

FOIRE AUX QUESTIONS AGILES

version : 24 septembre 2018



Source gallica.bnf.fr / Bibliothèque nationale de France

Table des matières

Le Lean Canvas ?	3
Scrum est-il éligible à tout ?	3
Comment faire de l'agile en Cobol ?	4
Comment faire quand le PO se désengage ?	4
Le Scrummaster est-il un chef d'équipe ? Y-a-t-il une relation hiérarchique avec le reste de l'équipe ?	5
Est ce qu'un scrum master peut transmettre des félicitations aux équipes, félicitations qu'il a reçues par ailleurs ?	5
On nous demande de travailler sur un projet agile. Quand le projet s'arrête, on arrête l'agile ?	5
Dans une organisation en mode spotify avec des tribus et des squads, la tribu doit elle faire une rétrospective en fin d'itération ? Qui y participe : les scrums master et po des squads ? les équipes de toutes les squads ?	6
Question : je découvre l'agilité, il paraît que c'est top presque miraculeux. Comment je peux commencer dans mon quotidien à faire de « l'agile » ? Q1 : Y a t'il des pratiques simples à mettre en oeuvre ? Q2 : Je peux demander de l'aide, à qui ?	6
Comment devient-on coach agile ?	7
Est ce que lorsqu'on a travaillé dans une petite entreprise (4-5 personnes) c'est une expérience Agile ?	7
Est ce qu'on peut être coach agile sans être passé Po, ou Scrum master ? IE est ce qu'on devient coach agile seulement en regardant les équipes qui fonctionnent en agile ?	7
Stratégie et tactique ?	7
Comment coacher un Scrum Master qui tombe rapidement dans la frustration (contexte inadapté, faible culture agile, passivité des équipes, ...)	8
Comment coacher un Scrum Master qui a un agenda personnel (le Cristiano Ronaldo qui marche sur les pieds des autres ?)	8
Comment on évite le multi-tasking qd on a des tâches qui peuvent durer des heures (genre une journée) en background (modifié)	8
Comment gère-t-on l'urgence ?	8

On a des flux différents suivant les tâches comment fait on pour le(s) visualiser avec 1 kanban?	9
Comment gère-t-on les dépendances? Comment peut-on avoir plus de visibilité?	9
Est-ce que l'agile c'est adapté aux gros projets?	11
Est-ce qu'on peut mettre en place des pratiques agiles dans une équipes où le chef d'équipe/manager n'est pas formé à l'agile/est réfractaire?	11
Comment on s'assure de la traçabilité, l'historique du projet?	12
commentaire	13
commentaire	13
Comment on assure la traçabilité? (réponse 2)	14
Est-ce qu'on doit/peut avoir des templates communs pour les livrables? c'est plus simple non si on travaille tous sur les mêmes documents?	15
Quelle (s) différence(s) vois-tu entre le Lean startup et l'Agile?	15
Qu'est ce que la loi de Conway peut apporter dans un programme de transformation, (et plus particulièrement pour les grandes structures qui ont adopté la démarche Follow-The-Sun)?	15
Quelle différence fondamentale existe t il entre un Stream Leader et un product owner.	16
Quelle différence fondamentale existe t il entre un Stream Leader et un product owner. (réponse 2)	18
Traditionnellement dans le cycle en V on mesure et on suit les risques. Comment fait on en mode Agile? D'ailleurs est ce qu'on gère les risques de façon independante en mode Agile?	19
I. Niveau produit	19
I.1 Cadrage - Ateliers risque	19
I.2 Cadrage - Estimation/Sizing d'équipe/Roadmap	19
II. Niveau programme	20
Autre réponse	20
Comment combiner des projets agiles et des projets non-agiles? (applications monolithiques dans un environnement "waterfall", la gouvernance de tous ces projets ensemble)?	22
Rappel schématique cycles itératifs et incrémentaux agiles	23
Rappel "waterfall", "cascade", "cycle en V"	23
Une autre vision du Waterfall	24

Le rôle du chef de projet et du scrummaster. Quelle différence?	24
Réponse 1	24
Réponse 2	24
Comment savoir qui peut prendre quoi comme décision?	25
Comment estimer la valeur d'une User story alors ... que j'entends dire depuis le début que par nature, les estimations sont toujours fausses.	25

1 PROBLEM Top 3 problems	4 SOLUTION Top 3 features	3 UNIQUE VALUE PROPOSITION Single, clear, compelling message that states why you are different and worth buying	9 UNFAIR ADVANTAGE Can't be easily copied or bought	2 CUSTOMER SEGMENTS Target customers
	8 KEY METRICS Key activities you measure		5 CHANNELS Path to customers	
7 COST STRUCTURE Customer Acquisition Costs Distributing Costs Hosting People, etc.		6 REVENUE STREAMS Revenue Model Lifetime Value Revenue Gross Margin		

Lean Canvas - cheminement

Le Lean Canvas?

Le lean canvas permet d'aligner l'ensemble des intervenants pour définir un problème, le segment d'utilisateur et la solution avec les features,... etc et tout cela en qlq heures max mais est ce que cela est suffisant pour définir la vision d'un produit d'une grande structure qui doit veiller sur sa réputation sur le marché et surtout comment convaincre les métiers de cela ?

(Pablo Pernot - février 2018) Plusieurs réponses à ta question. "tout cela en qlq heures max" : même pour une équipe autour d'un produit ce n'est sûrement pas quelques heures max. Un Lean Canvas se remplit avec du temps et au fil de l'eau. Et il se revoit constamment. D'ailleurs généralement on ne commence que par les premiers points.

On commence généralement par 1, 2 et 3. Après il y a des dynamiques entre les éléments du Lean Canvas

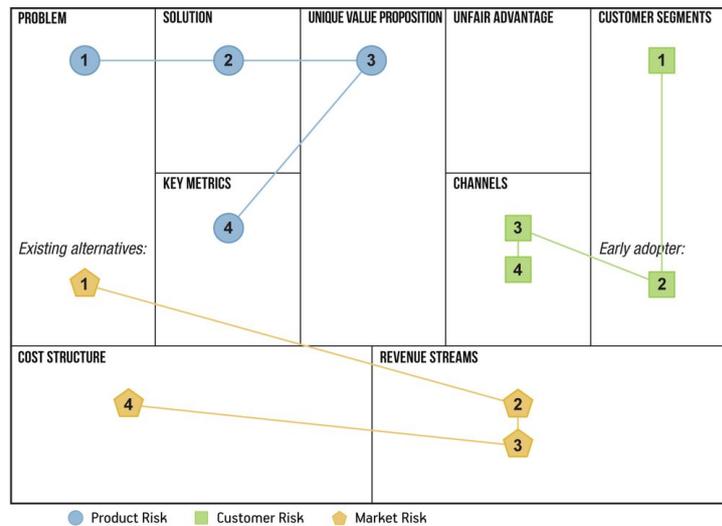
Un Lean Canvas cela se pense, se repense, se repense, se revoit, s'adapte, se clarifie, se communique, se communique, se communique, se repense, se revoit, se repense, se communique, se communique, etc.

Donc pas quelques heures.

Deuxième réponse. " vision d'un produit d'une grande structure" ... UN Lean Canvas c'est pour un produit. Pas pour la vision d'une grande structure. Ou alors tu réduis une grande structure à un seul produit. Et du coup pas la peine d'une grande structure. Moi j'imagine qu'une grande structure à une vision. des valeurs, potentiellement des principes. Et elle décline sa vision en produits. Qui eux mêmes peuvent commencer à envisager avoir un Lean Canvas Voilà mes réponses.

Scrum est-il éligible à tout ?

(Pablo Pernot - février 2018) Scrum est un cadre (et pas une méthode...) qui offre une façon de faire de l'agile. Il essaye de d'améliorer la prédictibilité en posant des contraintes : rituels / réunions avec des



Lean Canvas - dynamiques

éléments en entrée et en sortie clairs joués de façon cadencée (régulière, durée max, etc.). Il essaye d'améliorer les dynamiques en fournissant un bon équilibre de pouvoir entre le quoi, pourquoi et le comment (entre le PO et l'équipe). En une vingtaine de pages vous pourrez en savoir plus en lisant [le guide scrum](#). Mais est-il éligible à tout? Probablement pas. Des contextes qui ne le permettent pas oui. Mais c'est surtout beaucoup plus souvent les gens qui n'ont pas la rigueur de l'appliquer. Autre réponse sous-entendue : est-ce que l'agile est applicable à tous les contextes? Agile veut dire évoluer dans un monde complexe. C'est à dire incertain, en constante évolution. A vous de savoir si votre monde, votre contexte projet est incertain, en constante évolution. Facile : vous pouvez tout prévoir. Ne faites pas d'agile. Il y aura des choses inattendues, faites de l'agile (et pas forcément du scrum). Si quelqu'un peut tout prévoir qu'il me le dise, ça m'intéresse!!!

Comment faire de l'agile en Cobol?

