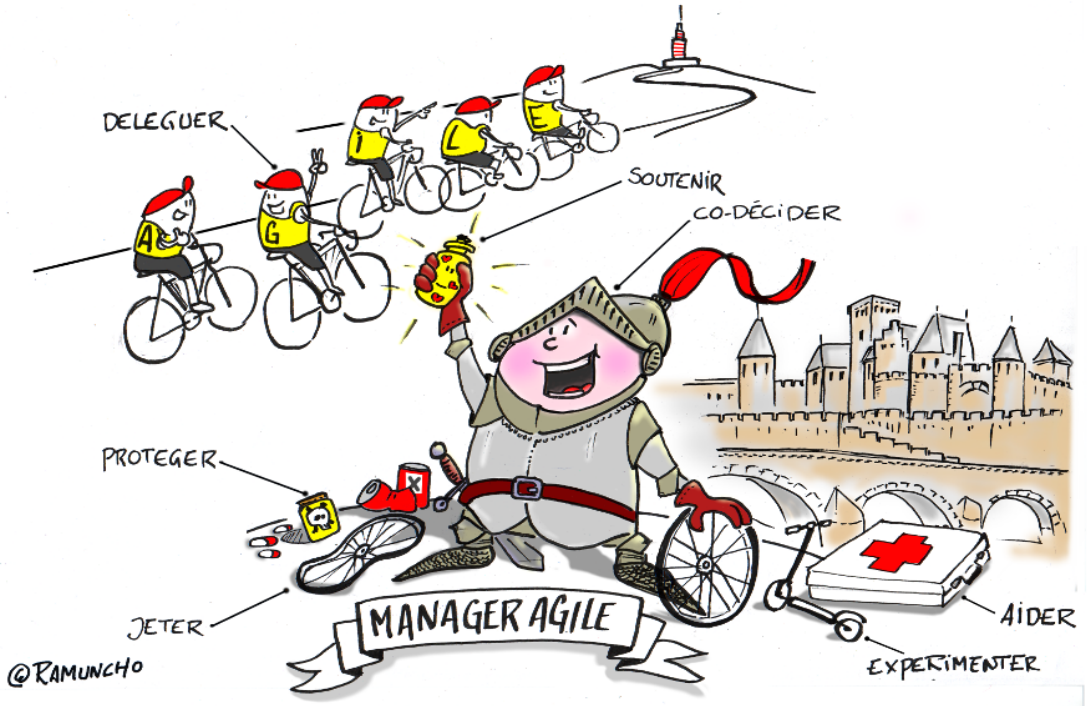
Thierry Cros

MANAGEZ AGILE

Co-fixez le Quoi, Soutenez le Comment.



Être Agile

Managez agile

Co-fixez le Quoi, Soutenez le Comment

Thierry Cros

Ce livre est en vente à http://leanpub.com/managezagile

Version publiée le 2017-10-30

ISBN 978-0-244-03266-1

Ce livre est publié par Leanpub. Leanpub permet aux auteurs et aux éditeurs de bénéficier du Lean Publishing. Lean Publishing consiste à publier à l'aide d'outils très simples de nombreuses itérations d'un livre électronique en cours de rédaction, d'obtenir des retours et commentaires des lecteurs afin d'améliorer le livre.

Aussi par Thierry Cros

Spécifiez agile

Le rocher du Pic de Nore

Ce livre est dédié aux Managers, femmes et hommes, qui réinventent jour après jour leur métier en toute humanité dans un monde de plus en plus complexe.

Table des matières

Introduction

Pourquoi ce livre

Le rôle de Manager agile a été identifié dès les débuts du mouvement agile, en particulier dans l'Extreme Programming. Disons brièvement qu'il est essentiellement **soutien** de l'équipe, elle-même constituée de personnes qui contribuent au **besoin** et d'autres à la **solution**. Le terme "équipe" représente, aujourd'hui, tout autant un groupe de quelques personnes dédiées à un produit qu'une organisation dans son ensemble. L'approche agile se généralisant, la question du management se pose alors de façon prégnante, d"autant plus que le Manager - qui détient généralement le pouvoir - est décisif dans une transformation agile.

