Convention de nommage

Programmation en binôme



L'amélioration continue (**Kaizen**) : au cœur de l'agilité

12ème principe du manifeste agile



À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

- Scrum préconise la **rétrospective**,
- le Lean veille à la **suppression du gaspillage**,
- Kanban se marie très bien à la **théorie des contraintes**, etc.
- Extreme Programming met en avant les **boucles de feedback**
- ...

L'amélioration en agilité revêt plusieurs formes, les petites avancées typiques du Kaizen constituent un puissant levier de transformation, que ce soit aux niveaux personnel, équipe, organisation.



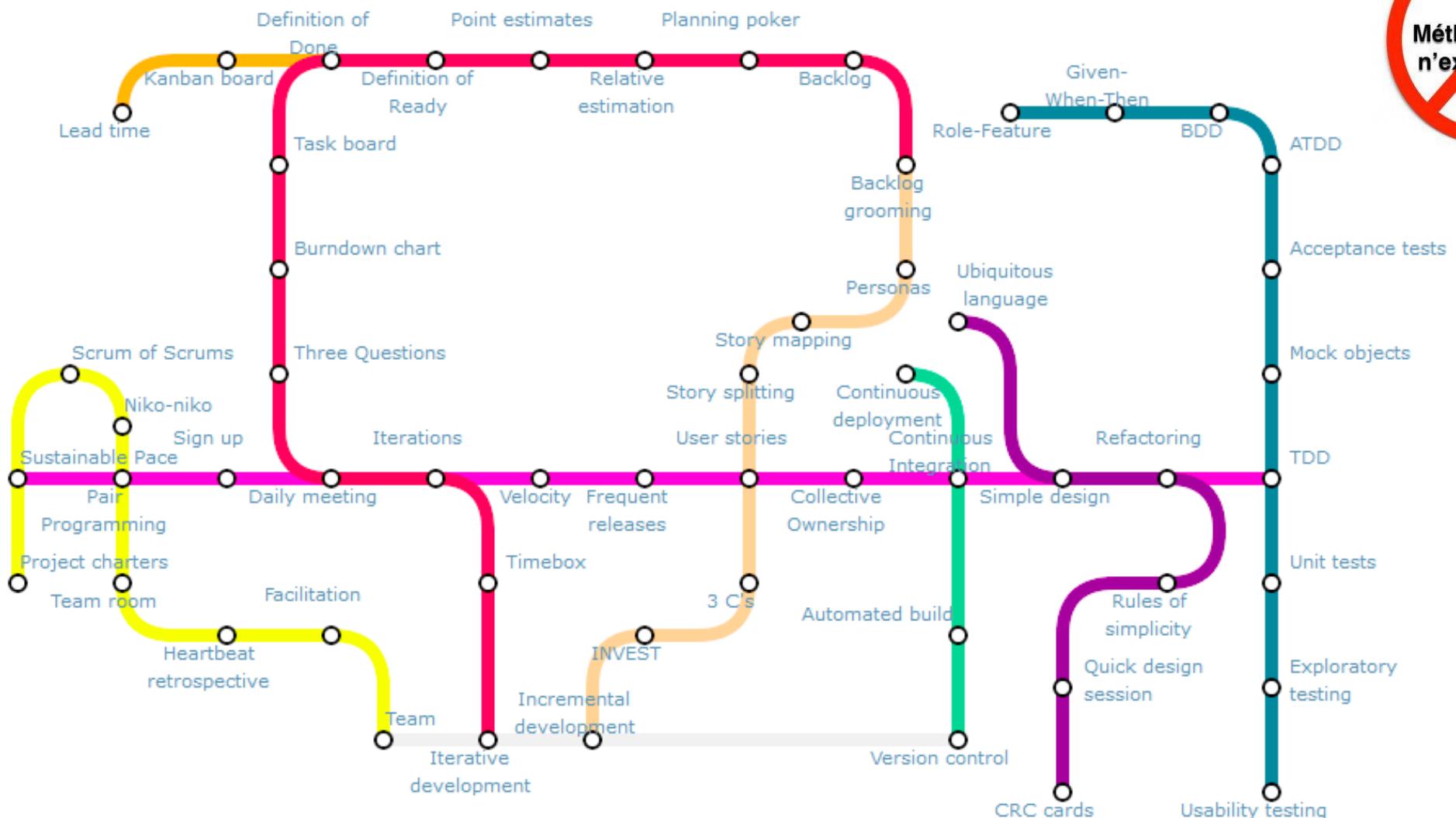
Image : http://blog.soat.fr/wp-content/uploads/2016/01/Kaizen-2.svg_.png

Voir aussi : <http://thierrycros.net/?post/2014/02/04/Le-pacte-Kaizen>

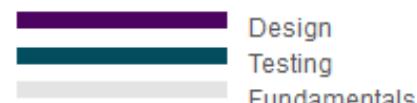
Isabelle BLASQUEZ

Alors le développement agile, C'est quoi ?

Ecosystème des techniques et pratiques agiles...



Lines represent practices from the various Agile "tribes" or areas of concern:

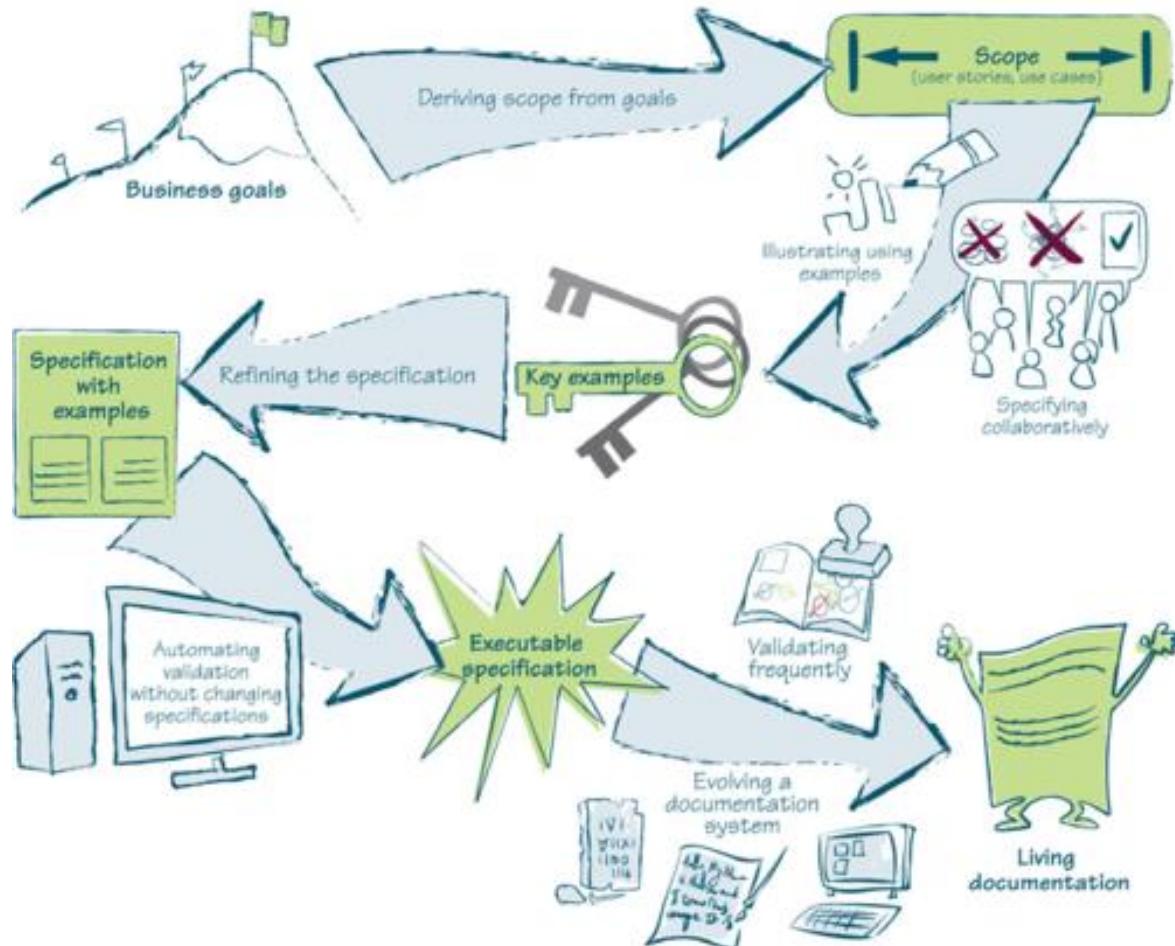


qui implémentent les valeurs et principes du manifeste agile ...

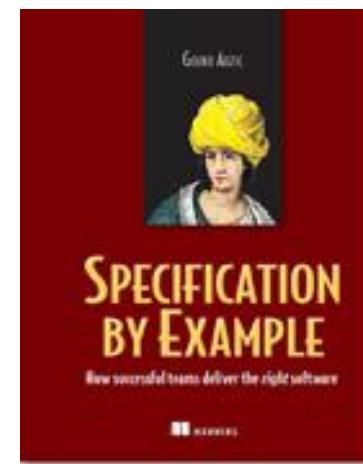


... pouvant être complété par une approche de spécification par l'exemple ...

Key process patterns of Specification by Example



Cheminement de l'**objectif** (métier)
à une **documentation vivante**
au travers d'un ensemble de 7 patterns,
qui permet de s'assurer que
le « bon » produit (right product)
sera effectivement livré.