(Pablo Pernot - février 2018). J'ai commencé par du Cobol, mais pas en agile c'est vrai. Agile c'est responsabiliser, rendre autonome, impliquer, mais aussi donc laisser un droit à l'erreur, et constamment essayer de s'améliorer, prioriser par valeur, découper en petits morceaux autonomes porteur de valeur, etc. (En lire plus [ici](#)). Tout cela est réalisable avec du Cobol. Pas forcément du "Scrum". Mais des cadres comme "Kanban" avec son management visuel sont tout à fait envisageables en Cobol. Faire du management visuel, des rétrospectives (amélioration continue), etc. sont des choses tout à fait faisables en Cobol et porteuses de valeur.

Comment faire quand le PO se désengage?

(Pablo Pernot - février 2018) C'est son projet, c'est son produit. Il devrait être fier de chaque avancée. Il devrait être impliqué, comme chacun de ses projets persos pour lequel il n'hésite pas à se donner les moyens de la réussite. Si le PO se désengage : arrêtez le projet. C'est qu'il n'est pas important, ou que le PO s'en fout, qu'il y a des choses plus importantes à faire. Bref : soit il s'en fout, soit ce n'est pas important,

et donc on ne se donne pas les moyens de réussir. Si vous préférez un projet raté plutôt que rien continuer, moi je préfère rien plutôt que de gaspiller un projet raté.

En 2009, Jean Tabaka (une jeune femme malheureusement décédée depuis) présentait une session qui avait fait date sur les 12 façons les plus communes d'échouer : [12 agile adoption failure modes](#). Façon 6 d'échouer : "None or too many Product Owners. Both cases look the same. Agile is yet another hat to wear and the person is already too busy. They check out and ask the team to just do Agile. Can't get past the 'this sucks' phase of adoption if the business is not bought in."

Le Scrummaster est-il un chef d'équipe ? Y-a-t-il une relation hiérarchique avec le reste de l'équipe ?

(Pablo Pernot - février 2018) Non le scrummaster n'est surtout pas un chef d'équipe. Non il n'y a surtout pas de lien hiérarchique entre le scrummaster et l'équipe. Surtout pas. Le scrummaster est un facilitateur, dont une des qualités essentielles sera d'être neutre. Il a besoin de cette neutralité pour bien faciliter. Impossible d'être neutre si on a un lien hiérarchique avec les personnes.

Concernant la hiérarchie, concernant le management, je vous invite à vous renseigner sur la posture de "host leadership". J'en parle rapidement [ici](#), ou sur [bel article de infoQ](#).

Est ce qu'un scrum master peut transmettre des félicitations aux équipes, félicitations qu'il a reçues par ailleurs ?

(Pablo Pernot - février 2018) C'est bien si il le fait mais, mais bon, c'est quand même une déviation, c'est encore mieux si il met en contact l'équipe avec la personne qui la félicite. Le scrummaster ne fait pas transiter les informations : surtout pas. Il met les gens en contact.

(Alexandre Thibault - février 2018) Le ScrumMaster, ce n'est pas un passe-plat d'info, d'ailleurs, j'ai eu cet anti-pattern il y a 4 ans et trop souvent les chefs espéraient que je passe les bonnes nouvelles. not good :slightly_smiling_face : on appelle ça la ScrumMum.

On nous demande de travailler sur un projet agile. Quand le projet s'arrête, on arrête l'agile ?

(Sabrina Palmieri - février 2018) Il ne faut pas réduire l'agilité à un simple processus de delivery, et se dire qu'en changeant la méthode on fera mieux, plus vite. Dans une transformation "agile", il n'y a pas de date de fin -> on change la philosophie, l'état d'esprit des organisations, et ça prend du temps. Changer le cadre méthodologique des projets peut être une première étape à cette transformation, mais ça ne s'arrête pas là. L'impact positif de l'agilité sur la chaîne de valeur d'une organisation c'est quand on essaye, on teste, on échoue, on apprend, on recommence et on s'adapte, en se focalisant sur la valeur produite. Donc non, lorsque le projet s'arrête, l'agile ne s'arrête pas.

- https://fr.wikipedia.org/wiki/Manifeste_agile
- <http://www.andwhatif.fr/2017/07/28/ne-reduisez-lagilite-a-ladaptation/>
- <https://www.areyouagile.com/2017/07/etre-agile/>

Dans une organisation en mode spotify avec des tribus et des squads, la tribu doit elle faire une rétrospective en fin d'itération? Qui y participe : les scrums master et po des squads? les équipes de toutes les squads?

(Pablo Pernot - février 2018) La tribu, c'est à dire un ensemble d'équipes autour d'une thématique donnée, devrait-elle faire des rétrospectives? C'est à dire de l'amélioration continue. Oui certainement. Probablement pas à toutes les itérations mais disons une fois tous les trois mois ou six mois, à chaque "incrément" (ensemble d'itérations faisant sens, et ne durant pas trop longtemps // gestion du risque). Qui participe? Plusieurs stratégies. Soit effectivement "juste" certaines personnes qui représentent l'équipe (ces personnes peuvent changer). Soit tout le monde pour que tout le monde se croise (c'est une bonne dynamique). Mais des fois c'est long et c'est compliqué. Selon la stratégie c'est d'ailleurs plus ou moins fréquent. Il y a aussi un besoin de synchronisation régulière des communautés de pratiques, des "guildes", guilde des architectes, guilde des scrummasters, etc. Cela peut être plus fréquent. Toujours penser qu'au delà de 7/8 personnes la réunion prend une autre tournure, une autre dynamique. Donc rétrospective : les équipes ou leurs représentants. Et synchronisation, amélioration de la pratique : les communautés de pratiques, les guildes.

Question : je découvre l'agilité, il paraît que c'est top presque miraculeux. Comment je peux commencer dans mon quotidien à faire de « l'agile »? Q1 : Y a t'il des pratiques simples à mettre en oeuvre? Q2 : Je peux demander de l'aide, à qui?

(Alexandre Thibault - février 2018) Q1 : Je commencerais par mettre en place la rétrospective d'équipe à intervalle régulier (2 semaines). Et puisqu'on veut de l'action et une amélioration continue, prévoir des boîtes de temps dans les 2 semaines qui suivent pour réaliser les actions déterminées lors de la rétro précédente. Pour s'inspirer sur les formats et la structure d'un tel exercice, voici ma référence accessible sur [google drive](#). Si vous n'avez pas accès à l'outil google drive, me demander en MP, et je vous enverrai la version PDF (mais vous n'aurez pas la dernière version). Le format le plus simple et qui est apprécié par expérience, c'est le [lean coffee](#).

Q2 : les coachs et gens qui ont démarré une réflexion personnelle sur l'agilité et qui travaillent dans ton contexte, et/ou chercher à faire partie de communautés (slack, meetups, conférences). Personnellement, je modère une communauté sur slack qui s'appelle enjoy agile, m'envoyer une demande perso pour recevoir une invitation.

(Pablo Pernot - février 2018) Miraculeux? Je me méfierais du breuvage miracle du Docteur Mabuse. Si c'est miraculeux c'est probablement du charlatanisme. Commencer à être agile dans son quotidien? Le mieux est de commencer sans même parler cadres ou méthodes (Scrum, Kanban, etc...). Mais plutôt de mettre en évidence deux points clefs : la création de valeur assez rapidement, et l'amélioration continue. La création de valeur rapidement? Obligez vous à délivrer en production et à mesurer l'impact de ce que vous délivrez tous les trois mois maximum, (tous les deux mois ou tous les mois si c'est possible et si cela fait sens dans votre contexte). C'est simple à dire, pas simple à faire. Délivrer régulièrement et mesurer. Deuxième point : tous les mois faite une séance d'amélioration continue : interrogez vous sur ce qui a bien marché le mois précédent, ce qui a mal marché. Choisissez un point clef sur lequel travailler pour le mois suivant. Là aussi c'est simple à dire, mais pas si simple à réaliser. Prenez ces deux pistes comme des choses qui demandent de l'entraînement. Cela se suit régulièrement.

A qui demander de l'aide? Comme le dit très bien Alexandre : ayez des conversations avec les coachs et gens autour de vous qui ont démarré une réflexion personnelle sur l'agilité et/ou chercher à faire partie de communautés (slack, meetups, conférences).

Comment devient-on coach agile?

(Pablo Pernot mars 2018) Facile. On écrit un blog et on indique dessus que l'on est "coach agile". Ou on change son intitulé dans LinkedIn. Comment devient-on un bon coach agile? Je ne sais pas. Une chose est sûre : les certifications font du mal et pas du bien. Une autre chose est sûre : cela doit impliquer beaucoup d'introspection et une compréhension du monde incertain qui nous entoure. Il n'y a pas de parcours typique.

Est ce que lorsqu'on a travaillé dans une petite entreprise (4-5 personnes) c'est une expérience Agile?

(Pablo Pernot mars 2018) Absolument pas. On peut très bien proposer beaucoup de "command & control", et un état d'esprit à l'antithèse d'une culture agile (imposer, croire à la prédictibilité forte du monde, croire aux processus plutôt qu'aux interactions, etc.) dans une petite entreprise de 4,5 personnes... comme dans une équipe de 4,5 personnes. La taille du groupe aide à sa dynamique, pas à en faire une expérience agile.

Est ce qu'on peut être coach agile sans être passé Po, ou Scrum master? IE est ce qu'on devient coach agile seulement en regardant les équipes qui fonctionnent en agile?

