

Guide de survie à l'agilité

Pablo Pernot ✖
Coaching Agile, Reverse mentoring,
Holocratie, Conduite du changement, Prospective
www.areyouagile.com
beNext

Dernière mise à jour le 1er mars 2017

LES DEVISES SHADOK



**IL VAUT MIEUX ÊTRE AGILE MÊME SI IL NE SE PASSE RIEN
QUE RISQUER QU'IL SE PASSE QUELQUE CHOSE
DE PIRE EN NE L'ÉTANT PAS.**

Table des matières

1	Trois tableaux	3
1.1	Modernité	3
1.2	Agilité	5
1.3	Engagé	7
2	Une transformation agile réussie?	10
3	Rappels sur les mouvements agiles	10
4	Guide Rapide	11
4.1	Il était une fois Scrum	11
4.1.1	La phase d'étude, de préparation, des fois appelée sprint zéro	11
4.1.2	Le début des itérations	18
4.1.3	L'itération continue	23
4.1.4	Fin de l'itération	29
4.2	Scrum, récapitulatif rapide	34
4.2.1	Diagramme	34
4.2.2	Sprint Planning	35
4.2.3	Daily Standup	35
4.2.4	Review	36
4.2.5	Retrospective	36
4.3	Kanban	37
4.4	Lean	37
4.5	Glossaire	38
5	Supports & Matériels	41
5.1	Supports de la formation	41
5.1.1	PDFs de <i>areyouagile.com</i>	41
5.1.2	Ateliers d'expression du besoin	42
5.1.3	Focus Lean Startup	42
5.1.4	Focus sur <i>Open Adoption</i>	43
5.2	Lectures	43
5.2.1	Livres relatifs aux méthodes agiles et outils	43
5.2.2	Livres pour appréhender la culture agile	44
5.3	Compléments venant d'autres blogs	47
5.4	Divers	48
5.4.1	Vidéos en ligne (culture agile)	48
5.4.2	Documents annexes	48

1 Trois tableaux

J'aimerais vous décrire trois tableaux, trois points de départ décrivant le monde dans lequel je pense nous vivons.

Le premier tableau traite des deux approches méthodologiques et organisationnelles qui ont façonnées les derniers siècles : celle de Ford et celle de Toyota. Le but de ce premier tableau est de surtout mettre en évidence que chaque époque a sa réponse propre, et qu'aucune réponse n'est la bonne isolée de son contexte.

Le deuxième tableau traite de l'agilité, ce terme qui devient galvaudé, mais qui représente encore beaucoup pour moi. L'agilité qui en soit est l'aboutissement et la synthèse de notre compréhension de notre monde nouveau, complexe.

Le troisième et dernier tableau traite de l'engagement des personnes, point clefs qui fait la synthèse des deux premiers tableaux.

1.1 Modernité

Au regard du dernier siècle et concernant l'organisation de nos entreprises on pourrait distinguer deux modes de pensée assez différents. À chaque époque (début du XXème siècle, fin du XXème siècle) des réponses adéquates mais très différentes car ne répondant pas aux mêmes problématiques ont été proposées. Ces différentes réponses sont assez bien mises en relief par la comparaison du Fordisme et Toyotisme.

Au début du XXème siècle, quand John Ford annonce « les gens peuvent choisir n'importe quelle couleur pour la Ford T, du moment que c'est noir », il a raison, personne ne songe encore à tous les progrès et les variations qui suivront. Ford est en train de révolutionner nos habitudes avec la voiture qui devient à partir de ce moment le meilleur ami de l'homme. Mais à ce moment personne ne s'interroge sur la couleur ou la forme de sa voiture, la question c'est d'en avoir une ou pas, tout simplement. D'autres traits caractérisent cette époque et cette façon d'aborder le travail : la main-d'œuvre est très bon marché, en surnombre, pas formée, Ford va donc baser sa force sur le quantitatif et non le qualitatif ; c'est l'heure de gloire du taylorisme. Ce taylorisme se justifie très bien compte tenu du contexte qui le voit naître.

Chez Ford on observe aussi que la dernière étape avant la sortie de la fameuse Ford T, c'est l'assurance qualité, le test, la recette en quelque sorte. Et on tri : cette voiture sort, celle-ci non. Rien de bien choquant pour

l'époque, la matière première ne manque pas.

Quand une voiture sort elle s'accompagne d'un guide concernant les 150 anomalies les plus probables ou fréquentes. Là aussi, rien de choquant, en ce début de siècle il suffit de se plonger dans ce manuel, de s'armer d'un bon marteau, et le tour est joué. La réparation de la voiture n'est pas un casse-tête chinois, et demeure accessible au commun des mortels. Il est donc préférable pour Ford et le consommateur de laisser ces anomalies à la charge de ses clients, le coût de revient est meilleur pour tous.

Quarante, cinquante ans après, chez Toyota les choses ont bien changées. C'est un pays détruit par la guerre que l'on découvre. La matière première est devenue un élément critique. On ne peut pas gaspiller ou stocker exagérément. Les machines outils sont aussi une denrée rare, il va falloir les exploiter au maximum de leurs possibilités. En un certains sens, tout comme les hommes, qui, à la demande du gouvernement pour redresser le pays, sont embauchés à vie par Toyota. La technologie a aussi changée, et elle continue de changer. Un marteau ne suffit plus pour réparer sa voiture. Enfin les attentes du public ne sont plus les mêmes. La firme japonaise est donc aux antipodes de la situation de Ford. Autre époque, autres mœurs.

La réponse de Toyota est révolutionnaire elle aussi : pour tirer le meilleur de ses employés, qui vont l'accompagner toutes leurs vies, la firme comprend que le meilleur moyen est de les respecter et les responsabiliser au maximum. C'est d'ailleurs en les responsabilisant qu'elle tire le meilleur parti de ses machines-outils. L'homme de terrain démontrant une capacité indéniable à adapter ces machines et à avoir des idées pour les exploiter au mieux. La réflexion pour l'amélioration continue devient une pratique courante chez eux. Cette pratique ne se limite pas en interne, les fournisseurs sont encouragés à trouver des moyens d'améliorer leurs produits ou leurs capacité de production : ils ne repercuteront cette amélioration que deux ans après celle-ci¹.

Pour palier au problème du stock, Toyota découvre que le meilleur coût provient de deux choses : a) la qualité intrinsèque : dès qu'une anomalie est détectée on stoppe la chaîne, l'impact et la correction de cette anomalie seront moindres, et le stock touché minime ; b) une gestion par flux tendu régulée par une participation active des employés, c'est eux qui tirent le flux (ils vont chercher et déclencher leurs actions qui s'adaptent ainsi parfaitement à leurs rythmes et à leur compétences, d'où aussi une forte responsabilisation), et ce n'est pas un flux poussé (là les employés

1. « The Machine That Changed the World », James P. Womack, Daniel T. Jones, Daniel Roos

subissent le rythme).

Il ne s'agit pas de réponses idéologiques en corrélation avec chaque époque. Nous parlons d'industrie, de compagnies, dont la raison d'être est de générer des bénéfices. Il s'agissait des meilleures réponses en terme de performance d'entreprise.

Aujourd'hui beaucoup de nos entreprises évoluent encore sous le modèle du Fordisme, mais c'est bien, même si il devient lui aussi peu à peu obsolète, le modèle de Toyota qui devrait nous inspirer : qualité intrinsèque (pas parce que cela fait, mais parce que cela coûte moins cher et génère plus de revenu), responsabilisation (pas parce que cela fait bien, mais parce que cela génère plus de valeur, de revenus).

J'insiste volontairement sur l'aspect purement économique. Ce n'est pas nécessairement ma tasse de thé mais j'évolue moi-même en environnement concurrentiel et je sais que c'est là le nerf de la guerre pour les entreprises qui font appel à moi. La bonne nouvelle c'est que pour faire réussir son entreprise aujourd'hui il faut engager et responsabiliser les personnes comme a pu le faire Toyota.

1.2 Agilité

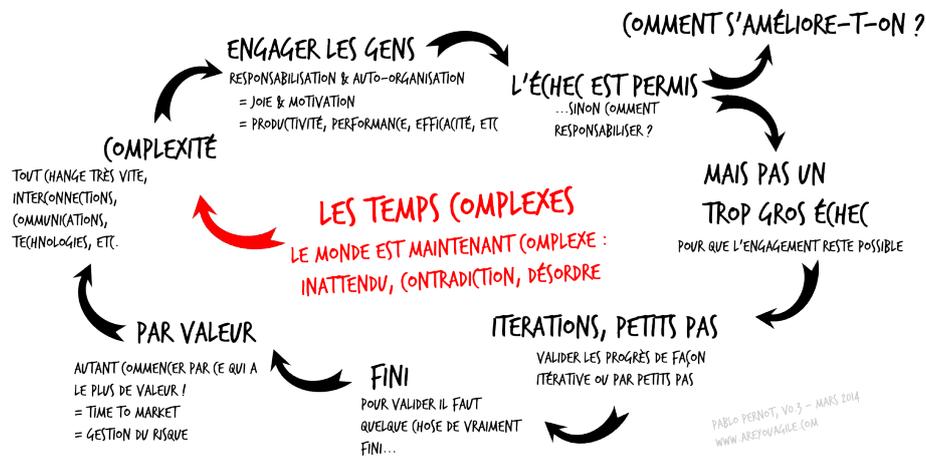
Cette réponse aux temps modernes de Toyota est ce que l'on appelle le « Lean ». L'héritier du « Lean » est le mouvement « Agile » : l'intensification de la communication, le renouvellement de la concurrence, des marchés et des technologies nous ont fait basculer dans les temps complexes. Pour comprendre l'Agile au regard du Lean aujourd'hui il faudra retenir l'adaptation et non pas la standardisation industrielle.

Nous sommes entré dans les temps complexes : tout va très vite, trop vite, tout est entrelacé, tout change, tout le temps, etc. La meilleure façon de répondre à cette complexité c'est de responsabiliser, d'engager les gens. Comme un général qui ne peut plus tout diriger comme il y a 200 ans, et qui doit se reposer sur l'auto-organisation de ses troupes, celles-ci restent fidèles à sa stratégie, à sa vision mais mettent en œuvre leurs tactiques.

Cette responsabilisation (et ainsi la libération de cette capacité de performance) n'est réelle que si on laisse le droit à l'erreur, sinon c'est juste un faux-semblant et les effet escomptés ne sont pas au rendez-vous.

Mais la responsabilisation est aussi possible si celle-ci ne met pas l'organisation en danger (ou le projet, etc.), sinon qui osera quoi que ce soit?

Figure 1 – Les temps complexes



Il faut donc soutenir la responsabilisation mais uniquement si l'erreur est possible, mais elle ne doit pas être « mortelle ». Pour cela il faut avancer par petits pas, de façon itérative. Et juger étapes par étapes de nos progrès. Une petite fièvre ça soigne, une grosse, ça tue !

Mais pour vraiment juger il faut des choses finies. Peu importe l'itératif si les choses ne sont pas finies (c'est alors de l'incrémental...). Interrogez vous très fort sur cette notion de fini (le mieux est de mettre sur le marché ! pensez au produit viable minimum). On aura un vrai *feedback*, de vraies réponses, qu'avec des choses finies.

Si vous réussissez à délivrer des choses finies, étape par étape, en étoffant, alors autant prioriser par valeur. Car a) on a jamais le temps de tout faire, donc autant faire uniquement les choses de valeur, b) cela permet de se donner les plus grandes garanties et les plus grands enseignements, et enfin c) d'avoir un sacré « time to market » (c'est à dire pouvoir mettre sur le marché de façon très réactive).

Ce « Time to Market » très utile en ces temps de grande complexité, car tout change si vite.

Naturellement ce cycle n'est apprenant (c'est nécessaire dans cette complexité ambiante) que si l'on s'interroge constamment sur comment s'améliorer.

