ARAMIS - 12 novembre 2012

Introduction à l'agilité

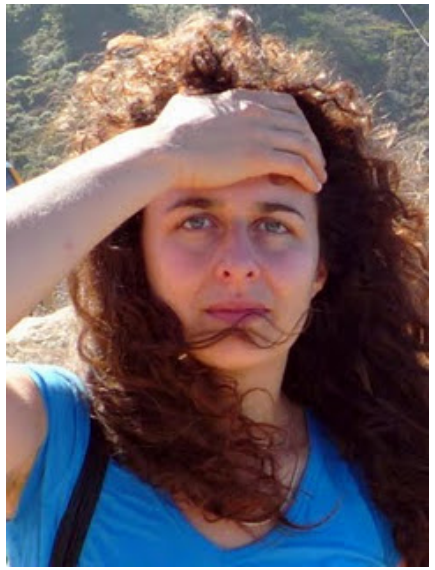
@Agnes_Crepet

@AlfredAlmendra

Agnès Crépet

@agnes_Crepet

agnes@ninja-squad.com



Java/JEE Architecte & Java Champion

- Laboratoires Boiron
- Ninja Squad

Java User Groups Leader: Duchess France & LyonJUG

Co-fondatrice de la conférence MIX-IT (Java, Agilité...)

Co-fondatrice du podcast Cast-IT (Java, Agilité...)

Alfred Almendra

@AlfredAlmendra

alfred.almendra@gmail.com

Freelance

- Scrum Master
- Coach agile
- Architecte Java/SOA

CARA Lyon

CARA

Club Agile Rhône-Alpes

Agenda

13h30	1 jeu de sensibilisation
14h	Introduction
14h45	Avantages et inconvénients

- 15h Pause -

15h15	1 jeu de sensibilisation
16h	Retour d'expérience Boiron
17h	Questions / réponses

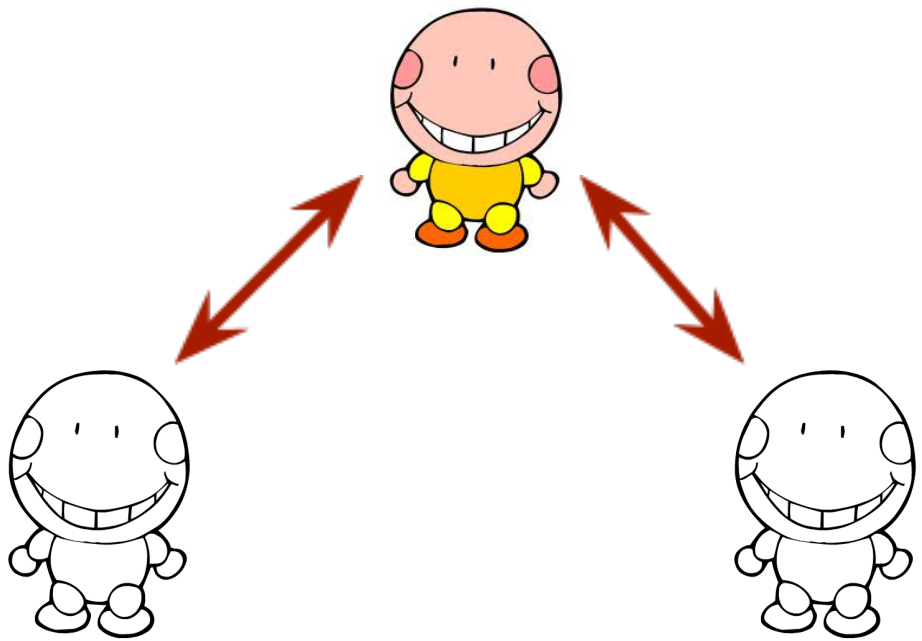
- 17h30 Fin - échanges libres -

Jeu : complex system

1. Tout le monde marche en groupe

2. Chacun choisit 2 personnes

3. Au top : on essaye de former un triangle isocèle



Introduction

Introduction

Manifeste agile et cycle itératif, incrémental

Manifeste agile

2001

http://agilemanifesto.org/iso/fr/

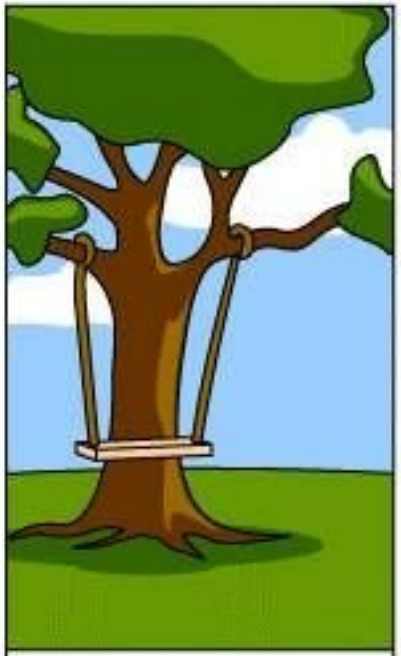
Savoir Être & Bon sens
Affronter les biais cognitifs
Embrasser le changement

Le manifeste agile : 4 valeurs, 12 principes

Privilégier :
- Les individus et leurs interactions plus que les processus / 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

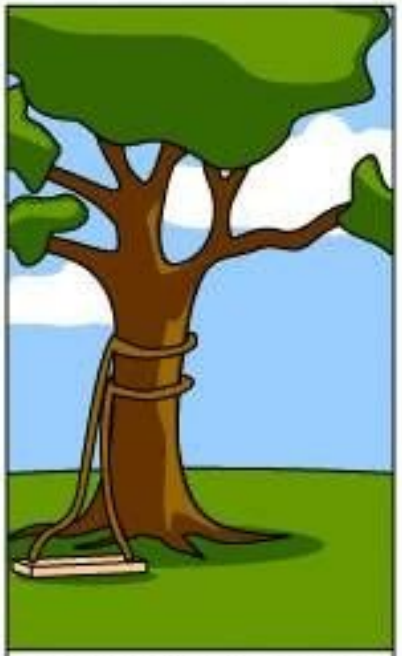
La gestion classique (prédictive)