Il y a quelque temps, un Manager me demande :
- Finalement, quelle est la journée d'un Manager agile ?

Cette interrogation revient régulièrement au hasard de mes missions d'accompagnement. En effet dans une transformation agile le Manager se retrouve face à la quadrature du cercle :
- sponsoriser et soutenir l'autonomie de l'équipe,
- déléguer une bonne partie de ses activités classiques,

- conserver (généralement) la responsabilité face à l'organisation, du moins dans un premier temps.

Hier, il était payé pour résoudre des problèmes. Aujourd'hui, il l'est pour aider l'équipe à les résoudre elle-même, dans la mesure du possible. Ce sont alors des questions **existentielles** qui se posent parfois. Je me souviens de cette Manager qui arrive décomposée à notre point mensuel (voir le chapitre "Posture").
- C'est bien beau l'agile[1]... Mais visiblement je ne sers plus à rien, je n'ai plus ma place ici.

Je fais la supposition qu'elle a eu le courage d'exprimer ce que bien des Managers - hommes et femmes - ressentent lorsqu'ils sont embarqués, bien souvent malgré eux, dans un changement vers l'agile.

De la pratique "One 2 One" du Management 3.0[2] au rôle de "Servant Leader" pour décrire le ScrumMaster dans le guide Scrum 2016[3], le Manager est pratiquement face à de nombreux fragments de son métier.
Quid de la synthèse ?
Quelle différence entre un Manager 3.0 et un Manager

1. Comme d'autres mots de la langue française, j'utilise "agile" tout autant comme substantif qu'adjectif.

2. Voir le site https://management30.com/. Le *one2one* est une rencontre fréquente entre le Manager et chaque membre de l'équipe.

3. Voir le site http://www.scrumguides.org/.

agile[4] ?

Cet ouvrage est simplement un essai : présenter les différentes facettes de ce nouveau métier, répondre à la question “mais que fait un Manager agile au quotidien ?” car l’agile est par essence pragmatique.

Mon expérience, à la base de ce livre

J’ai aujourd’hui (2016) *grosso modo* trente ans d’expérience professionnelle : création de sociétés[5], management de *Business Unit*…
Du recrutement au licenciement, de la gestion de projet à la stratégie d’une organisation en passant par le montage de partenariats, j’ai plongé bien souvent dans les eaux, tranquilles ou pas, du management. De collaborateur dans un grand groupe industriel à associé de “ma” SCOP[6] j’ai aussi expérimenté… différentes dynamiques.
La moitié de ce chemin, depuis 2000, a été parcouru en agile.

4. La différence est simple : dans “Manager agile” il y a “agile” ; un Manager agile se doit de *connaître* les valeurs et principes agiles (feedback concret et rapide…).

5. En particulier *Twam Informatique* en 1990, société de services puis de conseil en architectures et méthodes (J2EE, OMT, RUP, Extreme Programming…).

6. Société Coopérative et Participative. Voir par exemple https://fr.wikipedia.org/wiki/Soci%C3%A9t%C3%A9_coop%C3%A9rative_et_participative. Voir aussi http://capjaya.fr

Que ce soit en tant que Coach, Chef de projet, Qualiticien, Manager… l'intention est d"être agile.

Je découvrais l'Extreme Programming (XP) en 1999. Tout de suite je fus séduit par cette méthode à la fois pratique et révolutionnaire. Bien au-delà des techniques de développement (conception simple, binôme…) XP propose de maintenir un plan de *versions fréquentes*, des itérations courtes (trois semaines à l'époque alors que Scrum était à quatre), le *client sur site* (le Product Owner de Scrum qui travaille tous les jours avec les Développeurs)… Les principes du Manifeste agile reprennent d'ailleurs bon nombre de recommandations d'XP. Ma formation à XP en 2000 (Kent Beck, Ron. Jeffries, Robert Martin, Martin Fowler) a été complétée en 2006 par une formation au Coaching et plus récemment par un parcours SAFe.

Comme tout auteur je n'exprime jamais que *ma* compréhension. C'est bien à vous de forger votre propre opinion, surtout en pratiquant !