Voilà en quelques mots ce qui porte à mes yeux la notion d'agile aujourd'hui, une réponse adaptée aux temps complexes : responsabilisation, feed-

back, apprentissage, time to market. La complexité nécessite de l'émergence (feedback, apprentissage et ainsi time to market), qui nécessite de l'interaction (responsabilisation, amélioration continue, intelligence collective).

1.3 Engagé

Toyota, le Lean, puis l'Agile : on évoque constamment responsabilisation et l'engagement. Mais il est vain de dire « soit motivé ! », on connaît l'absurdité d'une telle démarche.

Les récentes études de Gallup indiquent que seulement 13% des employés dans le monde ne se sentent pas impliqués, ou n'ont pas envie de l'être². Voilà qui ouvre le débat sur la performance de nos organisations.

Dans le *Mythical Man Month*³ l'auteur exprime l'idée qu'une personne (en l'occurrence dans son cadre, un développeur informatique) a un ratio de productivité de un à dix selon le contexte. Sa capacité fluctuerait ainsi de un à dix selon le contexte. Mes observations -sur moi même et les autres- tendent à confirmer cette information. Et je pense que la grande part de cet écart vient de l'implication de la personne. Observez vous vous même à réaliser une tâche sur laquelle vous ne vous sentez pas impliqué, et une sur laquelle vous vous sentez impliqué, que vous vous êtes approprié.

On ne peut pas décréter la motivation, ni l'implication, ni l'appropriation. Observons donc les domaines où elles sont florissantes, un de ceux-ci est bien connu et a donné lieu à des études sur cette implication : les jeux vidéo, et plutôt les joueurs de vidéos.

On observe que l'implication dans un jeu vidéo est souvent bien supérieure à celle dans la vie d'entreprise. Tous ces joueurs de jeux en ligne qui payent pour échouer, recommencer, apprendre, refaire, fournir une documentation de qualité, ne pourrait-on pas avoir le même engagement en entreprise? L'industrie du jeu vidéo est florissante, comprenez qu'elle implique d'énormes sommes d'argent. Avant donc pour nous même d'en tirer des enseignements pour l'organisation, elle s'est interrogée sur elle-même car son succès est liée à l'implication qu'elle va engendrer. Les observations qui ont découlé de ces recherches sont une mine pour nous.

Mais alors que j'évoque ce sujet, je reçois le bulletin scolaire de mon fils aîné, Mathieu. Mathieu a de bons résultats scolaires mais le carnet indique

2. <https://hbr.org/2014/11/a-primer-on-measuring-employee-engagement>

3. Fred Brooks, <https://archive.org/details/mythicalmanmonth00fred>

que si Mathieu était impliqué il aurait de très bons résultats scolaires.

Je cherche Mathieu :

- Mathieu? Mathieu?
- (de l'autre pièce) Oui papa?
- Peux-tu venir que l'on parle de ton bulletin scolaire.
- Attend, là je ne peux pas, je suis en train de construire la quatrième tour de mon château fort⁴ et elle n'a pas exactement les mêmes décorations que les autres, ni la même constitution en fer, je ne peux pas la laisser ainsi.
- Oui c'est exactement de cela que je souhaite te parler...

En d'autres termes Mathieu est très impliqué dans son jeu, et ses résultats sont très bons. Qu'est ce qui manque à l'école vous récolter ce même engagement? Qu'est-ce qui manque à l'entreprise pour avoir ce même type d'implication? C'est le travail de synthèse opéré par Dan Mezick⁵.

Un dernier détour par l'école de mon fils aîné. Réunion parent/professeur, je suis en tête à tête avec la professeur principale, elle se plaint de la qualité générale de la classe, de son manque de sérieux et du niveau de son travail. Puis nous passons à la récente sortie pédagogique sur les fouilles archéologiques (plusieurs jours sur site). Elle se délecte de la réussite de cet événement, et dans les minutes qui suivent m'indique que la classe a fourni un excellent travail et dossier. Surpris je l'interroge donc sur l'écart entre le travail quotidien et ce travail là. Elle reste plusieurs secondes silencieuse comme si un piège s'était refermé sur elle puis me glisse, un peu désarçonnée : mais on ne peut pas faire des choses nouvelles intéressantes comme cela tout le temps. Je suis resté silencieux en pensant que c'était bien de l'éducation et de la vie de nos enfants dont on parlait alors.

Opérons là aussi un rapprochement évident avec nos entreprises : si vous faites des choses qui intéressent les personnes avec lesquelles vous travaillez, si vous réussissez à les impliquer, ce qui en résultera n'aura rien à voir.

Ainsi que nous enseigne ces études sur l'implication dans les jeux vidéos? Des choses simples, évidentes, mais trop souvent oubliées, ou délaissées. Pour qu'un jeu (ou une entreprise) génère de l'engagement il doit proposer :

1. Un **objectif clair** : a-t-on des objectifs clairs dans nos organisations?

4. Minecraft

5. The culture game, Dan Mezick, Reality is broken, Jane McGonigal

2. Des **règles claires** : a-t-on des règles claires dans nos organisations ?
Un cadre bien défini ?
3. Du **feedback** : a-t-on régulièrement du *feedback* ? Sait-on où nous en sommes ? Quel est le résultat de nos dernières actions ?
4. Le jeu est **invitation**, nul n'est obligé d'y jouer : cette notion d'invitation existe-elle dans nos organisations ?

En proposant ce cadre le jeu (ou l'organisation) va générer plusieurs sentiments qui sont la base de cette implication et de cet épanouissement.

1. Un **sentiment de contrôle**, dans le sens maîtrise, je peux agir et je sais comment agir.
2. Un **sentiment de progrès**, j'avance et j'observe que j'avance.
3. Un **sentiment d'appartenance** à une communauté.
4. Le **sentiment de travailler pour quelque chose qui me dépasse**, quelque chose de plus grand que soit, on se projette dans une réalisation qui nous dépasse.

Voilà des points simples mais cruciaux sur l'implication.

2 Une transformation agile réussie ?

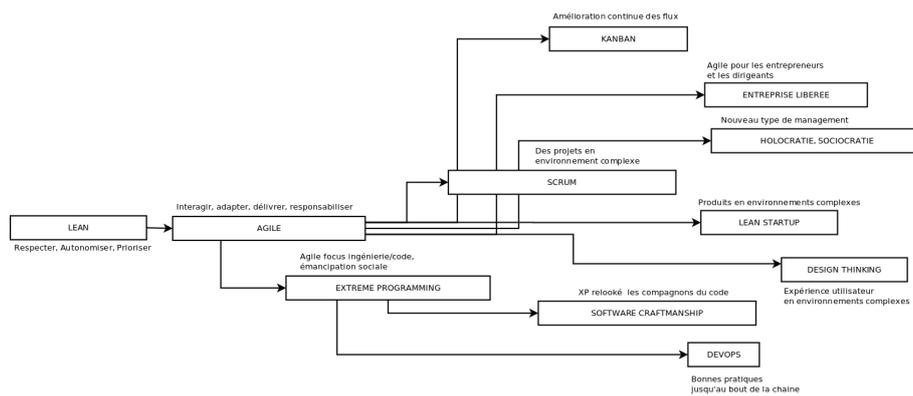
Articles :

- [Qu'est ce qu'une transformation agile réussie ?](#)
- [Paradoxes des transformations agiles](#)
- [Chemin d'une transformation agile](#)
- [Trois apprentissages clefs](#)

3 Rappels sur les mouvements agiles

Mouvement Agile

Figure 2 – Mouvements agiles



4 Guide Rapide

4.1 Il était une fois Scrum

Ce mois d'avril, grands chantiers, on démarre en Scrum chez Peetic. On a défini nos objectifs, ils sont simples et clairs. On sait que beaucoup d'argent gravite autour de ces petites bestioles (chiens, chats), et on se dit que faire un moteur de rencontre pour animaux peut nous permettre de vendre pas mal de services et de biens autour. Les six premiers mois on veut 2000 inscrits. Les six mois suivant 30000€ de vente. Avoir des objectifs clairs c'est essentiel si on souhaite que tout le monde les comprenne. Il s'agit des objectifs avec le plus de valeur : les plus significatifs. Mais il faut donner du relief, six mois c'est déjà long pour nous, on va se lancer dans des *releases*, des mises sur le marché, de trois mois. Car toutes nos idées demeurent des hypothèses avant de vraiment être confrontées à la réalité. « Aucun plan ne survit au premier contact avec l'ennemi » disait Von Moltke.

4.1.1 La phase d'étude, de préparation, des fois appelée sprint zéro

Comme toujours on veut tout, nos sponsors veulent tout, tout de suite. Mais pour réussir un produit et apprendre rapidement, sortir des hypothèses pour de vrais apprentissages terrains, il faut donner du relief à nos réalisations. Souvent on fait cela lors d'une phase d'étude, ou sprint zéro.

Fin avril, on prend 5 jours ouvrés sur une période de un mois pour aligner, harmoniser les points de vue et les attentes des différents acteurs. Normalement le critère clefs de succès est déjà défini comme annoncé plus haut. Cette période c'est le sprint zéro ou la phase d'étude, ou de préparation, ou l'avant projet, ou... 1 mois calendaire, 5 jours de travail, pour 6 mois de Scrum, humm, ça paraît correct.

Pour harmoniser les points de vue il faut des représentants de différents types d'acteurs, ainsi on se retrouve avec coach, product owner, équipe, designer/UX, experts métiers, sponsors, utilisateurs, bref tous les gens qui seront influencés et impactés ou qui tiennent à influencer le projet. Sur les 5 jours à passer on garde tout le monde les trois premiers. Il s'agit de faire de gros *brainstorming* pour que chacun amène sa vision : technique, métiers, ergonomie, etc. Que chacun amène ses contraintes et ses attentes, ses idées, ses craintes, son savoir. Et surtout que chacun entende l'autre et comprenne plus tard tel ou tel point de vue. Ce qui ne veut pas dire que tout le monde décide de tout.

Oui hein, je voulais dire, dans Scrum on parle de « dev team » dans l'équipe

scrum, et beaucoup pense qu'il ne s'agit que des développeurs au sens co-
deurs, pas du tout, c'est l'équipe qui développe au sens large, autant de la
documentation, du design, etc, que du code.

En savoir plus sur la constitution de l'équipe et la co-localisation :

- **Penser son organisation sur le terrain**
- **La horde agile**

Un gros *brainstorming* divergent ce premier jour. Que tout soit dit, affi-
ché. Le coach va demander durant le 2ème jour de commencer à conver-
ger, et ultimement, c'est le *product owner* ou le *product manager* (en tous
cas celui qui porte le produit) en accord avec la vision qui priorisera (si be-
soin le coach va utiliser des outils pour que les débats ne s'enveniment
pas). L'important est de mettre du relief : on fera tout le périmètre fonc-
tionnel? peut-être. On essayera de faire le mieux possible avec la capacité
octroyée. Grâce à cette mise en relief, à cette priorisation, on va mettre
en évidence des chemins clefs, des points stratégiques clefs. Toute cette
priorisation, cette mise en relief est de la responsabilité du product ow-
ner, qui peut naturellement écouter chacun des acteurs, mais les coaches
agiles se tourneront vers lui pour trancher. Il n'est pas seul, il peut bien
s'entourer. Mais on a besoin de choisir à la fin.

Inutile de parler de tout ce que va impliquer le produit (cela ne serait que
des hypothèses, et beaucoup se révéleraient fausses), il suffit de mettre
en évidence les points clefs et ensuite de les détailler pour travailler lors
de ces trois premiers mois. Idéalement ensuite on sortira en production
pour valider nos hypothèses : ce que nous développons fait bien croître le
taux d'inscription (et pérennise ces inscrits).

Ce n'est pas le cas? Et bien on aura *seulement* consommé trois mois pour
se rendre compte que nos hypothèses ne sont pas bonnes. Ouf, au lieu de
le réaliser à la fin du programme. On a réussi à avoir déjà pas mal d'ins-
criptions? Super. On a déjà commencer à atteindre une partie clef des at-
tentes du produit, et toute la suite du développement se basera sur moins
d'hypothèses et sur plus sur une vraie connaissance (autant technique,
que des retours d'informations sur les envies, les besoins et les compor-
tements utilisateurs).