How the customer explained it



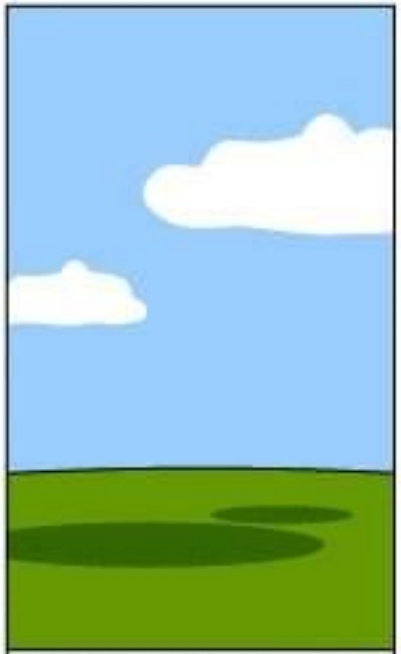
How the Project Leader understood it

How the Analyst designed it

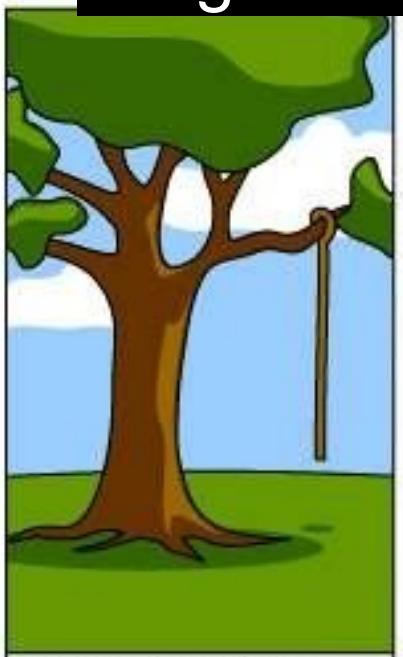


How the Programmer wrote it

How the Business Consultant described it

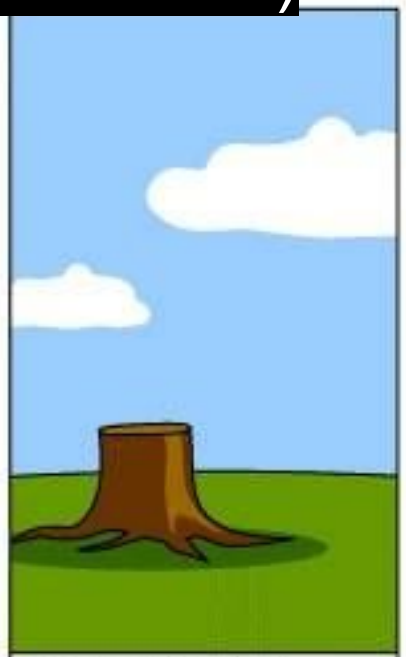


How the project was documented



What operations installed

How the customer was billed



How it was supported

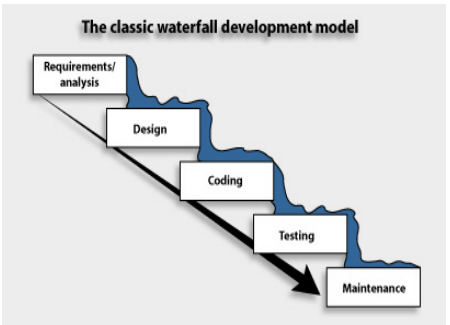
What the customer really needed

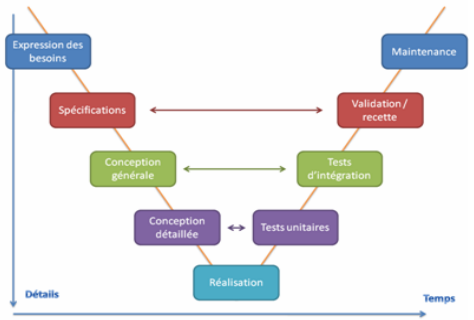
Cycle itératif, incrémental

Monalisa selon @jeffpatton

Effet tunnel

Processus continu de fabrication





L'agilité, principaux concepts

L'agilité recentre les processus sur les individus :

- le rythme de travail est soutenable
- l'équipe s'auto-organise
- une communication en face-à-face accrue!

Livrer tôt !

- Ne signifie pas livrer vite!
- L'objectif est de réduire au maximum
- l'effet tunnel

L'agilité, idées recues

"L'agilité ne fonctionne pas ou n’a pas fonctionné dans mon contexte”

La méthode est-elle bien utilisée ?
L'a-t'on adaptée/traduite par rapport à son contexte?

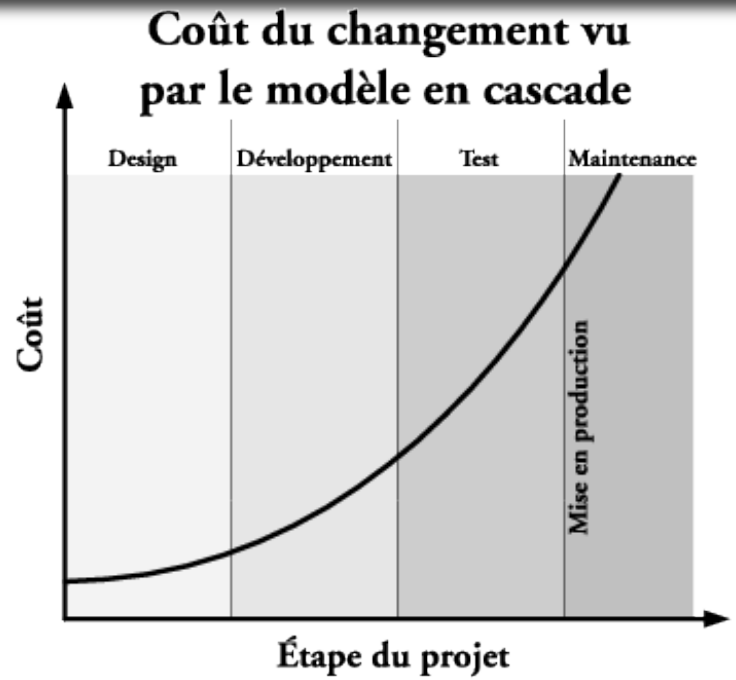
L'agilité, idées recues

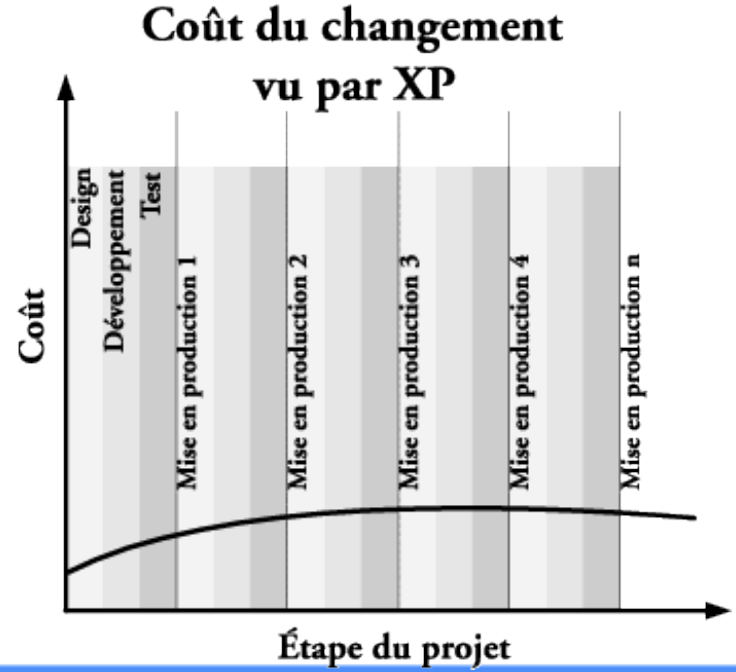
"L'agilté = Trop cher!"

Admettons à court terme, mais beaucoup plus rentable (investissement)

Non à moyen/long terme, suivi du ROI régulièrement (pour s'arrêter à temps)

Les itérations courtes ont un effet direct sur la qualité du produit fini





L'agilité, idées recues

"L'agilité, c'est un peu n'importe quoi, on fait ce qu'on veut!"

Ne pas confondre expérimentation & exploration avec le fait de "naviguer à vue"

Les méthodes agiles tendent à obtenir au plus tôt un retour en vue d'une amélioration

- Feedback user sur le produit
- Rétrospective sur la démarche de l'équipe

Elles se basent beaucoup sur des méthodes scientifiques et cartésiennes (ex: théorie des contraintes, loi de Little, courbes de cohortes, etc...)