Autrement dit, considérez cet essai comme une cartographie plus ou moins complète et détaillée, de votre nouveau territoire : le management agile. Ce sera alors à vous d'explorer, de compléter la carte, voire de la créer en expérimentant, ce qui est très agile. Je repense à cet accord toltèque :
Ne me croyez pas, ne croyez pas les autres, ne vous croyez pas, qui incite à remettre en cause nos croyances, nos ré-

actions stéréotypées[7]et en corollaire : apprendre à écouter, expérimenter soi-même.

Contenu

Dans les grandes lignes, la réponse à la question “Mais que fait un Manager agile ?” se construit à mon sens selon trois axes :
- Soutien opérationnel aux équipes
- Soutien aux personnes (pour le dire vite l’aspect “RH”)
- Posture : *Servant Leader*, Manager Coach.

Ceci étant, nous envisagerons une facette bien souvent négligée : le Manager est parfois **partie prenante** du produit, il est décisif quant à la stratégie, le budget…
Comment soutenir une équipe agile et ignorer ses buts, ses principes, comment être Manager agile et ignorer l’agile ? Je pose cette règle : un manager agile se doit de connaître l’agile.

Dans cet ordre d’idée, nous irons faire un tour du côté de la **planification**. Si cette activité est la souveraineté du Client ou Product Owner dans Scrum il reste que le Manager le soutiendra d’autant plus efficacement qu’il comprendra ce qu’est un plan véritablement agile.

Les équipes sont constituées d’individus. Le recrutement, le salaire, l’évolution de carrière… Autant d’activités géné-

7. Je vous suggère la lecture du livre *Le 5ème accord toltèque* http://www.editions-tredaniel.com/le-5eme-accord-tolteque-p-3875.html.

ralement dévolues au Manager. Mais que deviennent-elles dans une équipe agile ? Nous nous intéresserons alors à l'aspect RH du management agile.

Enfin, c'est dans sa **posture** que le Manager agile trouvera son efficacité, sa singularité. Un peu comme un "Coach agile" qui est pratiquement un *Consultant* (qui apporte des solutions) agissant dans une posture de Coach.

Avant de démarrer...

J'écris ce livre comme si je m'adressais à vous personnellement. Nous sommes par exemple en train de discuter de Management agile autour d'un verre dans cette belle et émouvante cité médiévale de Carcassonne qui me tient tant à coeur.

Vous êtes peut-être déjà Manager. J'utiliserai le masculin (un Manager) dans la suite de l'ouvrage, ce qui ne veut pas dire qu'il s'agit nécessairement d'un homme[8]. Quoi qu'il en soit, votre organisation se *transforme* pour devenir agile.

La diversité des organisations est extraordinaire. De la startup à la trans-nationale de dizaines de milliers de personnes en passant par des entreprises du tertiaire, secondaire ou

8. Soyons honnêtes : c'est aussi par "flemme" que je n'indiquerai pas systématiquement le masculin et féminin, par exemple sous la forme "il-elle était intéressé.e par cet ouvrage". Mes camarades de la SCOP "Maison de l'Initiative" à Toulouse, militant.e.s émérites de la condition féminine, voudront bien me pardonner.

primaire les situations sont extrêmement différentes. Je m'adresse donc plutôt à vous qui êtes dans une organisation de quelques dizaines à quelques milliers de personnes. Autrement dit, je vous imagine Manager parmi d'autres, dans une structure *hiérarchisée* à plusieurs niveaux. Cela signifie simplement que certaines situations ou considérations décrites ici vous paraîtront peut-être ubuesques si vous n'êtes pas dans ce cas.

Manager agile ?

Se pose tout d'abord la question du Manager. Vous êtes peut-être Manager fonctionnel, c'est-à-dire responsable de choses faites par d'autres. Ou bien Manager "RH", en charge du devenir d'individus dans l'organisation. Toutes les pratiques proposées ici ne correspondront peut-être pas à votre périmètre de management. Je suis persuadé que vous trouverez tout de même des éléments concrets et d'autres qui alimenteront votre réflexion.

De la gestion de projet à la direction d'une organisation, le management est multiforme. Ces différentes facettes sont donc à adapter : un manager de projet sera concerné par les obstacles qui se présentent au quotidien à l'équipe. Un directeur de département s'intéresse (entr'autres) à la raison d'être de l'ensemble formé par les collaborateurs. Un chef de service est attentif au plan de carrière des personnes. Autant d'exemples qui montrent la diversité des activités de management. Une caractéristique commune : le

Manager *invente* son métier, jour après jour. En agile cette caractéristique est d'autant plus importante.