Les deux jours suivants se font en comité plus réduit, disons l'équipe coeur :
on décline les chemins stratégiques mis en évidence précédemment avec
un plan de bataille terrain : on va développer cela, puis cela, etc. On pro-
pose une première projection sur le contenu à venir des trois mois mais
cela reste une estimation assez floue (comme toutes ces estimations réali-

sées au tout début d'une réalisation). De toutes les manières grâce à notre priorisation par valeur on aura le mieux possible pour le temps imparti. Si on estime que ce n'est pas assez on va le voir venir et on prendra une décision (changer le périmètre? Changer la date de sortie?).

- Plus sur les estimations : **Elles ne sont pas uniquement liées à la complexité.**
- Plus sur les estimations : **La malédiction du jour homme.**

A la fin de ces cinq jours ouvrés on a a) une fiche de cadrage global, avec 1 ou 2 premiers critères de succès clefs mesurables, b) des *personas* qui nous disent quelles sont les attentes, quels sont les comportements, les caractéristiques des profils cibles, c) un *impact mapping* qui met en évidence des chemins stratégiques clefs de réalisation, d) un *user story mapping* qui décline de façon opérationnelle (comment on réalise?) les chemins stratégiques relevés précédemment. On pensait mettre en relation des possesseurs d'animaux façon club de rencontres, patatras, de nos conversations a émergée l'idée plutôt de de gardiennages en commun, et de balades en commun. On spécifie aussi e) une définition de fini qui détermine nos critères de qualité : il faut la garder compact qu'elle ne sorte de la tête de personne. Et enfin f) Une première estimation globale qui vaut ce qu'elle vaut comme toutes ces premières estimations (réalisée par l'équipe), et du coup g) un premier *plan de release* : quel contenu devrait-on avoir et quand? Comme agile est assez binaire dans sa gestion d'une réalisation : c'est ou ce n'est pas fait, c'est fini ou ce n'est pas fini (et surtout pas : « on a 80% de la fonctionnalité F » ce que on le sait ne veut rien dire), c'est très facile de lire les *plans de release* et de se projeter sur des ensembles qui font sens sur le plan « business ». On connaît l'équipe puisqu'elle a participé à cette phase d'étude que l'on appelle parfois par abus de langage : sprint zéro. Et c'est elle qui estime (pas les managers, ni uniquement les techos, l'équipe et les profils hétérogènes qui la composent). On a aussi pris soin de commencer à préciser les éléments du haut de notre liste de fonctionnalités à développer (ce que l'on appelle notre *backlog*). Comme toujours c'est le *product owner* qui tranche et arbitre les débats, il peut s'appuyer sur toutes les compétences qu'il veut mais il porte le **quoi** et le **pourquoi**. L'équipe porte le **comment**. Le scrummaster et/ou les coachs agiles facilitent, ils sont donc neutres, et s'occupent du cadre pour permettre à chacun de s'y émanciper, de prendre l'espace.

- Plus sur la définition de fini : **Nul n'est censé ignorer la définition de fini.**
- Plus sur les **apprentissages des rôles de scrummaster, product owner, et équipier.**

Figure 3 – Un persona



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)

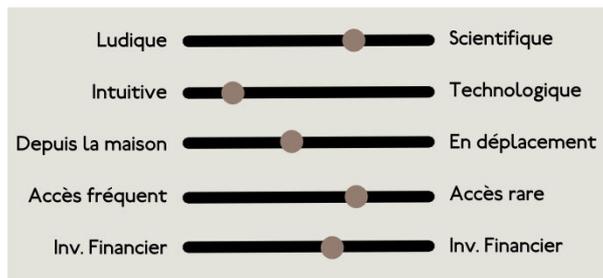


Figure 4 – Un impact mapping

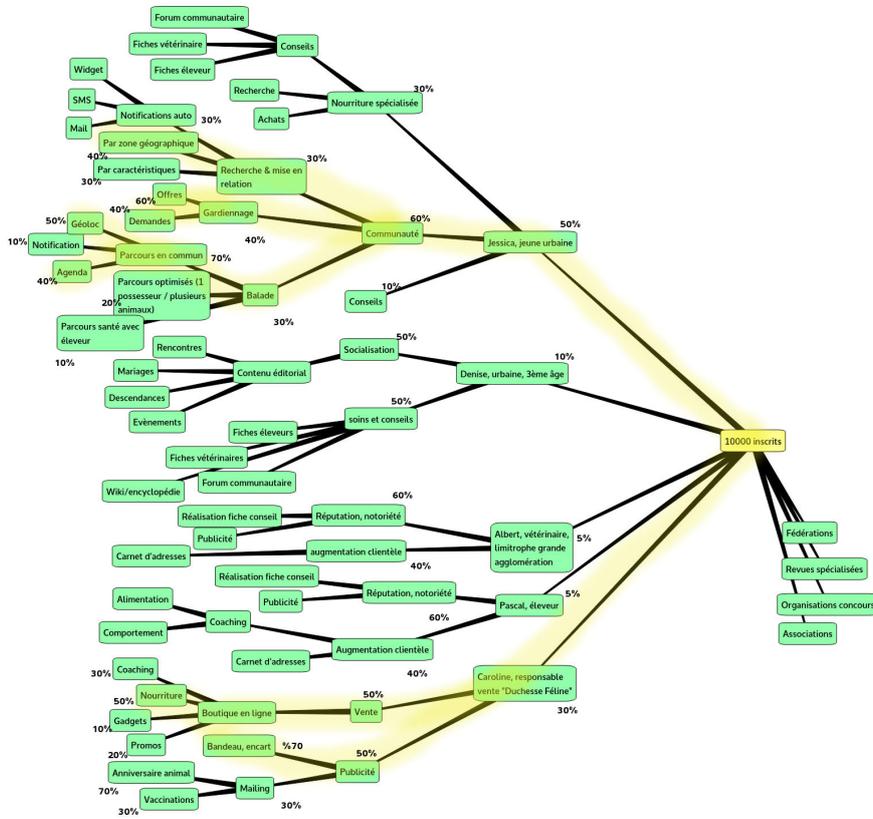


Figure 5 – Un autre impact mapping



Figure 6 – Ma pyramide à moi de l'expression du besoin agile

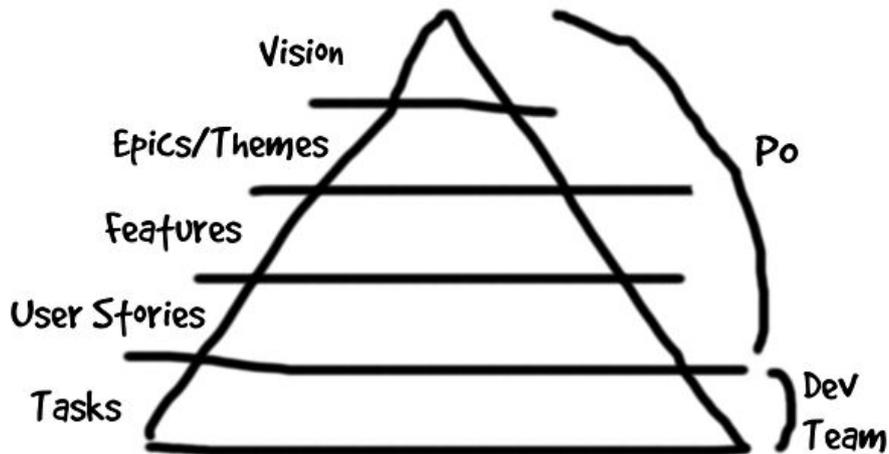
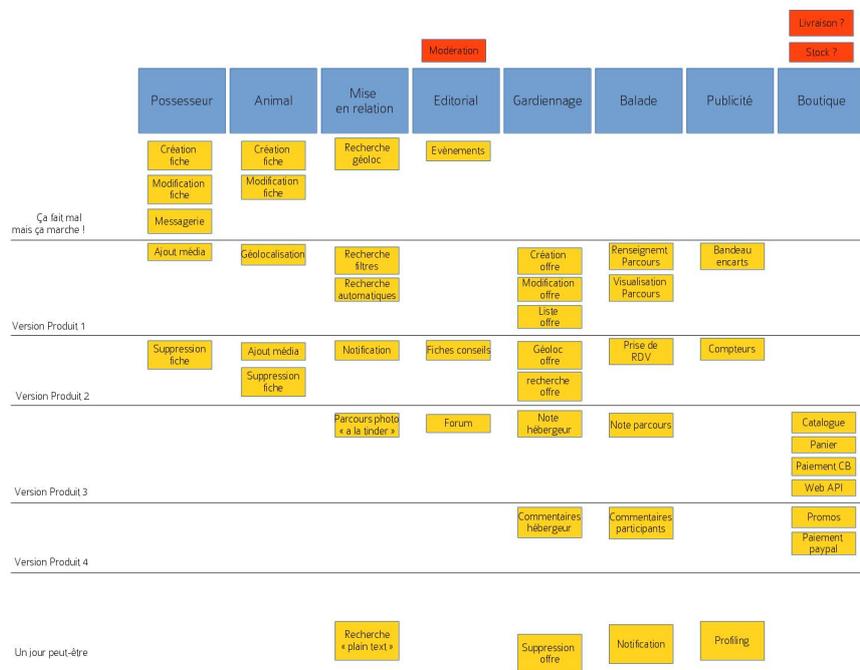


Figure 7 – User Story Mapping



4.1.2 Le début des itérations

On est prêt, on démarre la première itération (de deux semaines, elles feront toutes deux semaines, c'est important cette régularité pour nos calibrages. Estimation, engagement, et planification : répétition d'une cadence observée). Un calibrage, un réglage, cela prend trois à quatre itérations (quelques soit la durée de l'estimation... c'est vivre un cycle qui est important), donc autant les itérations sont courtes mieux c'est. On se calibre plus vite, on apprend plus vite.

Lundi la matinée est consacrée au **sprint planning**. Le *Scrummaster* s'est assuré que les invitations sont bien parties, que la salle est réservée, que tout le monde est au courant du contenu de la réunion, et que le matériel nécessaire est bien là (post-it, feuille, projecteur, etc). On va avoir une conversation sur les fonctionnalités du backlog. Une par une : dans l'ordre du backlog, celui de la valeur pour le *product owner*. Jusqu'à que le temps soit écoulé, ou presque. L'objectif est de bien détailler les attendus, de lever les inconnues, les quiproquos. On utilise les conversations autour de l'estimation pour les rendre évident, pour clarifier les points de vue de chacun. Le PO n'estime pas. Ni le *scrummaster* qui est neutre (il ne fait rien pour avoir le pouvoir de libérer la productivité de l'équipe). Les débats sont passionnants pour tous. Le métiers, le PO (*product owner*), découvre que les *techos* ne cherchent pas à les entuber, et comprend pourquoi parfois quelque chose de simple ne l'est finalement pas, et pourquoi parfois quelque chose de compliqué se révèle simple des fois. Les *techos* comprennent que le métiers, le *product owner*, n'a pas de problème de santé mentale et ne change pas d'avis pour les embêter, mais pour de bonnes raisons. Quinze minutes avant la fin de la réunion le *scrummaster* alerte tout le monde : il va falloir se projeter sur ce qui est réalisable dans les deux semaines à venir. Le *product owner* choisit l'ordre des *user stories*, des fonctionnalités, l'équipe annonce jusqu'où elle pense pouvoir aller. Ce qu'elle pense pouvoir finir. Car les agilistes sont des maniaques : ou c'est fini, ou cela ne l'est pas (autant ils sont perçus comme des **cowboys hippies**, autant ils sont binaires sur ce point fini/pas fini). Naturellement finir c'est respecter la définition de fini (les aspects qualité, techniques, etc), et les aspects fonctionnels métiers sur lesquels la conversation avec le *product owner* vient d'avoir lieu et dont l'arbitrage final reste au PO. Donc c'est le moment où l'on demande à l'équipe : jusqu'où penser aller? Disons que l'équipe nous dit les 5 premières. C'est le *sprint backlog*, le périmètre imaginé de l'itération.