Introduction

Les étapes d'un projet agile

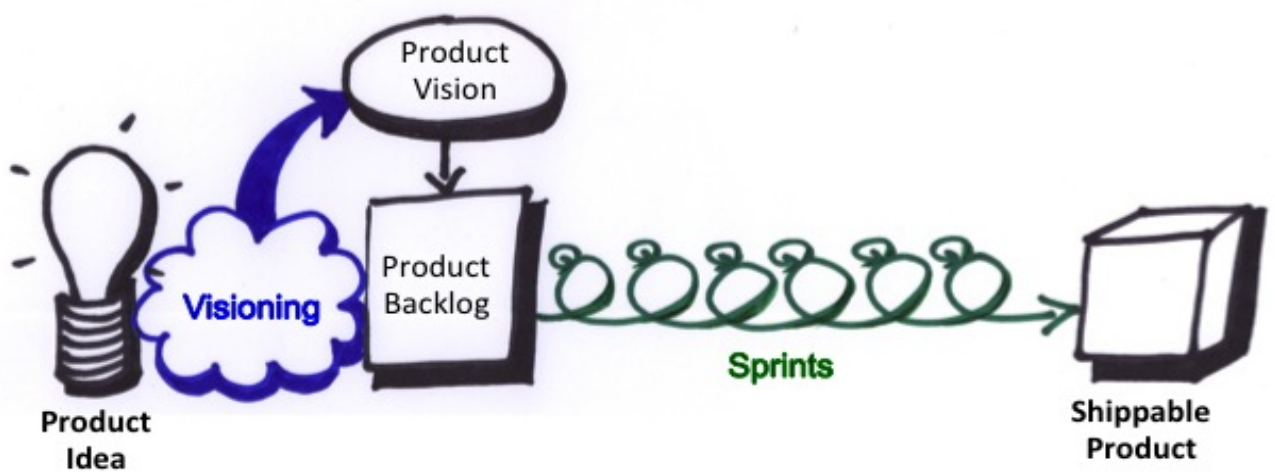
Les jeux et les étapes du projet

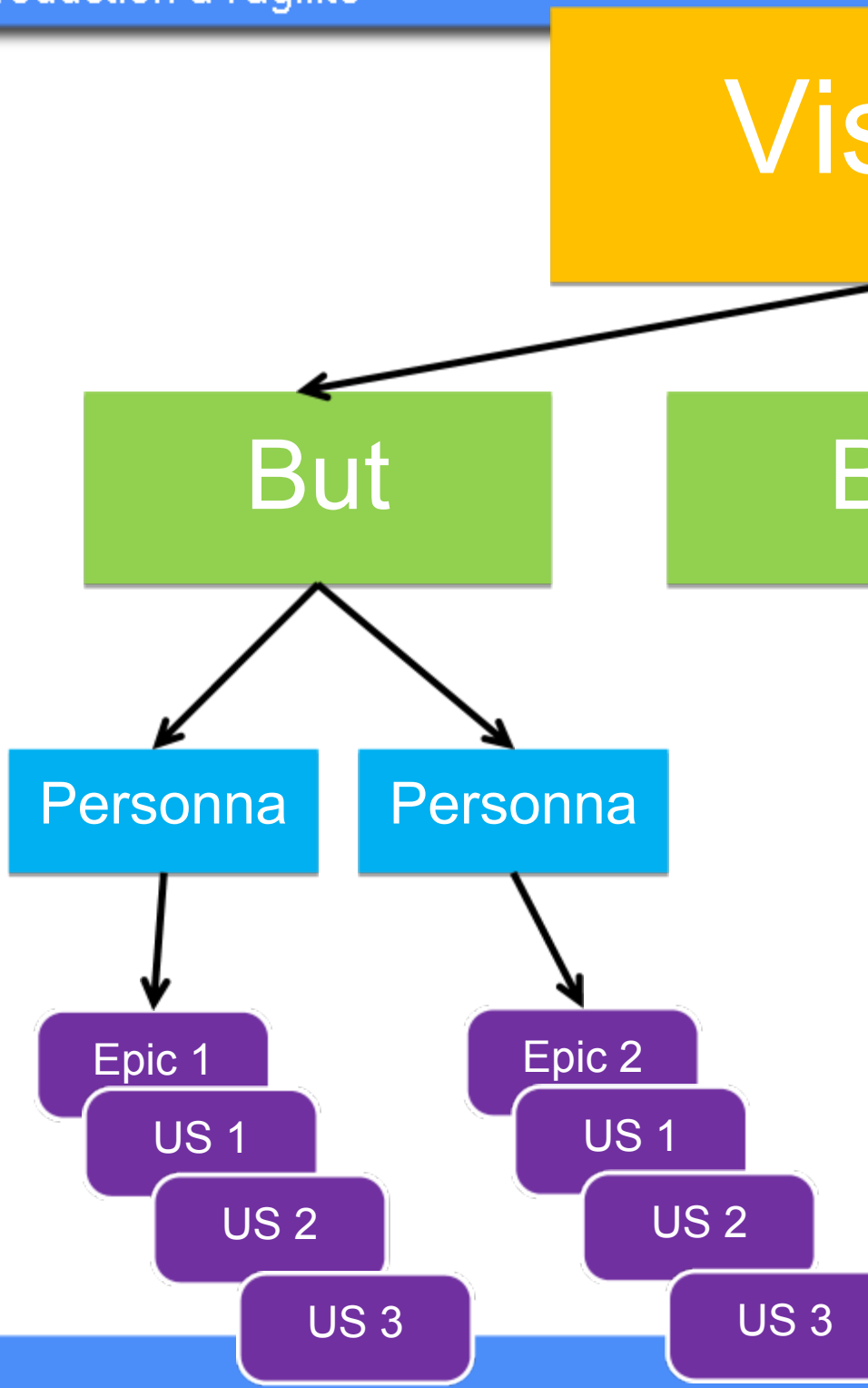
Extrait du calendrier du CARA Lyon (OUI, c'est de la PUB !)

Etape	Jeu
Définir sa vision produit	Product Box
Elaborer son backlog	User story mapping
Valoriser son backlog	Prune the tree
Planning et estimation	Zoo point, Planning poker
Définition of done	Artiste & spécifieur, Open-ended requirements
Daily meeting et auto-organisation	Marshmallow challenge
Rétrospective	SpeedBoat, Timeline

Vision : test de l'ascenceur (elevator statement)

POUR (les utilisateurs finaux du produit)
QUI SOUHAITENT (leurs besoins)
NOTRE PRODUIT EST (un résumé du produit)
QUI (le bénéfice majeur et l'utilité du produit)
A LA DIFFÉRENCE DE (produits concurrents)
PERMET DE (éléments différenciateurs majeurs)





Les personnas viennent de l'UX

Vision vers Epic
(~ fonction / UC)

Qu'est-ce qu'une User Story ?

Chef produit

Marketing

Développeur

En tant qu'Utilisateur

Je désire modifier une image

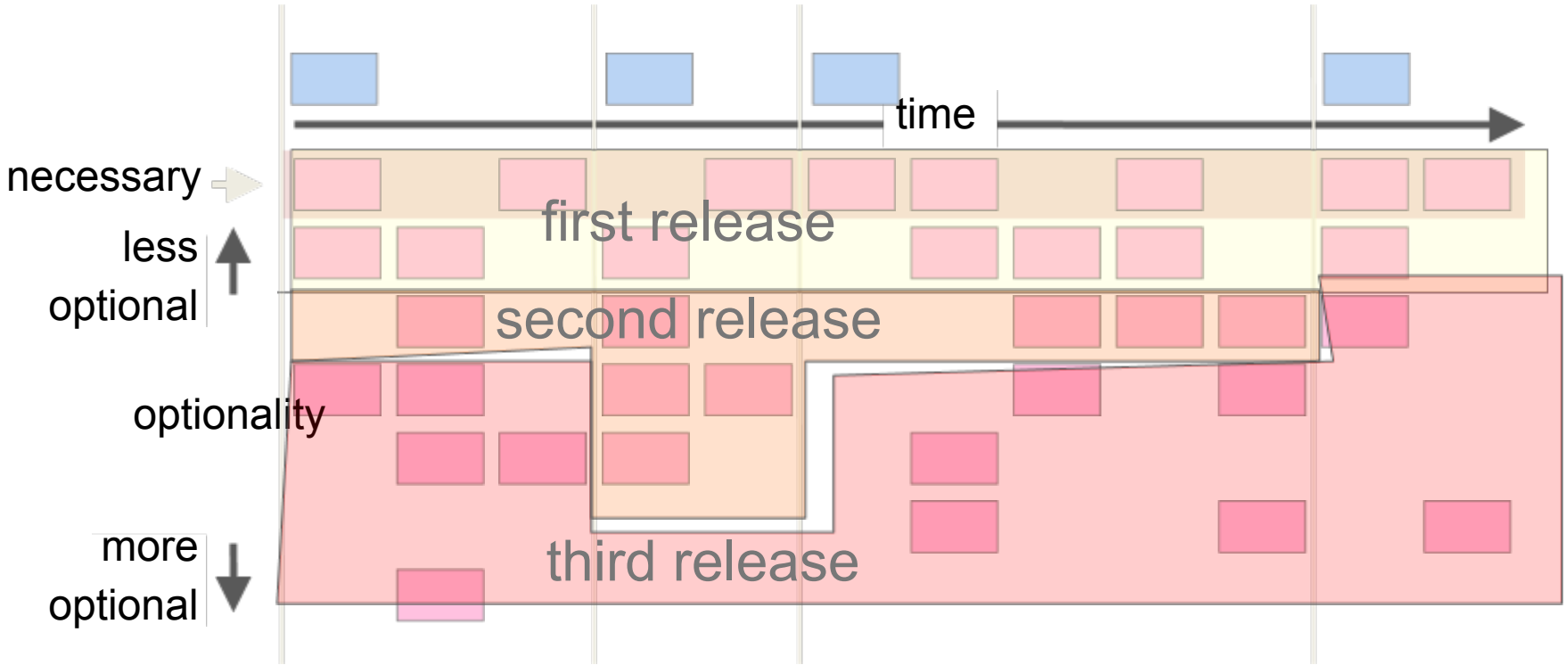
Afin de mettre sa description à jour

Client

Commercial

Ne pas oublier les critères d'acceptation !