Il est temps de nous retrouver dans cette belle cité de Carcassonne, entreprenons sans tarder notre aventure en Management agile.

Carcassonne,
le 14 juillet 2016.

1. Être agile

Nous voici au pied des remparts, côté Ouest, près de la rivière Aude. Je vous propose d'entrer dans la cité de Carcassonne, haut lieu de l'histoire occitane[1].

La cité côté Ouest

1. Pour un aperçu de l'histoire de la cité pendant le génocide des Cathares au XIIIème siècle, je vous suggère https://fr.wikipedia.org/wiki/Si%C3%A8ge_de_Carcassonne_%281209%29

Et donc vous êtes Manager agile

C'est une belle aventure que vous entreprenez !
Aventure collective et personnelle. Attendez-vous à être étonné, parfois désorienté. Et aussi émerveillé. J'insiste sur l'aspect "aventure" : inconnu et difficultés côtoient des découvertes extraordinaires, inattendues, y compris sur vous-même. Ne tombez pas dans le piège paralysant qui traduit *Capital Venture* par "Capital risque".
L'aventure est bien plus riche que ses seuls risques !

Agile... Qu'es aco ?

Commençons par le commencement. Dans "Manager agile" il y a "agile". Le Manifeste agile[2] est rédigé en 2001 sous l'impulsion de la communauté Extreme Programming[3]. Bien que certains aspects soient désormais obsolètes, il perdure en tant que définition originale - unique - de ce mouvement.

Qu'est-ce qui est, à mon sens, obsolète dans cette définition historique ?
D'une part les références au "logiciel". Aujourd'hui tout produit ou service, y compris au niveau de l'organisation dans son ensemble, est propice à être agile.

2. http://agilemanifesto.org/iso/fr/manifesto.html
3. Voir à ce sujet l'histoire du Manifeste agile (en anglais) : http://agilemanifesto.org/history.html rédigée par Jim Highsmith.

Une nuance importante : un produit peu complexe ne nécessite pas une boite à outils agile, nous y reviendrons très bientôt. Mais ce cas est-il d'actualité ?
D'autre part ce Manifeste s'inscrit dans ce que certains nomment "océan rouge"[4] (pour le dire vite le fonctionnement classique client-fournisseur) alors que nous évoluons vers un monde d'océans bleus, de "potentiels inexplorés". C'est d'ailleurs ce que suggère Kent beck dans sa nouvelle version du Manifeste en 2009.

> Initiating change over responding to change (or following a plan)[5]

Voici donc ce Manifeste, définition de l'agile.

> Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire. Ces expériences nous ont amenés à valoriser :
>
> > Les individus et leurs interactions plus que les processus et les outils
> >
> > Des logiciels opérationnels plus qu'une documentation exhaustive

4. Voir par exemple https://fr.wikipedia.org/wiki/Strat%C3%A9gie_oc%C3%A9an_bleu.

5. Voir http://jchyip.blogspot.fr/2010/06/kent-becks-evolution-of-agile-manifesto.html.

> La collaboration avec les clients plus que la négociation contractuelle
>
> L'adaptation au changement plus que le suivi d'un plan
>
> Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.

Ce sont les **quatre valeurs** du Manifeste agile. Notez que "valeur" est à prendre ici au sens de "croyance viscéralement ancrée". C'est ce qui vous fait réagir (et non pas répondre) immédiatement et quasi inconsciemment à une circonstance donnée, parfois de façon irrationnelle.

Prenons l'exemple d'une équipe qui fonctionne correctement pendant des semaines. Soudain, c'est la catastrophe ! Un imprévu remet sérieusement en cause le plan ! Dans ce cas, une réaction du Manager pourrait être "OK, j'ajoute tel contrôle qualité, tel document, tel reporting". C'est une réaction basée sur la valeur "Processus et Outils". Un Manager agile préfère *les individus et interactions* : "OK, arrêtons tout, réunissons-nous pour élaborer ensemble la meilleure réponse à cette situation". Ce livre ayant pour sujet le Management agile, je souligne la réaction du Manager. Bien entendu, c'est l'équipe dans son ensemble qui réagit ainsi. Je décrirai plus loin une pratique Lean - *Stop the Line* - qui est emblématique du rôle de Manager agile.