(Pablo Pernot mars 2018) Cela peut-être un plus d'avoir été PO ou Scrummaster, je crois que la grande majorité des coachs agiles l'ont été. On en devient pas en regardant, on devient en étant, en faisant. Attention cependant par faiseur j'entends : on incarne le rôle, on incarne et on explique l'attitude et la culture, en prenant des risques (protéger les équipes, passer des messages au management, aux métiers, etc.), en se vautrant dans la gadoue, en faisant des erreurs (pas en faisant à leur place). Et comme dit plus haut : ayez des conversations avec les coachs et gens autour de vous qui ont démarré une réflexion personnelle sur l'agilité et/ou chercher à faire partie de communautés (slack, meetups, conférences).

Stratégie et tactique?

Au fil de l'eau une équipe agile commence à faire face à des obstacles et les rendre visible (Scrumboard par exemple) pour les attaquer par la suite, ma question : Sur des projets de grande envergure, quelle est la personne la plus à même et qui a suffisamment de recul pour faire une analyse, une lecture diagonale de ces obstacles pour décorrélérer ceux qui relèvent de la mécanique/dynamique (le terrain) et ceux à fibre stratégique / organisationnelle? (J'ai travaillé dans une config 19H/24H (Follow the Sun), croyez moi le clivage est tellement tacite et difficile à "spotter")?

(Mars 2018, pablo pernot) Je ne sais pas. La stratégie et la tactique sont autant des choses qui peuvent appartenir au métier (Product owner, product manager), qu'à d'autres personnes. L'équipe peut avoir une

stratégie et une tactique. Si il s'agit de prendre du recul pour penser l'organisation l'erreur serait à mes yeux que cela repose justement sur une seule personne. C'est plutôt une vision partagée qui selon les feedback évolue. Après sur la stratégie quelqu'un doit "trancher", choisir (et naturellement pouvoir changer d'avis). Enfin je ne sais pas si il faut autant vouloir séparer les deux : j'aime l'idée défendu par Dirk Gently (le détective holistique) : "tout est connecté", ou comme le disait le Lean : il faut "voir le tout". J'ai l'impression que ta question appelle une réponse précise que tu as en tête. Mais tous les contextes sont différents.

Comment coacher un Scrum Master qui tombe rapidement dans la frustration (contexte inadapté, faible culture agile, passivité des équipes, ...)

(Mars 2018, pablo pernot) Lui souhaiter la bienvenue dans ce métier difficile de la facilitation et du coaching. Que cela est illusoire de vouloir amener les organisations où elles ne souhaitent pas aller (les organisations ou les personnes : [convaincre?](#) Et que le [métier est difficile](#). Et repartir sur tous les fondamentaux...

La frustration c'est bien cela démontre une envie. Trop de frustration c'est mal, cela éteint le sentiment de progrès nécessaire à une réelle motivation.

Comment coacher un Scrum Master qui a un agenda personnel (le Cristiano Ronaldo qui marche sur les pieds des autres?)

(Mars 2018, pablo pernot) Si son agenda personnel va dans le sens de la valeur pour les gens et l'organisation avec qui il est tant mieux. Si cela va au détriment des gens et de l'organisation, lui dire. Si il veut être coacher pour aller dans le bon sens, le coacher, si il ne veut pas, ne pas le coacher et lui dire pourquoi (il a de fortes chances d'échouer au passage). Mais il est intéressant avec lui de creuser par des questions le pourquoi de cette posture, le pourquoi de ses actions, les résultats obtenus, etc. Il pourrait être surpris. Le coach aussi d'ailleurs.

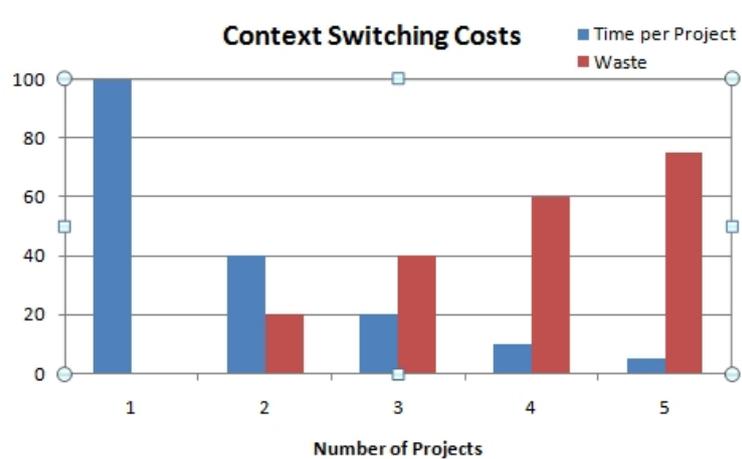
Comment on évite le multi-tasking qd on a des tâches qui peuvent durer des heures (genre une journée) en background (modifié)

(Mars 2018, benoit ricaud) En acceptant que nous sommes mono-tâche : je termine ce que je suis en train de faire, avant de passer à autre chose, je ne laisse rien en suspend ==> stop starting, start finishing.

(Mars 2018, pablo pernot) Oui, en témoigne cet article de INFOQ : <https://www.infoq.com/articles/multitasking-problems>. Il montre que le coût du changement de contexte est néfaste dès 3 choses que l'on réalise en parallèle.

Comment gère-t-on l'urgence?

(Mars 2018, sabrina palmieri) En accueillant le changement, l'équipe doit être capable de s'adapter. Quel degrés d'urgence? On change les priorités, ou on attend le prochain sprint si possible. Dans le manage-



Context Switching

ment visuel tu peux aussi ajouter une ligne urgence tout en haut, en limitant leur nombre parce que si tout devient urgent alors plus rien ne l'est. Et si une urgence n'est pas traitée ou ralentie à cause de la méthode, c'est que le problème vient de la méthode, pas de l'urgence. Promesse de l'agile = avoir des meilleurs résultats dans un monde complexe, incertain, donc on sait gérer les urgences.

On a des flux différents suivant les tâches comment fait on pour le(s) visualiser avec 1 kanban?

(Mars 2018, pablo pernot) Et bien tu n'utilises pas d'outils électroniques car ils ne savent pas faire. Mais sur un système kanban normal (management visuel et cognitif car physique) tu adaptes facilement ton système. Cf cet exemple de Kanban RH (la ligne sourcing). <https://www.areyouagile.com/2017/05/les-limites-dans-kanban/>. Attention si c'est trop morcelé cela veut dire que ton système ("ta vraie vie") est trop compliquée pour être bien géré, ou que tu devrais scinder les systèmes kanban (le management visuel). Suis-je clair?

Comment gère-t-on les dépendances? Comment peut-on avoir plus de visibilité?

(Mars 2018, pablo pernot) A priori tu évoques les dépendances entre équipes sur les fonctionnalités et les "user stories". Comment "voir le tout", ou "tout est connecté" de Dirk Gently (même si nous sommes d'accord que les dépendances sont toujours un frein et qu'il faut les limiter au maximum). Visuellement ce n'est pas simple. Si on possède un Portfolio Kanban (<https://www.areyouagile.com/2016/01/portfolio-projets-kanban-partie-1/>) on peut l'indiquer sur les cartes kanban.

Si il s'agit de se projeter dans un ensemble d'équipes cela peut se discuter "plus facilement" avec un plan de release agrégé de ces équipes. Mais ce n'est pas visuel, c'est dans le savoir de chacun. Je m'explique avec un schéma. Mais des postulats : On n'agrège pas plus de 5-6 équipes, après il y a trop de dépendances, plus rien n'avance. Si on souhaite des programmes avec 30 équipes (par exemple), on le divise avec groupes de 5-6 équipes max. Si toutes les équipes ont des dépendances avec toutes les équipes : c'est mort, c'est



Kanban

Titre Peetic, lotz, réalisation de la boutique en ligne		Version	Type : Legal, Innovation, Stratégique, Evolution, Tactique	S
Description Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore		Briques métiers/techniques clefs Lorem ipsum		
ACTEURS John (Po), Paul, Robert, Jimmy (Team) Kurt, Andrew Sponsors Lorem ipsum...	NB 4	Effort ★ ★ ☆ ☆ ☆		
	Valeur ★ ★ ★ ☆ ☆			
Dépendances Lorem ipsum dolor sit amet		Deadline officielle 26/01/16	Coût estimé 120k	
IDEES 21/06/15	SELECTION 12/09/15	ETUDE/PREP 19/09/15	REALISATION 19/11/15	VALIDATION
PACKAGING		DEPLOIEMENT		LIVE/PROD

Dépendances sur carte kanban

foutu. Donc pas le choix : on commence le programme avec uniquement 5-6 équipes, et quand on a dénoué les dépendances on fait grossir les effectifs si besoin.

Donc soit un plan de release :

Si tu agrèges ces plans de release (élément important à mes yeux de communication), les gens sauront assez les lier entre eux. Et du coup les "inutile de démarrer ça tant que" vont apparaître.

Tu as aussi l'option du mur des "quarterly planning", "pi planning", qui fonctionne un peu pareil. Le "bordel" engendré par la pelote de fils qui indiquent les dépendances envoie un signal très clair...



Release plan

Est-ce que l'agile c'est adapté aux gros projets ?