User Story Mapping



Valoriser et prioriser

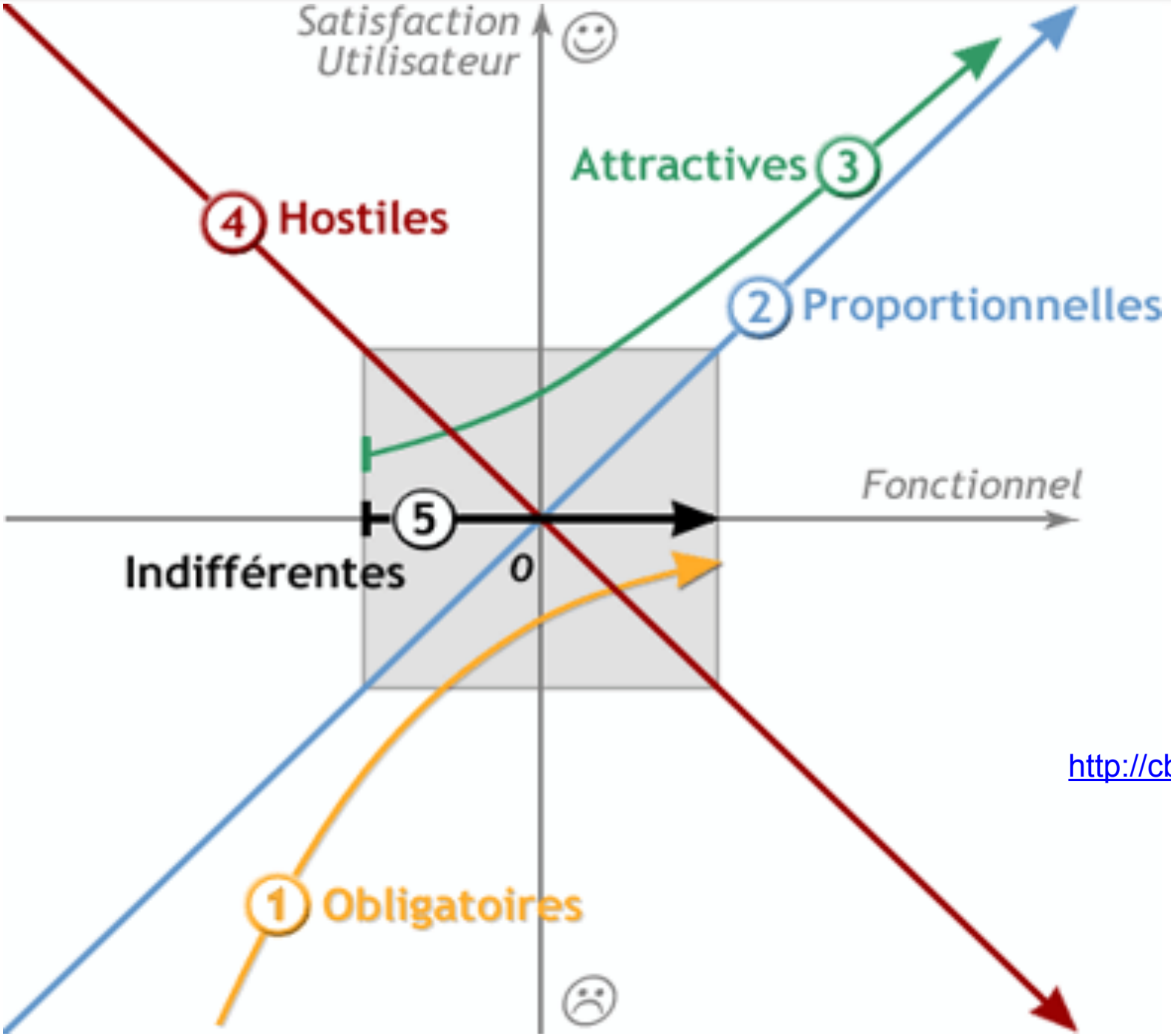
MoSCoW

- M - MUST have this
- S - SHOULD have this if at all possible
- C - COULD have this if it does not effect anything else
- W - WON'T have this time but would like in the future

Kano

Questions fonctionnelles et dysfonctionnelles

– Attractif
– Une dimension linéaire
– Obligatoire
– Indifférent
– Pénalisant



http://cbigot.net/kano

Estimation

Le budget n'est pas une somme d'estimation

Estimation relative par comparaison

- consensuelle, durable

Exemples de techniques de collaboration :

- Consensus du groupe
- Répartition de tâches
- Combinaison des estimations individuelles

Exemples de jeux

- Planning poker (1, 2, 3, 5, 8, 13, 20, 40, 100) (fibonacci)
- T-Shirt sizing (XS, S, M, L, XL)
- Zoo points (férocité), poids des chiens,...

Planification

Les itérations : de quelques jours, semaines à 2 mois max

Les releases : toutes les X itérations
- Idéalement toutes les itérations
- Maximum : toutes les 3

Scrum : périmètre figé du sprint en cours
Kanban : pérmiètre flexible

Gestion du risque, suivi et arbitrage au quotidien : daily meetings
- ++ estimation du reste à faire
- -- passer trop de temps sur un chiffrage trop fin

Retropective

Bilan des itérations
Objectif : s'améliorer et non juger!

Toute l'équipe participe à la réunion

En entrée

- le plan d'actions de la rétrospective précédente
- le backlog de problèmes

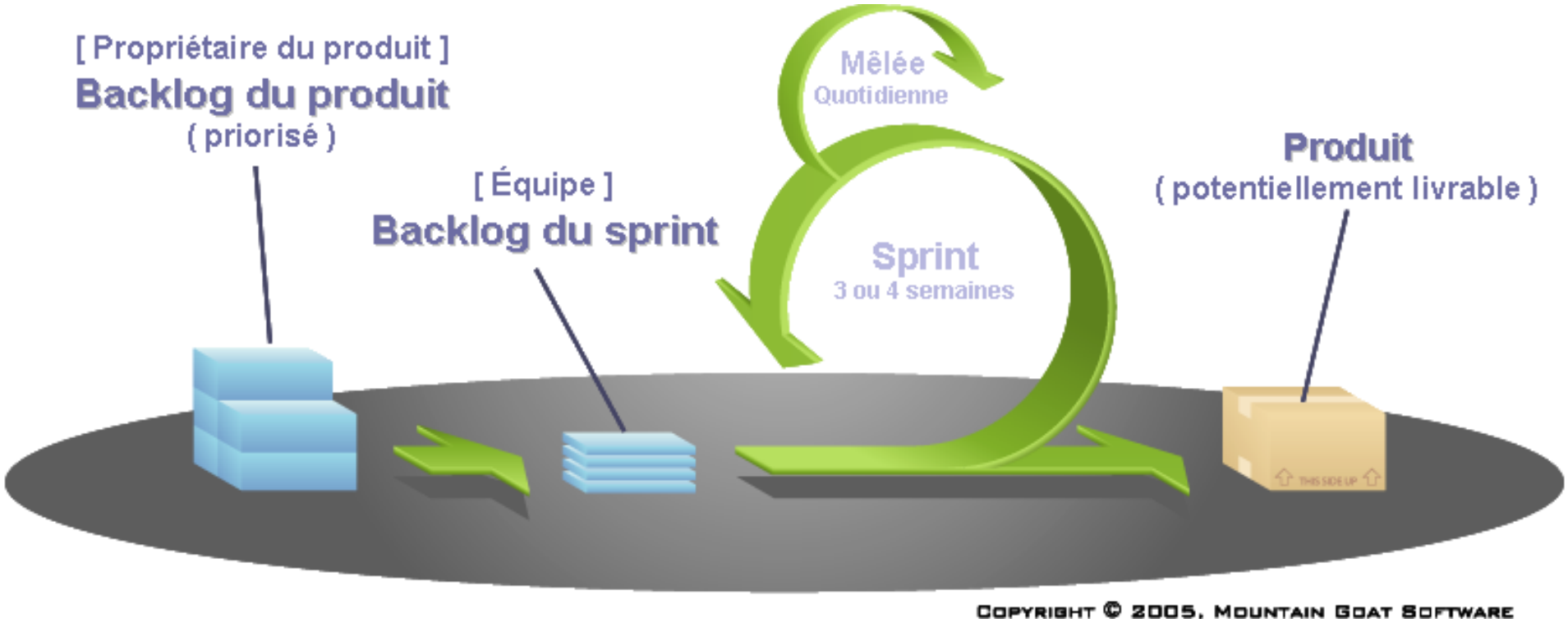
Etapes

- Créer un environnement propice à l'expression
- Collecter les informations relatives au processus
- Définir les priorités
- Planifier des actions d'amélioration

Introduction

Scrum, la méthode populaire

Scrum



Temps fixe des itérations, itération de refactoring, visibilité sur 1 ou 2 itérations