Au-delà des produits et services, les valeurs et principes s'appliquent à l'organisation qui, elle aussi, devient agile. Ce Manifeste est augmenté d'une douzaine de principes qui précisent ce qu'est véritablement l'agile. Je vous invite à lire, étudier, vous approprier (ou pas…) ces douze principes[6].

"Les individus et leurs interactions" est un synonyme d'équipe. J'insiste sur ce point car cette dimension **collective** en termes de résultats et de fonctionnement est bien souvent sous-estimée.

Le rôle de Manager est en filigrane du cinquième principe du Manifeste agile[7].

> Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.

Si la suite du livre vous permet de préciser votre nouveau rôle, comme tout Manager vous avez à *l'inventer* jour après jour. Connaître les "tenants et aboutissants" de l'agile me semble alors essentiel. Sinon vous allez être un Manager Coach, 3.0, collaboratif… Pour autant l'agile va plus loin.

Soutien, confiance, motivation (le 5ème principe)… Et "objectifs fixés". Votre nouveau territoire - le Management

6. Voir http://agilemanifesto.org/iso/fr/principles.html.
7. http://agilemanifesto.org

agile - se dessine en quelques mots, dès ce Manifeste en 2001.

Notez un point important : les cadres et méthodes agiles précisent le développement de produits, pas la gestion de projets. Adapter agile et gestion de projet (du moins budget, etc) est donc une étape nécessaire, normale et inéluctable dans une transformation agile, du moins tant que cette notion de projet existe.

Voyons rapidement comment le Management est intégré dans une méthode (Extreme Programming) et un cadre (Scrum) agiles.

Extreme Programming : le Project Manager

Dans le premier ouvrage consacré à l'Extreme Programing (*Extreme Programming Explained* première édition en 1999) Kent Beck décrit le rôle du *Big Boss*. Il insiste sur la posture de *soutien* à l'équipe. Les rôles et responsabilités changent, le Manager doit alors s'attendre à plus d'autonomie, de transparence de la part de l'équipe. D'une certaine manière, elle va lui parler *cash*, ce qui est probablement inhabituel.
Le *principe de réalité* est consubstanciel à l'agile.

Scrum : le ScrumMaster

Le Scrum Guide[8] définit le rôle de ScrumMaster. Il est garant de Scrum, supprime les obstacles, point de contact entre l'équipe et l'extérieur… C'est un *Servant Leader*. Si ce n'est pas le Project Manager, un tableau RACI-Délégation (voir chapitre Délégation) est très utile pour clarifier les activités et attributions de chacun. Mais nous pourrions dire que Scrum se contente de décrire une partie de ce rôle de Project Manager agile de proximité.

Définir "agile"

Parlons plutôt de **tentative de définition** de l'agile.

D'où vient l'agile ?

Le développement de logiciels s'inscrit historiquement dans un cadre prédictif, plus ou moins dérivé du "cycle en V". Cette approche pré-suppose
- la possibilité de fixer les besoins puis la solution
- la capacité à planifier les activités à long terme
- en dernière analyse, une maîtrise du développement qui autorise cette prédiction.

8. Vous trouverez ce document sur le site http://scrum.org (onglet Ressources).

L'expérience montre (voir les résultats d'enquête auprès des Utilisateurs) que cette démarche prédictive ne fonctionne pas correctement. La nécessité de feedback plus concret et plus rapide apparaît et prend la forme de maquettes d'interface ou bien de benchmarks de solutions matérielles et/ou logicielles. L'orthodoxie **waterfall** n'est plus de mise.
Les années 90 sont le théatre d'expérimentations basées non plus sur une approche prédictive mais *empirique*. Pourquoi ?

