

Lean Software Development

Mise en oeuvre

Pablo Pernet 

initialisation septembre 2014 - mise à jour septembre 2015 - version 0.8

<http://www.areyouagile.com>

Hommage à Mary Poppendieck

sans qui rien n'aurait été possible :)



Sommaire

Partie 1 : Histoire & Théorie

- Ateliers
- Les temps complexes
- Toyota versus Ford
- La maison et la pensée LEAN

Partie 2 : Principes du Lean

- #1 Éliminer les gaspillages
- #2 Amplifier l'apprentissage et la connaissance
- #3 Décider le plus tard possible
- #4 Montrer le plus tôt possible
- #5 Responsabiliser et engager les équipes
- #6 Qualité intrinsèque
- #7 Voir le tout

Partie 3 : Fils de Lean, descendances modernes dans le domaine informatique

- Management visuel
- Kanban Software
- Scrum
- Lean Startup

Le ROTI

Le ROTI

Return On Time Invested

ou **P.O.U.L.E.T.** en français

Performance Octroyée selon l'Utilité et la Légitimité du sujet En relation avec le Temps passé.



- 0 doigt : totalement inutile, perte de temps
- 1 doigt : pas terrible, aurait facilement trouvé mieux ailleurs
- 2 doigts : pas mal mais cela aurait pu être mieux
- 3 doigts : bien, intéressant
- 4 doigts : super
- 5 doigts : le nirvana, on recommence ?

Découverte du groupe

Découverte du **groupe** : jeu de la **constellation**

- Je connais Lean
- Je compte appliquer des choses dès lundi
- Je sais pourquoi je suis là
- Je connais beaucoup des autres participants
- Ma couleur préféré est le bleu

D'autres questions ?

Atelier Kanban

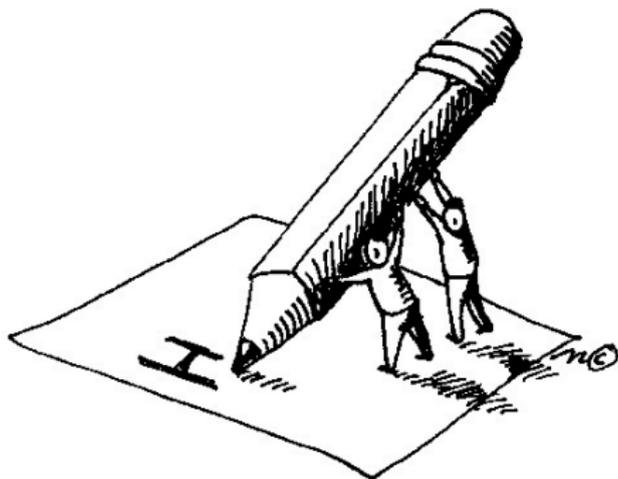
© 2014-2015 Kanban University. All rights reserved.



Atelier Pièces

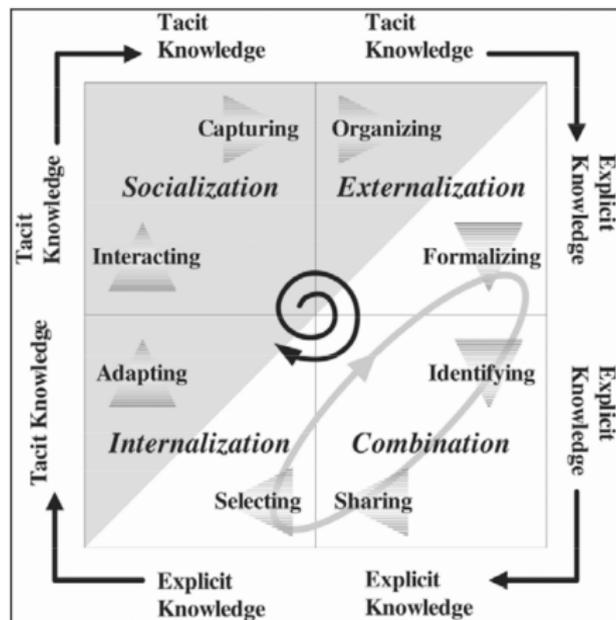


Atelier des noms



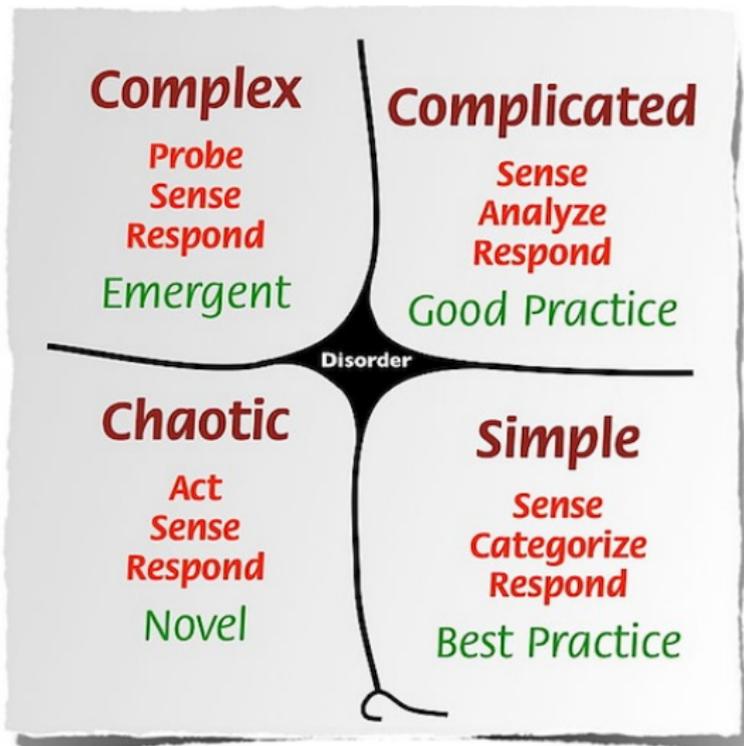
1

Atelier contexte



2

Des réponses de plusieurs types



Les temps complexes

« Agile » c'est quoi ?

Les devises Shadok



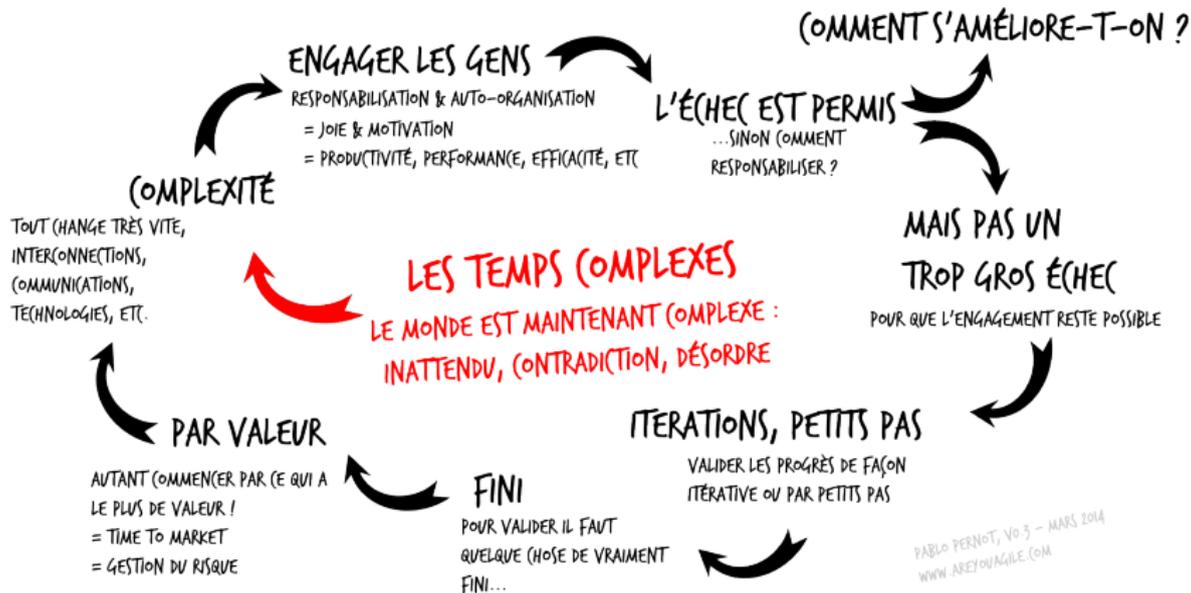
IL VAUT MIEUX POMPER MÊME S'IL NE SE PASSE
RIEN QUE RISQUER QU'IL SE PASSE QUELQUE CHOSE
DE PIRE EN NE POMPANT PAS.

Les temps complexes

Une **culture** de l'**adaptation** en milieu **complexe**



Les temps complexes



Responsabilisation & auto-organisation

Promouvoir l'**auto-organisation** et la **responsabilisation** pour répondre de la meilleure façon au défi de la complexité.

Comme tout système vivant, nous sommes naturellement disposés à l'auto-organisation.

« Chercher à organiser un système auto-organisé n'est pas seulement un oxymore, c'est surtout stupide ! »

– Harrison Owen



Le compromis effort/valeur

Un compromis **effort/valeur** .



L'amélioration continue

S'interroger, progresser, échouer, recommencer, etc.
L'amélioration continue .