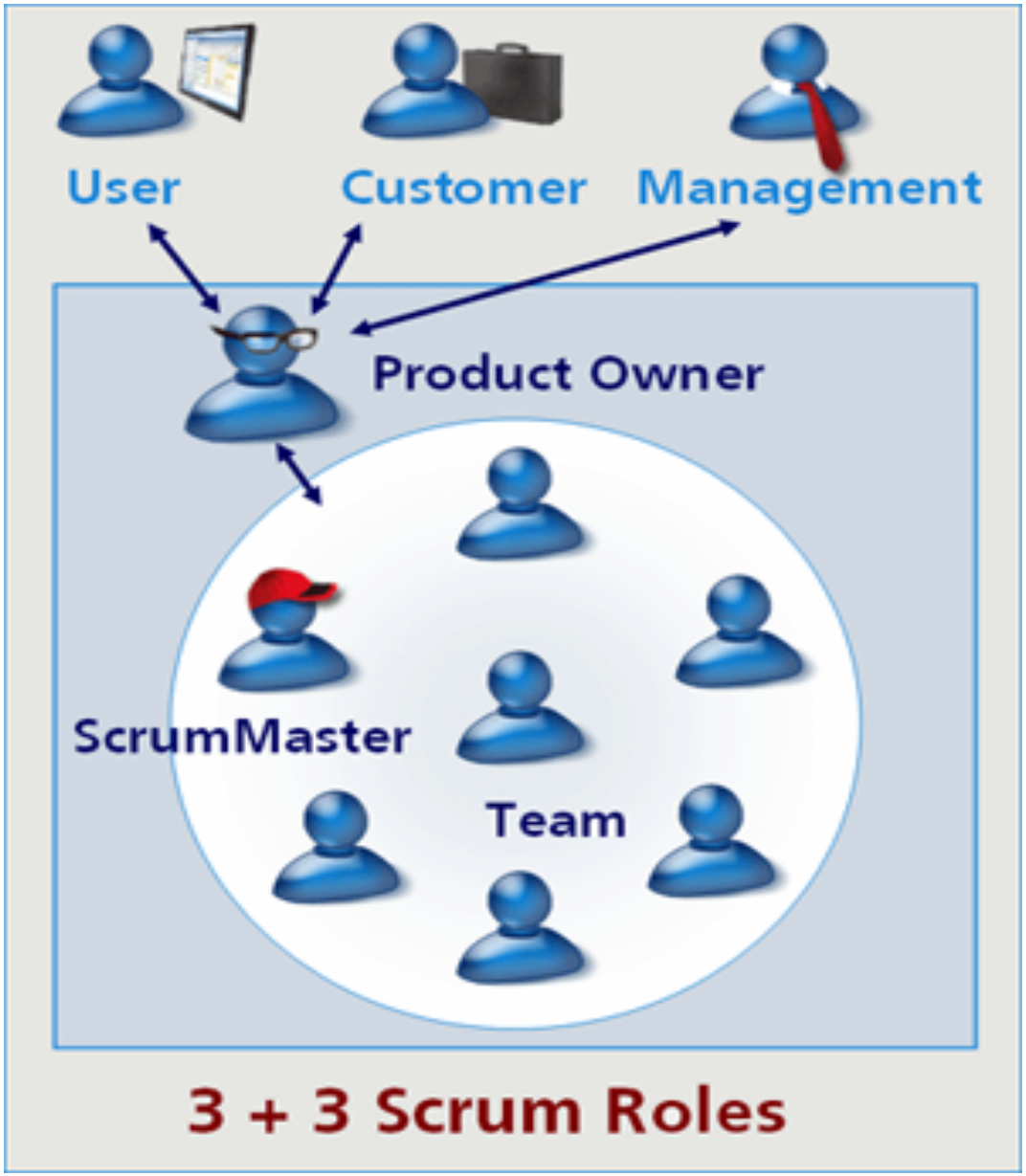
Attention d'éviter les goulots d'étranglement (spécs d'avance)
Présence PO : spécification, développement, recette

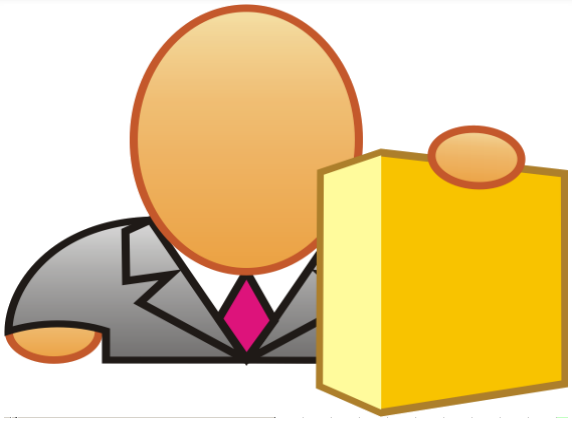
Rôles et auto organisation

Lâcher prise

Pilotage par les risques

Daily meeting (stand up)





Product Owner (PO)

Définit les **fonctionnalités** du produit

Définit les **priorités** dans le backlog en fonction de la valeur « métier »

Ajuste les fonctionnalités et les priorités à chaque itération si nécessaire

Teste les releases

Accepte ou rejette les résultats

Scrum Master (SM)

Vulgarise les valeurs et les pratiques de Scrum

Contribue à **améliorer** les outils et les pratiques de l'ingénierie

Facilite une coopération poussée entre tous les rôles et fonctions

Protège l'équipe des interférences extérieures

Met l'accent sur la **créativité** et l'**autonomie** des membres

Scrum : stand up (daily meeting)

3 questions :

- qu'avez-vous fait hier ?
- qu'allez-vous faire aujourd'hui ?
- qu'est-ce qui bloque l'avancement ?

Tous les "acteurs" parlent (au sens "task board")

- pas uniquement les développeurs

Time-boxing

- pas uniquement aux stand-up

Avantages et inconvénients

Un 1ier avantage : focus sur le besoin

Sans agilité, le projet peut vite être centré sur des livrables autres que l'application souhaitée par le client

Les méthodes agiles permettent de réduire l'effet tunnel
- Pour répondre réellement au besoin
- Maximiser la valeur fonctionnelle

On reste ouvert au changement !

Avantage : Pilotage par les risque

Pour maximiser les chances de succès du projet

Alertes au plus tôt si dérive

Possibilité d'arrêter un projet si le périmètre est satisfaisant

Indicateurs de risques
- simples à mettre en place
- compréhensibles par tous

Toute l'équipe
- participe à donner du feedback !
- est impliquée dans la réussite du projet

Avantage : réduction de la dette technique

Cultiver la qualité intrinsèque du code

Avec les méthodes classiques:

- Trop de bugs ou de projets non terminés!
- Un code coûteux à maintenir et à faire évoluer

L'agilité prône la qualité technique

Un Investissement qui vise à:

- Baisser la dette technique
- Pérenniser les produits

On garde du temps pour ajouter de la valeur au produit !

Des inconvénients?

Certains efforts sont demandés

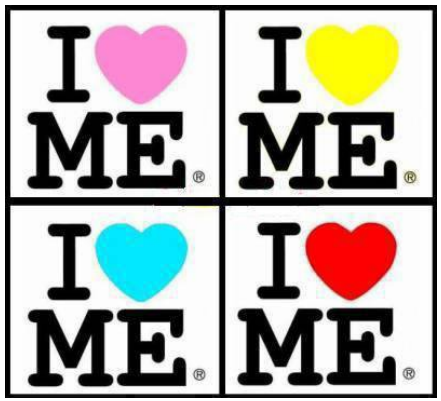
L'équipe doit faire preuve de courage, honnêteté, transparence, visibilité, engagement, respect...

Valeurs pas toujours faciles à partager:

- Propriété collective du code
- Responsabilité collective de la réussite du projet

On a hélas parfois tendance :

- à ne s'occuper que de son propre terrain
- à ne pas s'impliquer dans les problèmes
  - de son voisin pour l'aider à les résoudre



S'améliorer c'est d'abord sortir de sa zone de confort !

Certains efforts sont demandés

Collaboration active et impliquée du client, de l'utilisateur

Le client : "Pourquoi je dois m'impliquer?"

"Ce que je veux est pourtant simple !"

Lâcher-prise du manager

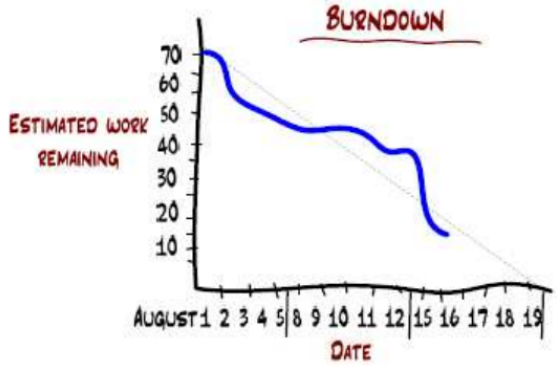
Préférer un management horizontal !

Moins de hiérarchie marquée !

Le rôle du manager est fortement remis en cause !

Attention aux dérives

Le daily meeting n’est pas du flicage : il s’agit de piloter les risques en identifiant les blocages quotidiens



La vélocité n’est pas un engagement de productivité : elle sert à estimer le périmètre réalisable dans la prochaine itération

Le sprint burndown chart n’est pas un indicateur de productivité : il permet de visualiser le reste à faire sur le périmètre initial de l’itération en cours

Jeux agiles

Jeux sérieux

Efficacité :

- Ludique : volontarisme, motivation
- Cadre et règles simples : focus sur l'objectif
- Créativité, collaboration, excellent taux d'apprentissage

Objectifs :

- Valeur humaine : pour sensibiliser
- Valeur produit : pour remplacer les réunions stériles
- Apprendre et s'entraîner (Playing Dojo)
- Diagnostiquer et résoudre un problème

http://prezi.com/n4s3cxjsxs-a/jeux-agiles-pour-professionnels-exigeants/

Jeu

Combien de temps faut-il pour écrire un prénom ?