(Mars 2018, Pablo Pernot). Non seulement il est adapté, mais il est fortement conseillé. Cherchez les statistiques sur internet si vous voulez (cependant on fait dire ce que l'on veut aux statistiques) mais tous les gros projets qui ne sont pas fait en agile échouent de plus en plus. Il faut d'abord comprendre que agile est la meilleure façon de gérer le risque. Exemple : si je commence par ce qui a le plus de valeur pour moi, et que je suis capable de découper les choses en petits morceaux, et que je suis capable de les mettre en production (je n'ai pas dit que c'était facile, mais vous voulez avoir de bons résultats?). Alors ma gestion du risque est optimale. Pourquoi? Parce que si je commence par de petits morceaux ordonnés par valeur, délivrés en production et que a) ça marche alors je touche très rapidement des bénéfices attendus, je valide mes hypothèses par du concret et du retour sur investissement. Je fais disparaître les principaux risques. Ou b) cela ne marche pas. Et vous avez réduit votre investissement au minimum pour découvrir que vos hypothèses sont fausses (la technologie n'est pas la bonne, le client n'est pas intéressé, etc.). Vous avez réduit au maximum le risque. Dans les gros projets sans cette approche vous maximisez les risques : vous attendez trop tard, vous investissez trop avant de savoir si vos hypothèses sont les bonnes ou pas.

Est-ce qu'on peut mettre en place des pratiques agiles dans une équipes où le chef d'équipe/manager n'est pas formé à l'agile/est réfractaire ?

(Mars 2018, Mostafa Elaibi) Bien sûr que OUI. Les pratiques Agiles sont multiples et charge à chaque équipe et chaque membre dans une équipe à pousser la pratique qui pourra améliorer la qualité de ce que l'équipe produit. Si je prend par exemple les pratiques craftsmanship, voici une slide qui présente les différents niveaux de pratique. Dedans toute la partie Dev et équipe, est à mon sens indépendante du manager (agile ou pas agile). Un management visuel ou un daily meeting peut être poussé par l'équipe même sans l'accord du manager (du moment que ça aide à fluidifier la communication). Ainsi que d'autres pratiques agiles. Ceci dit, un manager réfractaire doit être considéré par l'équipe comme « FEEDBACK ». La question à se poser : pourquoi il est réfractaire? en quoi on peut l'aider à mieux appréhender l'agile et les pratiques agiles...



Release plan

Comment on s'assure de la traçabilité, l'historique du projet ?

(Mars 2018, Pablo Pernot) Ah ah ! Quand j'entends cette question je m'interroge : pourquoi souhaite-t-on de la traçabilité et de l'historique... si ce n'est pour ... blâmer. Cela ne sert à rien de tracer les événements du projet, son historique. Il ne s'agit que de mesures pour que des personnes s'affrontent entre elles (contractualisation, qui est le coupable?). (Naturellement il est clair qu'on ne parle pas d'une fonctionnalité de traçabilité dans l'outil comme un journal -log- ou d'une fonctionnalité d'historisation de documents par hasard). Est-ce que c'est utile de passer du temps sur des mesures pour que des personnes s'affrontent? A vous de juger. La deuxième interrogation que je lis dans cette question c'est : comment je vais savoir comment le projet fonctionne. On parle de documentation. Parenthèse : c'est bien comment le projet fonctionne, et pas comment il a été bâti. Il est complètement inutile de savoir comment il a été bâti : aucun projet n'est le même, ne se déroulera de la même façon. Donc la véritable question qui aurait du sens c'est comment je sais comment mon produit, ma fonctionnalité, fonctionne? On parle de documentation (et pas de spécification, ou d'un journal sur comment le projet s'est déroulé, qui sont je le répète des objectifs vains voir nocifs). La documentation idéale en agile c'est : le produit parle par lui même : qui a été lire une documentation de Firefox? ou de Tinder? Sinon on peut envisager plein de documentations intelligentes comme : les tests automatisés métiers expliquent comment le produit fonctionne. Sur la documentation il y a un mythe qui voudrait qu'il n'y a pas de documentation en agile. C'est faux. C'est



Quarterly planning board

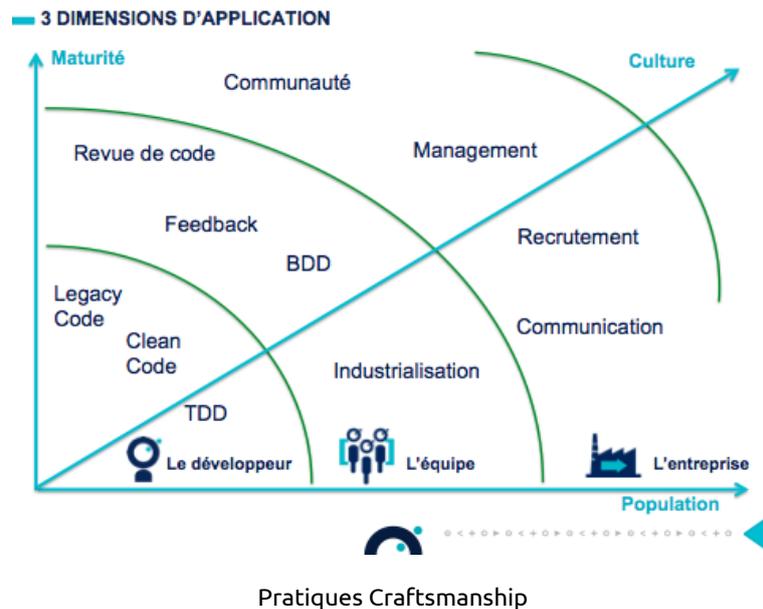
toujours la même question : a-t-elle de la valeur ? Quelle documentation ? Et ainsi, comme toujours selon le contexte vous aurez à vous interroger sur quelle forme de documentation a de la valeur dans votre cadre. Et aussi comment mesurer que cette valeur est bien au rendez vous (la documentation est-elle consultée par exemple ?).

commentaire

Merci Pablo pour cette réflexion. Pour voir les choses d'un angle positif, je pense que l'historique ou traçabilité n'est pas nécessairement utilisé pour blâmer ... l'historique ou la doc projet peuvent servir à la prise de relève, embarquer les nouvelles ressources rapidement... Pour Tinder et firefox ... personnellement en tant qu'utilisateur final, je n'ai pas besoin de ni doc sur comment on les a fait ni de doc sur comment ça marchent. Un produit doit être, à mon avis, simple et intuitif. Par contre il serait peut être intéressant de demander aux équipes en charge de la maintenance ou des upgrades de ces outils de quoi elles ont besoin pour accomplir leurs missions.

commentaire

Merci. Je pense que nous sommes en phase, je vais juste préciser certaines choses. Je pense que tu parles de "Documentation". C'est à dire quand tu as besoin d'information comme pour les exemples que tu cites : prise de relève, intégration de nouvelles ressources. Tu évoques donc probablement une documentation métiers (l'explication des algorithmes par exemple, idéalement au travers de tests automatisés : ils sont joués constamment donc constamment tenus à jour – le problème de la documentation –) ou une documentation technique (idéalement encore une fois : au travers de tests unitaires, etc.) ou encore d'autres documentation. Je voudrais juste ajouter : il faut vérifier constamment que cette documentation a de la valeur : c'est à dire qu'elle est consultée et utilisée : QUI mesure les accès à la documentation ? Faites le : et supprimez ou repensez 90% de votre documentation. Autre point clef : d'accord avec toi @MMZN mais jouons à un petit jeu : supprimons TOUS LES NOMS de cette "traçabilité", "historisation". Puisqu'on ne



cherche pas de coupable, puisque l'on veut de la reversabilité, de la documentation : inutile de savoir QUI et QUAND. Cela n'a pas de sens. On veut juste savoir QUOI et POURQUOI. Dans ces deux cas je te suis à 200% : a) on ne cite personne car cela n'est pas le but b) on mesure constamment que la documentation (ou la traçabilité et historisation pour reprendre tes mots) soient utiles (on mesure on n'écoute pas les gens qui donnent des réponses politiquement correctes) et le cas échéant on supprime ou on pense différemment cette documentation.

PS : réponse complémentaire : si on a besoin de savoir "qui" pour avoir plus de précision : facile : on se rend au milieu de l'équipe ou on donne un coup de fil.

Comment on assure la traçabilité? (réponse 2)

(Mars 2018, Benoit Ricaud) Si on s'en tient à la définition : la traçabilité est la possibilité d'identifier l'origine (pas forcément l'identification d'un membre de l'équipe mais surtout la détermination d'un point de départ) et de reconstituer le parcours d'un produit ou d'une feature d'un point A à un point B. (Ex : d'un ToDo au Done). Le mot peut faire peur car on peut s'en servir à des fins défavorables (et certains n'hésiteront pas..). Un kanban (visuel ou électronique) me semble un bon outil pour le suivi et par extension pour la traçabilité. Grâce à cela, l'équipe pourra obtenir des métriques sur le projet (ex : un lead time) permettant de construire des outils (ex : un CFD) connaître ses dysfonctionnements, dans l'objectif de s'améliorer, de déterminer sa vélocité (via un historique), et d'être de plus en plus prédictif.

et

Comment on "s'assure" de la traçabilité? En portant la responsabilité au niveau de l'équipe. C'est à l'équipe de s'assurer du bon maintien de son board pour donner et se donner de la visibilité.