La *complexité* du développement de logiciels est enfin reconnue. Des cadres tels que Scrum, ou des méthodes telles que l'Extreme Programming, naissent à ce moment-là. Sous l'impulsion de la communauté XP, les concepteurs de ces différentes approchent se réunissent en février 2001 et rédigent le *Manifeste agile pour le développement de logiciels*. En France, l'association Extreme Programming France est créée en 2000 à Toulouse (et deviendra Agile France en 2008). XP est fer de lance de ce mouvement jusqu'en 2006, date de la première formation "certifiante" à Scrum en France. Les questions d'agile à l'échelle se posent dès le départ, par exemple lors de la conférence internationale consacrée à XP en Juin 2000 en Sardaigne.

L'apparation de Scrum dans le PAF (Paysage Agile Français) permet le véritable lancement de l'agile, jusque-là confidentiel car exclusivement orienté développement. Pourtant, XP propose de nombreuses pratiques (versions fréquentes, client sur site, user stories…) qui seront peu à

peu intégrées dans Scrum. Mais la certication Scrum est une excellente idée marketting et nous vivons alors la *ScrumMania*. D'un point de vue agile, c'est une régression : quid du pilotage par la valeur, de la qualité logicielle, des versions régulières… ? Pourtant le Manifeste est, à bien des égards, une transcription d'XP.
- Les premiers principes reprennent "versions fréquentes" et "planning game"
- le 5ème principe du Manifeste est une description du rôle de Manager agile décrit dans XP
- l'émergence du besoin et de la conception est clairement posée
- le rythme soutenable est une généralisation de la pratique "40 heures par semaine".

Bien entendu, rien de tout cela n'est interdit dans Scrum qui a l'immense avantage d'être une coquille vide : vous pouvez la remplir avec ce que vous voulez. Prenons la "définition de fini", pratique importante dans Scrum, apparue au début des années 2000. Elle consiste à préciser les critères qui permettent de considérer une demande comme terminée. Ce sont les critères de transition entre la colonne "en cours" et la colonne "fini" d'un tableau Kanban. Maintenant, considérons une équipe qui n'est pas sensibilisée à la qualité logicielle. Cette définition de fini se résume à quelque chose comme
- le développement est terminé
- le Product Owner est d'accord.

Nous sommes à des années-lumière du *code propre*. Quid de

la gestion de configuration, des règles de codage, de revue de code… ? Sans parler de tests d'acceptation.
Attention donc si vous pratiquez Scrum : vous utilisez un cadre qui vous permet d'être agile sans vous l'assurer véritablement.

Le *Lean Sodftware Development*, *Kanban* débarquent vers la fin des années 2000. C'est un curieux retour au sources car XP s'est fortement inspiré de ces sources. Le nom lui-même indique un focus sur l'activité qui apporte directement la valeur en logiciel : la programmtion. Une *user story* dans XP est une étiquette de production Kanban. Ce dernier cadre, tout comme Scrum, vous permet d'être agile.

Le coeur du corpus agile n'évolue plus depuis une dizaine d'années. Mais l'agile est par nature perméable : le Management 3.0, les innovation games… Autant de boites à outils qui complètent le noyau initial.

Les cadres d'agile à l'échelle, tels que SAFe ou LeSS, constituent un saut quantique du corpus agile. SAFe en particulier est tout aussi révolutionnaire qu'XP il y a quasiment vingt ans. En posant les Program Increment "PI" - période d'une dizaine de semaines - ou bien la réunion du *build* et du *run*, en pilotant clairement par la valeur *via* le *Cost of Delay* et le WSJF (*Weighted Shortest Job First*), SAFe est une avancée décisive de l'agile dans les organisations. Autrement dit, c'est la rançon du succès : nous ne pouvons plus nous contenter de cadres tels que Scrum (une dizaine de personnes) et de voeux pieux tels que "inventons notre agilité".

Tout comme Scrum ou Kanban, SAFe peut être détourné. Attention donc à rester dans un esprit véritablement agile lors de leur mise en oeuvre.

Vingt ans après

Je crois que l'agile est, aujourd'hui, une réponse à une double évolution :
- le monde devient de plus en plus **complexe**,
- les **individus évoluent**, les générations Y et Z ne "fonctionnent" plus comme la génération X[9]. Plus généralement, la société elle-même est en mutation : rapport au travail, robotisation… "Le logiciel mange le monde".

Reprenons ces deux causes-racines du succès de l'agile.