Quels sont les facteurs de variabilité ?

Jeu

A remplir par le client (en secondes) :

- temps de début :
- temps de fin :
- soit une durée de :

Le fournisseur écrit le prénom de chaque client...

...mais en respectant la politique de notre entreprise !

Jeu

La politique de notre entreprise est :

Ne jamais faire attendre un client

Commencer tôt pour finir tôt

Satisfaire tous les clients en même temps :
on change de client après chaque lettre

Jeu

La politique de notre entreprise est

Limiter le travail à faire (TAF), l'en-cours
WIP = Work In Process

1 seul client à la fois :

on change de client une fois son prénom écrit

REX Boiron

Voir vidéo d'Agnès Crépet et Cyril Lacôte - 30 minutes
Retour d'expérience sur l'agilité chez Boiron
http://clacote.free.fr/
Interviews : DSI - Product Owner - CP - Développeurs, etc.

Exemple de mise en oeuvre

La DSI des laboratoires Boiron introduit en 2008 les méthodes agiles

Pour les projets de refonte du Système d'information sur la base d' architectures contemporaines (JEE, ESB, MDM, etc.)

Intérêts :

- introduire des demandes d'évolutions en cours de projet
- faciliter l'acceptation des nouvelles solutions informatiques par les utilisateurs finaux

Premier « vrai » déploiement sur un projets critique (10.000 jours)

Agilité chez BOIRON ?

- Un mix d'UP, XP et de Scrum / Kanban

Pratiques et outillages "agiles"

Processus itératif et incrémental
Recette Utilisateur à chaque fin d'itération
Stand-up quotidien / Tableau post-it
Gestion des exigences
Développement par les tests (JUNIT, DBUNIT, Mockito)
Refactoring régulier (par les patterns)
Bug Tracker (JIRA)
Intégration Continue (Maven, Jenkins, Nexus)

Coverage Report - All Packages

Package	# Classes	Line Coverage	Branch Coverage
All Packages	10	52 % 136/260	26 % 30/116
com.boiron.framework.security	7	51 % 99/193	25 % 26/102
com.boiron.framework.security.facade	1	84 % 26/31	100 % 2/2
com.boiron.framework.security.repository	1	83 % 10/12	50 % 2/4
com.boiron.framework.security.traceability	1	4 % 1/24	0 % 0/8

Report generated by Cobertura 1.9 on 24/04/08 18:36.

Agilité, modélisation et UML

Boiron

La modélisation agile peut-elle exister ?

- L'agilité se passe généralement de documentations volumineuses et de plus en plus d'UML

Mais Boiron a décidé néanmoins de garder UMl

- Traçabilité des exigences
- Analyse d'impact d'un changement
- Contrainte de validation pharmaceutique
- Communication inter et intra équipe

Stratégie Boiron pour pour la modélisation:

- Pas trop de doc
- Un peu d'UML

Voir :
http://www.slideshare.net/agnes_crepet/modelisation-agile-03122011

Exemple de mise en oeuvre

Boiron

Des itérations d'un mois calendaire

- Mais cela peut varier en fonction des phases du projet
  Un sprint est à durée fixe en Scrum

- Kanban

Des recettes utilisateurs
à chaque fin d'itération

En période pré-production :
recette toutes les 2 / 3 semaines

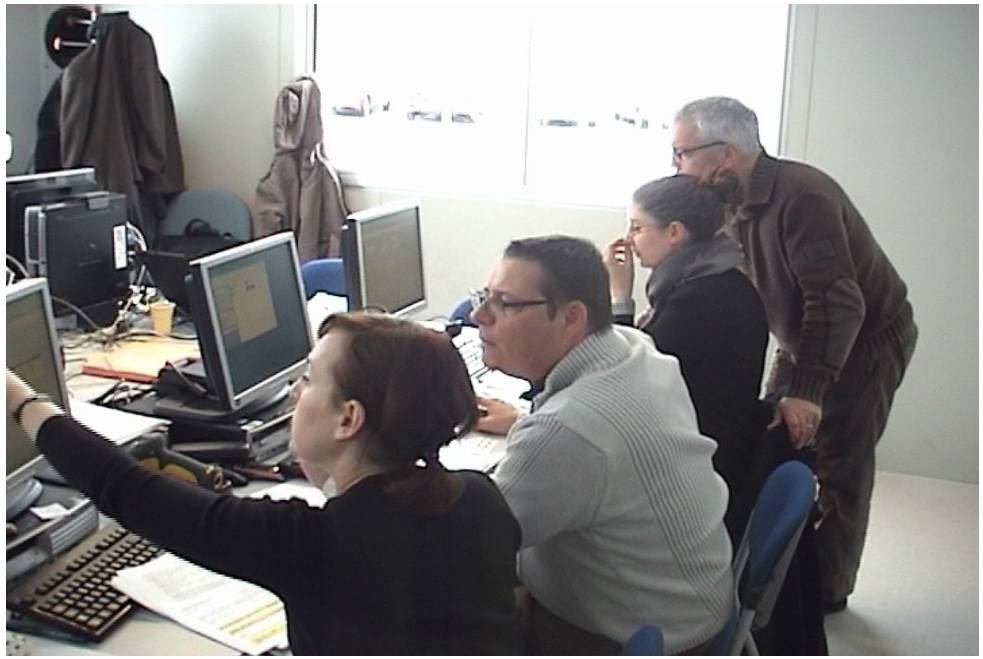


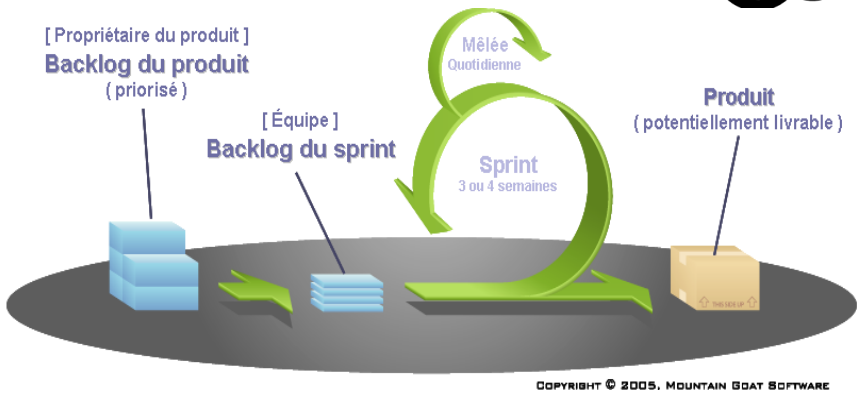
Photo : Recette Utilisateur
Boiron Janvier 2010