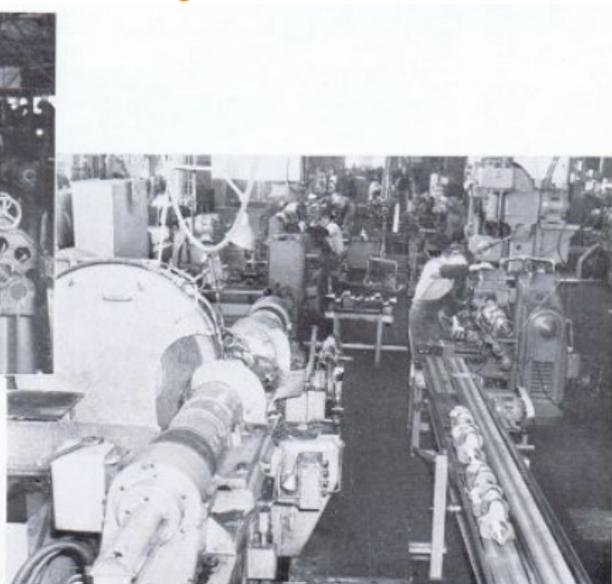
Histoire de la pensée Lean

Ford versus Toyota

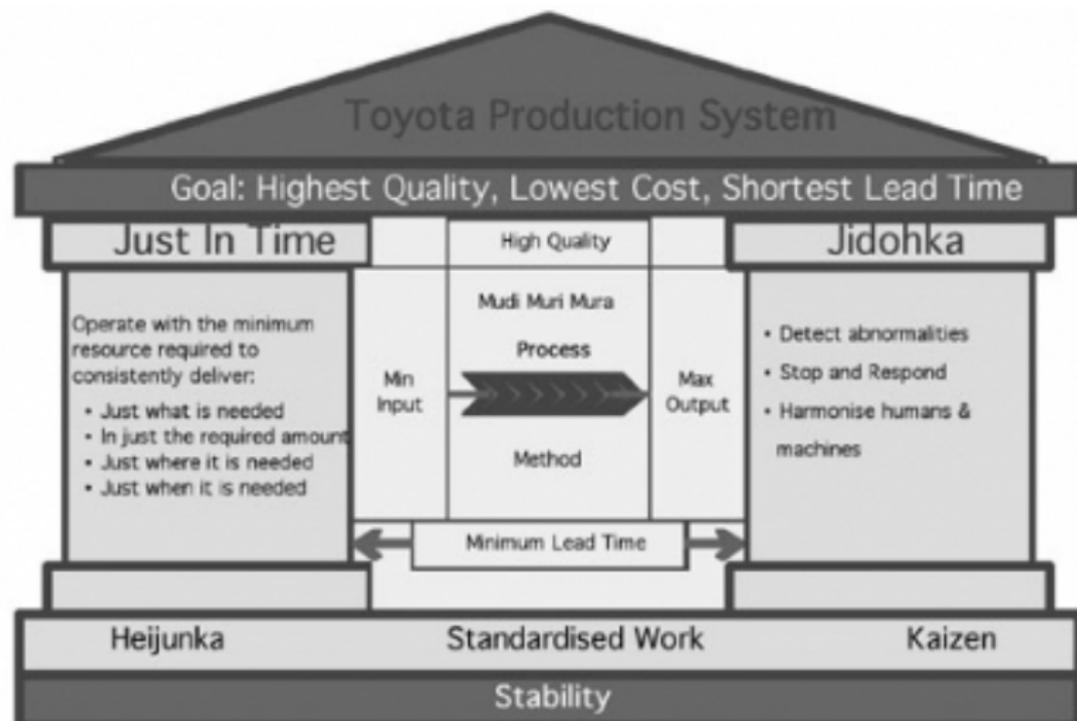


機械工場のクランクシャフト加工ラインの発展。

- (上) ロット生産のため工程中にストックが目立っています (昭和23年ごろ)。
- (右) これをコンベアによる流れ作業方式に改めて工程間の同期化をはかり、ストックをなくしました (昭和28年ごろ)。



la maison Lean



...en 1950

Le petit bréviaire du consultant/coach Lean

Jidohka ou Jidoka : détecter les anomalies et stopper immédiatement la chaîne pour les réparer.

Mudi,Muri,Muda : décrivent différentes formes de gaspillages

Gemba ou genba : Aller voir sur le terrain parmi les employés pour vraiment observer et trouver une amélioration.

Kaizen : Action de s'améliorer continuellement

Heijunka : gestion du flux tendu, juste à temps, de la quantité de travail à réalisé.

Yokoten : Diffusion horizontale des connaissances.

Obeya : salle projet.

Kanban : Management visuel ou étiquette permettant le flux tiré.

La base de la méthode Toyota est de ne pas se satisfaire du statu quo, vous devez constamment vous poser la question : « Pourquoi faisons-nous ça ? ». ⁷



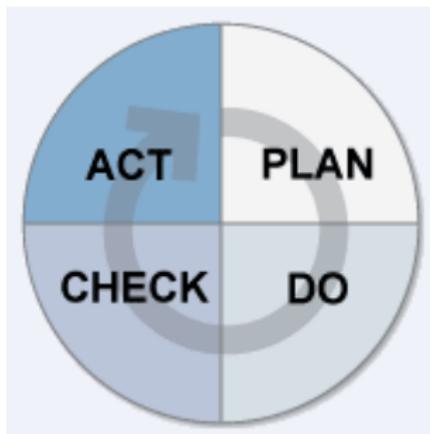
Une philosophie

- Respect des employés
- Une utilisation de toutes les compétences des employés
- Donner des responsabilités et avoir confiance dans les employés

La pensée Lean

- Amélioration continue
- Respect des personnes
- Remettre tout en cause
- Embrasser le changement

8



SHU-HA-RI

守 破 離

SHU : Apprendre les fondamentaux

HA : Trouver les exceptions, trouver de nouvelles approches

RI : Tout redevient permis (on oublie la règle initiale, qui est digérée)

Petit panorama

Lean (naissance 1950-1960)

Une pensée : focus sur le flux de valeur, respect des personnes, amélioration continue

⇒ **Lean Startup**

Extreme Programming (naissance 1996-1998)

Pratiques d'ingénierie logicielle et d'émancipation sociale

⇒ **Software Craftmanship, Devops**

Scrum (naissance 1993-1994, renaissance 2004-2006)

Framework de création de valeur dans un environnement complexe

Kanban Software (renaissance 2005-2009)

Système de gestion de flux de production et de management visuel poussant à l'amélioration continue

⇒ **Scrumban**

Pente douce ou pente forte ?



Le Pic Saint Loup

Les principes du LEAN

#1 Éliminer les gaspillages

#1 Éliminer les gaspillages

Gaspillages ?

- Tout ce qui ne contient pas de valeur
- Fonctionnalité ou code inutile
- Processus et pratiques lents ou inutiles
- Bureaucratie

#1 Éliminer les gaspillages

Gaspillages de l'industrie

Stock
Travail inutile
Surproduction
Transport (autre site)
Attente
Déplacement (même site)
défauts

Gaspillages du logiciel

Développements partiels
Documentation inutile
Fonctionnalité inutile
Mauvaise fonctionnalité
Attente des informations
Multitâches & déplacement
Défauts

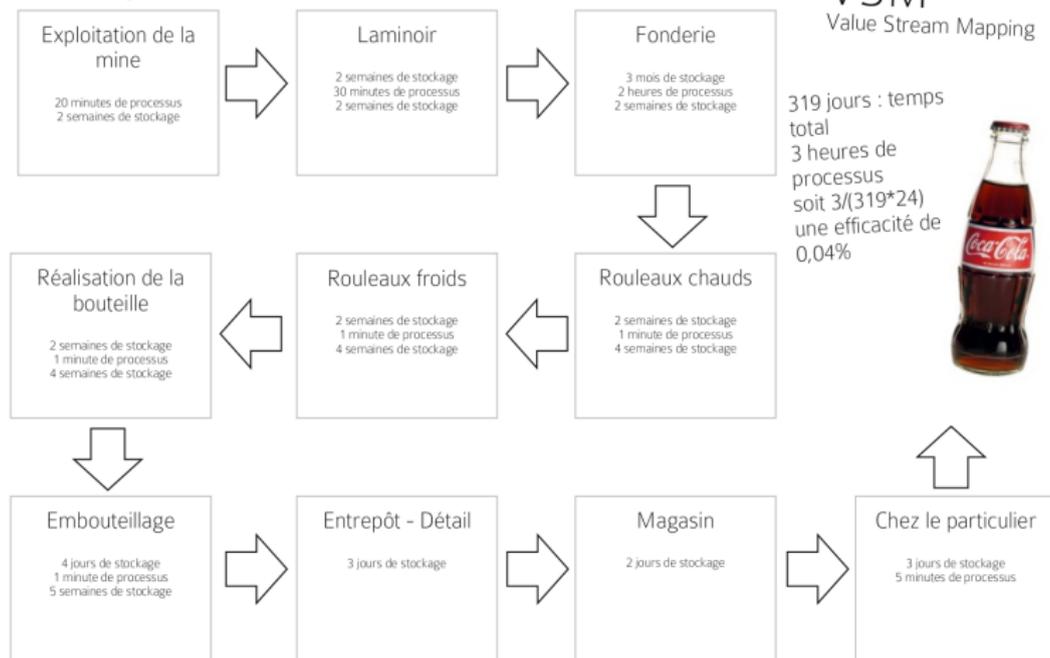
9

#1 Éliminer les gaspillages

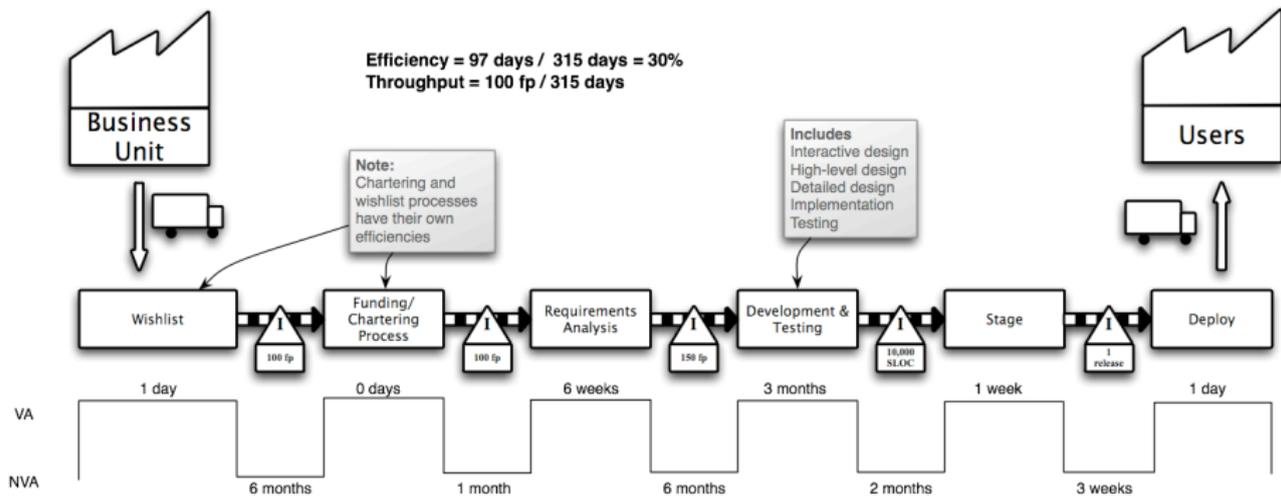
- 1 Faites une liste des 10/15 plus importantes activités dans votre organisation. Prétendez que vous êtes l'un de vos clients et notez les de 1 à 5 (1 = le client ne s'y intéresse pas vraiment, 5 = le client lui donne beaucoup de valeur).
- 2 Prenez les deux dont les scores sont les plus bas (gaspillage), et diminuez par deux le temps accordés à ces deux activités.
- 3 Lors d'une réunion équipe, discutez les 7 principes du Lean et demandez vous au sujet du gaspillage :
 - Êtes-vous d'accord pour dire qu'il s'agit d'un gaspillage ? pourquoi ? ou pourquoi pas ?
 - Estimez combien de temps y est consacré toutes les semaines
 - Qu'est ce qui peut être fait pour réduire ce temps ?