Une règle importante : on veut éviter les effets tunnels. Ainsi on ne veut des *user stories* qui puissent se finir dans le temps imparti, les deux semaines, et on en veut au moins trois. Cela induit une règle : ne peut être

pris dans l'itération que des user stories dont l'effort est inférieur à un tiers de la durée de l'itération. Lors du sprint 0 la première estimation aura permis d'alerter le PO sur des user stories importantes dont l'estimation dépasse allègrement cet effort maximum (je parlerai des estimations un autre jour). Pour le PO c'est un message important : il doit découper la user stories en plusieurs user stories plus petites (et croyez moi il n'aime pas ça) si il veut que le scrummaster autorise l'équipe à pouvoir les réaliser.

Voilà on a le contenu de l'itération : les cinq premières user stories du backlog. Elle quitte le backlog (qui appartient au PO) pour tomber dans l'es-carcelle de l'équipe. L'équipe passe l'après-midi (deux nouvelles heures) à découper en tâches les user stories : quelles seront les tâches nécessaires l'aboutissement de ces user stories selon les attentes de la définition de fini et les attentes métiers discutées précédemment avec le product owner : du code, de la documentation, des tests, du design, des jeux de données, etc etc etc. Autant le PO n'aime pas découper ses fonctionnalités (il veut tout naturellement! mais on aura tout on veut découper pour avancer sereinement, ressentir nos progrès, aborder mieux les questions, avoir du feedback, etc.), autant l'équipe n'aime pas découper en tâches non plus! Réfléchir avant de faire c'est une habitude à prendre car cette découpe c'est du *design*, de la conception finalement. Et cela va nous permettre de savoir où nous en sommes (« reste à faire ») durant les deux semaines au travers des *burndown* (de tâches) et *burnup* (de stories).

Voilà c'est parti, l'itération débute. Qu'est ce qui va se passer durant ces 9 journées restantes? Les membres de l'équipe vont essayer de produire, réaliser ce qui aura été défini durant le sprint planning. Tous les jours ils vont montrer l'avancé de leurs travaux avec un mur visuel de post-its. Ce mur sert à les synchroniser entre eux et à communiquer au monde extérieur. Ils vont aussi pour suivre leurs travaux réaliser un *burndown* : une courbe descendante du travail restant à faire (pour le burndown je compte les tâches et je décréméte quand elles arrivent à « fini »). Et un *burnup*, une courbe ascendante, des validations réalisées par le Product Owner. Car en effet on espère, on souhaite, que le product owner valide les user stories, les éléments définis lors du sprint planning, au fil de l'eau, pour limiter les risques et la pression. En validant au fil de l'eau pas de surprise, pas de pression. On obtient régulièrement du feedback, et on lève régulièrement des hypothèses. La qualité vient de la régularité.

Des compléments ici ou ailleurs :

- [La matinée typique d'un scrummaster](#), par Claude
- [L'après-midi révélatrice d'un scrummaster](#), par Claude
- [Le jour où j'ai découvert que j'étais un mauvais scrummaster](#), men-

songe éhontée de Géraldine.

- Du **scrummaster au coach agile**, un cheminement classique.
- **Comprendre le MVP** par Frank qui a traduit un texte de Henrik Kniberg.
- **Introduction à la stratégie du product owner**, toujours d'actualité avec Alexis, de Praxeo, ma période Jerry Garcia.
- La suite avec **Le product owner contre-attaque**,
- Et la dernière partie sur les **stratégies du product owner** (toujours dans ma période Jerry Garcia)
- Vous pouvez toujours jeter un oeil au **Scrumshot** (c'est du svg, c'est dynamique, on peut cliquer :)
- En savoir plus sur les **cowboys hippies**.

Figure 8 – Un plan de release

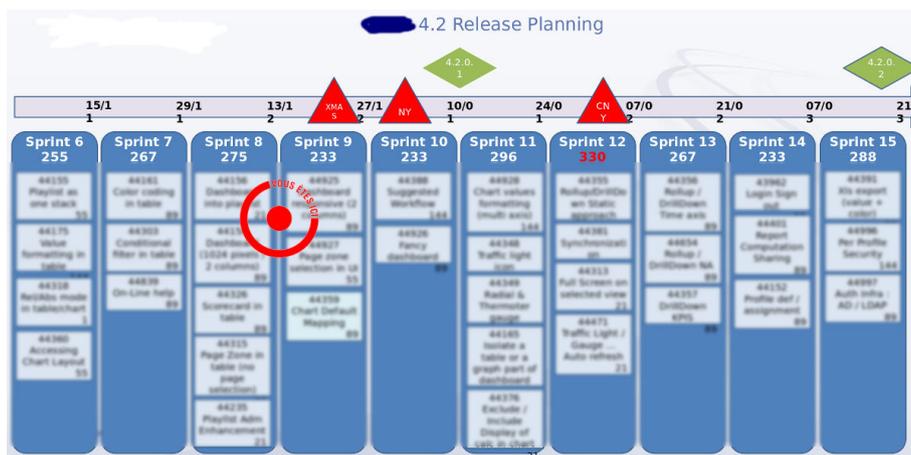


Figure 9 – Un mur scrum

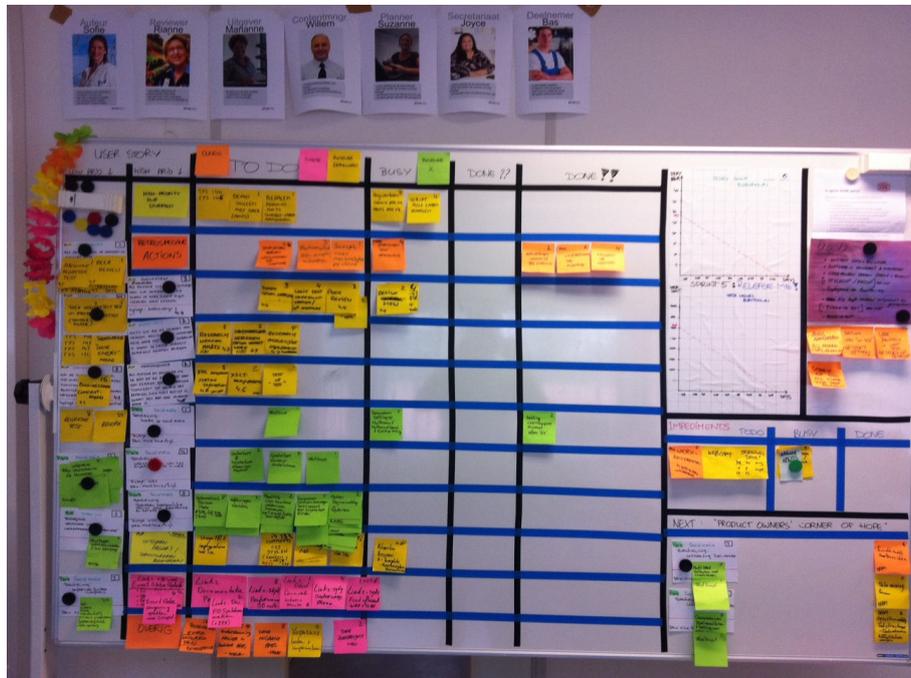
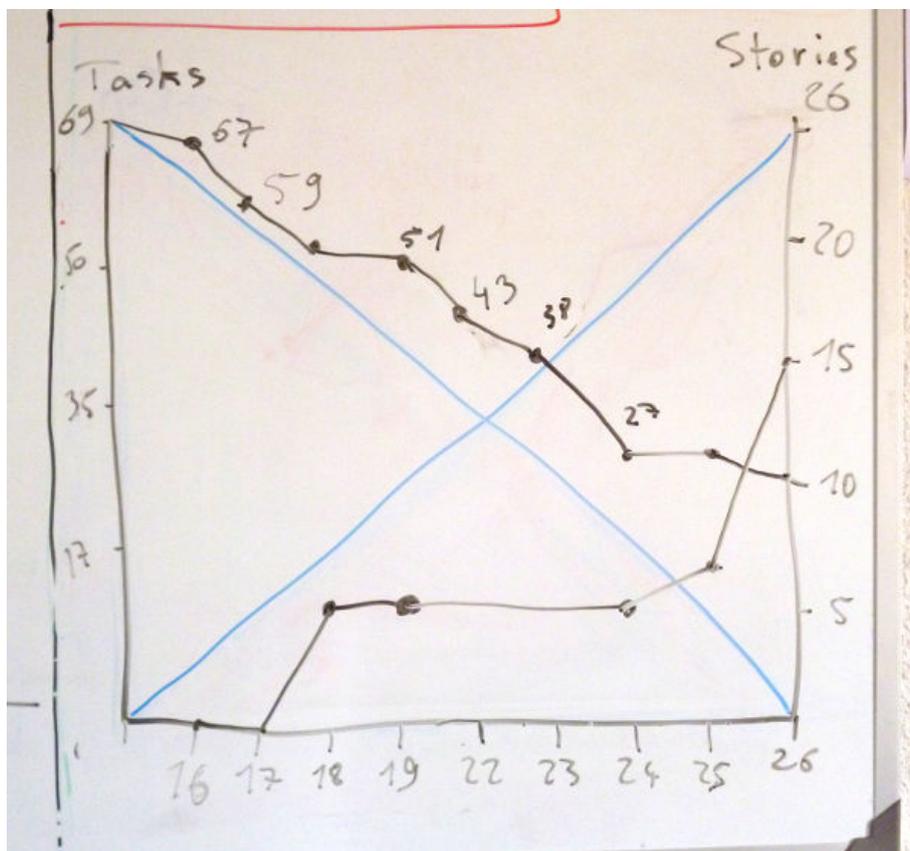


Figure 10 – Burndown et burnup



4.1.3 L'itération continue

Nous avons démarré l'itération lors de notre précédente histoire. Bref nous sommes dans la gadoue, au milieu du maul (car c'est bien d'un **maul** dont voulaient parler les japonais qui observaient un match de rugby et faisaient une métaphore avec leur dynamique d'équipe, et pas d'une mêlée (scrum)).

Durant cette itération le product owner valide au fil de l'eau quand c'est fini. Fini cela veut dire quoi? **Fini** c'est défini par la **définition de fini** (elle porte bien son nom).

En tous cas le product owner valide qu'il a bien obtenu ce qu'il avait demandé, il ne test pas si cela marche, il valide que c'est bien ce qu'il avait demandé. Rien ne l'empêche de faire tous les tests qu'il veut mais l'équipe avec ses différentes compétences a tout testé avant de proposer ces éléments au product owner. Tester cela veut dire que tout marche comme prévu, et que cela respecte la *définition de fini*. Tout roule, cela pourrait être mis en production aux yeux de l'équipe (ce n'est pas une obligation de mettre en production, c'est potentiellement une opportunité). Mais comme c'est le product owner qui porte le quoi et le pourquoi et qui est à l'origine de la demande c'est lui qui porte la validation finale : *oui c'est bien ce que j'ai demandé*.

Durant ces jours de travaux on va se rendre compte que l'on a oublié des tâches : on va les ajouter au mur de post-its (ou en retirer des obsolètes). Les courbes (*burndown...*) vont donc refléter ces changements. Pour les burndowns je demande à ce que l'on découpe en tâches, et idéalement ces tâches sont petites. L'intérêt n'est pas vraiment dans l'estimation, mais dans la clarification et dans la conceptualisation du travail à réaliser. Ensuite avec les *burndowns* on sait où sur ce chemin conceptualisé on se trouve. Il est primordial de toujours montrer l'état des lieux avec transparence, sans pipoter (de toutes les manières, cela se verrait). On n'incriminera pas l'équipe : que les choses changent c'est normal. Qu'il y ait des bonnes ou de mauvaises surprises c'est normal, mais ce que l'on veut c'est savoir : pour aider, pour arbitrer.