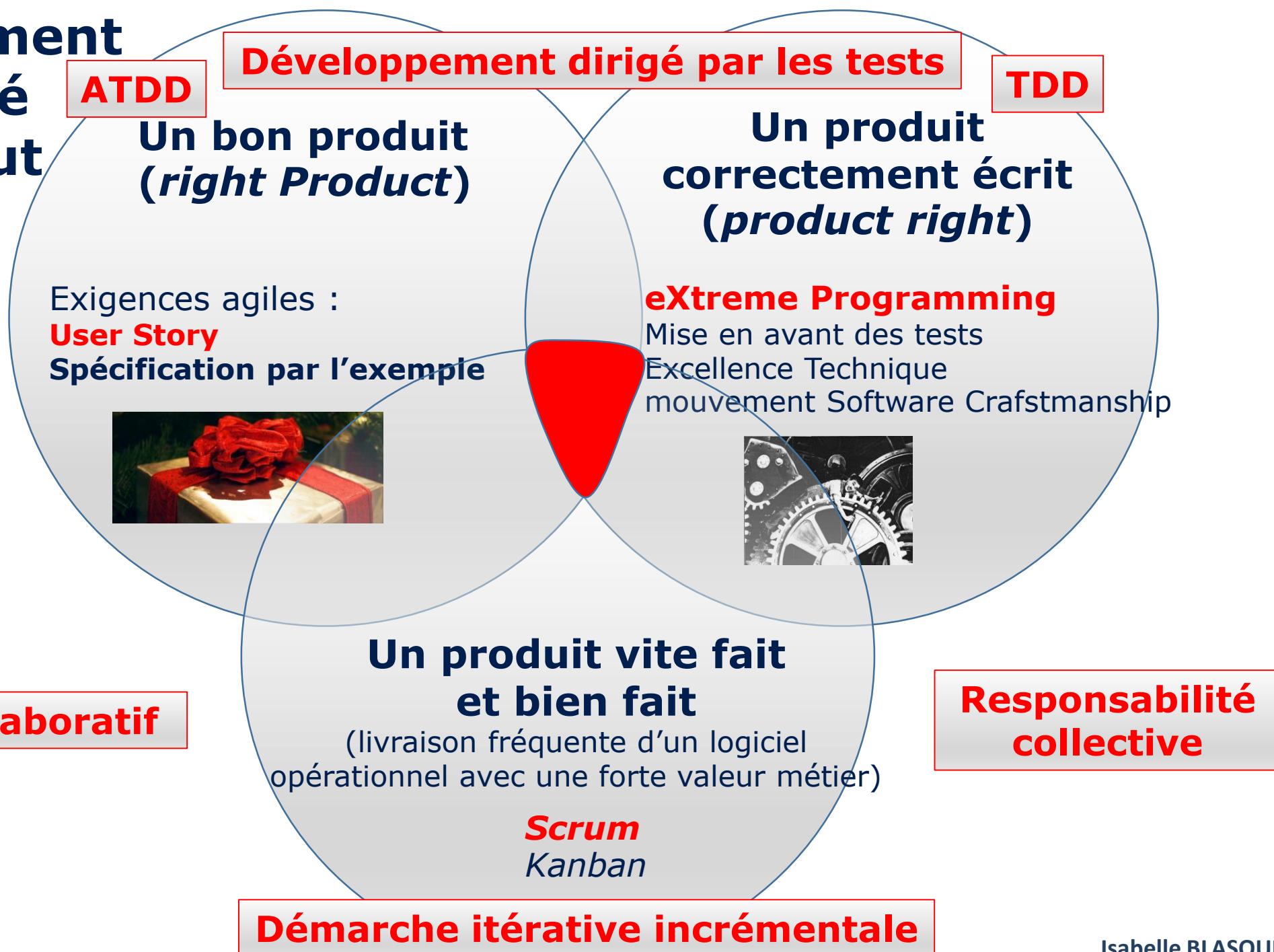


Specification by example (SBE) is a collaborative approach to defining **requirements** and **business-oriented functional tests** for software products based on capturing and illustrating requirements using **realistic examples** instead of abstract statements.

Isabelle BLASQUEZ

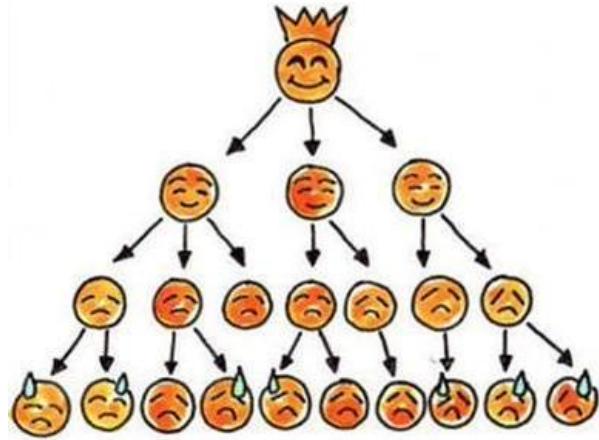
Un développement agile de qualité

C'est avant tout



Le tout dans un nouvel état d'esprit ... à la conquête d'une nouvelle culture d'entreprise...

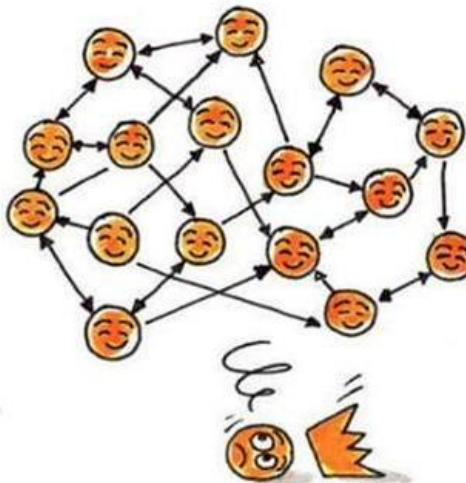
Depuis toujours, c'est ainsi...



<https://twitter.com/kaizenmag/status/551055626005401600>

Co-création
Co-décision
Collaboration

Mais un jour peut-être !



les règles de l'**INTELLIGENCE COLLECTIVE**



ÉCOUTER avec **ATTENTION**



PARLER avec **INTENTION**



ÊTRE **BIENVEILLANT**



SE FAIRE **CONFiance**



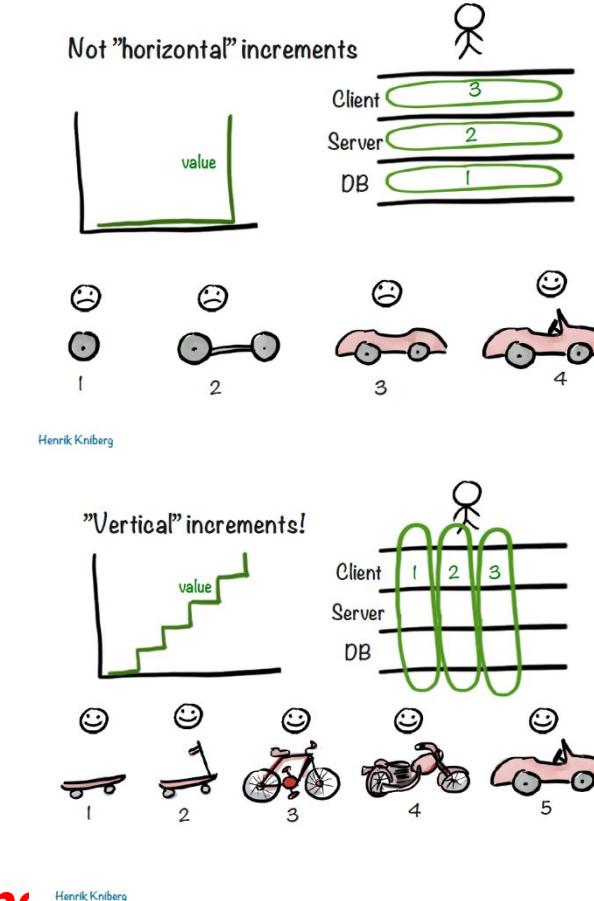
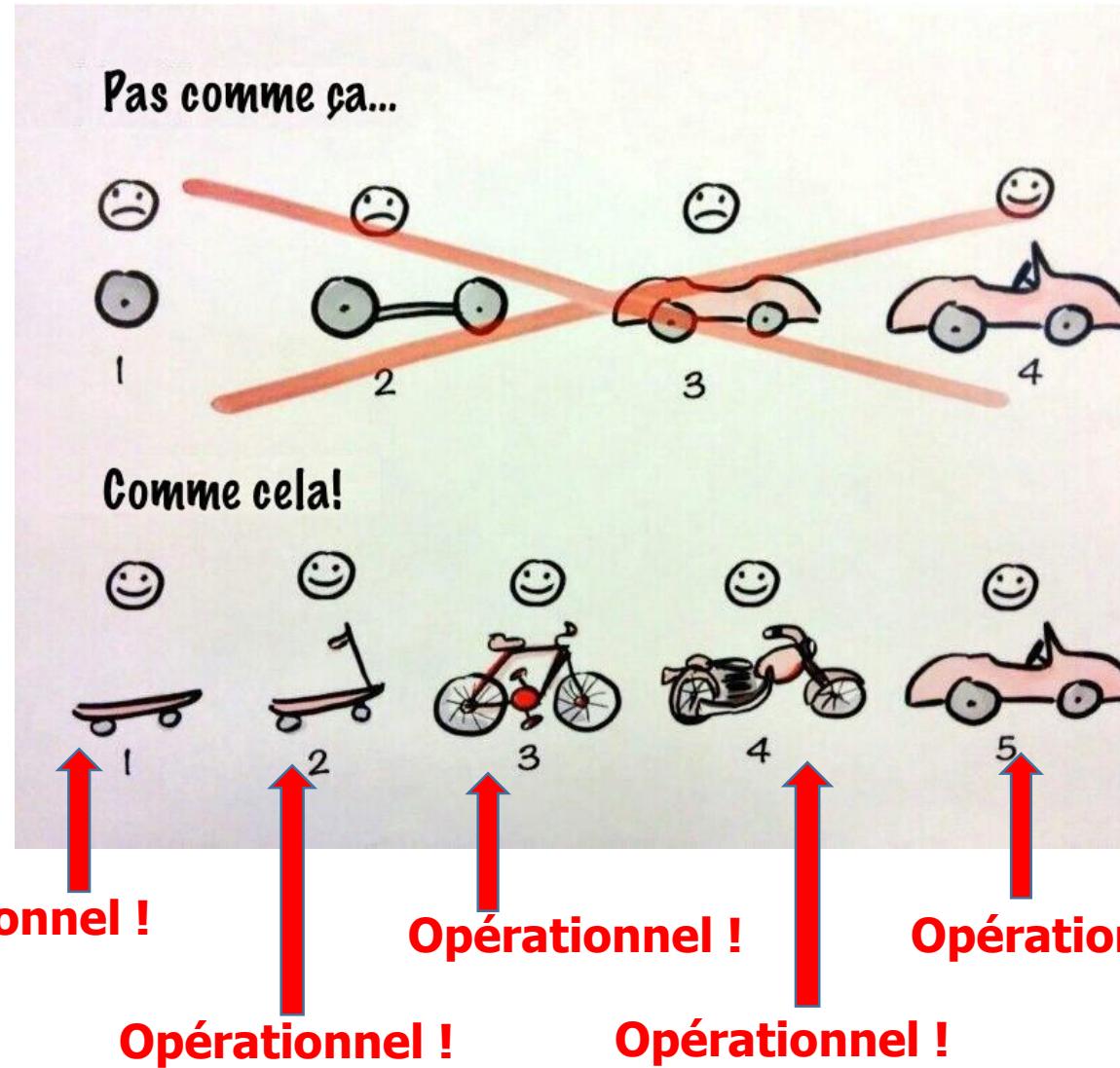
RESPECTER le **CADRE**

CC-BY-NC-SA

@heleneponville
& www.recompose.it

<http://www.recompose.it/2014/11/26/principes-intelligence-collective/>

Le tout en mettant en œuvre un développement itératif et incrémental ... avec de la valeur métier et un feedback positif !