Une itération

Boiron

Backlog de produit

Les exigences, les activités
- En UP : Use Case (Boiron)
- En XP : User stories



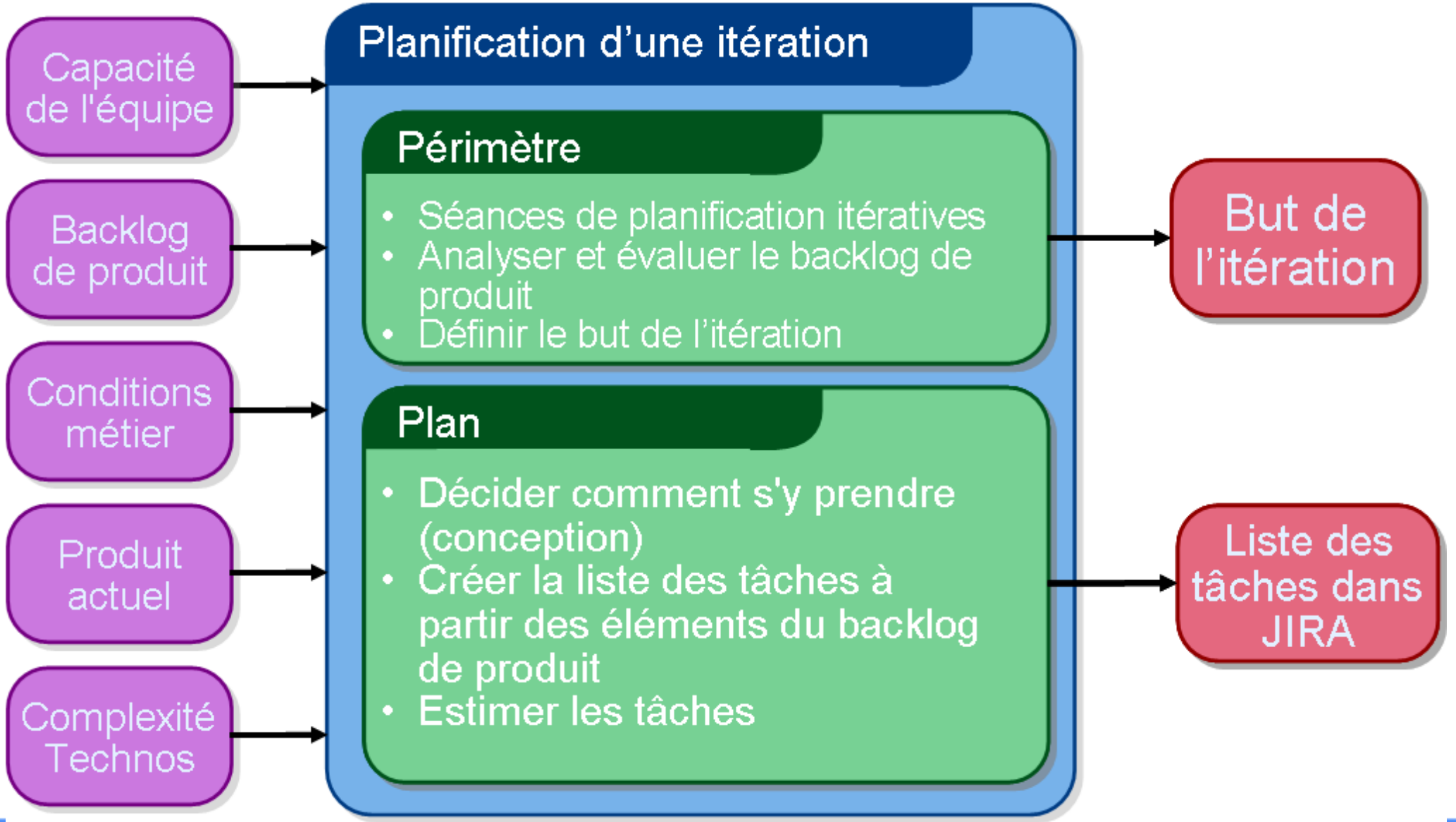
Une entrée du backlog de produit est un Use Case UML (inspiré d'UP)
- Un Use Case peut se dérouler sur 1 ou 2 itérations
  - ✗ en Scrum
  - ✓ en Kanban

Leurs priorités sont revues à chaque itération
- Définies par le Product Owner
- Mais également par le reste de l'équipe (différent de Scrum)

Exemple de mise en oeuvre

Boiron

Comment planifier une itération ?



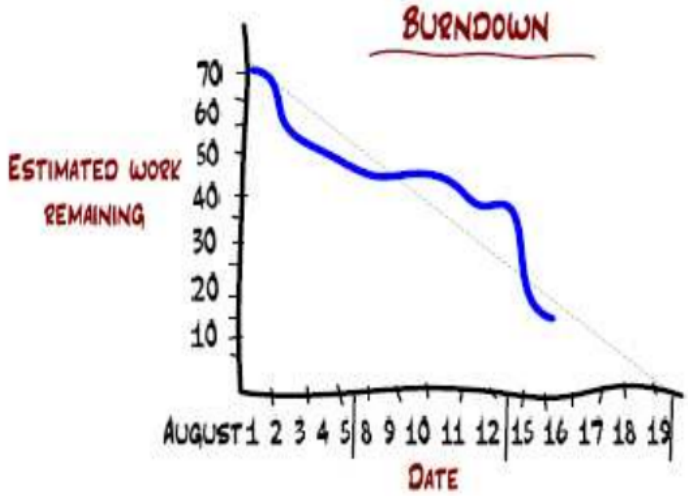
Exemple de mise en oeuvre

Boiron

Vie du backlog de l'itération

L'estimation du reste à faire est ajustée tous les jours (Stand-up / JIRA)

- Mise à jour du travail restant quand il est mieux connu



N'importe qui peut ajouter, supprimer, changer la liste des tâches en stand-up

Si un travail n'est pas clair, définir une tâche avec plus de temps et la décomposer après

	Changement en cours d'itérations	Estimation du reste à faire
Scrum	✗	Utilisation de Burndown Charts avec mise à jour quotidienne
Boiron	✓ (comme Kanban)	Utilisation de JIRA (quotidien)

Conclusion

Conclusion

Adapter l'Agilité à votre contexte
Appliquer les pratiques agiles qui semblent « pragmatiques » et adaptées à votre contexte

Outiller certaines d'entre elles

Bien **identifier les rôles!**

Et **vulgariser, former**…
Les personnes de l'équipe doivent d'approprier la méthode
Mieux que de l'imposer!

« Ne pas développer de dépendance spécifique à une arme ou à une école de combat »

Miyamoto Musachi, Samouraï du XVIIième siècle

Ressources

Ressources

Présentations de Claude Aubry (voir son Introduction à l'Agilité pour l'Inra) :
http://www.aubryconseil.com/media

http://www.extremeprogramming.org
Le site de référence XP, qui propose une excellente présentation de la méthode

http://www.xprogramming.com
site XP de Ron Jeffries. Articles très intéressants sur la méthode (rubrique "XP Magazine"), ainsi qu'une liste des frameworks xUnit disponibles pour divers langages ("XP downloads").

http://www.xp123.com
site de William Wake. Nombreux articles sur les pratiques concrètes de XP (les tests unitaires avec Java, le Planning Game, etc).

Ressources

http://www.clubagilerhonealpes.org/blogs
Des blogs d'agilistes!

http://institut-agile.fr/
Le site de l'institut agile (Laurent Bossavit)
et son référentiel : http://referentiel.institut-agile.fr/

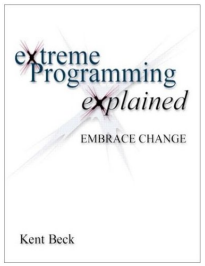
http://henrik-kniberg.developpez.com/livre/scrum-xp/
"Scrum et XP depuis les Tranchées"
Comment nous appliquons Scrum

http://lyon.clubagilerhonealpes.org/
Groupe Lyonnais du Club Agile Rhône-Alpes

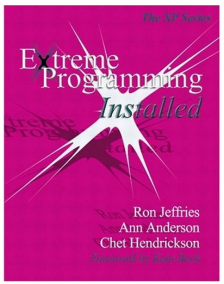
Ouvrages

L'Extreme Programming (avec deux études de cas),
Jean-Louis Bénard, Laurent Bossavit, Régis Medina,
Dominic Williams, chez Eyrolles, 2002.

Extreme Programming Explained : Embrace Change,
Kent Beck,
Addison-Wesley, 1999.

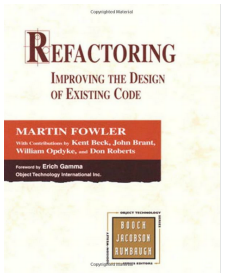


Extreme Programming Installed,
Ron Jeffries, Ann Anderson et Chet Hendrickson,
Addison-Wesley, 2000.



Planning Extreme Programming,
Ron Jeffries, Ann Anderson et Chet Hendrickson,
Addison-Wesley, 2000.

Refactoring : Improving the Design of Existing Code,
Martin. Fowler,
Addison-Wesley, 1999



Annexes

Qualité et rigueur du dév

Méthode XP et ses pratiques techniques

- Adaptée aux équipes réduites avec des besoins changeants
- But principal : réduire les coûts du changement
- Valeurs : communication, simplicité, feedback, courage, respect
- Pratiques :

Pair programming	Revue de code en permanence (binôme)
Test Driven Design	Tests faits avant chaque implémentation
Refactoring	Conception tout au long du projet
KISS (Keep It Simple Stupid)	Simplicité pour avancer plus vite
Métaphores	Faciliter la compréhension
Adaptatif	Intégration des modifications plusieurs fois par jour
Itératif	Evolution rapide des besoins, donc cycles de développement courts

Entrainements : code retreat, coding dojo

Mouvement du software craftsmanship (clean code, uncle Bob)