Value Stream Mapping

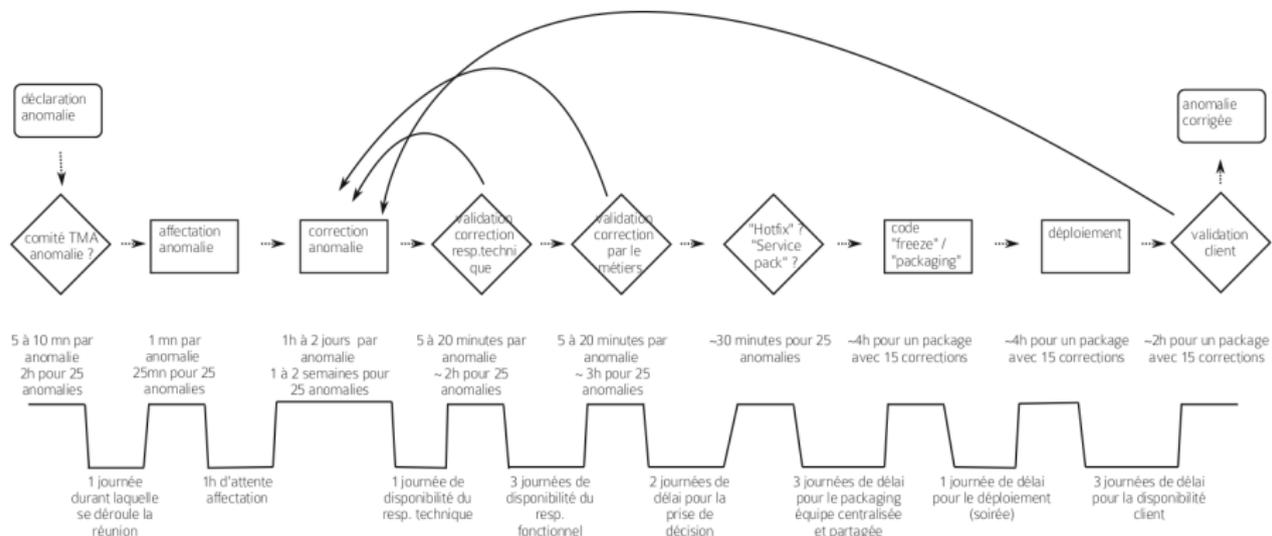
Lean Software Development, Mary Poppendieck
Lean Thinking, Womack & Jones



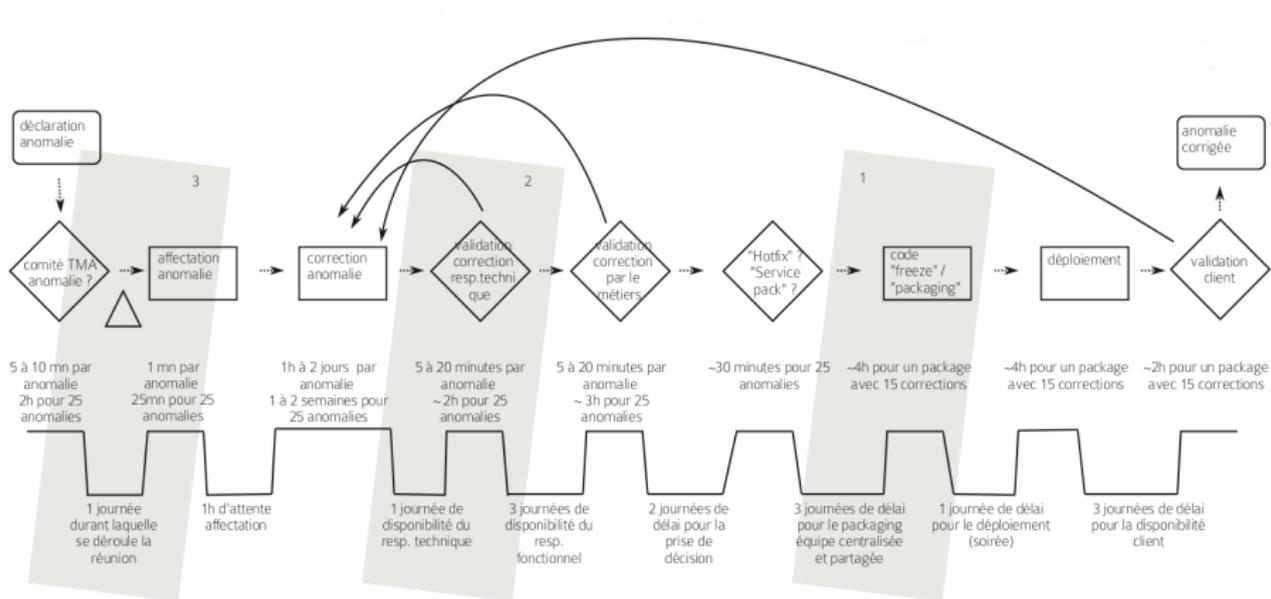
Value Stream Mapping



Value Stream Mapping



Value Stream Mapping



11155 minutes de valeur : ~185 heures -> 181heures -> 179 heures -> 178 heures
 20220 minutes de non-valeur : ~337 heures -> 265 heures -> 241 heures -> 217 heures
 Efficacité : 0,54% (sans aucun retour dans les étapes) -> 0.68 -> 0.74 -> 0.82
 En sortie : 15 anomalies délivrées corrigées

Les 5 pourquoi

L'obstacle n'est pas toujours celui que l'on croit.

Le « Washington Monument » s'érode et la firme responsable du ciment ne réussit pas à en trouver la cause (« root cause analysis »).



- 1 Pourquoi le bâtiment se désagrège-t-il ? Parce que l'on y applique trop de produits chimiques.
- 2 Pourquoi applique-t-on trop de produits chimiques ? Pour nettoyer les crottes de pigeons.
- 3 Pourquoi y a-t-il autant de pigeons ? Car ils mangent les insectes sur le bâtiment.
- 4 Pourquoi y-a-t-il autant d'insectes ? A cause de la lumière !

Solution : Réduire les horaires d'éclairages du monument... 11

Les 5 pourquoi

- 1 Pourquoi doit-on attendre 3 journées pour avoir le packaging de notre hotfix avec 15 anomalies ? Parce que l'équipe packaging, composée de 3 personnes, n'a pas assez de disponibilité.
- 2 Pourquoi l'équipe packaging n'a-t-elle pas assez de disponibilité ? Elle est composée de 3 personnes, fournit 6 équipes-projets-produits, maintient les architectures de "build" et packaging et surtout travaille elle-même sur un projet.
- 3 Pourquoi l'équipe packaging travaille-t-elle sur un projet ? Au début elle était une équipe projet, et elle a monté un système de build dont les autres équipes ont voulu se servir. Et du coup elle est devenue l'équipe de packaging.
- 4 Pourquoi est-elle devenue l'équipe de packaging ? Parce que personne n'a pris le temps d'apprendre à reproduire ces pratiques au sein de sa propre équipe, donc cette équipe a tout pris en charge.

Réponses :

- Former chaque équipe à maintenir son packaging ?
- Constituer une équipe packaging dédiée ?

#2 Amplifier l'apprentissage et la connaissance

La nature du développement logiciel

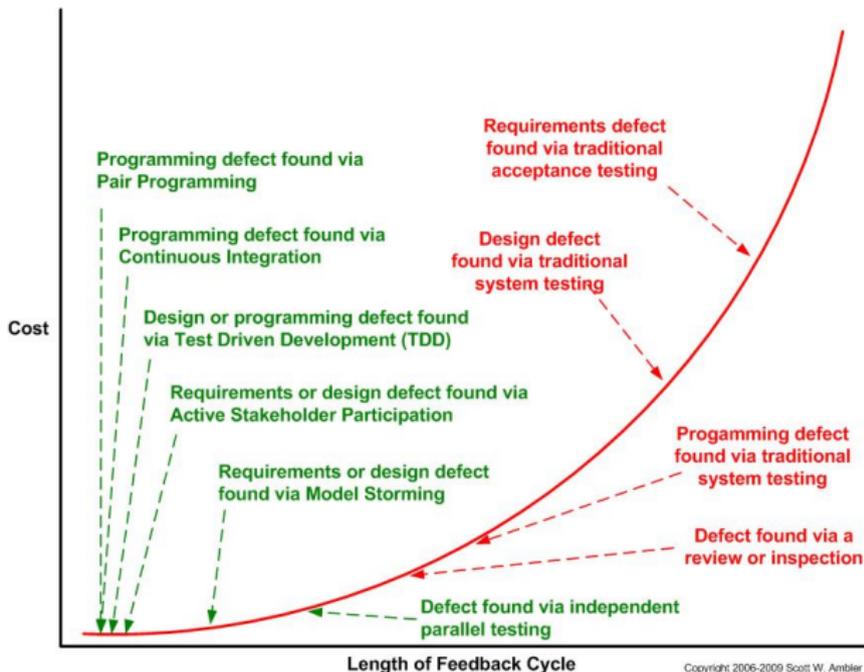
- Poursuite de la réalisation d'un besoin plutôt que le suivi à la lettre de spécifications
- Les résultats ne sont pas pensés pour être générique mais sont adaptés à un besoin utilisateur particulier.
- La meilleure conception est celle qui émerge : avec des cycles courts et répétés d'investigation, d'expérimentation et de validation.
- Nous aimons "essayons, testons, réparons".
- Créer de la connaissance au travers de boucles de feedback est essentiel

12

Le feedback

- Introduisez et augmentez les boucles de feedback au sein de votre développement.
- Lancez vos tests dès que votre code est achevé, ne laissez pas s'accumuler les défauts.
- Moins de documentation et plus du code en état de marche permettant du feedback.
- Plutôt que des sessions de réflexion sur les besoins, faites des sessions de revue des prototypes d'interface.
- Ne faites pas trop d'études sur quel outil sera le meilleur, comparez juste les 3 meilleurs.
- Encouragez et acceptez un feedback immédiat de vos clients finaux.