Extrait : <http://ayeba.wikispaces.com/Développement+d'un+Produit>

<http://fr.slideshare.net/RichardPDoerer/what-is-agile-henrik-kniberg-august-2013>

Isabelle BLASQUEZ

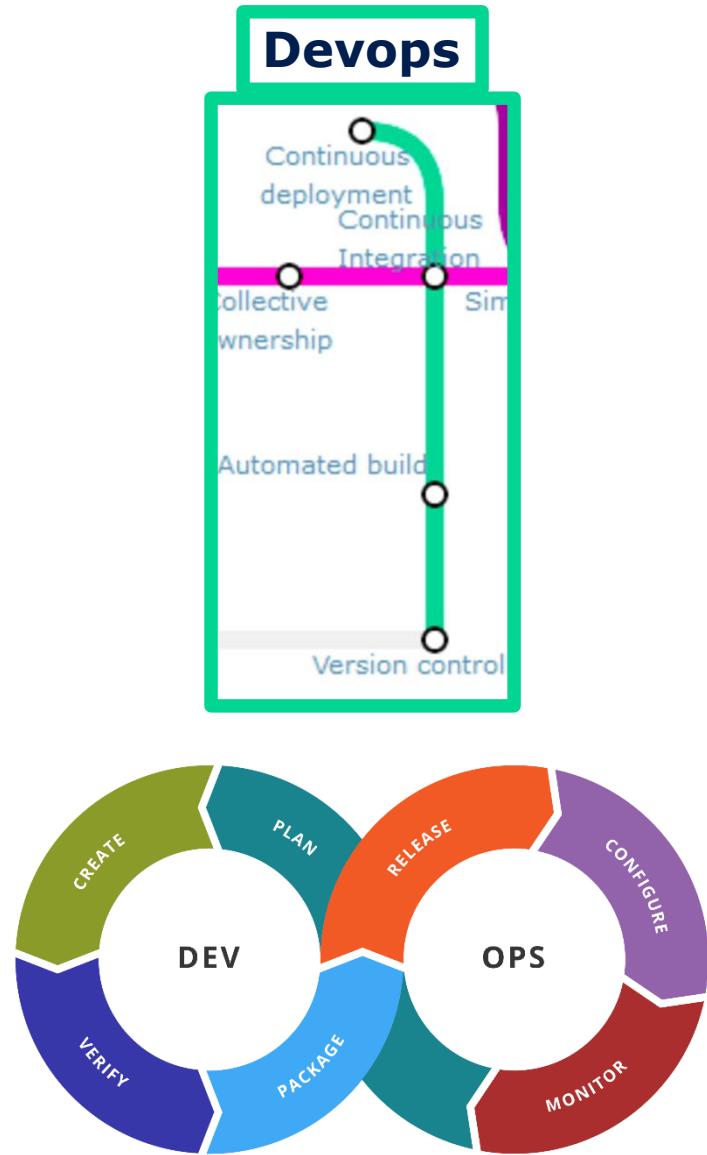
***Un développement itératif et incrémental ...
Oui, mais bien fait ... avec de la valeur métier !
(autre exemple...)***



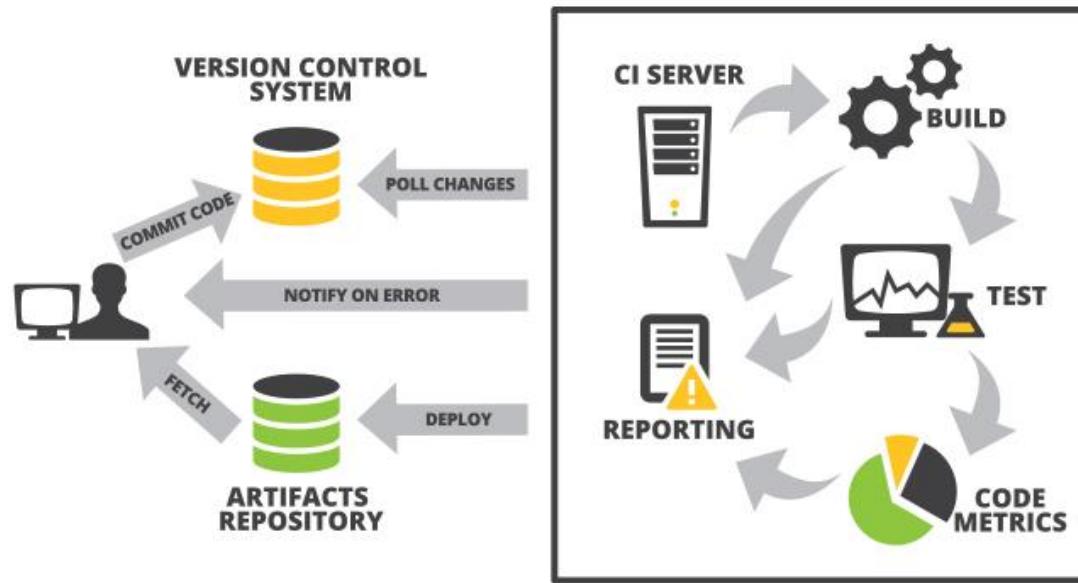
Illustration avec un côté plus réaliste
sur l'***aspect opérationnel***, car ...

....Même utilisation mais
expérience client amélioré à chaque fois.

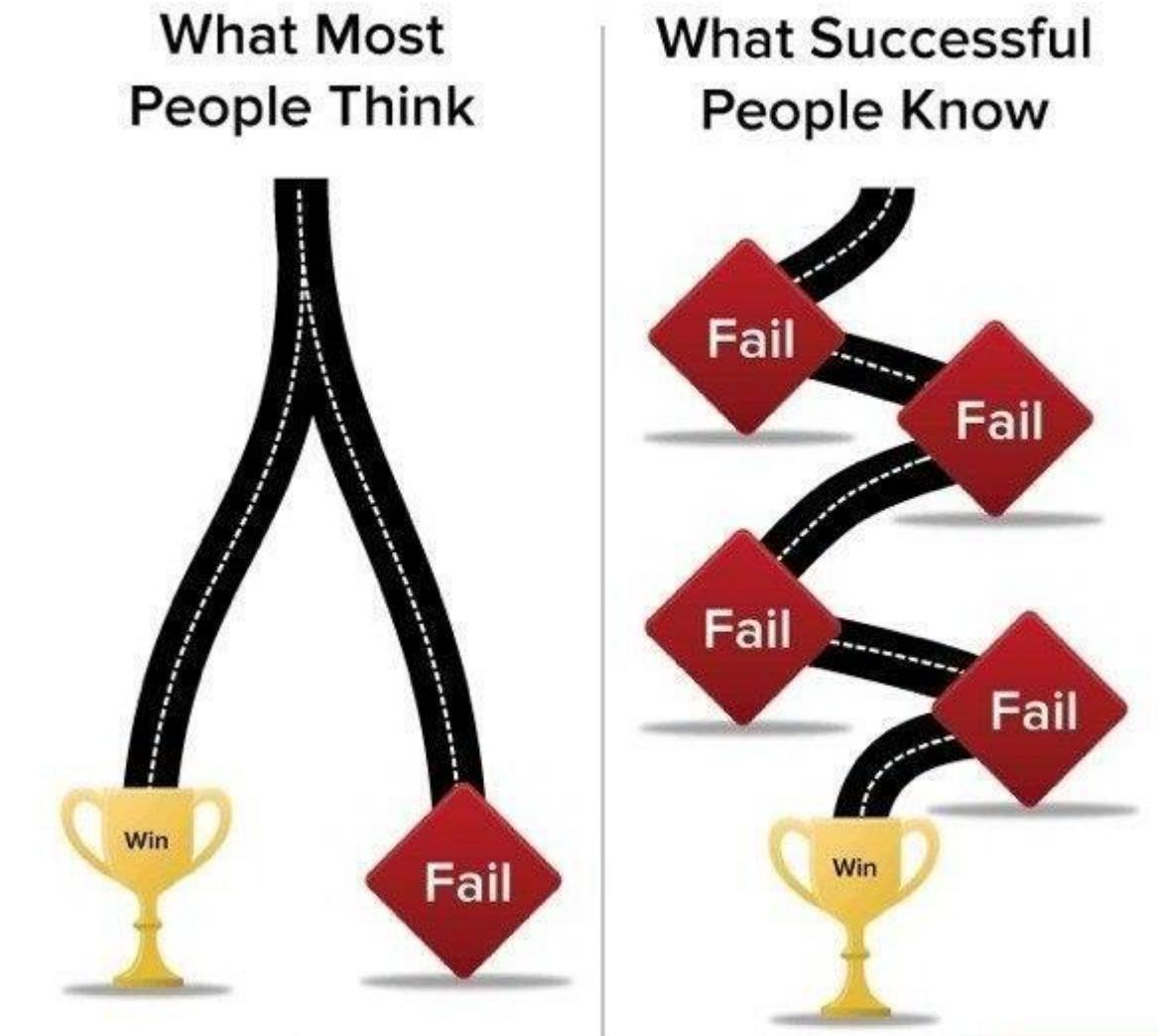
Etre agile : c'est raccourcir la boucle de feedback... ... et mettre en place de lignes de production *agile*



- ✓ Test en continu
- ✓ Intégration continue
- ✓ Inspection continue
- ✓ Déploiement continu



Car essai/Erreur : un chemin vers la réussite ... (about #Success and #Failure)

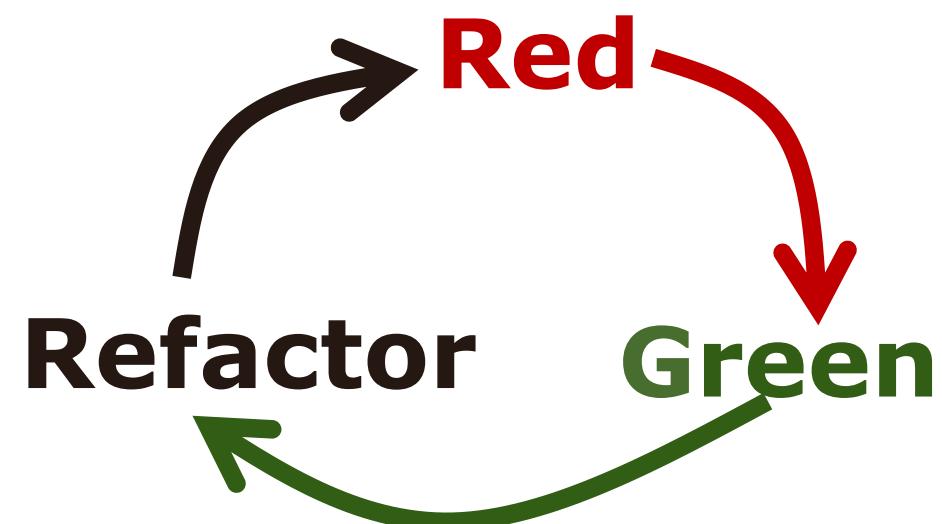
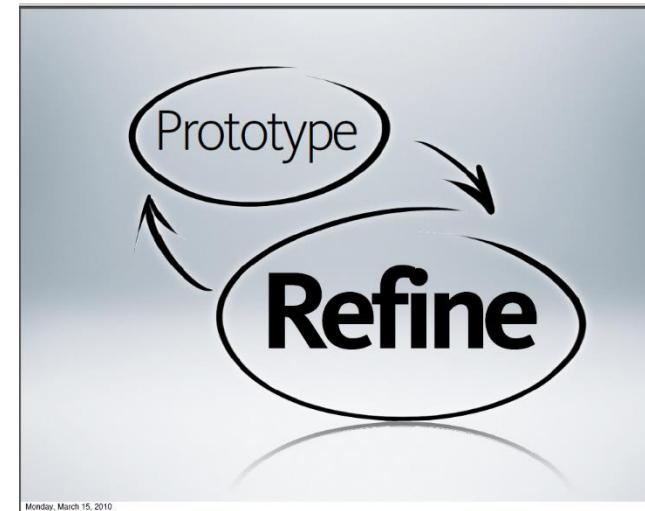
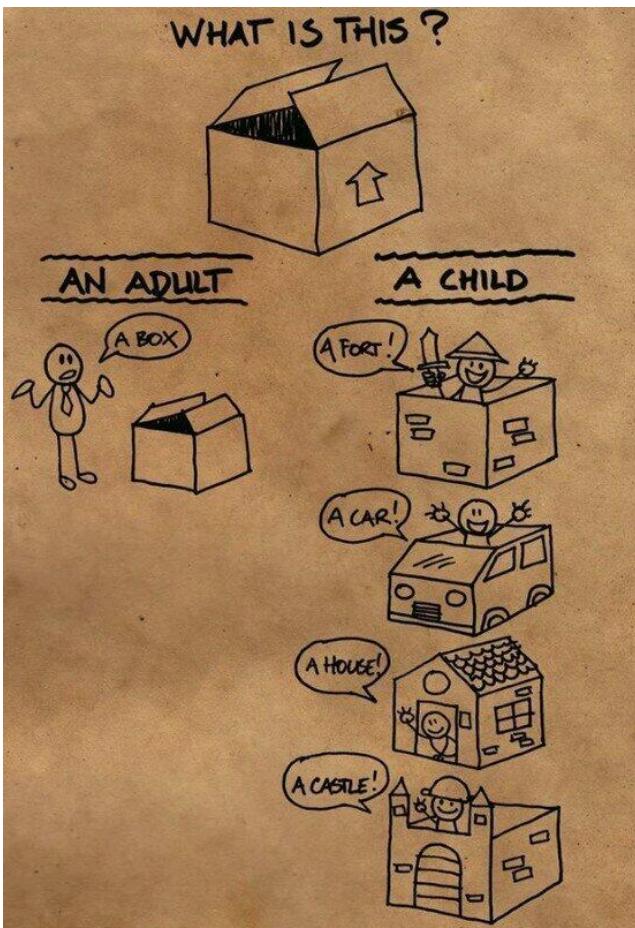