Est-ce qu'on doit/peut avoir des templates communs pour les livrables? c'est plus simple non si on travaille tous sur les mêmes documents?

(Mars 2018, Pablo Pernot). Faites ce que vous pensez être bien devrait être la règle. Si vous pensez que c'est bien d'avoir tous les mêmes templates faites le. La règle c'est plutôt que vous vous interrogiez régulièrement sur le fait de savoir si c'est bien, si c'est utile, si cela apporte de la valeur, et si vous changez d'avis que vous puissiez changer.

Quelle(s) différence(s) vois-tu entre le Lean startup et l'Agile?

(Avril 2018, Pablo Pernot). Agile est une compréhension et une réponse au monde moderne qui subit de constants changements difficilement prédictibles, l'agile est une philosophie, un état d'esprit, une culture, pour répondre avec les meilleurs résultats (disent ceux qui y croient) à ce monde complexe. Parmi les façons de faire, ou plutôt les cadres proposés chacun **répond** avec le même *mindset*, avec la même culture, à des contextes différents. (Je caricature un peu dans les affirmations suivantes) Comment est-ce que l'on fabrique du code dans un environnement agile? Extreme Programming, Software Craftmanship. Comment est-ce que l'on aborde, gère des projets ou des produits en environnement complexe? Scrum. Comment est-ce que l'on gère des flux dans un monde complexe? Kanban. Comment est-ce que l'on pense des produits et un positionnement marché dans un monde complexe? Lean Startup, Design Thinking. Lean Startup est ainsi une façon de penser son produit et son émergence dans un monde complexe, dans une culture agile. C'est un sous ensemble. L'auteur a donné le nom de Lean pour faire plus sérieux, plus lire "industriel" (le piège) : lire [les cowboys hippies](#) qui traitait exactement de cette question. Il faut savoir que initialement il devait s'appeler "agile startup". Une approche des "mouvements" agiles.

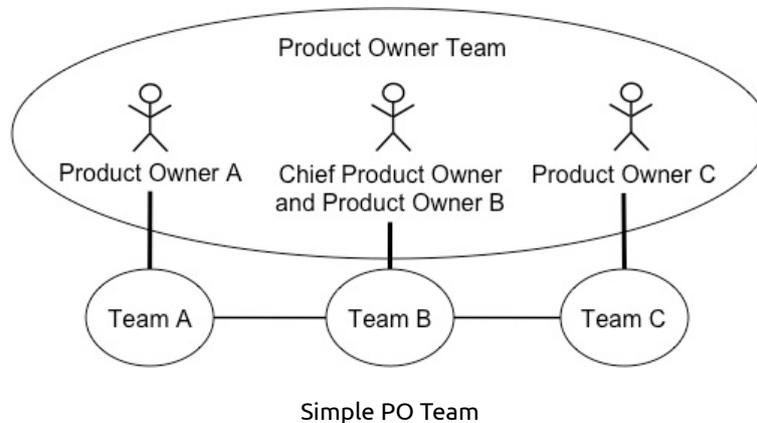
- <https://www.areyouagile.com/2014/11/les-cowboys-hippies/>
- <https://www.areyouagile.com/2015/08/mouvement-agile/>

Qu'est ce que la loi de Conway peut apporter dans un programme de transformation, (et plus particulièrement pour les grandes structures qui ont adopté la démarche Follow-The-Sun)?

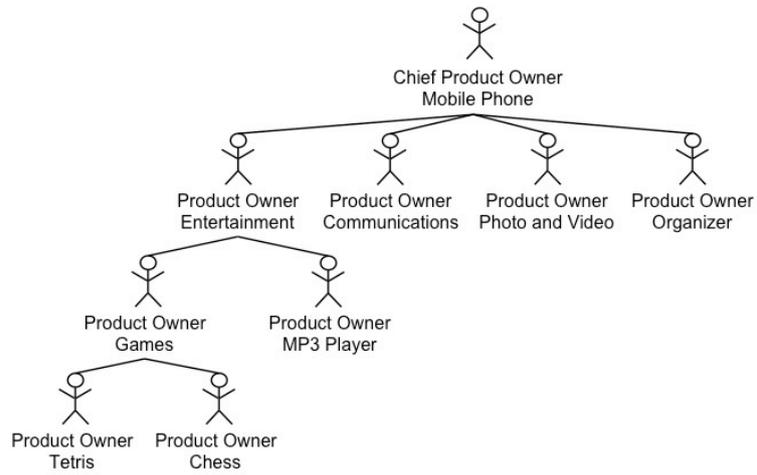
(Avril 2018, Pablo Pernot) La loi de Conway est passionnante. Elle dit *grosso modo* que la structure de votre produit (structure des composants), est la copie de votre structure organisationnelle. Si vous avez une équipe dédiée sur le paiement, vous aurez probablement dans votre produit un module dédié paiement. Je ne sais pas si c'est bien ou si c'est mal c'est à vous de choisir. Je ne sais pas exactement qu'elle est la question : programme de transformation c'est un moyen sûrement pas une finalité. (Et je ne sais pas ce qu'attend la question concernant la démarche follow-the-sun). En tous cas pour revenir à Conway, l'idée sous-jacente c'est peut-être que bien souvent dans un cadre agile on cherche des équipes pluridisciplinaires capables de délivrer des choses de façon autonomes (pour plein de raisons hors de cette question). Ainsi bien souvent il convient de revoir les organisations, et de les re-agréger autour d'éléments fonctionnels métiers. L'organisation épouse le besoin métier. L'organisation se reconstitue autour du sens métier. (D'où aussi les mouvements plus de conceptions techniques Domain Driven Design dont je n'ai pas la compétence pour les évoquer plus). Je ne sais pas si cela répond à la question mais je crois que c'est l'idée de fond.

Quelle différence fondamentale existe t il entre un Stream Leader et un product owner.

(Avril 2018, Pablo Pernot) Kezaco un Stream Leader? J'imagine que c'est un responsable d'une section du produit? On va attaquer la question différemment : On a besoin d'une personne qui incarne le désir de ce produit : il en explique le *pourquoi* (la valeur, le sens), et le *quoi* (comment cela se décline fonctionnellement). Je veux une boutique en ligne pour... (pourquoi) et donc je veux une gestion de panier comme ceci (quoi). Il travaille avec une équipe qui se charge du *comment* elle fabrique le *quoi*. Cette personne c'est le *product owner* : qui a un rôle de stratège (ma vision, comment je l'aborde), de tacticien (mise en oeuvre, mon quoi, de ma vision), et de facilitateur (expliquer, accompagner, faire comprendre, réunir, etc) de son produit. Après si le produit grossit on a peut-être besoin de plusieurs *product owner*. Car un *product owner* = une équipe. Dans ce cas apparaît le *chief product owner* ou le *product manager* ou le *meta product owner* (faites vous plaisir). Son rôle est de synchroniser une macro/meta vision qui rassemble ses *product owners*. Tiens je retrouve de vieilles images (simple PO team, complex PO Team). Alors le Stream Leader? Je lis leader il doit être le manager de l'équipe ou des équipes rattachées à un domaine métier. Par exemple le domaine ABRACADABRA possède 4 équipes, donc 4 PO avec 4 scrummasters, et 4 équipes. Il y a peut-être un "meta-po" (rôle peut-être tenu par l'un des PO). Et il y a un manager qui encadre toutes ces personnes, ou peut-être juste les membres des équipes selon le découpage de l'organigramme. Son rôle devrait être de faciliter le travailler de toutes ces personnes : sont-elles bien formées? Ont-elles les bons outils de travail? Sont-elles assez autonomes? Ont-elles assez de liberté pour s'impliquer? etc etc (et sûrement pas leur dire comment faire, ce n'est pas le rôle d'un leader ou d'un manager moderne). Le Leader doit s'assurer que le cadre dans lequel s'épanouit ses équipes est bon. Le *product owner* doit être fier de ce que tous ensemble ils fabriquent.



Ah , si on mélange ces deux rôles en une personne, c'est souvent la m***e.



Complex PO Team

Quelle différence fondamentale existe t il entre un Stream Leader et un product owner. (réponse 2)

(Avril 2018, Benoit Ricaud)

Product Owner vs Stream Leader - 1ère partie - Gestion du Backlog

Une des différences les plus impactantes entre un SL et un product owner est sur l'aspect décisionnel de son "produit" (on réfléchira sur la notion de "produit" dans une autre partie). Un des rôles du product owner est de pouvoir négocier, prendre des *décisions* sur son backlog en fonction de sa cible (utilisateurs finaux, marché, etc...) selon un mix optimisé time to market / valeur.

Un Product Owner choisit ses stories, un SL dispose d'une liste de tâches. Il semble qu'un SL ne peut définir ou même *prioriser* son backlog comme il le souhaite, il est figé à des deadlines et sans le respect de celles-ci, ainsi que sans la contribution de d'autres équipes liées, aucune valeur pour l'utilisateur ne peut en découler.