D'ailleurs pour les plus butés qui continuent à consolider les chiffres, si vous consolidez les tâches des *user stories* vous observerez que bizarrement les éléments jugés les plus conséquents sont souvent ceux avec le moins de tâches car c'est normal on les maîtrise mal! Et à l'inverse les éléments jugés les plus simples sont ceux souvent avec le plus de tâches et c'est aussi normal car on sait les découper sans souci! La découpe en tâches que je recommande à pour objectif non pas de de mesurer mais de

penser le travail à faire.

Au fait, pas de *business value* sur les *user stories*, c'est intenable. Vous devez scinder une user story vous divisez par deux votre *business value* ou pas? C'est le genre de questions qui font que je n'ai jamais réussi à maintenir une *business value* unitaire par *user story*. Un vrai casse tête chinois, et ce n'est pas franchement utile. Donc pas de *business value* sur les *user stories*. Vous pouvez représenter la *business value* a un niveau plus macro : epic/épopée, fonctionnalités/feature. La valeur aux yeux de product owner, cela sera sa priorisation.

Tous les jours un point au sein de l'équipe un point a lieu (le scrummaster comme le product owner n'ont pas besoin d'y participer, mais le scrummaster en tant que garant de la méthode doit s'assurer de son bon déroulement, et même potentiellement entendre les obstacles qui gênent l'équipe, bref, c'est paradoxal il est là sans être là). Donc tous les jours au sein de l'équipe ce point a lieu : au même endroit, à la même heure, maximum 15mn (debout pour s'en rappeler), pour partager l'état des lieux entre membres de l'équipe. C'est le **daily standup**.

Une pratique que j'aime bien c'est de demander à l'équipe un point rouge ou vert représentatif de leur confiance sur chaque élément tous les jours. Il est vert (on est confiant) ou rouge (on est pas confiant). Cela donne une tendance que tout le monde apprécie. Orange interdit. Normalement rien de rouge le premier jour (ou alors l'équipe n'a pas vraiment choisi le scope de l'itération...).

Après l'équipe est autonome et auto-organisée ; d'ailleurs je demande à ne pas écrire de noms sur les tâches. Les tâches appartiennent à l'équipe, pas a des personnes. Je ne veux pas entendre un Laurent (exemple) me dire « Moi j'ai fini, mais Julien est un boulet et il rame ». Primo, lors du sprint planning il s'agissait d'une projection d'équipe, donc je ne comprends pas cette distinction entre Laurent et Julien, et donc ils doivent se débrouiller. Deuxio, il faut dans un premier temps que l'équipe apprennent à se gérer elle même donc généralement je réponds : que pense Julien de cela quand tu lui as dit? Ne vous inquiétez pas l'auto-organisation de l'équipe est quelque chose qui marche très très souvent très bien. Comme tous les systèmes vivants l'être humain sait très bien s'auto-organiser. Naturellement on observe les étapes classiques d'une équipe quand elle se constitue (« forming, storming, norming, performing », voir **Tuckman**), et il faut 3 (voire 4 itérations) de calibrage, mais dans la grande majorité des cas cela roule. Jusqu'à ce que vous décidiez de changer l'équipe, et le calibrage doit reprendre. Si le problème persiste au delà de trois ou quatre itérations, consultez.

Ce calibrage de 3,4 itérations est aussi très perceptible concernant la relation et la communication entre le product owner et l'équipe, concernant la qualité des engagements des estimations des projections, etc. Et donc là aussi si vous changez l'équipe, il faudra se calibrer à nouveau durant 3,4 itérations pour la planification (par exemple).

Donc généralement le product owner passe la moitié du temps qu'il octroie au projet/produit à accompagner l'équipe dans sa réalisation : il lui donne du feedback, des informations, valide. Et l'autre moitié de son temps à anticiper la suite : il va rencontrer les autres product owner, product managers, les sponsors, les utilisateurs finaux, les gens qui influencent ou qui seront influencés par le produit/projet. Avec toutes ces informations il précise son backlog, son expression du besoin, le change, le réoriente.

Les codeurs codent, testent, documentent, discutent, épaulent le product owner pour ses user stories, osent des choses hors de leurs périmètres habituels. Les designers *designent*, codent, testent, documentent, discutent, épaulent le product owner pour ses user stories, etc. Les business analyst épaulent particulièrement le product owner, testent, documentent, discutent, etc. etc etc. Mais tout ça c'est l'équipe, ils sont auto-organisés. Ils veulent s'amuser à inverser les rôles, c'est leur tambouille, c'est comme ils veulent.

Si le besoin nécessite une attention particulière : un sujet compliqué mal maîtrisé, des adhérences avec d'autres entités par forcément très limpides, etc. On va faire des réunions d'affinage, du **backlog grooming**. Elles permettent d'anticiper les soucis, de préparer l'expression du besoin pour ne pas avoir des sprint planning en souffrance, elles permettent d'enrichir le propos du product owner.

- En savoir plus sur les niveaux de constitution d'une équipe par **Tuckman**.
- Le **backlog grooming**.
- Une **vidéo** sur un « mur » scrum de deux semaines
- Le **scrummaster que vous avez toujours rêvé d'être**, avec Jeff Sutherland dans le rôle du DG.
- Pourquoi j'aime que les **user stories restent métiers et pas techniques**, et la **partie 2** de l'article.

Et la semaine avance, les tâches (sur les post-its) avancent, des user stories, éléments fonctionnels, sont validés par le product owner.

Naturellement toute la validation se déroule sur la plate-forme d'intégra-

tion : un serveur propre, sans anomalie (si il y a une anomalie la user story n'est pas validée et remise à l'équipe).

Donc oui : il est juste impossible de valider une user story avec un bug, une anomalie. Soit l'anomalie est mineure, cosmétique, et il est rapide de la corriger, donc pas de raison de la valider avec cette anomalie. Soit l'anomalie est importante et donc pas de raison de la valider avec cette anomalie. Oui les agilistes sont intransigeants sur ce point (ou devraient l'être). Nous ne voulons pas de bug, aucun. Ce n'est pas que nous ne faisons pas de bug, nous en faisons autant voire plus que les autres (car nous aimons essayer), mais nous n'acceptons pas de valider comme fini quelque chose avec des bugs, nous n'acceptons de sortir quelque chose avec des bugs. Ce n'est pas une religion! Nous ne sommes pas les chevaliers blancs du bug! C'est tout simplement que tous ces bugs coûtent plus cher et plus de temps. Nous ne voulons pas de bugs car nous voulons gagner du temps et de l'argent. Tout le monde a déjà vécu, a observé, que un bug qui se corrige dans l'instant va coûter deux minutes, deux heures ou deux jours, et qu'un bug qui se corrige plus tard, va coûter le double, le triple, le quintuple... Et le 8ème bug c'est un nouveau bug où il est lié au 4ème? ...bref une vraie gadoue.

Ensuite comme nous travaillons itérativement impossible de demander au product owner de valider toute la première itération la première fois, puis la première et la seconde la deuxième fois, puis la première, la seconde, la troisième...cela deviendrait fou.

Impossible aussi d'imaginer valider et tester durant une itération+1 le résultat de l'itération, avec une autre équipe, tout en commençant une nouvelle itération. Cela ne marche pas, c'est une fuite en avant suicidaire. Dès que l'on détecte une anomalie, et cela sera le cas sans validation ni test préalable, comment la traiter? Interrompre l'équipe qui a commencé l'itération suivante? Impossible on se prend systématiquement les pieds dans le tapis. Une seule pratique : quand c'est fini, c'est fini, un seul rythme, une lecture clarifiée de l'état des lieux.

Il est donc aussi obligatoire que ce qui est validé reste en bon état constamment. Pour cela nous faisons tourner régulièrement (très régulièrement : plusieurs fois par jours), des tests automatisés à tous les niveaux : unitaire, technique, métiers, d'intégration, de performances, d'interface, etc. Cela demande donc des serveurs (oui Agile coûte plus cher en infrastructure, mais compte tenu de la qualité et de son optimisation de la valeur sont coût global est meilleur n'oubliez pas, je n'y crois pas : je l'observe). La vraie difficulté dans les tests automatisés (ils sont faits par l'équipe naturellement!) c'est d'abord d'apprendre à les réaliser, à en intégrer l'importance,

mais surtout les jeux de données adéquats, simili prod idéalement (sinon comment vraiment développer?). Jeux de données que l'on va déployer, modifier, supprimer, re-déployer, constamment... Une vraie réflexion doit avoir lieu sur le ratio effort/valeur, mais voilà la direction.

Un burndown

A cette époque (2010), je comptais encore les points par tâche (en quart de journée : 1=0,25, soit normalement 4 max, car une tâche ne devrait pas excéder un jour, mais j'acceptais facilement 8, jusqu'à deux jours). Je ne compte plus les points par tâche, cela n'a pas d'intérêt, je compte juste le nombre de tâches pour les burndowns, pour voir si on avance sur le chemin prévu, où si il faut agir, prendre une décision, etc. Je demande simplement à ce que les tâches restent petites « moins de deux jours ».

Attention sans *burnup* nous pourrions nous faire piéger : disons que nous avons dix tâches par *user story*. Elles avancent toutes sauf une par *user story*. La ligne serait presque parfaite sauf qu'à la fin nous n'aurions...rien. Or, pas de valeur dans les tâches (qui font parties du *comment*). La valeur réside dans les *user stories* (c'est pourquoi aussi j'aime que les **user stories restent métiers et pas techniques**, et la **partie 2** de l'article). Un *burnup* de validation des user stories m'aurait montré que l'on ne valide rien au fil de l'eau.

Figure 11 – Burndown 2

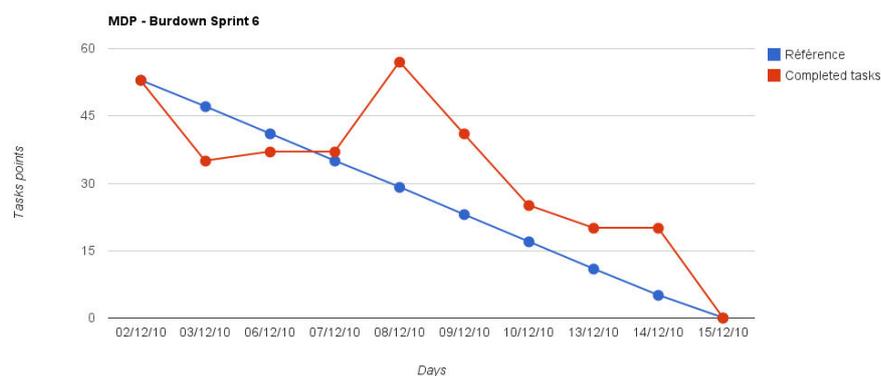


Figure 12 – Autre plan de release

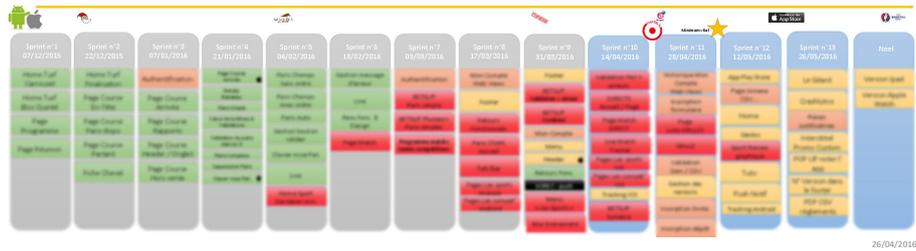
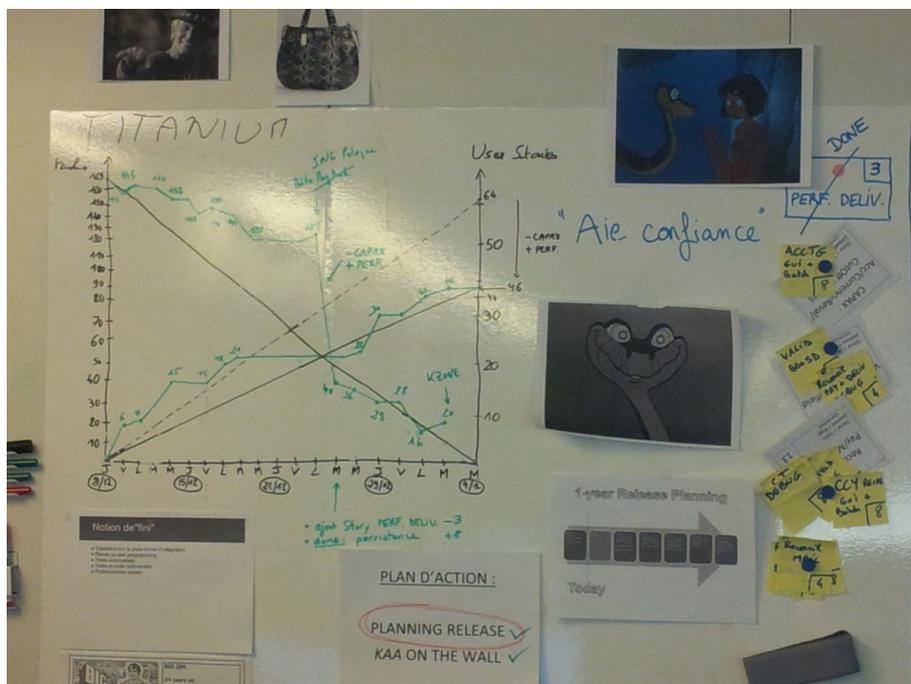


Figure 13 – Un bout de « radiateur d'information ». C'est comme cela que l'on appelle les murs « chauds » agile : il diffuse de l'information. Ici on chambre le coach en l'associant avec le serpent KA (et on lui dit aussi qu'on va le transformer en sac si besoin), c'était bien moi. Cela date de 2011? On y voit un plan de release, une liste d'actions post-rétro, une définition de fini, un burndown, un burnup?