Plus le *feedback* est rapide, moins cela coûte !



Intégration continue

- Requiert un système de gestion de configuration.
- Requiert des "builds" et des "smoke test" journaliers.
- Requiert une automatisation des tests.
- Requiert une approche verticale des applications : "avoir la capacité de planter un clou à travers tout le système".
- Requiert la capacité à développer des composants de façon modulaire.
- Requiert un haut niveau de communication

15

Sur quoi baser le développement

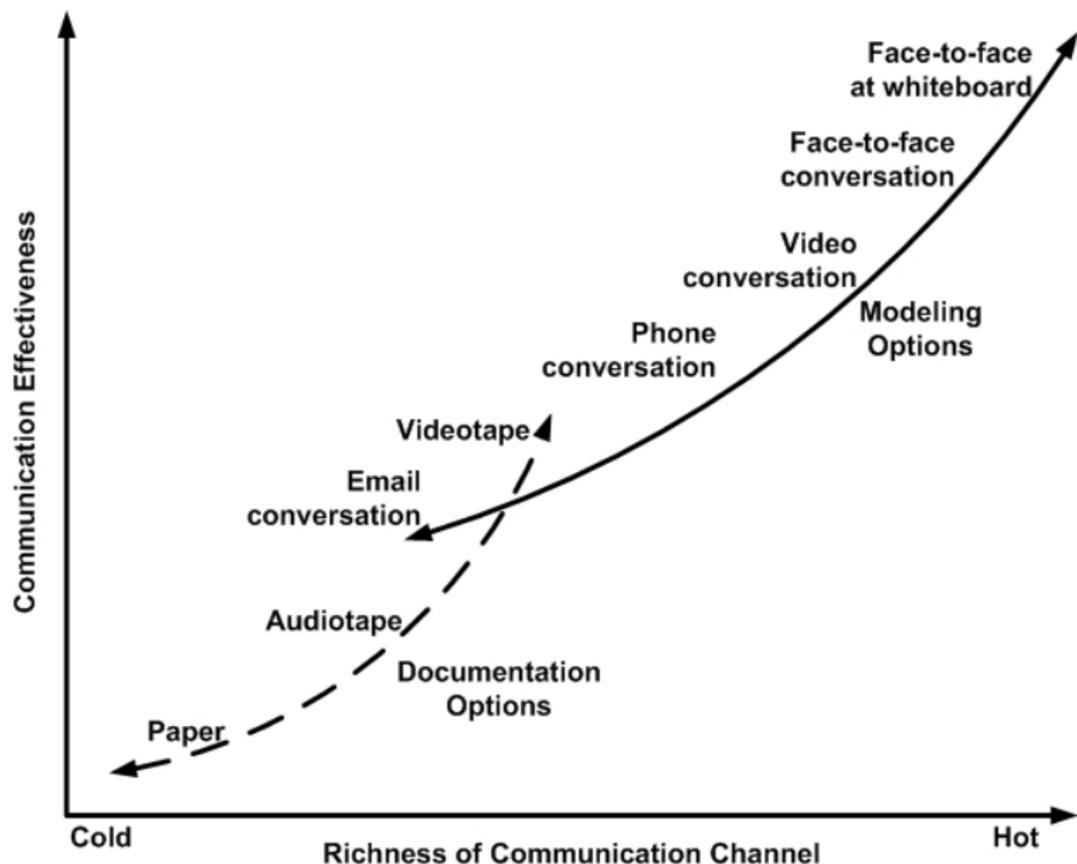
- Il s'agit de communiquer les contraintes pas les choix ou les solutions
- Vous imaginez, développez plusieurs options, vous les observez au regard des contraintes, et vous laissez les solutions émergées
- Vous projetez les itérations suivantes pour développer le reste du système

16

Atelier sur les modes de communication



Modes de communication



Atelier Marshmallow challenge



#3 Décider le plus tard possible

Approche séquentielle ou concurrente

Approche séquentielle

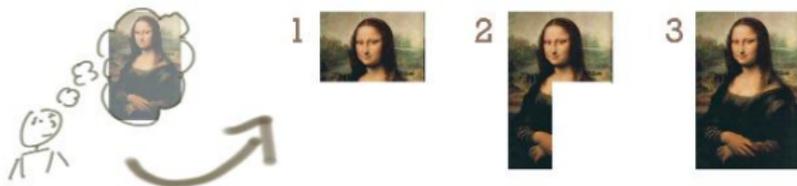
- Approfondissement dès le début (on plonge immédiatement dans tous les détails)
- Ordre de création figé (le jour 1, exemple : nous créons les spécifications).
- Rigide : le changement est coûteux.
- Escalade des coûts quand les défauts sont trouvés tard
- Les décisions importantes et impliquantes sont basées sur des hypothèses et donc avec des incertitudes.

Approche concurrente

- Vue d'ensemble, vision.
- Réalisation des éléments ayant le plus de valeur d'abord
- Le changement est possible et peu coûteux.
- Escalade des coûts est faible, les anomalies sont trouvées au fil de l'eau.
- Les décisions importantes et impliquantes sont possibles au dernier moment responsable.

Approche incrémentale ou itérative

incrémentale



itérative



Conception émergente



20

Options ouvertes

- Un droit pas une obligation de faire quelque chose dans le futur
- Les attentes de nos clients ne sont pas toujours claires ou compréhensibles
- On ne peut pas prédire le futur, il faut donc garder de la souplesse, jusqu'à que l'incertitude disparaisse.
- Les options sont comme des compromis elles ont un coût, ni faciles ni gratuites.



Stanley Kubrick's
**FULL
METAL
JACKET**

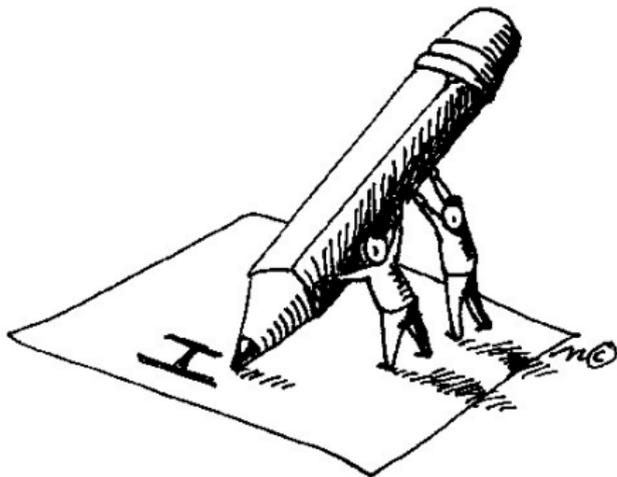
Dernier moment responsable

Le moment où ne pas prendre de décision élimine une alternative importante. Ce n'est pas de la procrastination !

Comment le saisir ?

- Partager les informations de "design" qui sont achevées
- Organiser la collaboration
- Développer un état d'esprit rompu au changement
- Faire de la conception orientée objet ou orientée composant métier
- Savoir ce qui est critique dans le domaine métiers concerné
- Développer une capacité à comprendre *quand* les décisions ont besoin d'être prise.

Atelier *Les 2mns*



Prendre des décisions

Approfondissement dès le début

- S'engager tôt sur des éléments
- Besoin d'un accord tôt sur un périmètre précis
- Rationalisation
- Prédications & hypothèses guident les décisions

...ou **Vue d'ensemble, vision**

- Permet de retarder les engagements
- Nécessite quelqu'un de bon sens pour comprendre comment les détails émergeront et quand il sera temps de s'engager
- Décision intuitive
- Information en temps réel et le feedback guident les décisions

#4 Montrer le plus tôt possible

Montrer le plus tôt possible

Pourquoi montrer, délivrer rapidement ?

- Nos clients aiment être rapidement livrés.
- Montrer, délivrer rapidement laisse moins de temps à nos clients pour changer d'avis
- Des produits en cours de fabrication, ou finis partiellement peuvent contenir des anomalies encore ignorées
- Le plus souvent il est possible de montrer, délivrer, le moins nous avons besoin de choisir de prendre des décisions car cette flexibilité nous laisse du temps.

Systemes "Just in time", systemes "flux tirés"

- Permet aux gens de se responsabiliser sur le travail à réaliser. Le travail devient auto-géré.
- Les cheminements complexes peuvent être stoppés rapidement dès qu'un problème est repéré.
- L'approche "just in time" permet aux décisions d'être prise aux moments opportuns, en temps réel, et pas en avance.

Théorie des files d'attente

La clef est de réduire le temps de cycle, c'est à dire le temps que prend élément à réaliser pour traverser tout le processus.

Cadence régulière du temps de cycle

- Recherchez une cadence régulière d'arrivée et contrôlez régulièrement ce que contient votre file d'attente.
- Délivrer, découper en petits éléments permet de diffuser facilement le travail à réaliser au sein des équipes.
- Placez des priorités et choisissez les éléments les plus critiques.
- Délivrer régulièrement le travail produit est encore plus important.

Cadence régulière de service

- Enlever les variations liées aux processus
- Avec la livraison de petits éléments les erreurs sont aussi plus petites
- On peut travailler en parallèle sur plusieurs éléments pour éviter les blocages
- Assurez constamment des livraisons

Prioriser

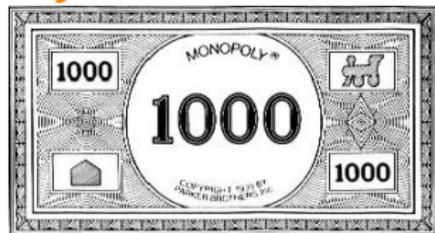
Exemples d'approches pour prioriser.

MoSCoW

- Must Have
- Should Have
- Could Have
- Would Have

26

Buy a feature



Autre outils : *Dot Voting*

Gérer le *Slack* (évitez le plein emploi)

100% d'utilisation n'est pas forcément le plus efficace, on a besoin de moments de respiration.

Pratiquez la règle des 80/20 concernant la charge de travail.

Les "gros morceaux" mettent plus de temps à être réalisés dans les files d'attente, demandent plus de personnes, etc.