Exemple : Je suis un SL d'une filière, je suis entre autre responsable d'un certains nombre d'interfaces, ces interfaces dépendent de de la contribution de mon équipe mais également de la contribution de d'autres filières dont je n'ai pas la main. Pourtant ceci est dans mon scope de responsabilités car c'est ma filière qui contribue le plus à ces interfaces. (?) Qu'est ce qu'une interface? Un ensemble d'appels. Peut-on considérer une interface comme une API as a product? Peut être.. Et nous sommes déjà dans un cadre de responsabilités ou le SL aurait à minima deux produits et plusieurs équipes dont certaines non maîtrisées (+ l'équipe de développement). Rappelons nous qu' 1 PO = 1 équipe = 1 produit.

Traditionnellement dans le cycle en V on mesure et on suit les risques. Comment fait on en mode Agile? D'ailleurs est ce qu'on gère les risques de façon indépendante en mode Agile?

(Avril 2018, Benoit Ricaud)

Hello on fait les 2, le suivi des risques fait partie intégrante d'un projet agile tant au niveau produit que programme.

I. Niveau produit

Un des objectifs d'un cadrage est d'identifier les risques en amont du build afin de les suivre en continu.

I.1 Cadrage - Ateliers risque

Généralement, un atelier risque se déroule de la façon suivante :

I.1.1. Identification des risques On identifie principalement 4 typologies de risques : Business, people, schedule et technique lors d'un atelier de brainstorming

I.1.2. Classement des risques On classe ensuite ces risques en fonction de l'impact et de la probabilité

I.1.3. Définition des solutions de mitigation L'équipe trouve ensemble des solutions de mitigations qu'elle peut classer tel quel

- risque résolu : il n'y a plus de risque - c'est réglé
- risque géré : on a déterminé la procédure à faire si le risque arrive
- risque accepté : on ne peut rien faire, on accepte le risque et on fera avec s'il arrive
- risque porté : on a trouvé un porteur du risque (soit pour déterminer le flow à réaliser et qu'il devienne un risque géré soit pour identifier la personne qui prendra en charge le risque s'il survient)

Le suivi des risques se fait en continu sur un projet lors des comités de suivi (généralement toutes les 2 semaines sur un projet scrum "classique")

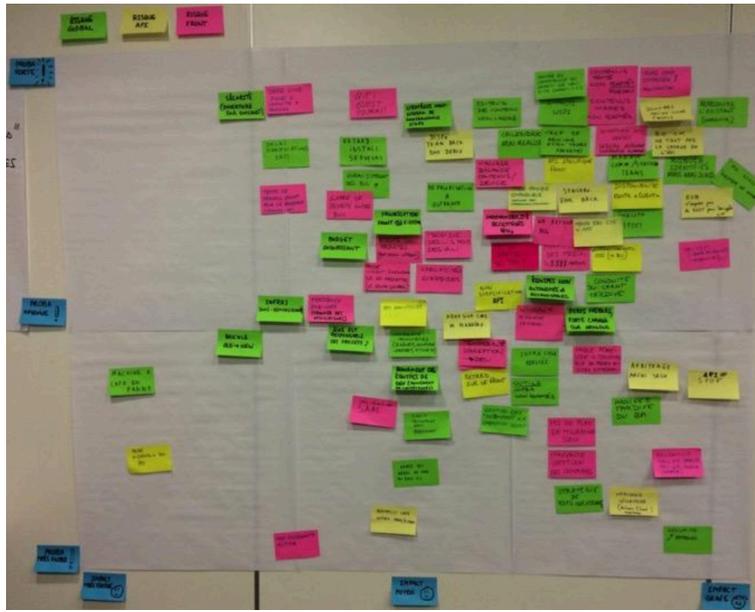
I.2 Cadrage - Estimation/Sizing d'équipe/Roadmap

Lors d'un cadrage agile, on fait généralement un rapport entre l'estimation de l'ensemble des Epics évaluées et la vélocité désirée pour atteindre l'objectif. En fonction de cette vélocité, on adaptera les solutions palliatives si nécessaire dans les instances adéquates (cosui ou copil)

Important, il faut plusieurs sprints (généralement 3 à 4) pour connaître la réelle capacité à faire d'une équipe agile et commencer à être prédictif.

II. Niveau programme

Les release planning days ont une partie dédiée à la gestion des risques au niveau de l'incrément. L'atelier peut suivre le même format que l'atelier risque de cadrage (avec un step au dessus) En complément, on révisera en RPD la roadmap programme en fonction des changements d'items au backlog des équipes (en prenant compte des dépendances) évaluée en fonction de la capacité à faire des équipes.



Risque 1 ; réponse Benoit

Autre réponse

(Juin 2018, Pablo Pernot)

Ma compréhension est très différente. D'une part je suis un grand défenseur du "maximiser la valeur et minimiser les efforts" (de Jeff Patton). Dans cette gestion du risque tel que vous l'énoncez je vois juste une perte de temps, comme les gros backlog que l'on doit constamment nettoyer aménager parce que l'on se refuse à rester simple et efficace, et que l'on garde une liste sans fin. Qui c'est sûr génère du travail, donc j'imagine un alibi, quand à savoir si cela a de la valeur... Donc ne gérons pas des risques inutiles, ne gérons pas des risques qui sont intellectuellement plaisant mais dont la gestion n'amènera rien. Evitons toutes ces réunions évoquées. Evitons toutes ces listes évoquées. Vous ne vivez qu'une fois bon sang, faites des choses avec de l'impact.

Cela ne veut pas dire qu'il n'y a pas de gestion des risques en agile.

Agile = the best risk management *strategy* currently known for Software development –
Vasco Duarte

Pourquoi est-ce que Vasco écrit ceci? Avec une pensée agile on va prioriser par valeur et découper en petits morceaux. Puis, par exemple, on va se lancer dans un cadre "Scrum". Alors je reprends les risques présentés par la précédente réponse :



Risque 2 ; réponse Benoit

“Disponibilité équipe client”, “Certaines applis BO ne sont pas accessibles”, “Synchro evol back”, “Arbitrage archi/secu”, “B.O CM ne tient pas la charge”.

“Disponibilité équipe client”? Cela va remonter lors de la rétrospective. Cela va devenir évident lors de la rétrospective, lors de la phase d’amélioration continue. En fait cela va remonter dès le début du projet. De par les règles même du jeu agile dont la collaboration est une donnée clef. C’est possible de le régler : faisons le, ce n’est pas possible : trouvons une solution. Toujours en rétrospective.

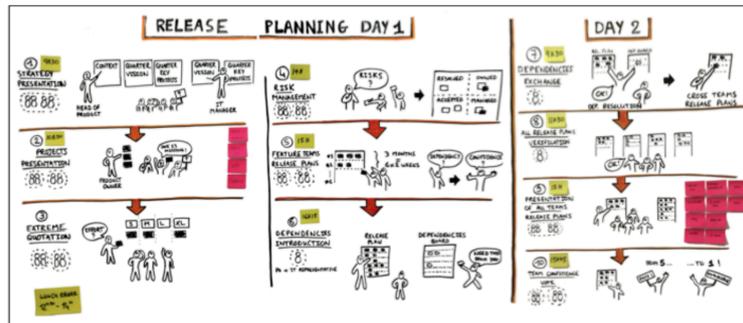
“Certaines applis BO ne sont pas accessibles” : idem que plus haut.

“Synchro evol back” : idem que plus haut.

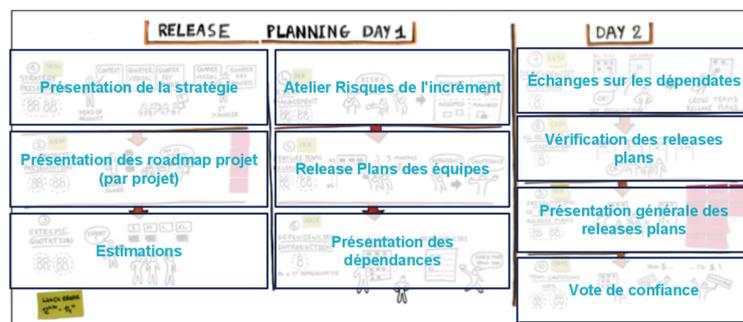
“Arbitrage archi/secu” : rien de mieux que d’avoir un logiciel qui fonctionne pour montrer ce qui marche ou ce qui ne marche pas. “Aucun plan ne survit au premier contact l’ennemi”. Si la question est importante, c’est qu’elle a de la valeur (business, apprentissage, etc.). Ainsi probablement les éléments du backlog qui portent cette question ont une forte priorisation, et ainsi la résolution de cette interrogation sera rapide : les éléments vont vite être traités. L’arbitrage deviendra très vite concret : on va délivrer du code le mettant en oeuvre et on aura une réponse bien plus adéquat à apporter sur une base concrète. Si ce n’est pas important, autant ne pas traiter.

“BO CM ne tient pas la charge” : Ici on évoque la **définition de fini**. Dans un cadre agile on développe des choses finies qui fonctionnent pour le contexte cible. C’est à dire on développe dès le début dans des environnements simulants les conditions de production par exemple. Si la charge apparaît comme un souci : très bien, on le découvre **dès le début** et ainsi on peut apporter une réponse à moindre coût.

En priorisant, en délivrant des petits incréments en situation cible, on aborde tous les vrais risques et ceux auxquels on n’avait pas pensé.