4.1.4 Fin de l'itération

Notre itération avance. Nous arrivons le vendredi de la deuxième semaine. L'après-midi c'est la **revue** que l'on appelle aussi la **démo**, mais on ne fait pas une démo, on fait la revue du travail réalisé, c'est différent, donc je préfère revue. Non pardon : pas du travail réalisé, du travail fini. Fini : testé, validé, qui respecte la définition de fini et qui a été validé par le *product owner*.

À 14h et pendant disons 45 minutes on va donc faire la **revue** (que l'on peut préparer un peu avant, à la fin de la matinée, si on est bon élève, quel-qu'un de bien, pas un vaurien, un *bad boy*). Durant celle-ci on montre le travail fini. Et seulement celui-ci. Pas question de montrer quelque chose en cours. Ce qui est en cours ne sera peut-être jamais fini, ou différemment. Et pour avoir de bons débats, de bons arbitrages, une vraie visibilité, un vrai *feedback* (rigolo le *feedback*, on n'a pas de terme français vraiment adapté, c'est symptomatique. J'aime bien le lien avec l'apprentissage) : bref, on montre ce que l'on a et PAS ce que l'on a PAS. Mais pas d'intermédiaire. On nous prend toujours pour des cowboys hippies mais sur ce point nous sommes binaires. La lecture du projet/produit est ainsi éminemment facilitée. Et comme les éléments finis peuvent potentiellement être mis en production on sait exactement ce que l'on a et ce que l'on a pas. Pas de flou artistique sur 80% d'une fonctionnalité en cours : cela ne veut rien dire.

Ainsi que se passe-t-il durant les 30/45mn de la revue? Disons qu'il y a une ouverture de 5mn : « *voilà l'état des lieux, voilà sur quoi nous nous sommes projetés pour cette itération, voilà l'état de nos plate-formes et de nos tests automatisés. Naturellement à tout moment vous pouvez vous connecter à la plate-forme d'intégration continue pour jouer avec notre produit* ». Puis on fait la revue des choses finies, le mieux c'est que cela soit le *product owner* qui le fasse (du coup les premières itérations a minima, qu'il sente bien qu'il appartient à l'équipe, et que quand il valide il pense à sa revue...). Donc la revue surtout au début est portée par le *product owner* (après une autre personne peut le faire, quand le pli est pris). Naturellement aucun slide, on veut des choses finies, on montre des choses finies, sur du réel. Si on ne peut pas montrer, on ne peut pas faire. Soyez inventifs.

Si dans la **définition de fini** il y a marqué **haute disponibilité** comptez sur moi pour débrancher le serveur en direct avec tout le monde, normalement la revue peut continuer sur un autre serveur puisque nous disons que nous faisons de la haute disponibilité. Ou si il y a marqué dans la **définition de fini** : apps mobile et site web, comptez sur moi pour arrêter la revue sur le site et demander à la voir sur l'application mobile. Je sais ces

scénarios sont durs pour l'équipe, mais c'est tellement mieux de se trouver en défaut pendant une revue que en production. Et on se retrouve tellement plus zen quand on a poussé le niveau de qualité jusqu'au bout. Et on gagne tellement en vitesse quand cette qualité offre une base solide. Ainsi pendant 20 minutes le *product owner* fait la revue des réalisations finies (qu'il a validé au fil de l'eau lors de l'itération), et demande du *feedback*. C'est le lieu du *feedback* la revue. Que les gens parlent maintenant ou se taisent à jamais (pour le meilleur et pour le pire...). Ils ne pourront ne devraient pas dire qu'ils ne savaient pas. Les 5/10 dernières minutes de la revue s'ouvre sur la suite : on propose un plan de release (qui est remis à jour toutes les itérations) et qui projette la réalisation estimée sur un calendrier. On a observé une cadence, on la reproduit, et cela nous dit où nous devrions être à telle ou telle date. C'est très efficace, et cela marche relativement bien : pas d'estimation mais la reproduction d'une cadence observée, cela marche relativement bien aussi car la lecture est claire : ce que l'on a on l'a, et ce que l'on a pas, on ne l'a pas. Et comme tous les éléments sont découpés par avoir du sens de façon assez autonome, la lecture de l'état des lieux du projet/produit est efficace. À cela je rajoute qu'il ne devrait pas y avoir de bug, et que nous appliquons la définition de fini qui prend en charge les éléments clefs du projet/produit (charge? performance? etc.) donc pas de surprise.

Et si vraiment on découvrait un bug lors de la revue? Ca arrive, rarement normalement, mais ça arrive. Il vient se placer en tête du backlog, de l'expression du besoin, et on le traite donc en priorité 1. Enfin, c'est ma philosophie, après le *product owner* peut décider différemment tant qu'il assume.

Autre sujet, la revue est l'occasion pour tous les gens concernés notamment les sponsors d'être au courant. Une demi-heure toutes les deux semaines. Si malgré tout personne ne vient, proposez d'arrêter le projet/produit. Il va s'ensuivre un dialogue de ce genre avec le sponsor : « Nous avons arrêté le projet », « Quoi? Pourquoi? », « Personne ne vient aux revues, c'est qu'il n'est pas important, autant se concentrer sur les choses importantes », « Mais si il est très important! », « Attendez je ne comprends pas soit il est très important et les gens viennent une demi-heure toutes les deux semaines, soit il ne l'est pas et autant l'arrêter ». Naturellement c'est plus facile à porter comme dialogue pour un externe comme moi, mais croyez moi, ça marche.

45 mn, 30mn, les 45mn sont une boîte de temps hein. Tous les rituels de scrum sont des boîtes de temps : on peut faire plus court, mais on ne dépasse pas (la durée de l'itération, le sprint planning, etc.). On apprend à fonctionner de façon optimale dans ce cadre, dans cette boîte. Et chaque

boite se déroule au même moment, avec le même objectif, les mêmes éléments en entrée, et les mêmes éléments en sortie. D'où l'idée de rituels. Pas la peine de sacrifier de chèvre.

Figure 14 – Là une sorte de mur Scrum et Kanban qui paraît bordélique mais qui marchait assez bien



Petit moment de relaxation après la revue, puis c'est le tour de la **rétrospective**.

La rétrospective est un sanctuaire, elle est réservée à l'équipe (les développeurs – dans le sens réalisateurs – du produit – dans son ensemble, pas que le code –, le *product owner*, le *scrummaster*). Pour s'améliorer il faut se parler franchement, pour parler franchement au sein de l'équipe, il faut savoir que les débats vont rester entre nous, et qu'aucune personne extérieure ou manager n'est présente. Cependant si l'équipe le demande, tout reste permis.

La **rétrospective** c'est le lieu donc de l'amélioration continue, pour cela il faut s'interroger. Généralement pour deux semaines d'itération, la rétrospective dure 2h. Les premières 20mn sont consacrées à se rappeler ce qui s'est passé durant les deux semaines écoulées (une ligne de temps avec les faits par exemple), puis l'heure suivante sert à s'interroger, faire émerger les questions, les succès, les échecs, les frustrations, les plaisirs, etc. On prend 5/10mn pour choisir ce qui nous interpelle le plus : soit le point douloureux que l'on veut résoudre, soit le point positif que l'on veut propager. Puis on passe la dernière demi-heure à trouver comment le résoudre, ou le propager, de façon factuelle, mesurable, réaliste durant les deux prochaines semaines. Les embûches de la **rétrospective** : a) vouloir changer le monde en deux semaines, se lancer dans cinq améliorations dans les dix prochains jours, irréaliste et frustrant, mieux vaut se focali-

ser sur un seul point; b) ne jamais changer de format, toujours les mêmes conversations et points de vue reviennent, irritant et frustrant, c'est le boulot du *scrummaster* de manipuler différents formats qui feront voir les choses sous des angles nouveaux régulièrement; c) trop se focaliser sur les aspects négatifs à améliorer : se focaliser sur les points positifs et les propager c'est tout aussi efficace, voire plus!; d) Se donner un plan d'action simple et réaliste qui puisse être porté par l'équipe, et pas un plan compliqué, non mesurable, qui est lié à des gens externes, c'est frustrant et inutile.

Voilà, lundi ça recommence : sprint planning première partie, 2h30, on se projette sur un ensemble d'éléments à réaliser, l'après-midi l'équipe découpe en tâches, on fait le *burndown* et le *burnup*, que l'on suit dans la semaine, tous les jours on a notre *daily standup*, le *product owner* valide au fil de l'eau les *user stories*, une ou deux réunions d'affinage (*backlog grooming*) permettent d'anticiper le prochain *sprint planning*, notre corpus de tests automatisés s'enrichit, deuxième vendredi revue et rétrospective. Puis le lundi après sprint planning, et ça repart. Un Scrum bien appliqué dégage une forte impression de cadence et de rigueur, presque comme le Taylorisme. Sauf que les équipes sont maîtres de leur destin. Ca change tout. J'ai l'habitude de m'amuser à dire que si vous débranchez des équipes scrum vous avez des poules sans tête : où est mon daily? Sur quoi on travaille pour les deux semaines à venir? pour renforcer l'idée de cadence. Cette cadence on l'aime ou pas, il faut des cycles de relâche. Ne soyez pas surpris si tous les six mois, comme une équipe de sport, votre équipe Scrum a besoin de repos. Laissez là faire. N'oubliez jamais le rythme soutenable. Et que l'on ne va pas plus vite en allant plus vite mais en allant différemment.

Cette rigueur est présente car l'outillage Scrum est compact, dans la tête.

Quand plusieurs équipes cohabitent? Et bien je suggère de ne pas les synchroniser au niveau calendaire. C'est mon vécu, et je n'ai pas vu l'inverse fonctionner aussi bien. Pourquoi? Je fais primer la présence des sponsors et des utilisateurs finaux lors des revues avant toute chose : l'apprentissage, le feedback. Impossible d'avoir ces personnes qui suivraient 10 équipes, soit 10 fois 45mn dans une même journée. Trop lourd, indigeste. A ceux qui défendent l'idée qu'il suffit de faire 2 heures pour l'ensemble des équipes : j'y trouve un manque de respect et manque de responsabilisation des équipes, c'est mon expérience, mon point de vue. Donc je m'oriente plutôt vers des alternances de semaines, de jours, pour permettre à toutes les revues d'avoir une vraie existence, avec de vrais sponsors, client finaux (potentiellement), et un vrai feedback/apprentissage.