Source : <https://twitter.com/MatthieuGarde/status/420123879609499648/photo/1>

Isabelle BLASQUEZ

Et le prototypage libère la créativité (les idées) donne **confiance** et l'envie d'essayer

If you want to be more creative,
try to see the world through
a child's eyes

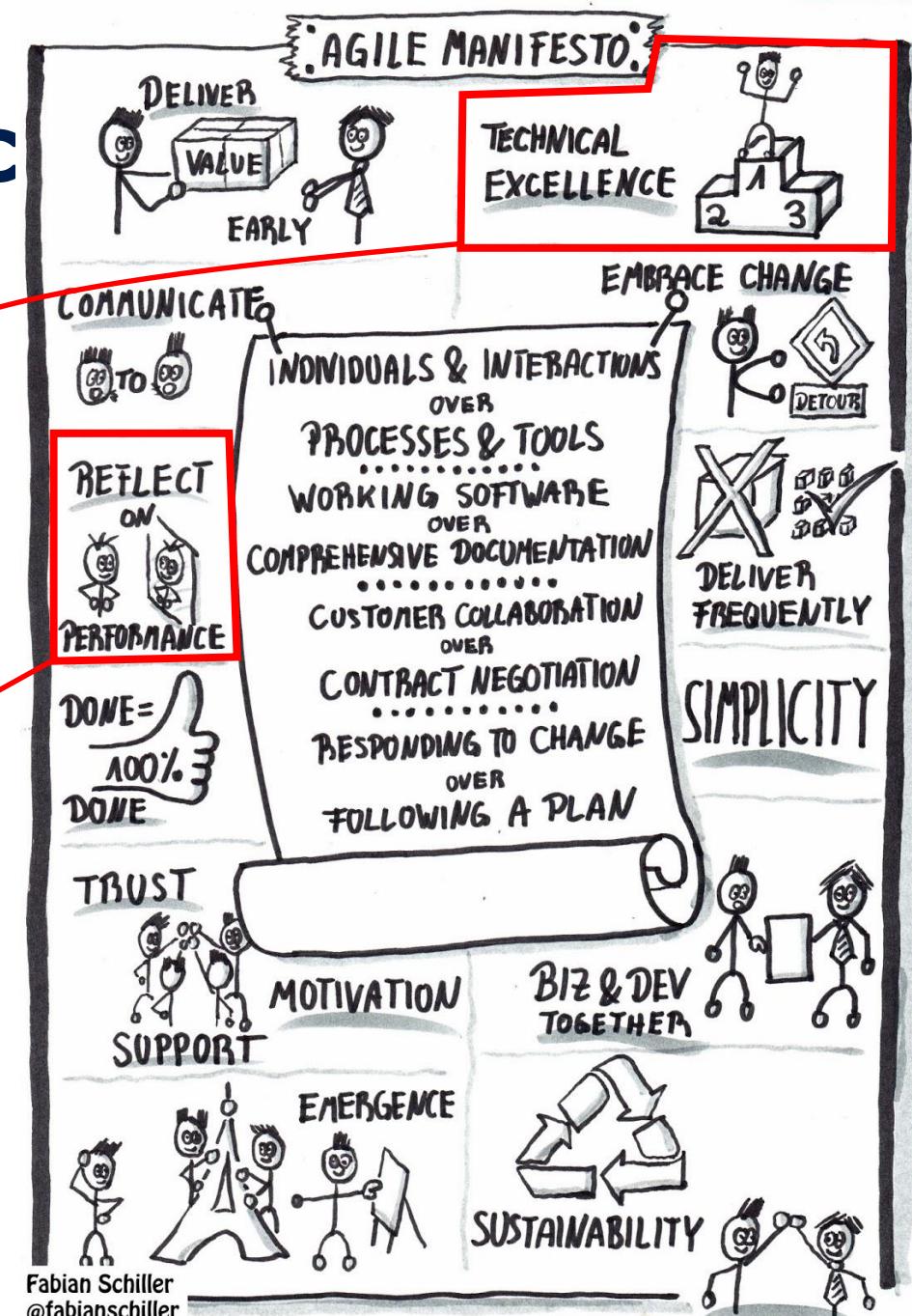


... Outre le classique manifeste agile de 2001 avec ses 4 valeurs et 12 principes

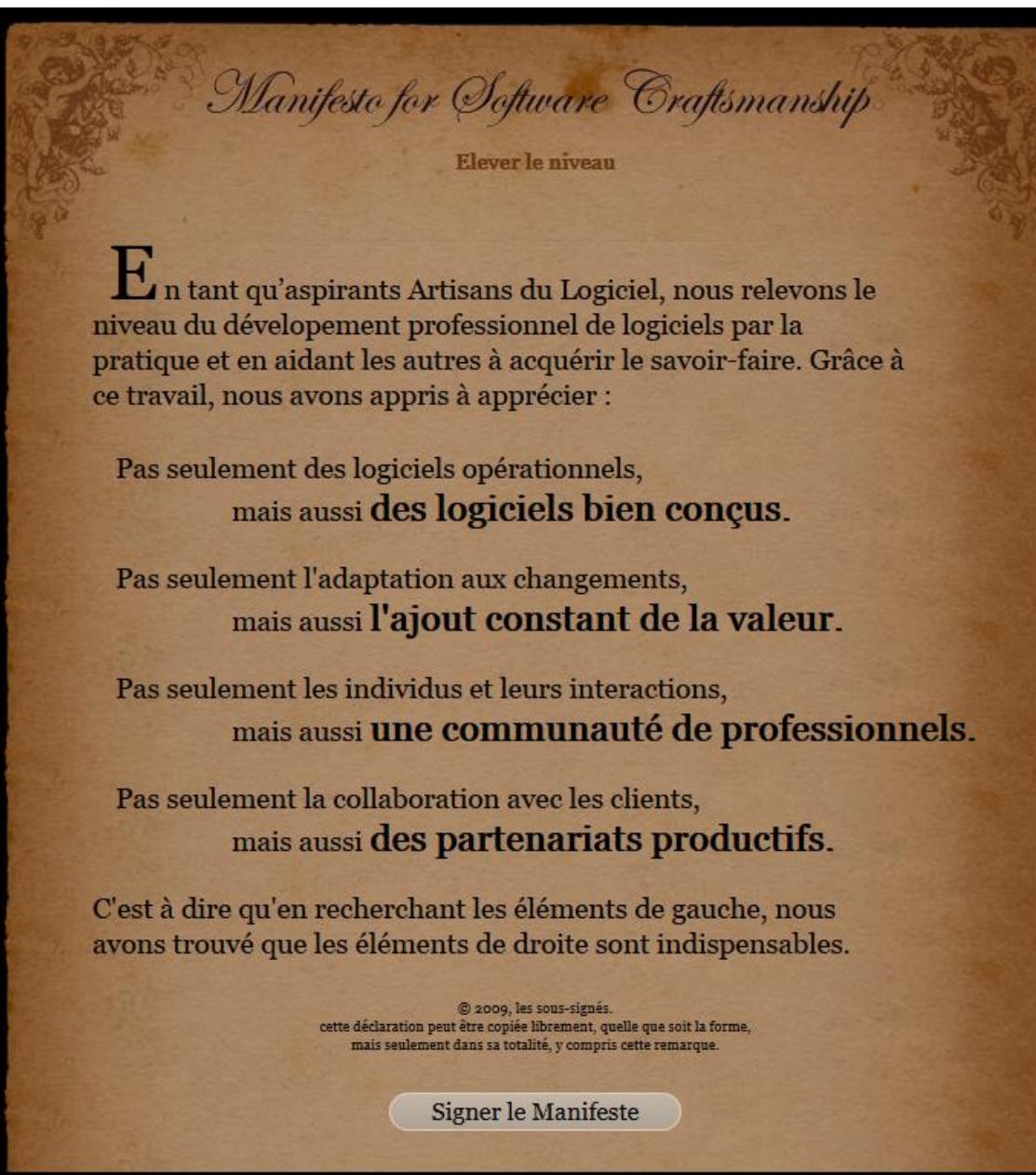
Une attention continue à l'excellence technique et
à une bonne conception renforce l'Agilité.

...