Risque 3 ; réponse Benoit



Risque 4 ; réponse Benoit

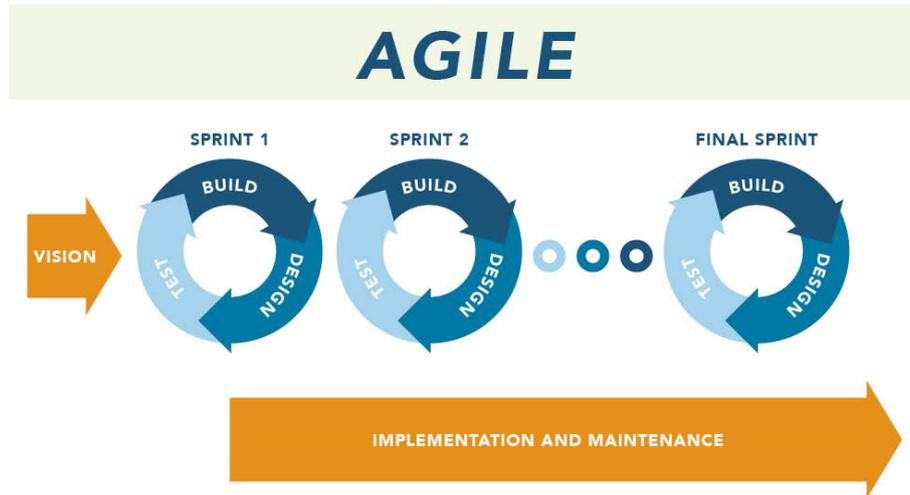
Comment combiner des projets agiles et des projets non-agiles ? (applications monolithiques dans un environnement “waterfall”, la gouvernance de tous ces projets ensemble) ?

(Pablo Pernot, août 2018). En un mot : avec douleur. Et Agile va perdre. Agile va perdre car il va essayer de délivrer des petits morceaux autonomes faisant sens et priorisés par valeur pour délivrer et valider au plus vite (et ainsi gérer les risques). Mais si l'équipe agile est dépendante d'une équipe non agile, en mode “waterfall” (comme une chute d'eau, une cascade, dont chaque palier est différent : la définition, les spécifications, les développements, les tests, etc contrairement à une approche agile dont l'itération limite le scope mais fournit le tout), je reprends : mais si l'équipe est dépendante d'une équipe non agile, cela ne fera pas de sens car elle ne pourra pas délivrer des petits morceaux autonomes plein de sens et de valeur. Elle devra attendre et pédaler dans le vide bien souvent. Autant arrêter. Pour anticiper ces problèmes et voir cela un “portfolio Kanban” peut être très utile.

Un mot d'ordre : il faut limiter les dépendances ! Au maximum. D'où l'organisation autour des *feature teams*.

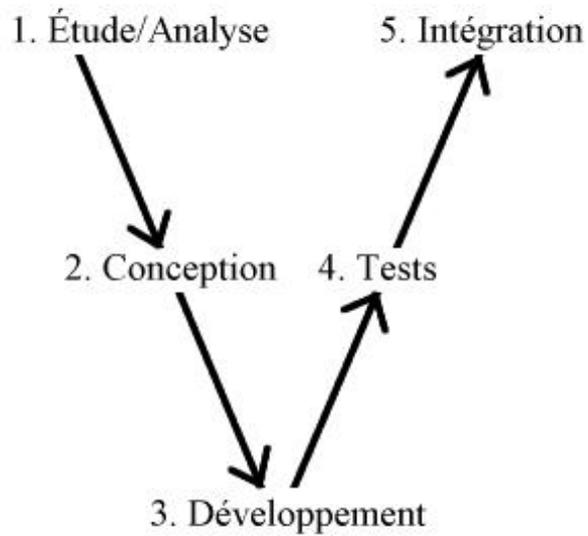
- Kanban Portfolio 1
- Kanban Portfolio 2

Rappel schématique cycles itératifs et incrémentaux agiles



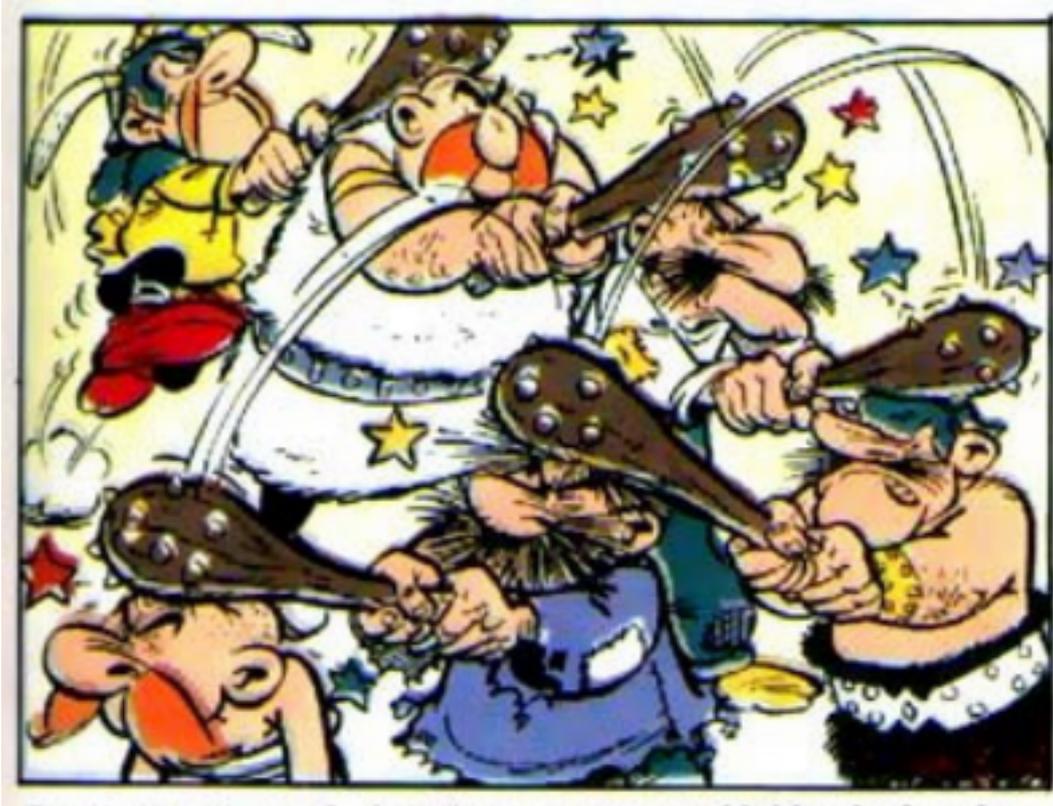
Cycles itératifs/incrémentaux agiles

Rappel "waterfall", "cascade", "cycle en V"



Cycle en V

Une autre vision du Waterfall



Waterfall selon Astérix

Le rôle du chef de projet et du scrummaster. Quelle différence ?

Réponse 1

(Pablo Pernot, août 2018) La différence est cruciale. Dans le monde moderne dans lequel nous évoluons nous devons répondre à la complexité (tout est connecté, tout est lié, tout est très rapide, trop rapide). Ainsi nous ne pouvons nous appuyer sur une seule personne. Une seule personne est trop lente. Une seule personne se trompe. Nous nous appuyons sur l'intelligence collective, sur la dynamique d'une équipe (pas trop grosse, pas plus de 8/10 personnes). Aujourd'hui nous délivrons régulièrement, pour valider, pour apprendre dans ce monde complexe. Nous n'avons donc pas besoin d'une personne dont le rôle serait de contrôler ou d'organiser le travail des autres. Nous avons besoin d'une personne qui va faciliter autour de l'équipe pour l'aider à délivrer de la valeur. La différence est cruciale.

Réponse 2

(Benoit Ricaud, août 2018) C'est un peu contextuel, tout dépend des responsabilités que l'on se donne, celles que l'on désierait, la théorie vs ce que l'on voit sur le terrain. Parlons-nous d'un chef de projet classique? d'un "chef de projet" agile (aka un ScrumMaster, non ? :)), d'un Delivery Manager? Il en va de même

pour du rôle d'un ScrumMaster. Le plus important me semble-t-il est de s'accorder sur des définitions communes pour les comparer.

By the book :

- [What is a scrummaster](#)
- [Duties effective project manager](#)

Quelques billets sympa de Jean-Claude Grosjean :

- [Chef de projet : "Command & control" VS "Protecteur & facilitateur", Un chef de projet qui déménage](#)
- [Chef de projet et Agilité : une évolution... nécessairement en contexte!](#)

Comment savoir qui peut prendre quoi comme décision?

(Pauline Garric, août 2018) Le *delegation board* est un outil de management visuel issu du management 3.0. Il est évolutif dans le temps. Il permet à une équipe de se mettre d'accord, ensemble, sur le niveau de délégation attendue par tâche ou par projet et ce qu'il faudrait pour aller de plus en plus vers l'autonomie.