27

Pensez aux Lab Day de Atlassian, Google ou 3M.

#5 Responsabiliser et engager les équipes

Qu'est ce qu'une organisation mature ?

Hypothèse 1 du Lean

Une organisation mature s'attache à l'ensemble du système. Elle ne s'attache pas à optimiser une seule sous-partie ou plusieurs sous-éléments du système.

Hypothèse 2 du Lean

Une organisation mature s'attache à un apprentissage efficace en son sein permettant ainsi une responsabilisation, un engagement de ses équipes (ceux qui font le travail) pour prendre des décisions.

28

Auto-organisation

Transférer les pratiques d'un contexte à un autre est souvent une erreur. Laissez l'équipe définir elle-même ses procédures. Rappelez vous que le rôle du management est de coacher, de former, d'assister les équipes. Il est essentiel de comprendre les fondements principaux qui composent les pratiques, et de transformer ces principes en nouvelles pratiques. Les managers doivent s'améliorer autant que les équipiers. Une boucle de feedback est essentielle entre le management et les équipes pour mener les améliorations.

29

Motivation

Définir un objectif, un but au travail. Les gens s'attachent plus à un but qu'à eux mêmes.

Il doit être clair.

Il doit être atteignable.

L'équipe doit avoir accès aux clients.

Laissez l'équipe faire ses propres choix d'engagement.

Le rôle du management est de fournir du support, des ressources matérielles, des conseils et de la protection.

30

Engager les équipes

Responsabilisation , Haut niveau d'exigence , implication ,
quelles sont les clefs ?

Qui s'implique avec un très haut niveau d'exigence plus de 20h par semaine pour échouer, s'améliorer, réaliser une documentation aux petits oignons, et en plus en payant !? Ils sont 20 000 000, qui sont-ils ?

Engager les équipes

Les **jeux de vidéos** ...

...ne pourraient-ils nous enseigner quelques leçons ?



Pourquoi mon fils n'a-t-il pas le même niveau d'exigence pour son travail scolaire qu'avec ses constructions dans *minecraft* ?

Un bon cadre (de jeu)

Un bon cadre (de jeu)³¹ pour votre entreprise :

- 1 Un **objectif clair**
- 2 Des **règles claires**
- 3 **Feedback**
- 4 **Invitation**

De nouvelles réunions

Imaginez que toutes vos réunions soient basées sur l'**invitation** ³² ?
...qu'elles s'achèvent par un **R.O.T.I.** (amélioration continue) ?
...proposent des **objectifs clairs** et des **règles claires** ?

Une bonne implication (dans un jeu)

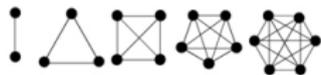
Une bonne implication (comme dans un jeu)³³ nécessite :

- 1 Un **sentiment de contrôle**
- 2 Un **sentiment de progrès**
- 3 **Appartenance à une communauté**
- 4 Agir pour un **but qui nous dépasse**

Organisation de l'entreprise

Debrief

- Des objectifs divergents sans affrontement
- Casser les silos
- Des équipes hétérogènes (et petites)
- Approche organique et non industrielle

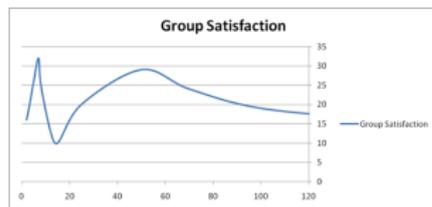


People	2	3	4	6	6	(n)
Interfaces	1	3	6	10	15	$\frac{n(n-1)}{2}$

34 35

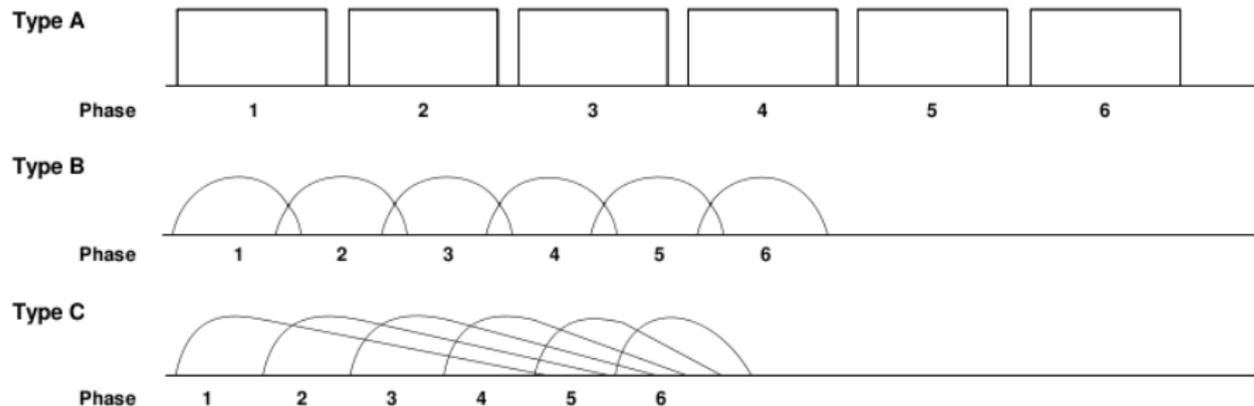
À connaître

- Le chiffre de Dunbar (150-220)
- La satisfaction d'appartenance à un groupe (7-50)



Organisation de l'entreprise

Sequential (A) vs. overlapping (B and C) phases of development



36

Atelier Leadership

Leadership

Manager

- Gère la complexité
- Planifie et budgetise
- Organise et assigne
- Supervise et contrôle

Leader

- Gère les changements
- Donne une vision
- Aligne, inspire les personnes
- Rend possible la motivation



Atelier du devin



Responsabilisation et engagement

Debrief

- Effet Pygmalion (ou Rosenthal)
- Processus de responsabilisation de Christopher Avery
- Leader-Mentor et pas Manager-Control Freak
- Facilitateur de vos équipes et de votre organisation (*Servant Leadership*)



Expertise

- Développez et faites fructifier des groupes d'expertises.
- Faites la promotion du mentorat et du "pair programming"
- Encouragez la formation et l'auto-formation
- Développez des standards et pratiquez les
- Proposez l'approche "Lab Day" (80/20 : Google, 3M, Atlassian)

37

Amélioration continue

■ Des outils de *root cause analysis*

- les 5 pourquoi
- la liste Phoenix
- le rapport A3 de Lean

■ Des outils de type *Solution Focus*

- Remember the future

■ Des outils de rétrospective

- Tour de table classique
- L'étoile de mer
- Le *speedboat*
- La rétrospective *Jeopardy*
- A la façon des cadavres exquis
- En se basant sur *Ishikawa*
- Provocative : qu'avez vous fait d'inutile ?
- ...

Il faut chercher à **voir les choses différemment** .

La liste phoenix

The problem

- Why is it necessary to solve the problem ?
- What benefits will you receive by solving the problem ?
- What is the unknown ?
- What is it you don't yet understand ?
- What is the information you have ?
- What isn't the problem ?
- Is the information sufficient ? Or is it insufficient ? Or redundant ? Or contradictory ?
- Should you draw a diagram of the problem ? A figure ?
- Where are the boundaries of the problem ?
- Can you separate the various parts of the problem ?
- Can you write them down ?
- What are the relationships of the parts of the problem ?
- What are the constants of the problem ?
- Have you seen this problem before ?
- Have you seen this problem in a slightly different form ?
- Do you know a related problem ?
- Try to think of a familiar problem having the same or a similar unknown.
- Suppose you find a problem related to yours that has already been solved. Can you use it ?
- Can you use its method ?
- Can you restate your problem ?



La liste phoenix

- Can you solve the whole problem? Part of the problem?
- What would you like the resolution to be?
- Can you picture it?
- How much of the unknown can you determine?
- Can you derive something useful from the information you have?
- Have you used all the information?
- Have you taken into account all essential notions in the problem?
- Can you separate the steps in the problem-solving process?
- Can you determine the correctness of each step?
- What creative thinking techniques can you use to generate ideas?
- How many different techniques?
- Can you see the result?
- How many different kinds of results can you see?
- How many different ways have you tried to solve the problem?
- What have others done?
- Can you intuit the solution?
- Can you check the result?
- What should be done? How should it be done? Where should it be done? When should it be done? Who should do it?
- What do you need to do at this time?
- Who will be responsible for what?
- Can you use this problem to solve some other problem?



Le rapport A3 de Lean

○

THEME: "What are we trying to do?"

○

Background

- Background of the problem
- Context required for full understanding
- Importance of the problem

Current Condition

- Diagram of current situation (or process).
- Highlight problem(s) with storm bursts.
- What about the system is not IDEAL.
- Extent of the problem(s), i.e., measures.

○

Cause Analysis

- List problem(s)
- Most likely direct (or root) cause:

Why? Why?
Why? Why?
Why? Why?