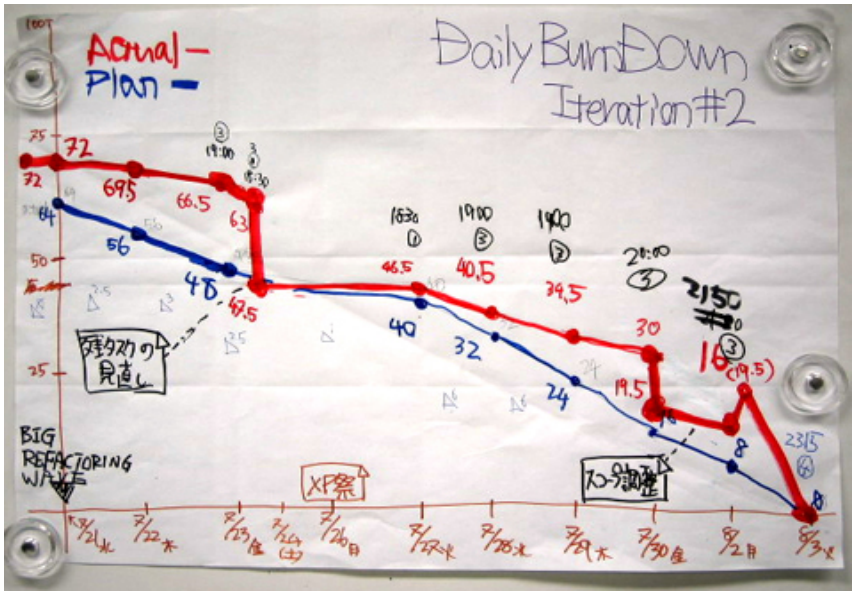
Efficacité du dév collaboratif

Usine de Qualité Logicielle :

- Versionning
- Mesure automatique de la qualité du code
- Automatisation des tests, intégration continue
- Gestion des tâches
- Déploiement continu

Visual Management

task board / DoD



burndown chart

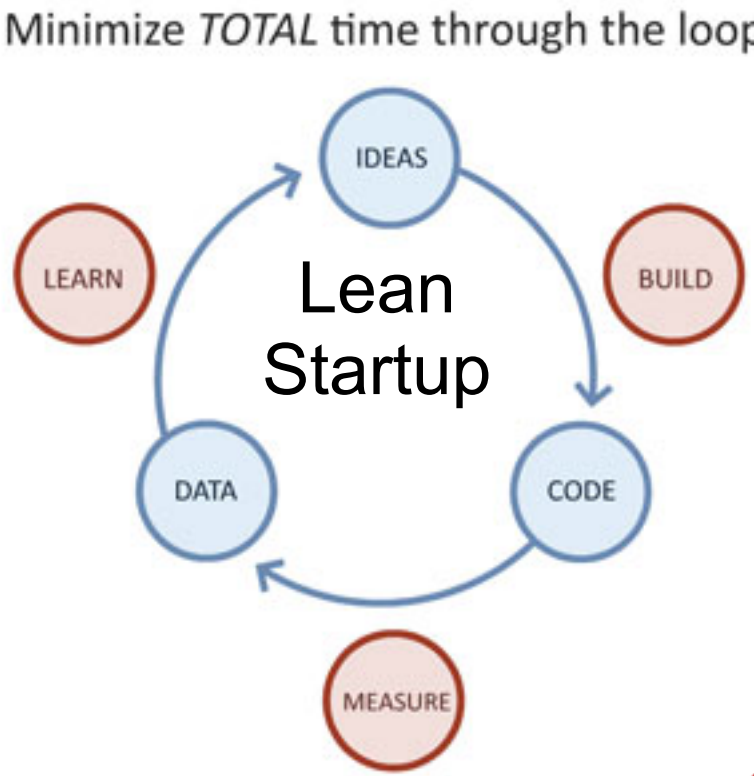
plan d'itération

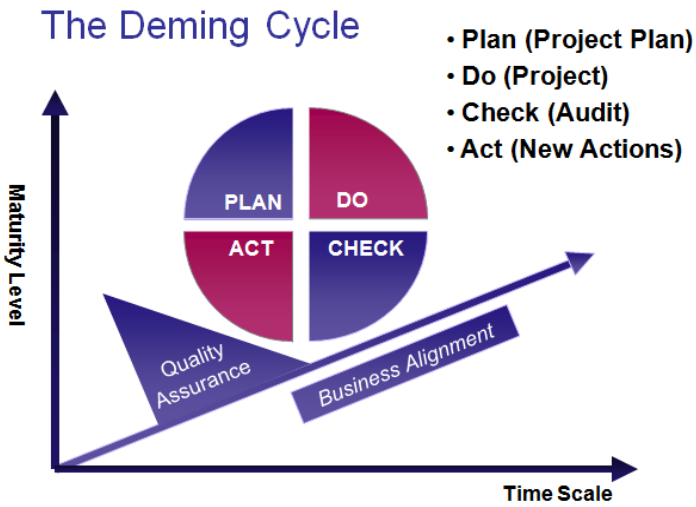
Story	To Do	In Progress	Done
Story A		Task	Task
Story B	Task	Task	Task
Story C		Task	Task

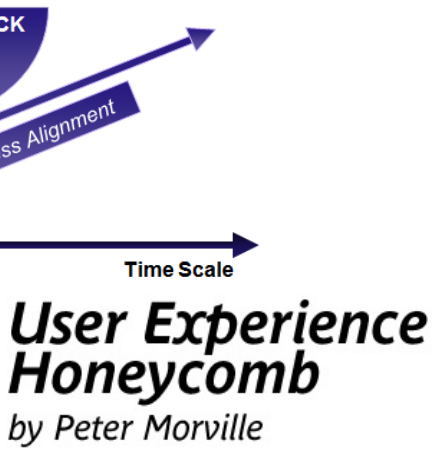
story mapping

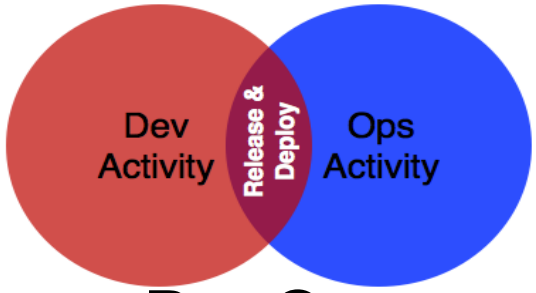
Niki-niko

Mouvements, tendances, convergence









Contractualisation

Contractualisation

Entorse à l'agilité :

- privilégier la collaboration avec le client (VS contrat)
- adaptation au changement (VS périmètre prédictif figé)
- cycle itératif (VS effet tunnel)

Agilité ?

- contractualiser aussi sur la forme (moyens)
- réduire l'effet tunnel
- faire son marché

Exemple concret, français et libre (fruit de la collaboration avec le département IP/IT du cabinet d'avocats Alérion et le cabinet de conseil LCA) :
http://contrat-agile.org/

Contractualisation

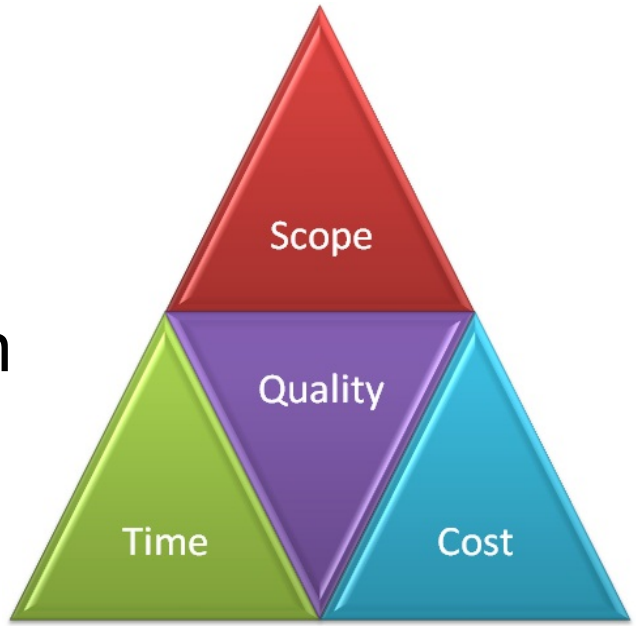
Différentes situations :

- client agile / non agile
- équipe avec ou sans collaborateur du client
- régie / forfait
- appel d'offre / marché public / éditeur progiciel
- sur site / multi sites / nearshore / offshore

Contrat = engagement

Actionner des leviers différents :

- Marché de niche, appli web : exploration
- TMA : industrialisation



Evaluation itérative de la satisfaction client : Done criteria

Contractualisation : exploration

Périmètre fonctionnel et marchés inconnus ou flous

- expérimentations itératives
- engagement sur la forme plutôt que sur le fond

1 contrat ou avenant pour l'itération N+1

- fixation du coût du sprint
- périmètre : fixe ou avec part variable (mini cycle en V)
- satisfaction client : stop ou encore
  - le meilleur indicateur (principe agile)

Client : agile (et demandeur de l'agilité !)

Scrum : PO client (métier) ou proxy-PO fournisseur (présent)

Contractualisation : TMA

Application existante

- périmètre fonctionnel connu
- priorisation claire (bloquant, majeur, mineur)

Taille des demandes limitée : estimation facilitée

Qualité maîtrisée : usine de qualité logicielle

Client : agile ou non

Coût annualisé : % projet, forfait, pool de jours

Kanban : industrialisation