Il existe 7 niveaux de délégation :

- Tell : le manager prends la décision et informe l'équipe
- Sell : le manager prends sa décision et convainc l'équipe que sa décision est la bonne
- Consult : le manager prends la décision après avoir consulté l'équipe
- Agree : le manager prends la décision en accord avec l'équipe
- Advise : le manager influence la décision prise par l'équipe mais ne pourra pas prendre lui même la décision
- Inquire : l'équipe demande du feedback au manager après avoir pris la décision
- Delegate : l'équipe fonctionne en autonomie

L'équipe et son manager détermineront ensemble la liste des points sur lesquels l'ensemble envisage d'éventuelles délégations. En exemple, nous pourrions avoir des décisions sur :

- la gestion de projet
- les outils à utiliser
- la stratégie fonctionnelle du produit

Il est possible de la faire en double entrées : projet et personne et d'attribuer un post-it donnant le niveau de de délégation attendu.

[Management 3.0 : delegation board](#)

Comment estimer la valeur d'une User story alors ... que j'entends dire depuis le début que par nature, les estimations sont toujours fausses.

(Pablo Pernot, septembre 2018) Oui les estimations sont fausses, surtout si on considère qu'elles sont justes.

On pourrait dire que les estimations sont un bon repère de maturité d'une équipe, d'un département, dans une approche moderne.

Au début on estime tout, pour de gros projets. On estime tout et précisément. Et comme vingt années d'observation le confirment, nos estimations sont fausses et bien souvent inutiles. Enfin ce qui est inutile c'est la précision du chiffre et le temps passer à le calculer. La conversation et la réflexion qu'il y a eu autour ont pu nous faire apprendre beaucoup de choses. Mais souvent on apprend plus en essayant qu'en imaginant. Avec le temps donc, l'expérience, ce que j'appelais maturité, on se rend compte que les estimations sont fausses, mais que les conversations et les réflexions engagées ont pu se révéler intéressantes, mais que la réalité n'a jamais été comme nous l'avions imaginé et donc que l'intérêt de ces conversations demeure limité. On se rend compte que l'on perd beaucoup de temps. À estimer, surtout avec précision. À discuter de choses qui se révéleront différentes ("*aucun plan ne survit au premier contact avec l'ennemi* – Von Moltke"), surtout si ces choses se déroulent dans un futur non proche.

Autre problème en délivrant des estimations on répond à un besoin de se projeter, compréhensible, mais aussi souvent à un besoin de contractualisation. Celui-ci beaucoup plus incompréhensible dès que l'on découvre que nos estimations ne sont pas justes. Mais l'expérience venant (très rapidement sur ce sujet), on devine aussi que peut importe la justesse de l'estimation, c'est plus un levier de pression et de désresponsabilisation qui surgit ici. Qui sera coupable? Qui s'engage sur quoi sur un périmètre que l'on ne maîtrise pas? Ou surtout l'inverse : ce n'est pas ma faute ils avaient dit que...

C'est toujours un problème quand la réponse ne répond en fait pas à la question. Vous me suivez? C'est un problème de demander des estimations – qui seront fausses – pour savoir "à peu près" le temps et les moyens engagés, quand souvent une partie de la question est : qui sera responsable? C'est donc toujours un vrai problème d'avoir un service achat. Il ne pose pas la même question. Et la même réponse sert à deux questions très différentes...

Bref, dans un premier temps il sera difficile de faire lâcher ces personnes sur les estimations (qui vont les protéger, qui vont leur permettre de mettre la pression, etc.), on va donc probablement réduire le gros projet en petits projets que l'on va estimer par sous-parties. On garde les estimations. On en arrive à ces itérations agiles. De petits espaces dans lesquels on estime des *user stories* ou autre. Ça permet déjà de ne plus estimer des choses trop inconnues (qui se produiront plus tard), ou trop inutiles (dont nous découvrirons qu'elles ne se feront jamais). C'est déjà un progrès.

On se met à délivrer (ou pas) des choses régulièrement. Et si justement ce n'est pas régulièrement, cela va gratter... Dans cette dynamique les estimations vont nous servir à découper les choses en petits éléments pour nous permettre de délivrer des choses, éviter l'effet tunnel. Puisque l'on est entré dans une approche incrémentale/itérative, les estimations nous aident à percevoir si c'est gros, moyen, ou petit. Et on tend à viser du "petit", voire du "moyen", pour s'assurer de pouvoir finir les choses. Si on garde des estimations précises on va se rendre compte que même dans les semaines à venir les surprises et les changements seront présents. **La précision se révèle donc une perte de temps.** Et la précision, ou plutôt le chiffre, poursuit des buts cachés :

- Quelle équipe fait le plus de points? (même si cette question est absurde, car isolée comme cela cette notion n'a ni queue ni tête).
- Notre client paie pour tant de jours/hommes, ou tant de points, est-ce que nous les lui fournissons? (c'est toujours absurde, vous percevez bien comme on fait de quelque chose de très flou quelque chose de très précis, à tort, mais tellement aisé pour négocier).

Ici on peut avoir la chance de sortir des chiffres. Puisque l'intérêt c'est de découper en petits éléments, et d'avoir une conversation, une réflexion sur ce qui doit être fait. Et que l'on a appris que l'on perdait du temps à être trop précis (rappel pour l'étourdi : les estimations sont fausses surtout si on pense qu'elles sont justes). On réussit alors à découper en éléments sans chiffre, ni nombre, (*tee-shirts* : XXL, XL, L, M, S, *animaux* : Mammouth, Rhinocéros, Ours, Chien, Chat, Souris, *macro* : Impensable, Très gros, gros, moyen,

petit, etc. Inventez le votre). Il y a toujours des gens qui se croient malin et qui font un temps des conversions. Si ils ont du pouvoir vous verrez vite que vous n'êtes pas encore sorti du système précédent (les dialogues tourneront autour du temps, de l'estimation, et pas du contenu, de la valeur). Laissez le temps au temps? Limitez encore plus la lecture : trop gros, gros, moyen, petit. On gagne du temps à ne pas aller dans des précisions inutiles, et nos estimations sont plus réalistes car plus floues.

Si cela fonctionne, vous commencez à délivrer des "moyennes" et des "petites" *user stories* ou autre. En ayant perdu beaucoup moins de temps avec des choses inutiles comme estimer précisément, on lève la tête et on a du temps pour se poser d'autres questions. Comme l'homme préhistorique a pu se lancer dans plein de nouvelles activités car il s'est mis à cuir sa viande, et donc son temps de mastication et digestion a été fortement réduit. L'art rupestre? La philosophie? On ne saura jamais vraiment. Ainsi en gagnant du temps les bonnes questions arrivent : Pourquoi cette fonctionnalité? Et pourquoi pas celle-ci? Laquelle marche comme prévue? etc. On passe doucement d'une approche projet, à une approche produit. Le but n'est pas de délivrer un ensemble de demandes, mais de travailler sur une dynamique entre votre produit et son utilisation. Autre effet de cette période : on se met à ne plus estimer par avance, on perçoit une cadence : tiens on délivre 4 "petites", et 2 "moyennes" par itération, en moyenne. À partir de cette cadence on se projette sur une capacité. On a une capacité de production relative à notre produit, et non plus une somme estimée de livrables pour un instant t. (D'où aussi la disparition des backlogs, qui eux aussi sont des puits à perte de temps dès qu'ils deviennent un peu chargés).

La réflexion s'étant déplacée vers le produit, la valeur. L'estimation s'est réduite d'un côté à : Quelle est notre capacité de production? Qu'est ce qui serait le plus intéressant à délivrer? Comment délivrer quelque chose d'exploitable au plus vite (donc savoir découper en petits morceaux autonomes qui portent du sens)? Et la véritable estimation a grandit : Qu'est ce qui a de la valeur pour nous, pour le produit? Comment valider nos hypothèses de la meilleure façon? À ce moment on priorise par valeur vis à vis du produit, et on découpe en petits morceaux nos premières priorités (et juste cela) pour avoir assez de souplesse, pour rendre possible une articulation qui nous permette de bénéficier aisément de cet investissement : valider nos hypothèses, et potentiellement gagner de la valeur. Avant de faire le pas suivant. On maximise l'apprentissage, on minimise le temps perdu. On garde une vision, une direction. On obtient les agendas en observant notre cadence, notre capacité.

On pourrait résumer quatre phases :

- Grand mensonge et perte de temps.
- Petits mensonges entres amis.
- On est plus efficace sans chiffre.
- De quoi a besoin mon produit?

Enfin quelques articles liés :

- [Beyond budgeting et #noestimates.](#)
- [La malédiction du jour homme.](#)

Cette FAQ est maintenue lors d'accompagnement par les coachs, scrummasters & product owners qui interviennent autour de DSI RA.

Ils sont (ordre alphabétique / prénom) :

- Alexandre Thibault
 - Benoit Ricaud
 - Mostafa Elaibi
 - Pablo Pernot
 - Pauline Garric
 - Sabrina Palmieri
-

Table des figures

1	Lean Canvas - cheminement	3
2	Lean Canvas - dynamiques	4
3	Context Switching	9
4	Kanban	10
5	Dépendances sur carte kanban	10
6	Release plan	11
7	Release plan	12
8	Quarterly planning board	13
9	Pratiques Craftsmanship	14
10	Simple PO Team	16
11	Complex PO Team	17
12	Risque 1 ; réponse Benoit	20
13	Risque 2 ; réponse Benoit	21
14	Risque 3 ; réponse Benoit	22
15	Risque 4 ; réponse Benoit	22
16	Cycles itératifs/incrémentaux agiles	23
17	Cycle en V	23
18	Waterfall selon Astérix	24