To: _____
By: _____
Date: _____

Target Condition

- Diagram of proposed new process
- Countermeasures noted as fluffy clouds
- Measurable targets (quantity, time)

Implementation Plan

What?	Who?	When?	Where?
Actions to be taken	Responsible person	Times, Dates	
Cost:			

Follow-Up

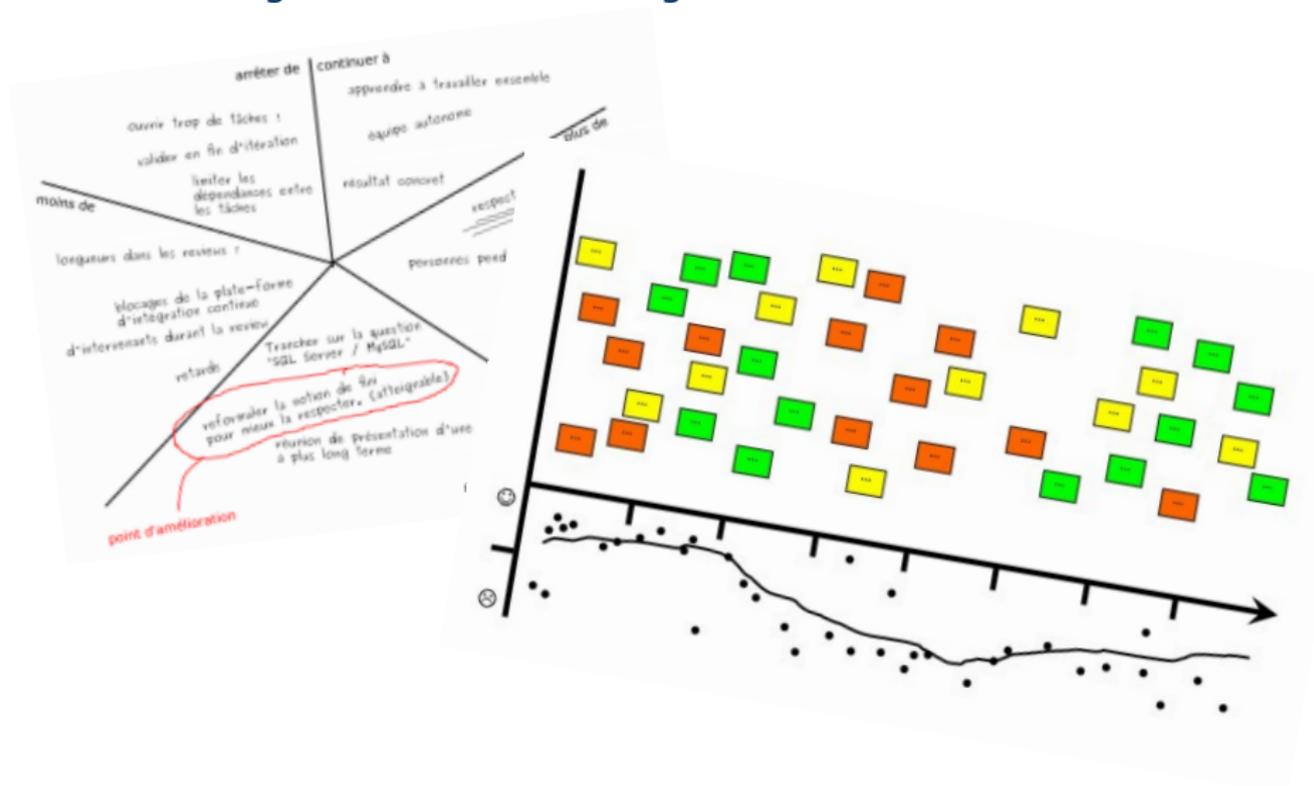
Plan	Actual Results
<ul style="list-style-type: none">• How will you check the effects?• When will you check them?	<ul style="list-style-type: none">• In red ink/pencil.• Date check done.• Results, compare to predicted.

Atelier Remember the future



Rétrospective

Une réunion régulière où l'on s'interroge sur comment s'améliorer. ³⁹



Rétrospective

3 étapes d'une rétrospective ⁴⁰

- 1 Rappelez le contexte et le fait (utilisez une ligne de temps?)
- 2 Collectez les choses positifs et négatifs
- 3 Choisir l'élément que l'on souhaite résoudre (aspect négatif) ou répandre (aspect positif)

Un seul plan d'action, un seul élément, sinon vous risquez de vous disperser et de n'obtenir aucun résultat.

#6 Qualité intrinsèque

Qualité ?

Qualité perçue

est celle perçue par l'utilisateur final face à l'ensemble du système.

- Est-ce que le système est intuitif ?
- Comment le système réagit-il aux changements du domaine (métiers) ?
- Le système résoud-t-il des problèmes métiers ?
- Combien de part de marché possède-t-il ?
- Quelle est sa popularité ?

Qualité conceptuelle

est celle au coeur du système : ses concepts principaux devraient proposer une approche globale cohérente.

- Existe-t-il un équilibre entre flexibilité, maintenabilité, efficacité et performance ?
- Le système peut-il évoluer facilement ?
- Possède-t-il un ensemble de principes de conception ?
- Est-il convivial ? ergonomique ? intuitif ?

Qualité perçue

- Les petits systèmes devraient être développés par une seule équipe et montrer au plus tôt aux utilisateurs finaux.
- De courtes itérations devraient être réalisées pour avoir du feedback d'un maximum de profils impactés par le système.
- Les tests utilisateurs proposent un excellent processus de communication autour du projet
- Les systèmes complexes devraient être représentés en proposant des modèles et des définitions qui soient compréhensibles par tous et utilisés tel quel par les développeurs.
- Les grands systèmes devraient être suivis par un développeur senior possédant un gros bagage technique et une vraie connaissance métiers.

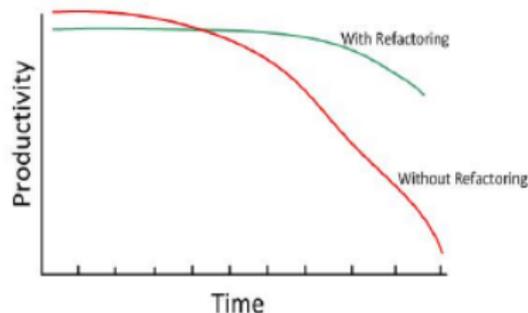
Qualité conceptuelle

- La qualité de la communication pour toutes les décisions est essentielle
- Ne complexifiez pas dès le début la conception.
- Conception émergente, résolution des problèmes
- Comprendre et résoudre le problème au même moment
- Les informations préliminaires sont immédiatement communiquées et n'attendent pas d'être complètes pour être fournies.
- L'information est communiquée morceau par morceau et pas d'une façon monolithique.
- L'information est bi-directionnelle, pas uni-directionnelle.
- La meilleure façon de communication est face à face et pas uniquement de la documentation.



Refactoring

- Les systèmes complexes ont des propriétés qui ne peuvent être anticipées lors de la conception.
- L'architecture doit rester saine alors que le système s'étoffe.
- La qualité conceptuelle doit rester saine
- Simplicité : les conceptions simples sont les meilleures
 - Clarté : le code doit rester compréhensible
 - Fait pour être utilisé : le code est destiné à un usage
 - Pas de répétition : ne jamais dupliquer de code
 - Pas de chose inutile : ne fabriquez pas des choses inutiles



KISS

Keep It Simple Stupid

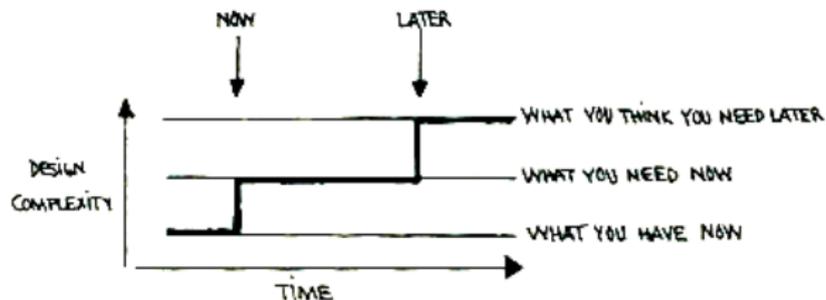
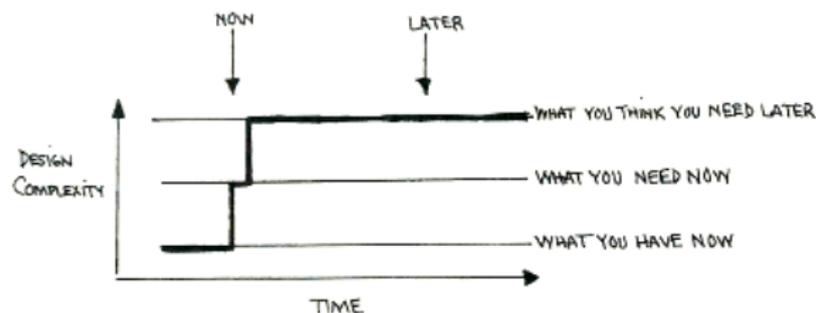
YAGNI

You Ain't Gonna Need It

DRY

Don't Repeat Yourself

Refactoring



Tests

- Permet de communiquer comment les choses "devraient" marcher.
- Permet de savoir si le système fonctionne comme cela a été prévu.
- Donne une structure qui permet aux développeurs d'opérer des changements lors de la réalisation.
- Donne de l'autonomie aux développeurs

Tests unitaires, systèmes, d'intégration, de non-régression, métiers, d'interface, de performance, d'acceptation, utilisateurs, sauvages, manuels, etc. Le problème des tests c'est leurs noms.

Tests & communication

- Les tests développeurs communiquent comment le système devrait fonctionner (Documentatin technique ?)
- Les tests utilisateurs communiquent par l'exemple comment le système à besoin de fonctionner.
- Les tests utilisateurs peuvent remplacer les spécifications (une forme de Test Driven Development, les tests font le "design").
- Les conceptions ne sont pas achevées tant que les tests ne sont pas exécutés (Design Driven Development)

45

Tests & feedback

- Feedback immédiat (grâce aux tests) est essentiel au développement
- L'écriture des tests utilisateurs est réalisé avant la fin du développement.
- Les tests automatisés sont la clef des intégrations continues

46

Tests : une approche protectrice

- Un socle qui doit vous permettre de faire des choses qui autrement seraient dangereuses
- L'approche LEAN nécessite cette protection (décisions prises au dernier moment responsable, changement de code tardif, etc.)
- L'automatisation des tests est essentielle
- L'ensemble automatisé des tests utilisateurs et développeurs vous garantit la bonne santé de votre système
- Pour réduire le coût de la maintenance, maintenez cet ensemble de tests automatisés

Quels indicateurs ? Quelles mesures ?

Critères pour un indicateur et une mesure :

Actionnable : démontrer une cause et un effet clair.

Accessible : lisible et visible.

Auditable

Les « **actionnable metrics** » contre les « **Vanity Metrics** »⁴⁷

#7 Voir le tout

Pensée systémique

Un système n'est pas la somme de ses composants, mais le produit de ceux-ci .

Quand un système commence à disfonctionner on lui applique généralement des règles séquentielles assez rigides. une approche séquentielle rigide peut soigner un symptôme mais pas la cause originelle. Il va devenir de plus en plus difficile à maintenir.

Motifs récurrents de la pensée systémique

- **Limite de croissance** : même quand un processus délivre la valeur attendue il génère d'autres contraintes qui impacteront son effet et limiteront sa croissance.
- **Théorie des contraintes** : vous pouvez supprimer une contrainte pour libérer une croissance, mais cette contrainte se reportera à d'autres endroits.
- **Déplacement de charge** : si on adresse les symptômes plutôt que la cause source (*root cause*).
- **Sous-optimisation** : Plus un système est complexe, plus vous êtes tentés de le traiter par morceaux, c'est une erreur.