Articuler plusieurs équipes? On utilise généralement des plan de release consolidés et/ou un **kanban portfolio**.

Articuler une organisation autour d'un produit ou projet en scrum? de l'**holocratie**? votre propre saveur d'entreprise libérée, au choix. Mais n'oubliez pas les **bases de ce monde complexe**.

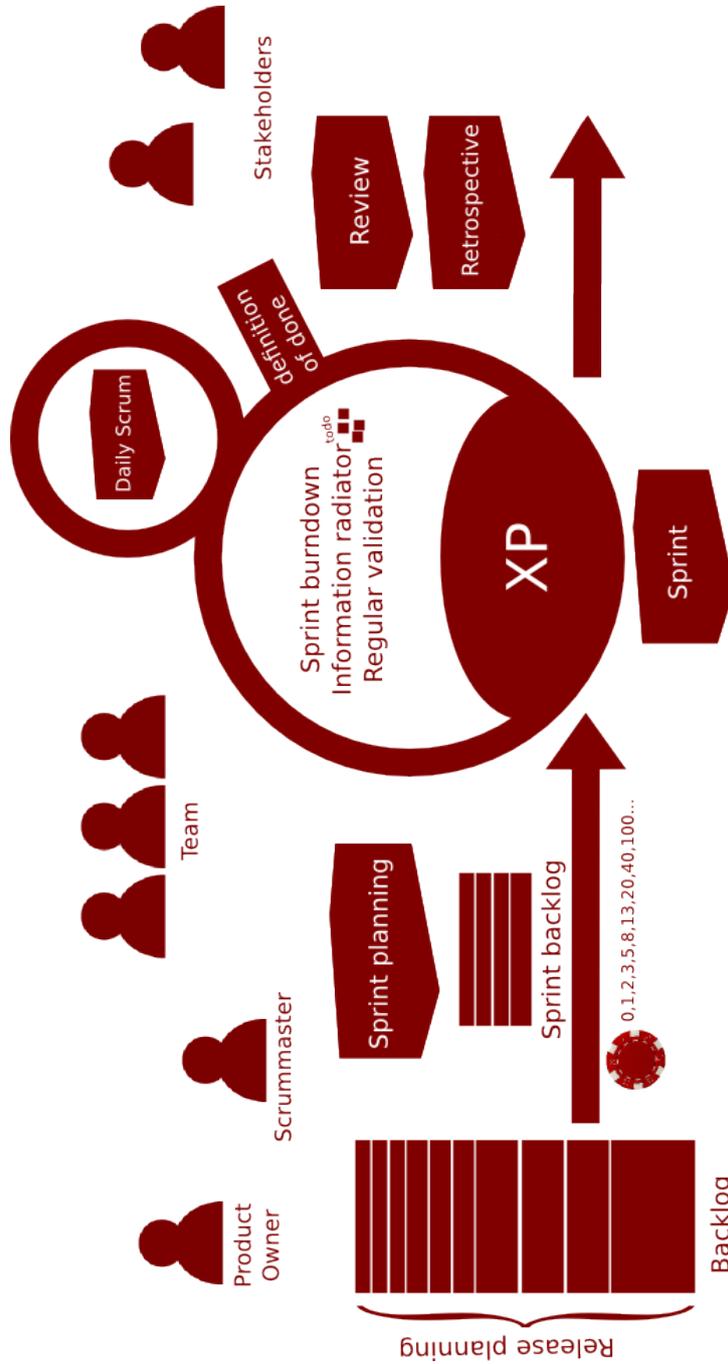
Des rétrospectives?

- **Festival de rétrospectives.**
- **L'art de la rétrospective'**
- **Deux formats de rétrospectives**
- **Rétrospectives osées**

Pour finir, Scrum, on s'en fout. Kanban on s'en fout. XP on s'en fout (mais ça tout le monde le dit malheureusement). Avant toutes choses pensez aux fondations : l'intention et l'attitude de ceux qui ont le pouvoir dans l'entreprise et les moyens physiques de vos équipes et produits : co-localisation ou véritable espace virtuel, type de produit, de marché, de déploiement, etc. Après Scrum, Kanban, XP sont des outillages aboutis, aiguisés, qu'il est très bien d'utiliser pour faire la différence.

4.2 Scrum, récapitulatif rapide

4.2.1 Diagramme



4.2.2 Sprint Planning

Ou encore ... la réunion de lancement de l'itération. Le *sprint planning* c'est deux étapes :

Qui? : sur quoi devons nous travailler? définition du contenu de l'itération à venir (avec estimation & engagement).

Comment? : comment allons nous aborder notre travail (découpage en tâches).

Qui? : Toute l'équipe Scrum : le product owner, le scrummaster, l'équipe.

Boite de temps? : Pour une itération de 2 semaines, disons 3-4 heures. 2 heures la première partie, 1h30 la seconde partie.

Entrée : Un backlog avec les user stories en tête assez détaillées.

Sortie : L'engagement de l'équipe sur un ensemble de user stories.

4.2.3 Daily Standup

Ou encore ... la réunion quotidienne de l'équipe. Le daily standup c'est tous les jours à la même heure, dans un même lieu, le moment où on partage nos avancées, notre travail et l'état des lieux du projet au sein de l'équipe.

Qui? : L'équipe uniquement (mais le scrummaster doit s'assurer que le daily standup se déroule correctement). Public silencieux mais autorisé.

Boite de temps? : 15 minutes maximum c'est impératif.

Entrée : Présenter à chacun le travail effectué depuis le dernier daily standup, le travail sur lequel on travaillera jusqu'au suivant, et si l'on rencontre des difficultés.

Sortie : Une connaissance partagée, une information circulante au sein de l'équipe.

4.2.4 Review

Ou encore ... le passage en revue des réalisations de l'itération. C'est le lieu où l'on montre les réalisations de l'équipe lors de la dernière itération, mais uniquement ce qui est réellement fini. C'est aussi le lieu où l'on reçoit le feedback des parties prenantes.

Qui? : Public. Tout le monde est le bienvenu. Il faut même s'assurer que les personnes clefs sont bien présentes (« parlez maintenant ou taisez vous à jamais... »).

Boite de temps? : La durée officielle est 1h30. Je préconise 30/40mn car il n'est pas aisé de déplacer nos parties prenantes clefs, or c'est impératif. Si vraiment il faut allonger la durée pour discuter du feedback, je suggère de provoquer une réunion dédiée.

Entrée : Le résultat de l'itération (mais rien qui ne soit fini et validé par le product owner). Des éléments d'informations, et une projection (release plan) sur la suite.

Sortie : L'assurance que les parties prenantes soient au courant de l'avancée des travaux du projet, et qu'elles aient eu l'opportunité de donner leur feedback.

4.2.5 Retrospective

C'est le moment où l'on s'interroge sur comment s'améliorer. En 3 étapes : on observe le déroulement de l'itération, on définit ce qui s'est bien passé, et ce qui s'est mal passé, et nos axes d'améliorations. Enfin on sélectionne un (et un seul) élément à améliorer et comment y parvenir.

Qui? : C'est un espace sanctuarisé, réservé uniquement à l'équipe (product owner, scrummaster, l'équipe).

Boite de temps? : 2h pour une itération de deux semaines. Il faut du temps pour bien approfondir les sujets ou laisser émerger les idées.

Entrée : Qu'est ce qui s'est bien passé, qu'est ce qui s'est mal passé, que peut-on améliorer?

Sortie : Un et un seul élément d'amélioration qui soit atteignable et mesurable.

4.3 Kanban

Cinq principes de Kanban :

1. Visualiser votre flux de création de valeur
2. Optimiser votre flux
3. Limiter le travail en cours
4. Clarifier les règles et les processus
5. S'améliorer de façon collective

Article [Portfolio Projets Kanban - partie 1](#)

Article [Portfolio Projets Kanban - partie 2](#)

Support pdf sur Kanban : [Kanban](#)

4.4 Lean

Sept principes du Lean :

1. Eliminer les gaspillages
2. Amplifier l'apprentissage et la connaissance
3. Montrer le plus tôt possible
4. Décider le plus tard possible
5. Responsabiliser les équipes
6. Fabriquer de la qualité intrinsèque
7. Voir le tout

4.5 Glossaire

Backlog

Les choses à faire, ordonnées par valeur. Le product owner le bichonne, c'est son jardin. Comme une plante, c'est vivant.

Backlog grooming

Faire le toilettage de son backlog pour favoriser les travaux à venir. En savoir plus : « **Pour que le backlog grooming ne soit plus une corvée** » par Géraldine.

Daily Scrum

Point journalier au sein de l'équipe.

Définition de fini

Les critères de qualité que l'on se fixe pour estimer que quelque chose est fini. En savoir plus : « **nul n'est sensé ignorer la définition de fini** ».

L'équipe

Elle réalise le comment, les estimations et l'engagement. Elle est heureuse car elle est auto-organisée et elle peut agir.

Gherkin ou BDD Behavior Driven Development

Une façon d'exprimer les tests d'acceptance dans les user stories qui propose une situation, une action, un résultat.

Impact Mapping

Approche pour découper stratégiquement son besoin.

Kanban

Le management visuel qui pousse à s'améliorer.

Open Agile Adoption

Une façon d'appréhender le changement et l'amélioration sans s'imposer une méthode particulière et en invitant les gens à prendre en main la transformation.

Personas

Personnages virtuels qui représentent les personnes impactées par le produit ou le projet et qui permettent d'éviter de se perdre en route.

Plan de release

Vision synthétique des itérations passées et des itérations à venir.

Planning Poker

Une façon de faire les estimations en convergent et en comparant.

Product Owner

Le propriétaire du produit, le porteur du pourquoi, du quoi, de la vision et de la valeur. Il est fier de son produit.

Retrospective

La réunion où l'on s'interroge sur l'itération ou nous même et comment s'améliorer.

Review

La réunion publique où l'on observe le résultat de l'itération et où l'on écoute les retours des parties prenantes.

Scrum

La mêlée en anglais, aussi le nom de la méthode agile la plus populaire. Si un membre de la mêlée est en difficulté, la mêlée s'effondre.

Scrummaster

Le garant de la méthode et le facilitateur du produit, du projet. Il est totalement neutre.

Sprint

La boîte de temps qui définit une itération. Malheureusement trop souvent appelé et vécu comme un sprint.

User Stories

Les histoires qui constituent le backlog.

User Story Mapping

Approche pour découper tactiquement son besoin.

5 Supports & Matériels

Les titres en bleu indiquent un lien vers une ressource en ligne. Je vous indique la langue d'origine du livre ou du support, mais vous pouvez peut-être trouver une version dans une langue qui vous plaît plus.

5.1 Supports de la formation

5.1.1 PDFs de *areyouagile.com*

Le grenier de *areyouagile.com*

L'endroit où l'on trouve les liens vers les publications et autres supports en pdf.

Formation agile

Des slides pour une formation agile sur 3 jours autant de Scrum, Kanban & Extreme Programming.

Initiation agilité

Des slides pour une initiation à l'agilité sur une journée.

Vision & sens

Un module demi-journée concernant le sens et la vision (approche vision du projet ou de la solution).

Expression du besoin

Un module demi-journée concernant l'expression du besoin (User stories, personas, impact mapping, etc.).

Zoom Scrum

Un module demi-journée qui zoom sur Scrum.

Peetic, des exemples

Peetic, le site factice de rencontres en ligne pour animaux : des exemples de personas, de backlog, d'impact mapping, de rétrospective, etc.

Autres slides / Speakerdeck

Slides en ligne sur les différentes conférences ou interventions que je peux réaliser : Open Agile Adoption, Open Office365 Adoption, La horde agile, La stratégie du product owner, etc.

Scrumshot, animation scrum

Petite animation colorée type « prezi » pour animer une discussion autour

de Scrum.

Kanban

Une introduction à Kanban (support encore *beta*).