L'acronyme anglais *VUCA*[10] caractérise notre monde :
- Volatil (dynamique des changements…)
- Uncertain (difficulté à prévoir, imprévus…)
- Complex (conditions multi-factorielles…)
- Ambiguous (interprétations erronées…).

La genèse de l'agile est probablement, dans les années 90, la prise de conscience de la *complexité* du développement de logiciel. Face à cette complexité, une réponse prédictive (type cycle en V) est inappropriée.

9. Voir à ce sujet les articles de Wikipedia : https://fr.wikipedia.org/wiki/G%C3%A9n%C3%A9ration_Y https://fr.wikipedia.org/wiki/G%C3%A9n%C3%A9ration_Z.

10. Voir https://en.wikipedia.org/wiki/Volatility,_uncertainty,_complexity_and_ambiguity.

De même, plusieurs signes montrent l'évolution des personnes, l'individualisation, l'envie d'autonomie : le succès de l'auto-entreprenariat, le besoin des organisations de miser sur l'intraprenariat, le déclin (inexorable ?) du salariat classique (CDI)… Le rapport aux Sachants et plus généralement au Savoir change radicalement : pourquoi apprendre alors qu'une recherche sur votre moteur préféré vous donne la réponse en quelques clics ?

Face à ce monde VUCA, une réponse qui miserait uniquement sur des processus est inadaptée, insuffisante. Comment décrire dans des processus opérationnels, des activités par nature complexes, donc sujettes à des inconnus, des impondérables, des changements… bref : imprévisibles ?

La proposition agile est d'adopter une approche empirique.
Je vous invite à vous souvenir de cette définition lapidaire de l'empirisme. **Apprendre en faisant**.

Une réponse par essence agile - empirique - à une situation complexe produit deux nouveaux équilibres :
- Valeur et Qualité,
- Responsabilité et Respect.

Équilibrer ajout de *valeur* et niveau de *qualité*[11] du produit ou service, tel est le défi du Manager agile. Répondre aux demandes du présent sans hypothéquer l'avenir par une dette technique ingérable. Harmoniser le Client, qui veut

11. Par "qualité" j'entends en particulier *maintenabilité* autrement dit capacité à évoluer sans avoir à payer un prix exorbitant.

toujours plus de valeur et le Développeur, qui veut être fier de son produit et s'assurer que son évolution ne sera pas un chemin de croix.
Afin d'éviter toute confusion, vous pouvez remplacer "Qualité" par "Dette technique".

Équilibrer nouvelles *responsabilités* et *respect* dans l'équipe. L'agile est une évolution du pouvoir dans l'organisation. Les équipes sont plus responsables et le Manager lâche prise. Mais une véritable transformation agile s'intéresse aussi au *bien-être* des personnes et les respecte. Dans les premiers temps, nous parlions d'hédonisme[12]. Aujourd'hui le "bonheur au travail" est un sujet d'actualité. J'utiliserai "respect". Ce n'est pas uniquement par charité chrétienne. Il se trouve que le bien-être des personnes favorise les activités intellectuelles - les métiers de la connaissance et de la création - et donc l'efficacité. Respecter quelqu'un, c'est ne pas porter atteinte à son intégrité physique, psychologique. Dans cet ordre d'idée, il s'agit - paradoxalement - de ralentir pour aller plus vite. Un Manager agile se doit d'être un agent de ralentissement.

Ces équilibres sont autant de nouveaux états stables. Une transformation agile consiste à passer d'un état stable actuel (qui a le mérite d'exister…) à un nouvel état caractérisé par les équilibres que j'évoquais précédemment. Ce changement d'état suppose une rupture plus ou moins

12. Voir http://thierrycros.net/?post/2013/10/24/H%C3%A9donisme-LeMotDuVendredi ou bien cette présentation http://thierrycros.net/public/docs/ManifesteHR/img1.html.

chaotique, instable. Il ne s'agit donc pas de "résistance au changement" mais plutôt d'homéostasie (fonction d'un système dont l'objectif est de conserver son état d'équilibre) dans cette transformation.