Indicateurs

- Quand vous demandez des mesures de performances, notamment des compétences des développeurs, vous faites une erreur.⁴⁹
- Les gens vont optimiser les réponses attendues par vos indicateurs et ainsi ne plus adresser le système comme un tout.
- Ne cultiver pas ce genre de comportement. Si vous ne pouvez pas mesurer le tout, n'essayez pas.
- Un défaut dans le code n'est pas toujours la faute d'une personne. Blamer un développeur ne règle pas la cause source (root cause) du problème.
- Les indicateurs devraient encourager une optimisation du tout, de l'ensemble, et pousser l'équipe à s'améliorer en ce sens.
- La culture moderne pousse à un succès individuel, c'est une sous-optimisation qui va à l'encontre de la pensée systémique.

50

49. Et d'ailleurs comment vous mesurez vous vous-même ?

50. M & T Poppendieck, Lean Software Development

Fils de Lean
Descendances modernes
dans le domaine informatique
Kanban Software,
Scrum, Lean Startup
et d'autres

Kanban

Rendez **visible** votre création de **valeur**, rendez visible votre **activité**, laissez vous bousculer par un affichage physique et tactile mural.



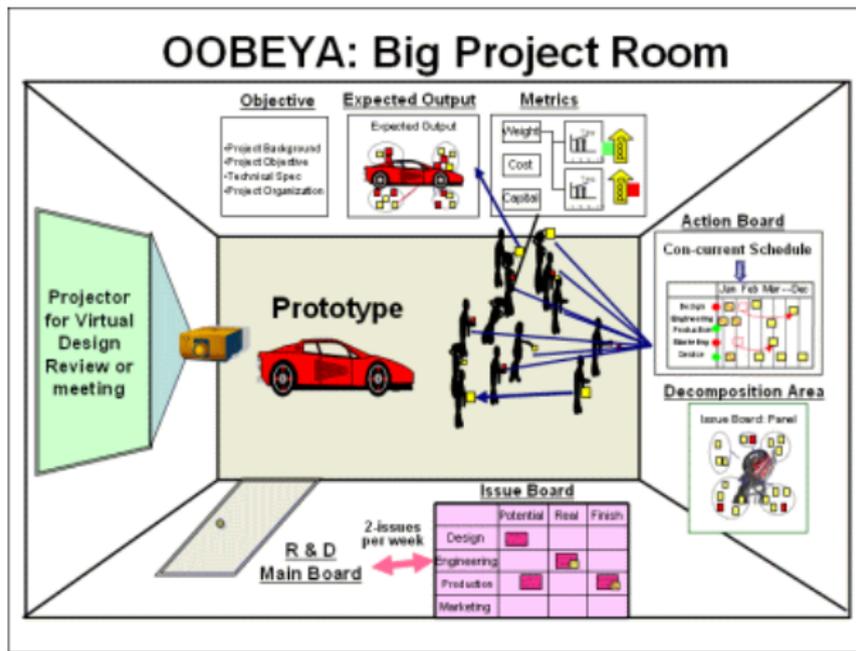


Rappels

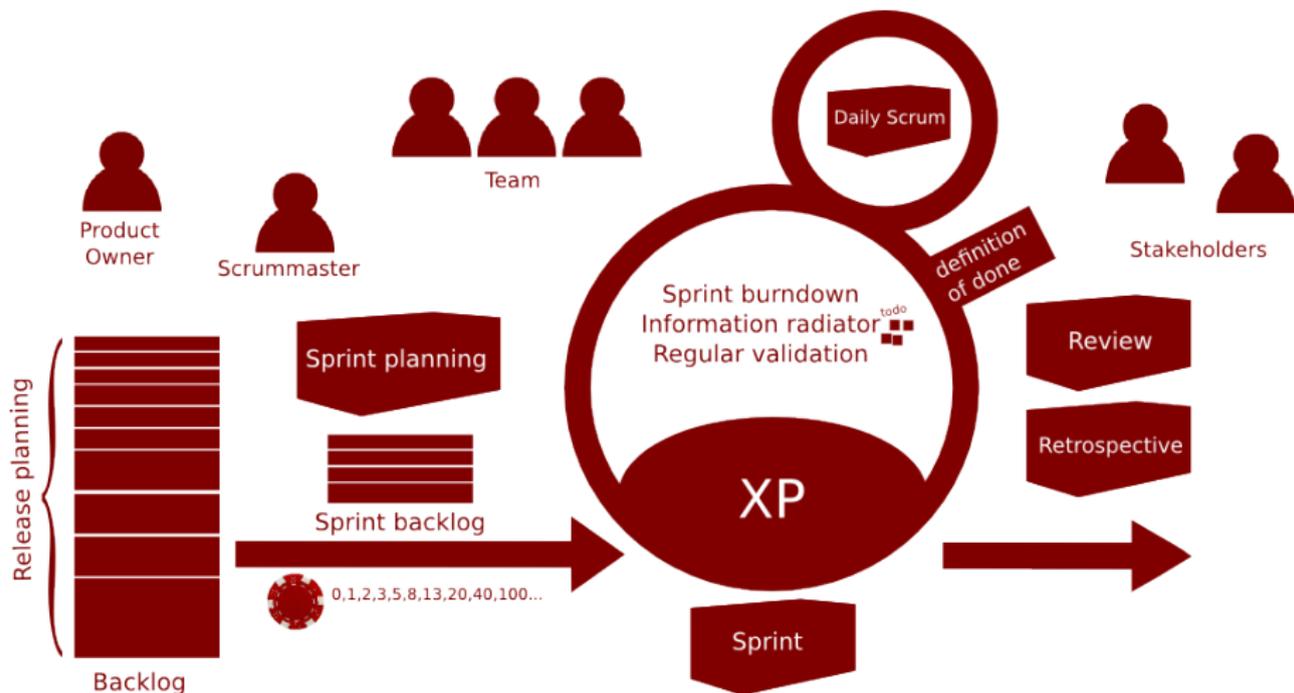
- 1 Commencer par ce que vous faites maintenant : rendez visible votre activité
- 2 Respecter le processus actuel, les rôles et responsabilités, ne changez rien
- 3 S'engager à changer de manière incrémentale et évolutive

Pratiques

- 1 Visualiser la création de valeur.
- 2 Gérer le flux de travail (en changeant les choses peu à peu).
 - Rendre explicites les règles de gestion des processus.
 - S'améliorer de façon collaborative.
- 3 Limiter le flux pour que celui-ci s'écoule plus vite.



Scrum



52

Vision & sens

1 Business Model Canvas

2 Elevator Pitch⁵⁴

3 Carte projet

NOM :

PORTEUR :

MAXIME COMMERCIALE :

3 CRITERES DE SUCCES :

*

*

*

MESURABLE COMMENT ?

*

*

*

*

Qui achète ? Pourquoi ?

*

*

Qui utilise ? Pourquoi ?

*

*

*

Lean Canvas

<p>PROBLEM 1 Top 3 problems</p> <p>Répondre aux spécificités de son animal (alimentation, reproduction)</p> <p>Difficulté vie sociale en zone rurale</p> <p>Difficulté avoir un animal en milieu urbain</p>	<p>SOLUTION 4 Top 3 features</p> <p>Rencontrer des possesseurs d'animaux comme le mien</p> <p>la garde partagée par la communauté</p> <p>l'expertise de la communauté (possesseurs et vétérinaires)</p> <p>KEY METRICS 8 Key activities you measure</p> <p>AARRR Pirate metrics pour le site</p> <p>Cash par semaine pour la boutique</p>	<p>UNIQUE VALUE PROPOSITION 3 Single, clear, compelling message that states why you are different and worth buying</p> <p>Peetic, le meetic de l'animal</p> <p>On n'est jamais mieux servi que par les gens comme soi</p> <p>une communauté de personnes avec les mêmes attentes que moi concernant mon animal de compagnie</p>	<p>UNFAIR ADVANTAGE 9 Can't be easily copied or bought</p> <p>On a passé beaucoup de temps à travailler pour Duchesse Féline On connaît très bien les interlocuteurs + fournisseur proche</p> <p>CHANNELS 5 Path to customers</p> <p>Pub en ligne Pub papier revues spécialisées website/mobile vétérinaires fédérations</p>	<p>CUSTOMER SEGMENTS 2 Target customers</p> <p>Personnes 20-50 ans urbaines (spécificités)</p> <p>Personnes 40-70 ans rurales (sociabilité)</p> <p>Porteur d'affaires pour : Vétérinaires, fabricants, Eleveurs</p> <p>Early Adopters</p> <p>Jeunes femmes 20/30 ans pouvoir d'achat + spécificités aiguisées</p>
<p>COST STRUCTURE 7 Customer Acquisition Costs</p> <p>Distributing Costs</p> <p>Hosting</p> <p>People, etc.</p> <p>Publicités en ligne Infrastructure et développement visites fédérations, vétérinaires, fabricant Publicités papier revues spécialisées (traces, vétérinaires, fédé...)</p>		<p>REVENUE STREAMS 6 Revenue Model</p> <p>Lifetime Value</p> <p>Revenue</p> <p>Gross Margin</p> <p>Pub Marge sur la boutique : nourriture, tintoys, etc Abonnement/pub des vétérinaires Accès club privilégié : accès contenu spécialisé Revente inscrits</p>		

Personas

Bien se projeter vers l'utilisateur (et l'acheteur...).