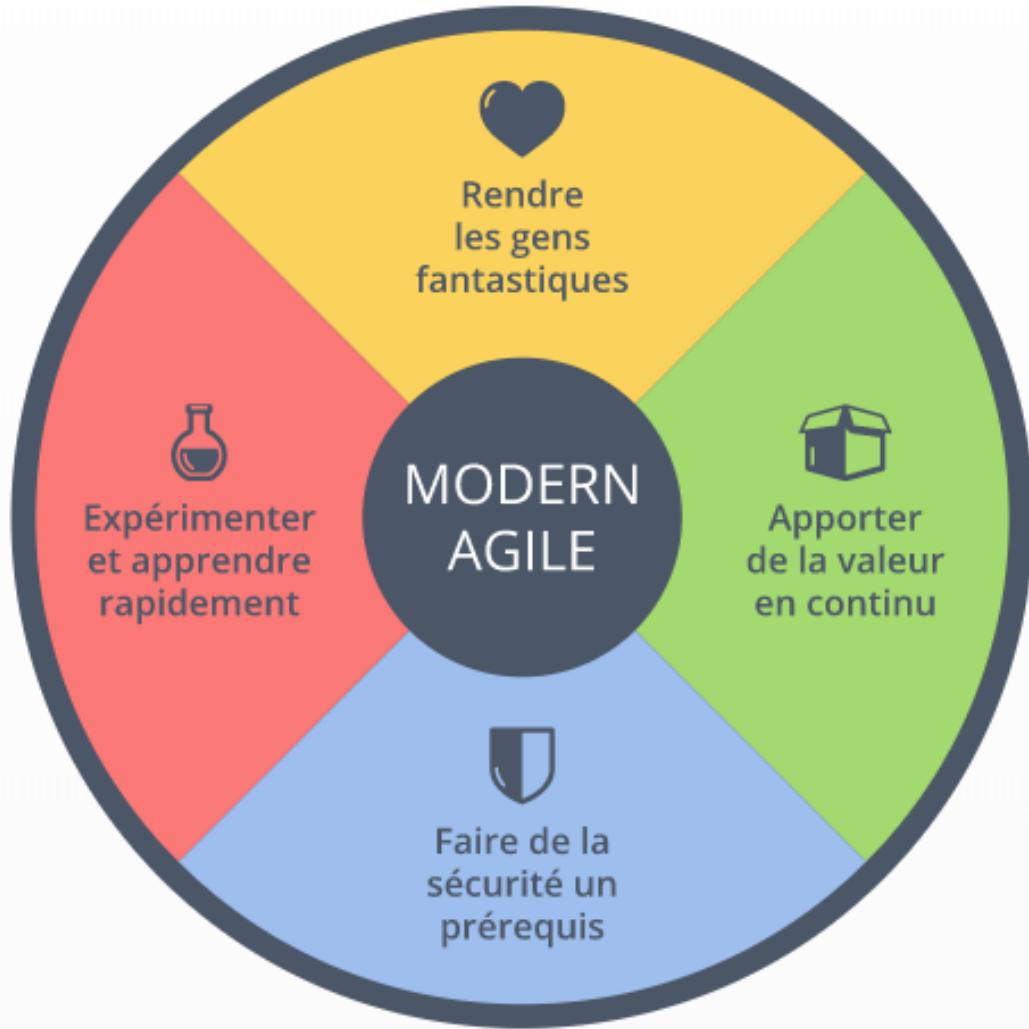
À intervalles réguliers, l'équipe réfléchit aux moyens
de devenir plus efficace, puis règle et modifie son
comportement en conséquence.



... un prolongement avec le Software Craftsmanship (2008)



... et une nouvelle « itération » du manifeste (2016)



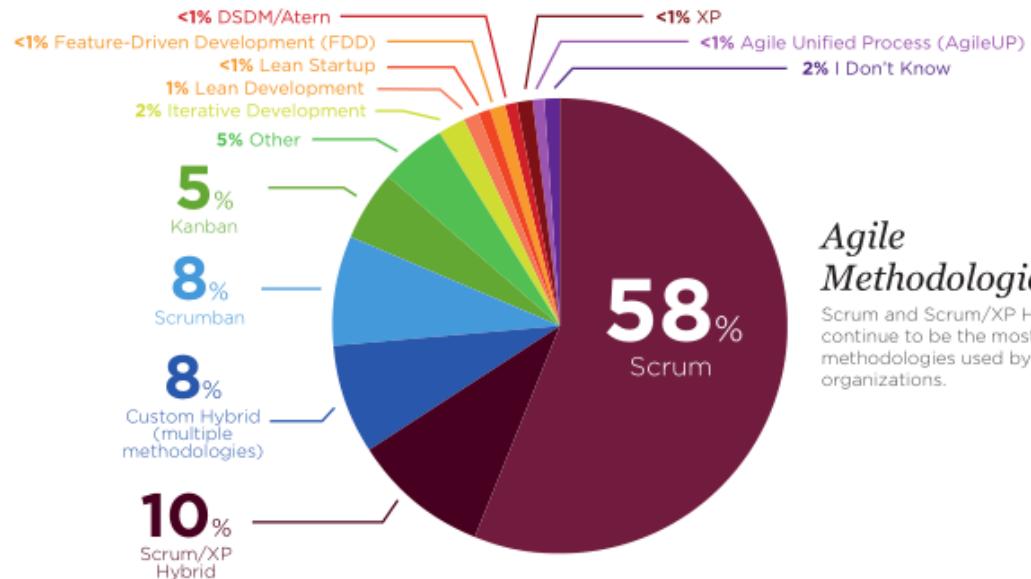
World famous organizations like Google, Amazon, AirBnB, Etsy and others are living proof of the power of these four principles. However, you don't need to be a name brand company to leverage modern agile wisdom.

Pour suivre l'évolution du développement agile (1/2)

Consultez le rapport annuel de VersionOne
<http://stateofagile.versionone.com/>



AGILE METHODS AND PRACTICES



Agile Methodologies Used

Scrum and Scrum/XP Hybrid (68%) continue to be the most common agile methodologies used by respondents' organizations.

TOP 5 AGILE TECHNIQUES



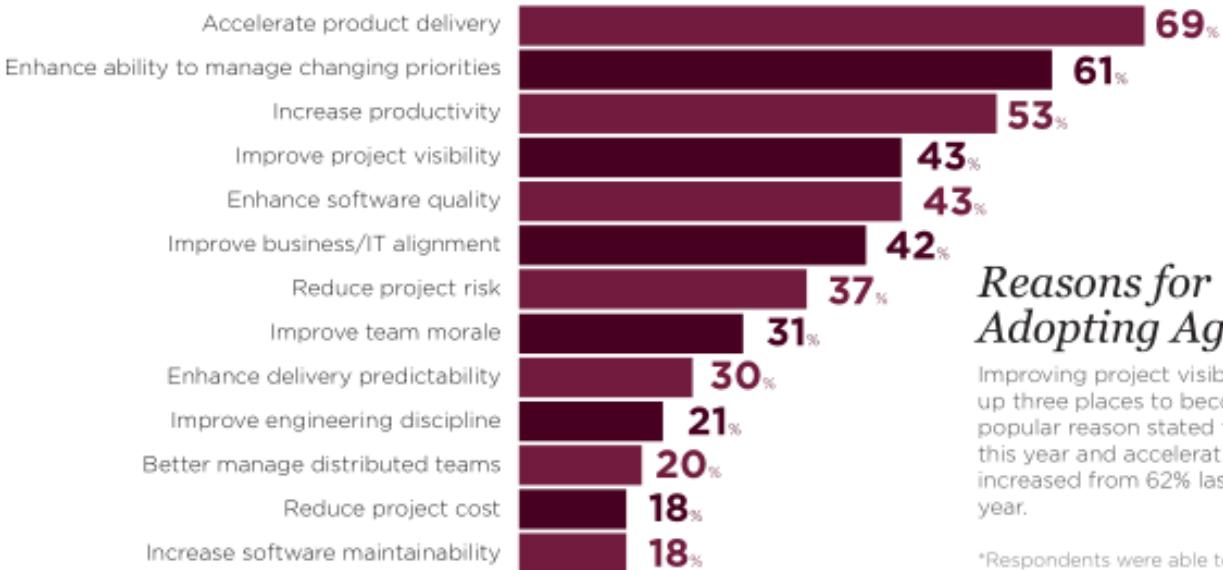
Agile Techniques Employed

From 2015 to 2016, the use of Kanban grew from 39% to 50%; iteration reviews increased from 54% to 81% and iteration planning went from 69% to 90%.



*Respondents were able to make multiple selections.

Pour suivre l'évolution du développement agile (2/2)



Reasons for Adopting Agile

Improving project visibility (43%) moved up three places to become the fourth most popular reason stated for adopting agile this year and accelerating product delivery increased from 62% last year to 69% this year.

*Respondents were able to make multiple selections.

Engineering Practices Employed

While the usage of XP as an independent methodology continues to decrease (<1%), the practices associated with XP are still prevalent.

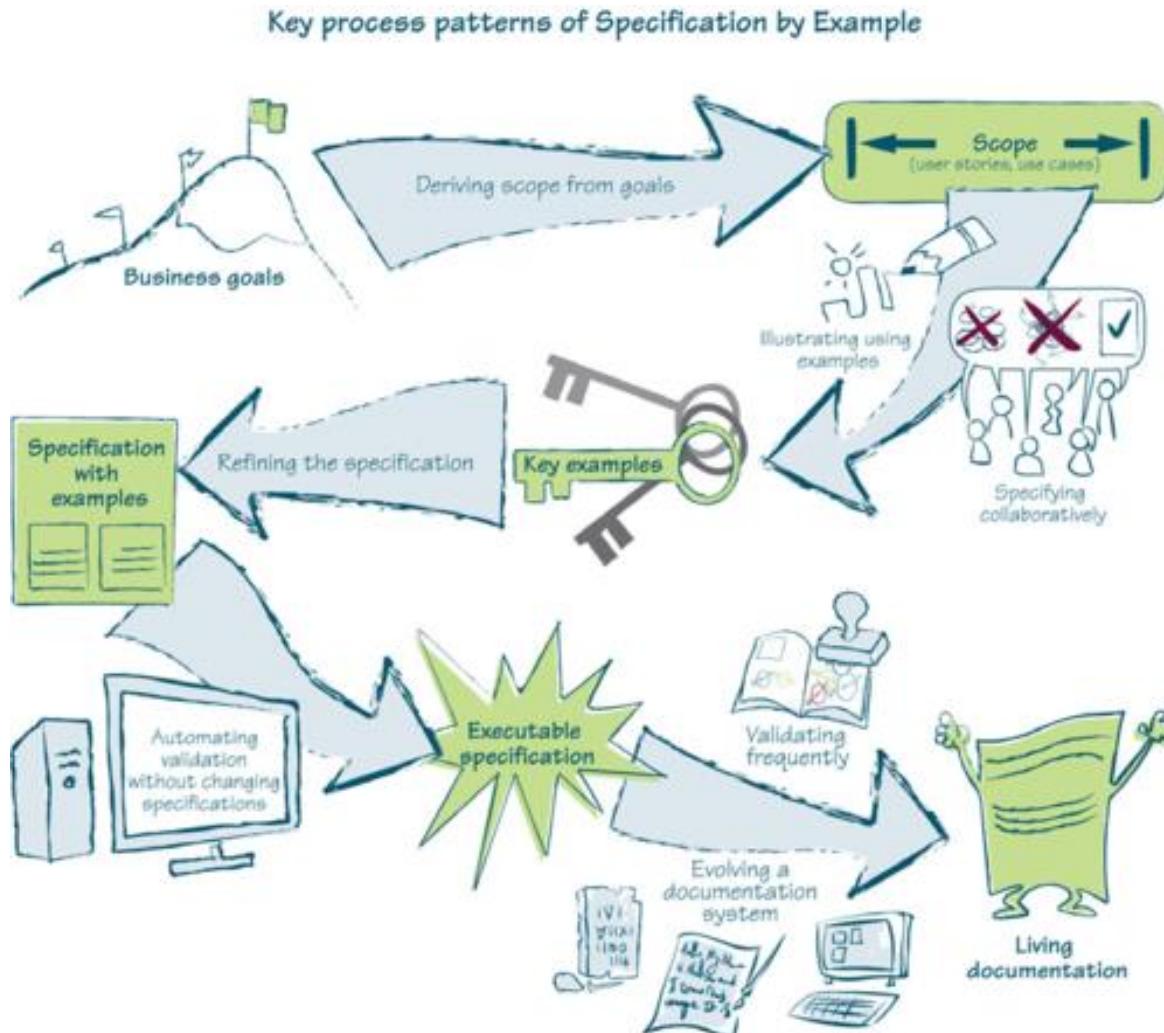


*Respondents were able to make multiple selections.