À lire plutôt :

Article [Portfolio Projets Kanban - partie 1](#)

Article [Portfolio Projets Kanban - partie 2](#)

Product Owner

Une introduction au rôle de Product Owner (support encore *beta*).

Lean Software Development

Une introduction au Lean Software Development (support encore *beta*).

5.1.2 Ateliers d'expression du besoin

Remember the future

Atelier de projection sur la vision qui produit qui permet de s'extraire des contingences journalières (Priorisation et mesure du produit, du projet).

Product Box

Atelier de priorisation et de mise en évidence de la valeur attendue du produit.

Buy a feature

Atelier de priorisation sur le produit quand plusieurs visions s'affrontent.

User Stories Mapping

Atelier d'organisation des histoires utilisateurs (User Stories), surtout selon une approche pratique.

5.1.3 Focus Lean Startup

Impact Mapping

Atelier d'organisation des histoires utilisateurs (User Stories), surtout selon une approche stratégique.

Article [Cartographie de stratégie, l'impact mapping](#)

Article [Cartographie de stratégie, l'impact mapping, hors des sentiers battus](#)

Article [Impact map : recette de cuisine](#), par Géraldine

Article [Cartographie de plan d'action : le User Story Mapping](#)

Article [Cartographie de plan d'action\(user story map\) revisitée](#)

Article [De l'idée au plan de livraison](#)

Article [Éléments de Lean Startup : croquettes Peetic](#)

[Pirates Metrics](#)

De sympathique et concrets métriques en environnement web (par Jurgen de Smet).

5.1.4 Focus sur *Open Adoption*

[Forum ouvert](#)

Origine et définition du forum ouvert par wikipedia.

[Guide du forum ouvert](#)

Guide du forum ouvert par son inventeur Harrison Ownen (en).

[Open Agile Adoption \(articles\)](#)

Articles de Dan Mezick sur *Open Agile Adoption*.

[Open Agile Adoption \(slides\)](#)

Des slides sur *Open Agile Adoption*.

[Open Adoption \(slides\)](#)

Des slides sur Open Adoption (Darefest 2015).

5.2 Lectures

5.2.1 Livres relatifs aux méthodes agiles et outils

Guides Scrum

Guides de Scrum (à peu près 20 pages) de [scrum.org](#), en français, en anglais, en polonais....

Scrum, Claude Aubry

Scrum, point d'entrée classique et efficace sur la méthode la plus populaire actuellement. Par Claude Aubry, un copain toulousain avec qui je fais

le **raid agile**, une **formation agile différente dans les Cévennes** (ou ailleurs) ; en français. (fr)

Kanban pour l'IT, Laurent Morriseau

Scrum, point d'entrée classique et efficace concernant Kanban en environnement IT. Par Laurent Morriseau, un copain breton, en français. (fr)

Spécifier agile, Thierry Cros (prononcez « crosse »)

Pour approfondir les spécifications en environnement agile. Par Thierry Cros un copain occitan du sud. (fr)

User Story Mapping, Jeff Patton.

Mener une découpe de votre vision en histoires utilisateurs avec une approche pleine de sens et de richesse : le user story mapping de Jeff Patton. À rapprocher de Impact Mapping, qui est stratégique, alors que User Story Mapping est tactique. (en)

Retrospective, Esther Derby & Diana Larsen

(Très) bon petit ouvrage sur les rétrospectives. Pour creuser cette étape essentielle de l'amélioration continue. (en)

Lean Startup, Eric Ries

Un livre inspirant sur les principaux concepts du Lean Startup. Lecture à compléter avec celle de *Running Lean*. (en)

Running Lean, Ash Maurya

Un livre inspirant sur les principaux concepts du Lean Startup. Lecture à compléter avec celle de *Lean Startup*.(en)

Impact Mapping, Gojko Adzic

Approche outils Lean Startup pour viser juste dans votre création de valeur (**Impact Mapping**). À rapprocher de User Story Mapping, qui est tactique, alors que Impact Mapping est stratégique. (en)

Agile Lean Toolkit, Mary Poppendieck

Le Lean Software Development par sa grande prêtresse Mary Poppendieck. A longtemps été mon favori. (en)

5.2.2 Livres pour appréhender la culture agile

Freedom Inc (Liberté & Compagnie en français), Isaac Getz

Une très bonne introduction à l'entreprise libérée, cette nouvelle façon

de vivre et penser l'entreprise dans notre monde agile. Lecture à compléter avec celle de *Tribal Leadership*.(fr ou en).

Tribal Leadership, Dave Logan

Tribal Leadership c'est un peu la matrice de lecture des histoires de *Freedom Inc*. Indéniablement inspiré par la Spirale dynamique. Lecture à compléter avec celle de *Freedom Inc*(en).

Reinventing Organizations, Frédéric Laloux

Holocratie, entreprise libérée, un suivi et des pistes opérationnelles pour devenir une organisation de nouvelle génération.

The Culture Game, Dan Mezick

Là aussi on pense l'entreprise avec de nouveaux paradigmes et de nouveaux principes. Essentiel concernant ce que nous pourrions apprendre des jeux vidéos, et concernant Open Agile Adoption. Dan est un copain et un homme extraordinaire (comme sa femme Roberta). Lecture à compléter avec celle de *Reality is broken*. (en).

Introduction à la pensée complexe, Edgar Morin

Ce n'est pas dur à lire, c'est un petit bouquin, et cela permet de comprendre le monde complexe qui nous entoure, et le nouveau paradigme qu'il propose. (fr).

La revanche du rameur, Dominique Dupagne

Les histoires d'un médecin confronté aux dérives kafkaïennes de la médecine française et qui soudain découvre l'agilité. Très inspiré par Henri Laborit. (fr).

Drive, Dan Pink

Mais finalement qu'est ce qui nous motive ? Sûrement pas les primes. Lecture à compléter avec celle de *The Culture Game* pour les questions sur l'implication.(en).

Switch, Chip & Dan Heath

Changement et résistance au changement. Comment appréhender et mener le changement face à nos deux cerveaux (l'émotionnel et le cérébral). Lecture à compléter avec celle de *Thinking Fast & Slow*.(en).

Thinking fast & slow, D.Kahneman

Nos deux cerveaux (le flemmard intuitif, et le flemmard rationnel) sont la source de bien des biais cognitifs. (en).

Reality is broken, Jane McGonigal

Sur l'implication des joueurs de jeux vidéos et tous les enseignements que nous pouvons en tirer pour nos entreprises. Lecture à compléter avec celle de *The Culture Game*.(en).

The fifth discipline, Peter Senge

Une somme sur les entreprises organiques (systèmes complexes adaptatifs) et l'émergence dans ces nouvelles organisations. (en).

Unfolding the napkin, Dan Roam

Un dessin vaut mieux qu'un long discours. La pierre rosette a foutu le bazar, on fonctionnait mieux avec l'Aurochs sur les murs. Une nouvelle façon de communiquer, par le maître. (en).

The machine that changed the world, Womack, Jones, Roos

L'histoire comparée de Ford et Toyota pour expliquer le changement de paradigme entre le taylorisme et le lean. (en).

To the desert en back, Mirvis, Ayas, Roth

L'histoire du changement conduit chez Unilever. Une transformation d'entreprise conduite au travers de performances émotionnelles. Lecture à compléter avec celle de *Seven habits of highly effective people*. (en).

Seven habits of highly effective people, Stephen Covey

Être efficace, ou tout simplement pertinent, ce livre sur son propre management est très intéressant. Lecture à compléter avec celle de *To the desert and back*.(en).

Change your questions,change your life, Marilee Adams

Revoir sa posture, revoir sa façon d'adresser ses relations, ses problématiques, son attitude. Simple mais très efficace. Pour du développement personnel.

Mindsight, Daniel J.Siegel

Mémoire explicite, implicite, le fonctionnement de notre cerveau, notre intégration dans le sens notre équilibre. Siegel propose des réflexions qui remettent au goût du jour certains concepts plus anciens en les associant avec la science du cerveau et en nous faisant percevoir notre capacité à nous transformer. Pour du développement personnel.

Primal Leadership, Goleman

Nouveau management, nouveau paradigme de l'intelligence émotionnelle. Ce qui fait vraiment les grands leaders. Différents types de management, et les étapes du changement.

The Information, James Gleick

Un grand parcours romancé sur l'information depuis le premier dictionnaire jusqu'aux bits informatiques en passant par le télégraphe. Tout ce qui fait de l'information la denrée la plus intéressante des 100 dernières années. Ce livre historique, romancé m'a passionné cet été 2015.

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

Qu'est ce qui fait de l'homme, l'homme? Observons nos ancêtres qui parcouraient la Savane. Nous n'avions ni dents acérées, ni muscles proéminents, encore moins une rugueuse carapace. Et pourtant nous avons émergé et finalement dominé le monde. Pourquoi? parce que nous avons cette culture de l'adaptation en milieu complexe. Avec Descartes et les Lumières nous avons un peu oublié cela. Nous avons une réponse cartésienne à chaque question. Et nous pensons que le monde marche ainsi. Mais ce n'est plus le cas : interdépendance, connectivité totale, immédiateté, simultanéité, globalisation, le monde est de nouveau imprédictible. Aujourd'hui que peut nous enseigner notre vieille horde?

5.3 Compléments venant d'autres blogs

Le [blog de Claude Aubry](#) avec qui je fais le [raid agile](#), une **formation agile différente dans les Cévennes** (ou ailleurs).

Les **traductions pléthoriques** mais tellement utiles de **Fabrice** : [Wiki Ayeba](#)

Les introspections de **Géraldine** (la plus prometteuse des coachs agiles) : [The Observer Self](#).

Très bonne traduction de Henrik Kniberg concernant le **MVP** de l'ami **Frank** : [Comprendre le MVP](#).

Petite introduction rigolote à la **résonance limbique** : [Tripote moi le limbique](#).

5.4 Divers

5.4.1 Vidéos en ligne (culture agile)

Français

Jean-François Zobrist

Malgré un côté un peu misogyne ou vieille France, une excellente vidéo sur l'entreprise libérée. Cette façon nouvelle de penser les entreprises.

Marc Halévy

Là aussi si les livres de Marc Halévy sont parfois déséquilibrés entre une excellente prospective et des idées douteuses, cette vidéo est un vrai moment de plaisir et de fraîcheur sur les entreprises modernes.

Anglais

The role of leadership in software development , Mary Poppendieck

Une passionnante intervention sur le management ancien et moderne par la prêtresse du Lean Software Development.

Robert C. Martin - The Land that Scrum Forgot

Excellente intervention de l'un des pionniers de l'agilité sur les dérives de certains mouvements, et sur la source de l'agilité.

Dan Pink, On motivation

Une intervention synthétique de Dan Pink sur son ouvrage (*Drive*) concernant la motivation.

Intelligence sociale par Daniel Goleman.

Toute cette intelligence émotionnelle qui fait la différence.

How great leaders inspire par Simon Sinek.

Le cercle doré du pourquoi, du quoi, du comment.

5.4.2 Documents annexes

Le cahiers des charges de la deux chevaux

Étonnant cahier des charges de dix lignes concernant l'un des symboles de la voiture française.

La culture chez Netflix, juste 16 millions de vues pour cet excellent document

Table des figures

1	Les temps complexes	6
2	Mouvements agiles	10
3	Un persona	14
4	Un impact mapping	15
5	Un autre impact mapping	16
6	Ma pyramide à moi de l'expression du besoin agile	16
7	User Story Mapping	17
8	Un plan de release	20
9	Un mur scrum	21
10	Burndown et burnup	22
11	Burndown 2	27
12	Autre plan de release	28
13	Un bout de « radiateur d'information ». C'est comme cela que l'on appelle les murs « chauds » agile : il diffuse de l'information. Ici on chambre le coach en l'associant avec le serpent KA (et on lui dit aussi qu'on va le transformer en sac si besoin), c'était bien moi. Cela date de 2011? On y voit un plan de release, une liste d'actions post-rétro, une définition de fini, un burndown, un burnup?	28
14	Là une sorte de mur Scrum et Kanban qui parait bordélique mais qui marchait assez bien	31