Jessica & Tom
son petit brabançon



Quand

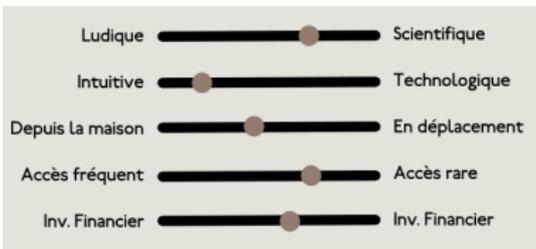
- Le matin au petit déjeuner durant le café (mobile)
- Le soir au coin du feu avec Corsair sur les genoux (tablette)

Comment

- Téléphone mobile (android)
- Tablette (android)

Pourquoi

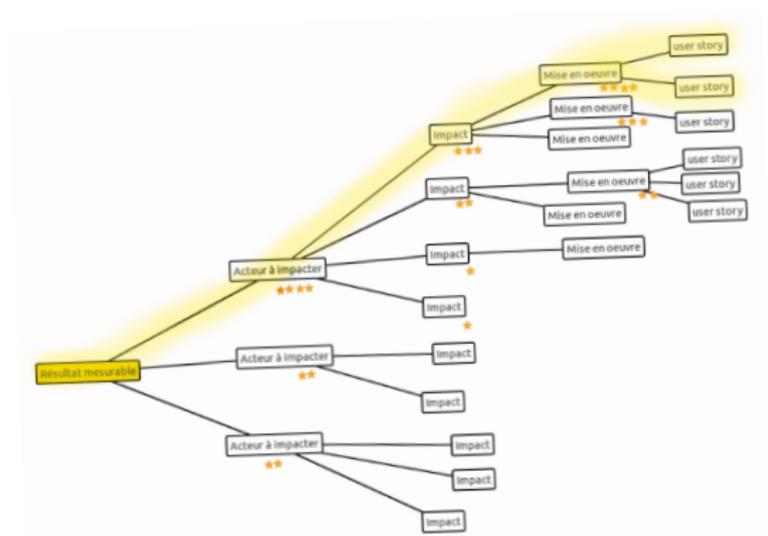
- Trouver de la nourriture adéquat pour son tout petit animal
- Avoir des conseils pour l'entretien d'un si petit animal (notamment dans son contact avec les autres chiens ou chats)



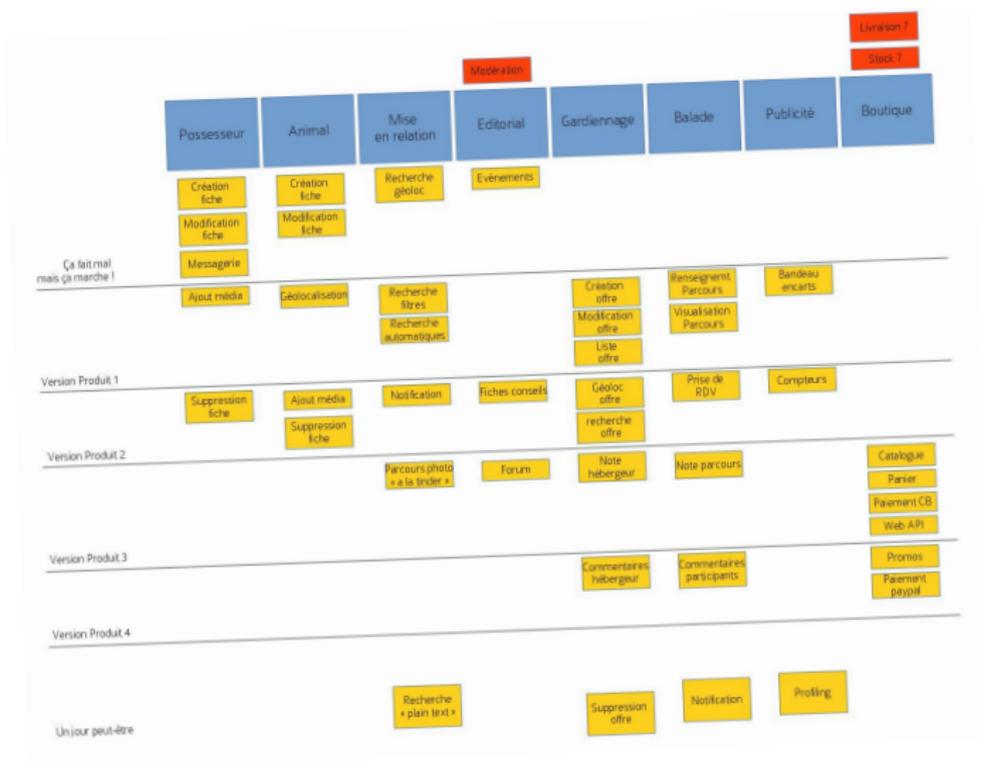
Impact Mapping

...de la fiche projet au *chemin critique*⁵⁵

- 1 Quelle est ma cible ?
- 2 Qui dois-je impacter pour l'atteindre ?
- 3 Quel impact sur cette personne me permettra de l'atteindre ?
- 4 Quelle implémentation pour cet impact ?



User Story Mapping



Open Agile Adoption

Conduite du changement basée sur l'**invitation** ⁵⁶ et le **forum ouvert** ⁵⁷.

- Invitation
- Forum Ouvert
- Liminalité
- Storytelling
- Auto-organisation

56. Open Agile Adoption par Dan Mezick

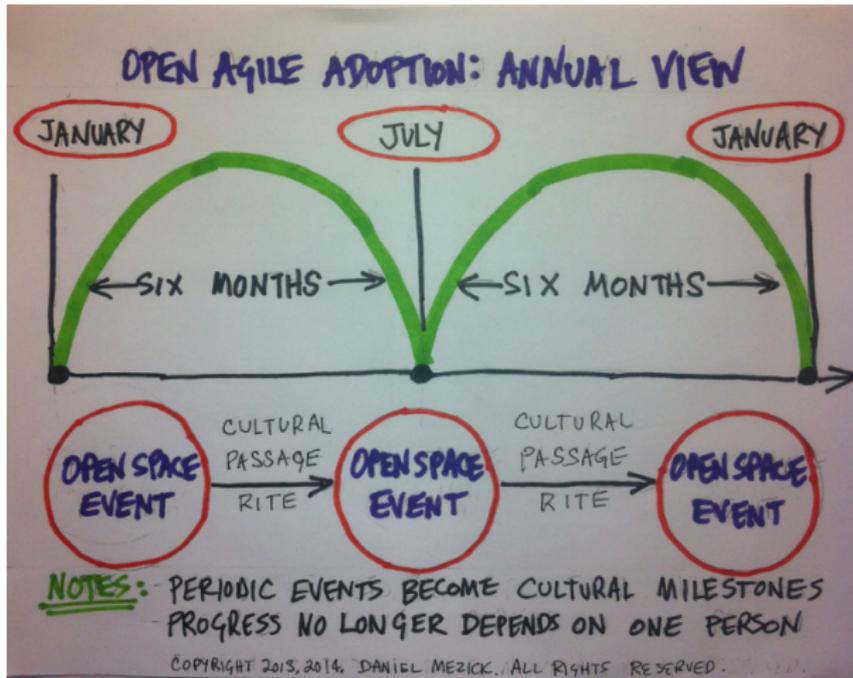
57. Forum ouvert, Guide du forum ouvert

Conduite du changement agile



A posteriori de l'*Open Agile Adoption* (appelez ça « conduite du changement agile » c'est plus simple) et tout au long de la vie de l'entreprise le **storytelling** est essentiel : quelles sont les histoires qui se racontent à la machine à café ? Voilà une question cruciale pour l'entrepreneur. Alimenter ces histoires avec celles des succès de votre conduite du changement.

Open Agile Adoption



58

Références

Le guide rapide pour survivre à la formation

<http://www.areyouagile.com/pdf/guiderapide.pdf>

Sur Lean et les méthodes agiles

- [Guides scrum](#) de [scrum.org](#), en français, en anglais, en polonais...
- *Scrum*, Claude Aubry (fr)
- *Kanban pour l'IT*, Laurent Morrissette (fr)
- *Retrospective*, Esther Derby & Diana Larsen (en)
- *Lean Startup*, Eric Ries (en)
- *Agile Lean Toolkit*, Mary Poppendieck (en)

Sur Kanban

- [Kanban pour l'IT, de Laurent Morriseau \(fr\)](#)
- [Kanban, David Anderson \(en\)](#)
- [Retrospective, Ester Derby & diana Larsen \(en\)](#)
- [Lean Software Development, an agile toolkit, Mary Poppendieck \(en\)](#)
- [The machine that changed the world, Womack, Jones, Roos \(en\)](#)
- [Priming Kanban, Jesper Boeg \(en\)](#)
- [Kanban vs Scrum, Henrik Kniberg \(en\)](#)
- [Kanban for sceptiks, Nick Oostvogels \(en\)](#)
- [Holy Land Kanban, Yuval Yeret \(en\)](#)

Sur la culture agile

- *Freedom Inc*, Isaac Getz (Liberté & compagnie en français)
- *The culture game*, Dan Mezick (en)
- *Tribal Leadership*, Dave Logan (en)
- *Introduction à la pensée complexe*, Edgar Morin (fr)
- *La revanche du rameur*, Dominique Dupagne (fr)
- *Drive*, Dan Pink (en)
- *Switch*, Chip & Dan Heath (en)
- *Reality is broken*, Jane McGonigal (en)
- *Lean Startup*, Eric Ries (en)
- *The fifth discipline*, Peter Senge (en)

Et ma petite contribution : [pdf](#) ou [epub](#) - [La horde agile](#)

Ateliers d'expression du besoin

- Remember the future
- Product Box
- Buy a feature
- User Stories Mapping
- Impact Mapping

Supports sur areyouagile.com

- Le grenier de areyouagile.com
- Formation agile
- Initiation agilité
- Vision & sens
- Expression du besoin
- Zoom Scrum
- Peetic, des exemples
- Autres slides / Speakerdeck
- Scrumshot, animation scrum
- Kanban

Lean Startup

- *Lean Startup*, Eric Ries (en)
- [Slides](#) par Christophe Monnier
- [Pirates Metrics](#) par Jurgen De Smet

Open Agile Adoption

- [Forum ouvert](#)
- [Slides en français sur l'*Open Agile Adoption*](#)
- [Guide du forum ouvert](#) par son inventeur Harrison Owen (en)
- [Open Agile Adoption](#) par Dan Mezick
- [Open Agile Adoption](#) article

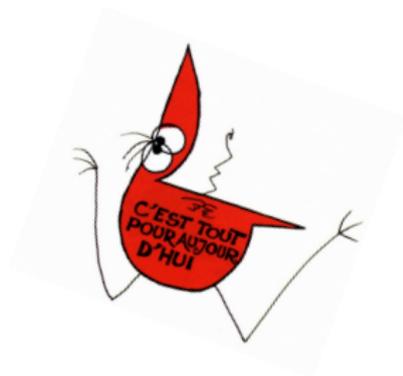
C'est tout pour aujourd'hui !

Pablo Pernet 

Blog : www.areyouagile.com

Twitter : [@pablopernot](https://twitter.com/pablopernot)

Linkedin : [Linkedin](#)



Réalisé avec \LaTeX

Généré la dernière fois le 10 septembre 2015

Ce travail est placé sous licence : <http://creativecommons.org/licenses/by-nc/4.0/>