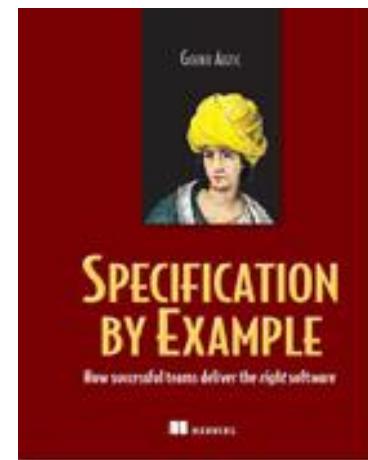
Dans le module M3301 ...



Une démarche *agile* de spécification par l'exemple ...



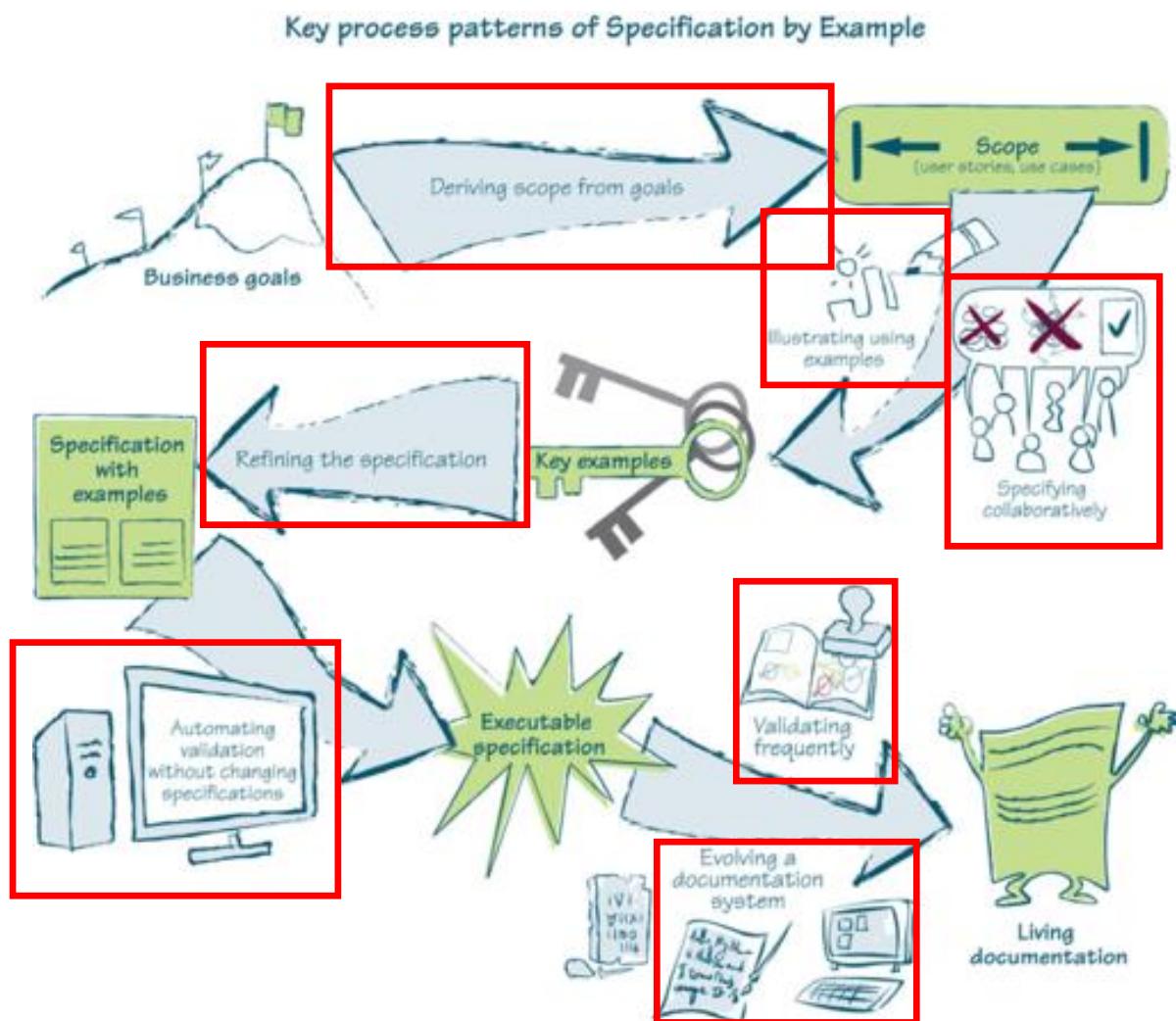
Cheminement de l'**objectif** (métier) à une **documentation vivante** au travers d'un ensemble de 7 patterns, qui permet de s'assurer que **le « bon » produit (right product)** sera effectivement livré.



Specification by example (SBE) is a collaborative approach to defining **requirements** and **business-oriented functional tests** for software products based on capturing and illustrating requirements using **realistic examples** instead of abstract statements.

Extrait : https://en.wikipedia.org/wiki/Specification_by_example

Une démarche agile de spécification par l'exemple ...



Deriving scope from goals
Travail autour de la vision

Specifying collaboratively

Compréhension commune. La Collaboration permet aux équipes de produire des spécifications qui sont faciles à comprendre.

Illustrating requirements using example

Exemples précis, complets, réalistes

Refining specifications

Création d'un contexte concret et précis pour le développement et le test

Automating validation without changing specification

Une **Spécification avec des exemples** automatisés (tests) qui est compréhensible et accessible par tous les membres de l'équipe devient une **spécification exécutable**

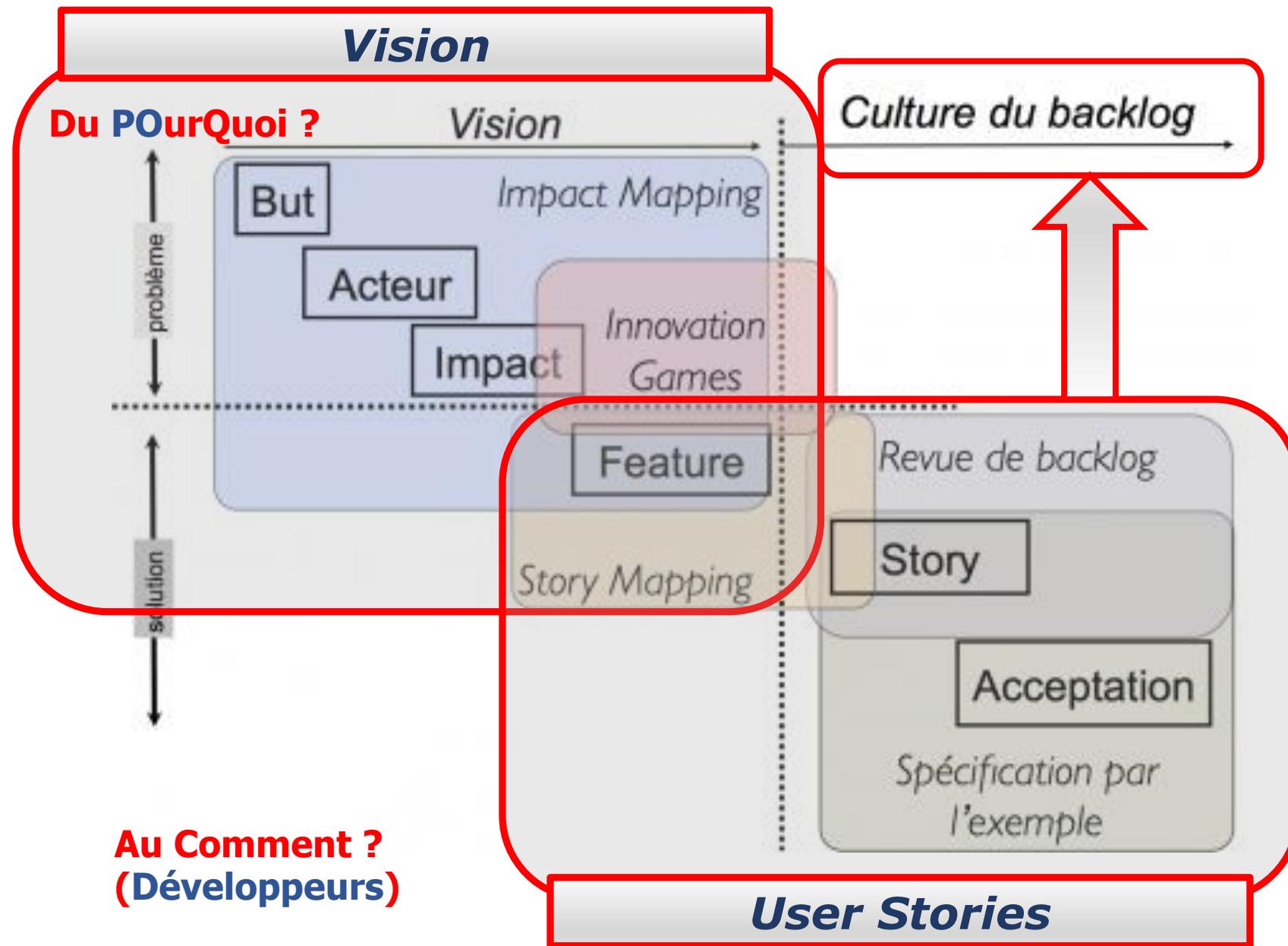
Validating frequently

Evolving a documentation system

Proposer une **documentation vivante** : facile à comprendre, cohérente et organisée

Extrait : https://en.wikipedia.org/wiki/Specification_by_example

... allant de la Vision à l'écriture des User Stories (1/2)



... allant de la Vision à l'écriture des User Stories (2/2)

Establishing a shared understanding

Du PourQuoi ?

Why?

Impact, Goal

Impact Mapping

Capability

Story Mapping

Feature

Epic

User Story

Acceptance Criterion

Example

Specification-By-Example

Isolated,
formalized
example

Reminder
for a
discussion

Bug
report

Code

How?

Au Comment ?
(Développeurs)

Easier to define upfront

Harder to define upfront

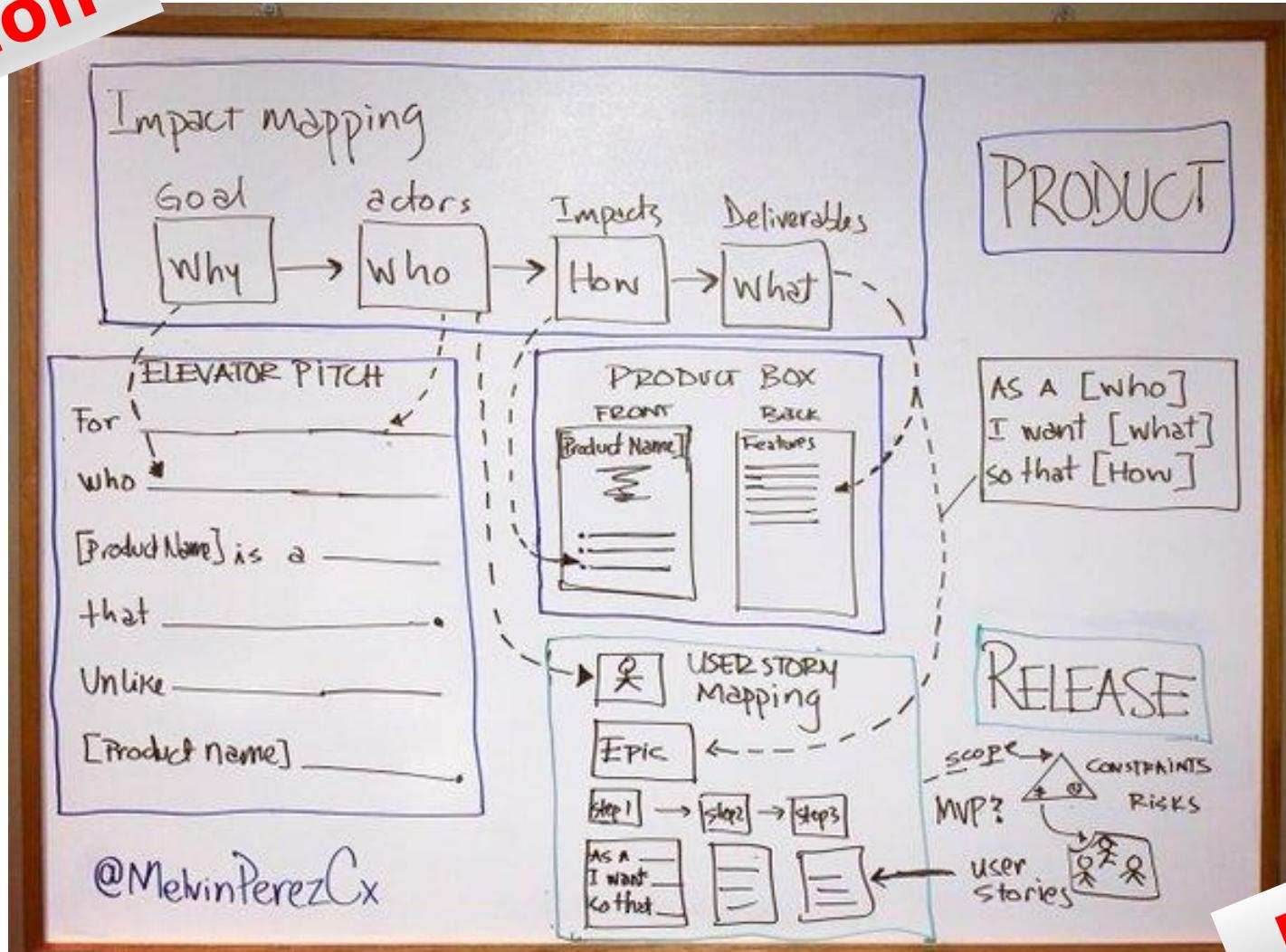


Extrait de « Specification-By-Example with Gherkin »

<http://fr.slideshare.net/chassa/2013-0603specification-byexamplewithgherkinchristianhassa>

... et axée autour des outils du Product Owner (right product)

Vision



User Stories

Les TD/TP permettront de capturer de manière non ambiguë les exigences

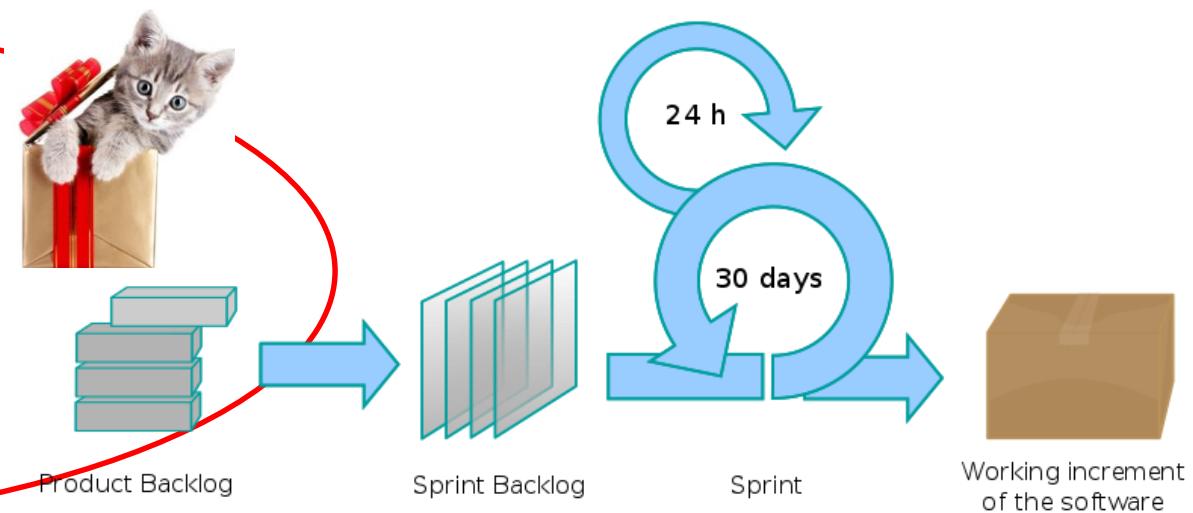
Why ? (Pourquoi ?)
(business problem)

Spécification

What ? (Quoi ?)
(explications claires sur la solution à développer)

- ✓ **Les methodologies agiles** fournissent un framework pour développer des produits et suppose l'existence d'un backlog de produit initial

Mais ... Comment construire une première version d'un "bon" premier backlog de produit correspondant aux exigences du client ?



Une Séquence d'ateliers collaboratifs allant de la vision du produit à l'identification et la description des users stories permettant de développer le *bon* produit

- 
- 1. Construire la vision: De l'objectif aux premières fonctionnalités (features)**
(en découvrant le problème et identifiant les premières features)
 - 2. Affinage des features en epics et découpage en stories**
 - 3. Conversation - Confirmation**
(détailler les user stories,
partager une compréhension commune
avoir un langage métier commun)

Based sur la
Specification par l'exemple et des
techniques d'expérience Utilisateur (UX)

Une Séquence d'ateliers collaboratifs allant de la vision du produit à l'identification et la description des users stories permettant de développer le *bon* produit

Why ?

Elevator Pitch

Product Box

Personnas

Impact Mapping

Story Mapping

Wireframes

Telling and expressing scenarios
in domain language

What ?

1. Construire la vision: De l'objectif aux premières fonctionnalités (features)
(en découvrant le problème et identifiant les premières features)

2. Affinage des features en epics et découpage en stories

3. Conversation - Confirmation
(détailler les user stories,
partager une compréhension commune
avoir un langage métier commun)

Annexes

Réflexion de l'été 2015 sur le(s) mouvement(s) agile(s)

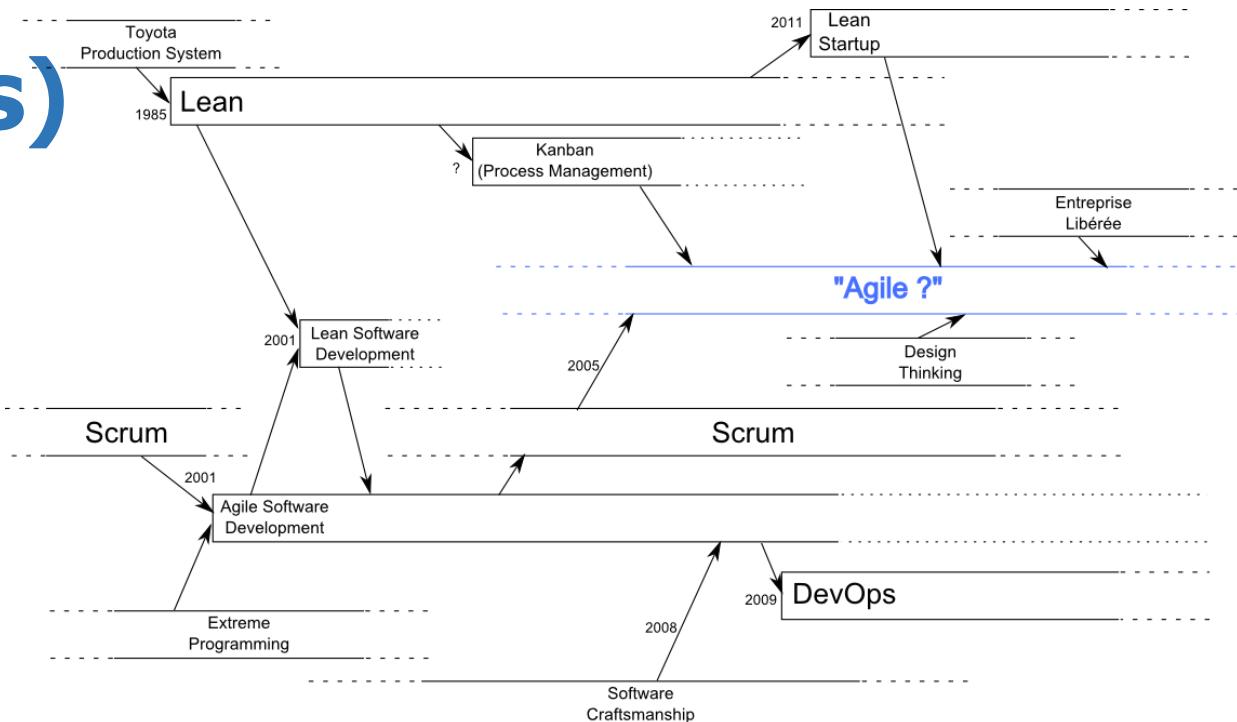
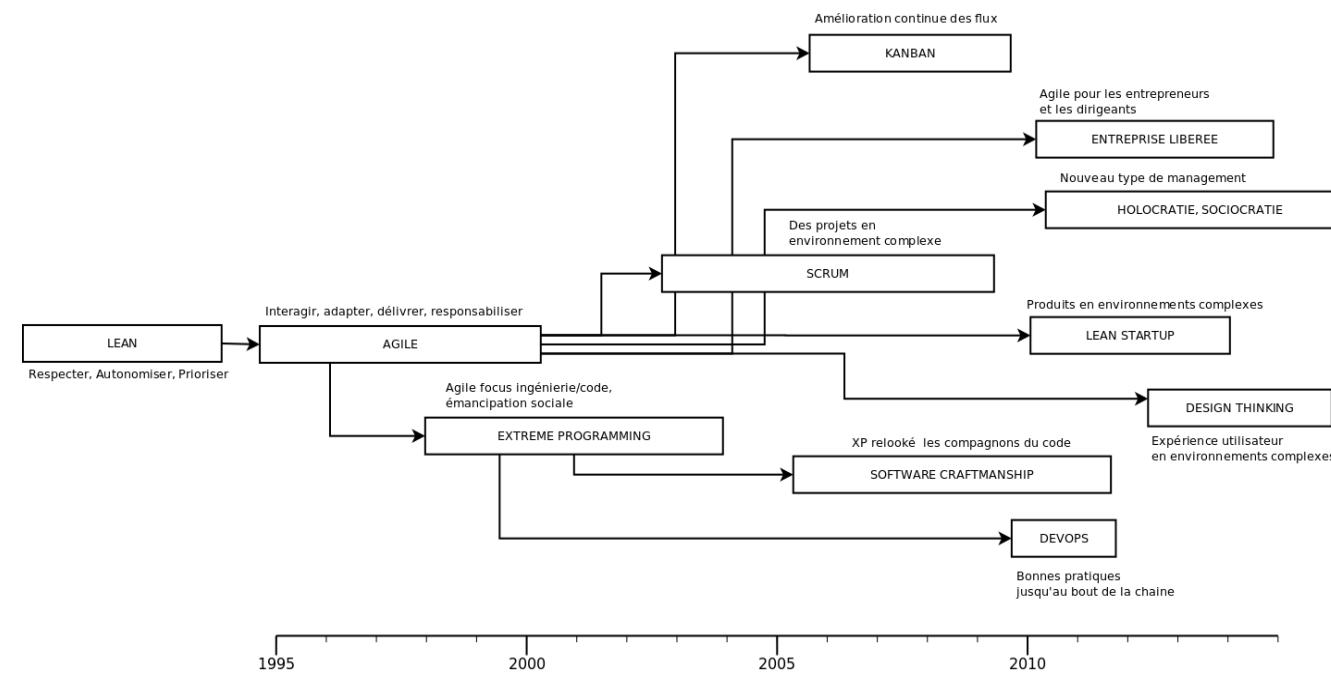
Les mouvements agiles du mouvement Agile

<http://www.areyouagile.com/2015/08/mouvement-agile>

un seul et unique "mouvement agile"

qui vient du lean et se décline en de multiples sensibilités ?

OU



« Ensemble de flux qui, potentiellement et ponctuellement, convergent vers quelque chose de commun, tout en gardant leur mouvement propre. S'il y a une "racine" commune à l'ensemble des approches agiles, elle sera dans le futur ». <http://agilitateur.azeau.com/post/2015/08/16/Mouvement%28s%29-Agile%28s%29>

Réflexions sur le développement agile en 2018 ...

The State of Agile Software in 2018

transcription of my keynote at Agile Australia

On the surface, the world of agile software development is bright, since it is now mainstream. But the reality is troubling, because much of what is done is faux-agile, disregarding agile's values and principles. The three main challenges we should focus on are: fighting the Agile Industrial Complex and its habit of imposing process upon teams, raising the importance of technical excellence, and organizing our teams around products (rather than projects). Despite the problems, the community's great strength is its ability to learn and adapt, tackling problems that we original manifesto authors didn't imagine.



<https://martinfowler.com/articles/agile-aus-2018.html>

Martin Fowler
25 August 2018

The Clean Code Blog
by Robert C. Martin (Uncle Bob)

The Tragedy of Craftsmanship.
28 August 2018

You look pensive.

Yes, I just read a transcript of Martin Fowler's keynote at Agile Australia, 2018. He called it *The State of Agile in 2018*.

Ah, yes, a great talk.

- Beware the Agile-Industrial-Complex.
- Maintain Technical Excellence.
- Products over Projects.

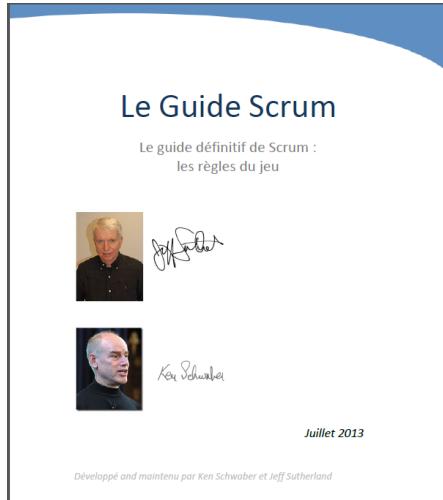
Great stuff! So what bothers you?

3 points...

*il est temps de laisser les équipes s'auto-organiser,
de respecter les développeurs
et de passer de la pensée projet à la pensée produit*

<http://blog.cleancoder.com/uncle-bob/2018/08/28/CraftsmanshipMovement.html>

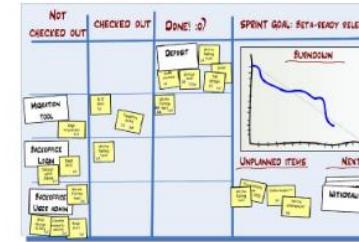
Quelques lectures de référence sur Scrum



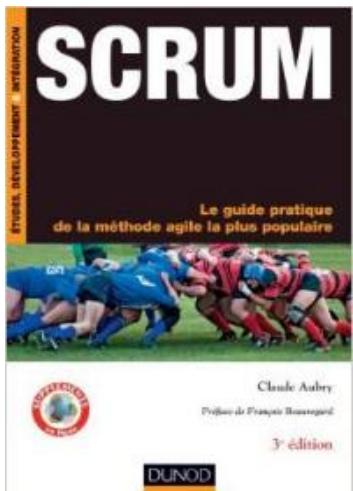
Le **scrum guide** de Ken Schwaber et Jeff Sutherland : <https://www.scrum.org/scrum-guide>
(version française disponible)

Scrum et XP depuis les Tranchées

Comment nous appliquons Scrum



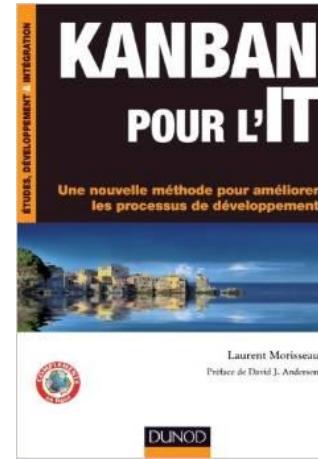
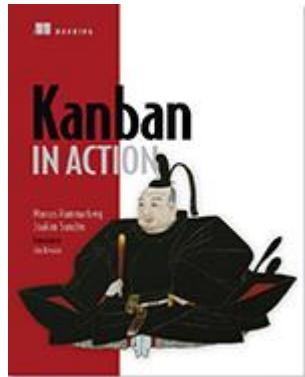
Écrit par :
Henrik Kniberg



Scrum : Le guide pratique de la méthode agile la plus populaire
<http://www.aubryconseil.com/pages/Livre-Scrum>

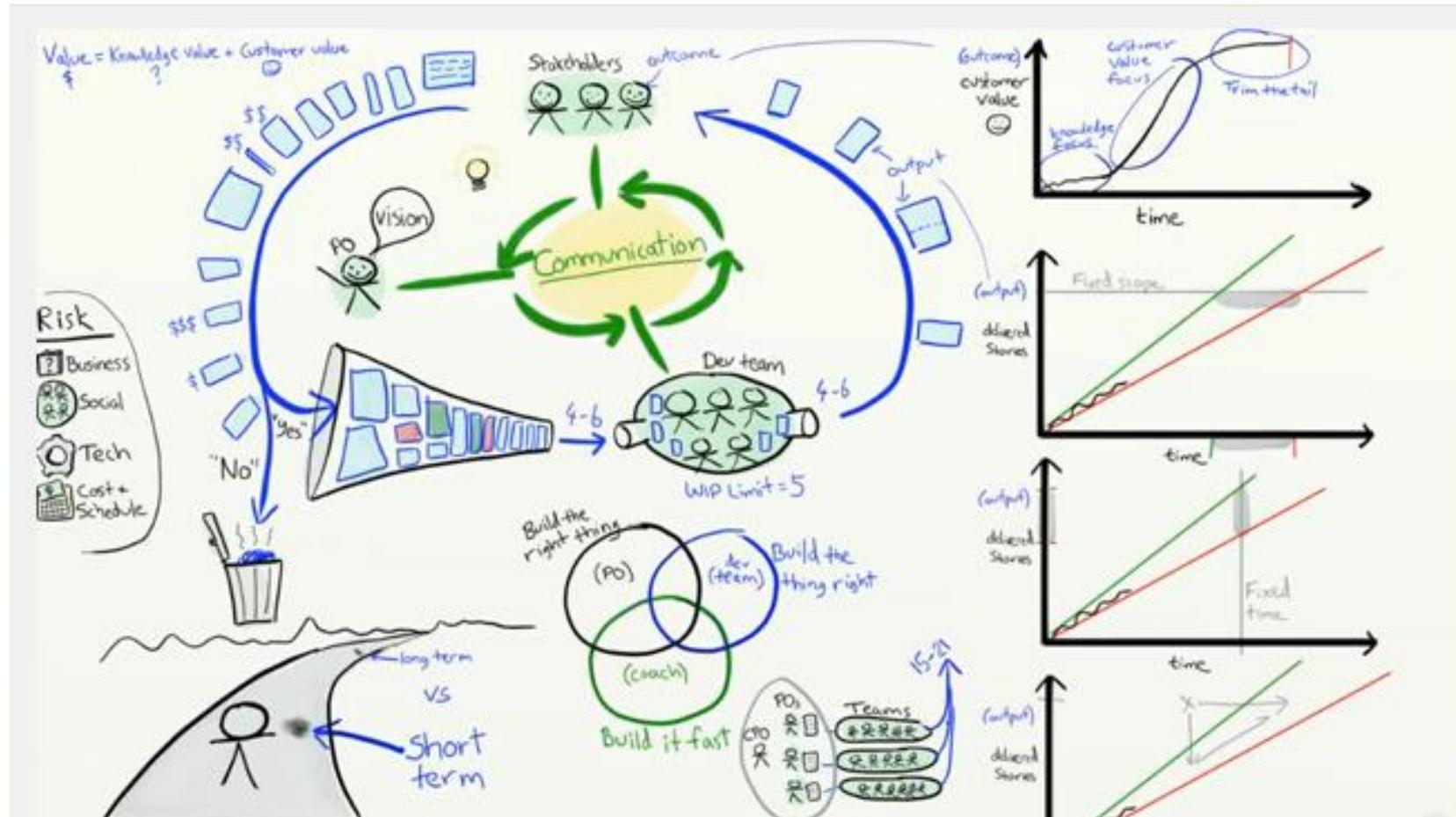
Blog Scrum, Agilité... et Rock'n roll : <http://www.aubryconseil.com/>

Quelques lectures sur Kanban



- ✓ De nombreux liens sur :
<http://www.crisp.se/kanban>

La gestion du produit agile en deux mots



<http://www.youtube.com/watch?v=3qMpB-UH9kA>