Plus précisément, une approche agile repose à mon sens sur quatre piliers, peut-être plus concrets que les valeurs du Manifeste et plus synthétiques que les douze principes. Je vous invite d'ailleurs à concocter votre propre expression de l'agile, à partir du Manifeste, des méthodes…

Premier pilier : "centré valeur"

C'est clair dès le premier principe agile.

> Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.

"Livrer rapidement" c'est du *Lean* (réduire le Lead Time). Rapidement et régulièrement c'est plus précisément du *Lean Development*. Profitez-en pour jeter dès maintenant une pratique ancestrale : "LA" livraison en fin de développement ou fin de projet. Préférez plusieurs livraisons incrémentales. Cela permet un Retour sur Investissement partiel mais *plus rapide*. Surtout, le feedback obtenu des Utilisateurs et Opérateurs permet de bonifier le produit.

Attention, ne confondez pas "versions fréquentes" d'une approche agile et lotissement d'une approche classique. Si

les deux se rejoignent en termes de mises en exploitation successives, leur contenu, leur rythme, leur pilotage, sont probablement sensiblement différents.

Ce pilier "centré valeur" signifie aussi qu'il est important de privilégier les pratiques qui apportent une réelle *valeur*, les pratiques qui ont un intérêt démontré. Autrement dit jeter les pratiques à "non valeur ajoutée" ou du moins limiter leur impact. Comme l'écriture de documents. Je ne parle pas ici de documents dont l'utilité est évidente, par exemple un "plan de validation" dans un environnement certifié. Dans ce cas, la question qui se pose est celle d'une élaboration *agile* d'un tel document. Il est terrible de constater qu'un document est bien souvent justifié par de nombreuses personnes sauf celles directement concernées. Ou simplement rédigé parce que… C'est comme çà !

Centré valeur : il se trouve que c'est l'origine du nom "Extreme Programming" : concentrer l'effort sur l'activité qui apporte la valeur, la programmation dans le cas du logiciel. Nous retrouvons ici une autre caractéristique corollaire du Lean : éliminer les gaspillages.

La question à se poser est celle de l'origine et de l'utilité des pratiques. J'ai commencé à programmer avec des cartes perforées. Nous avions des pratiques spécifiques, comme tracer au feutre une diagonale sur le paquet de cartes au cas où… Ou bien réfléchir sérieusement à la solution globale car le cycle de production du logiciel était (très) long : perforer les cartes, les déposer dans le bac à l'attention de l'opérateur. Celui-ci les récupérait et nous obtenions le ré-

sultat de la compilation (et éventuellement de l'exécution) parfois plusieurs heures après. À comparer aux analyses syntaxiques à la volée des environnements et machines actuels. Mais avons-nous vraiment jeté tout ce qu'il y a à jeter ? La "Grosse Conception du Début" ? La "Grosse spécification" ? Cela avait du sens… Avant.

Notez ce principe agile :
> Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées.

Ici, l'émergence est en quelque sorte une déclinaison de l'empirisme : apprendre en faisant, **spécifier et concevoir grâce à l'utilisation du produit**.

Deuxième pilier : feedback plus concret et plus rapide

Probablement l'apport le plus original de l'agile. Le *feedback* est d'ailleurs élevé au rang de valeur dans l'Extreme Programming. Il est en filigrane des piliers "Inspection" et "Adaptatation" de Scrum.

De quoi parlons-nous ? Kent Beck dans *Extreme Programming Excplained* utilise l'analogie suivante.
La Bande d'Arrêt d'Urgence est séparée de la voie de circulation normale par un dispositif particulier : une ligne discontinue blanche, souvent granuleuse ou striée. La fonction de cette ligne est de donner du feedback au conducteur, un feedback suffisamment concret et rapide pour éviter

toute conséquence facheuse à une déviance de trajectoire. Nous pourrions dire de ce point de vue qu'un cadre agile constitue un dispositif de feedbacks concrets et suffisamment rapides pour éviter un dérapage incontrôlé. C'est typiquement la fonction d'une **démonstration** dans Scrum. Mais il existe de nombreux dispositifs de feedbacks plus ou moins concrets et rapides.

Quels sont les dérapages dont nous parlons ?
- budget dépassé
- livraison retardée
- qualité insuffisante
- maintenabilité difficile
- …

Si le véritable feedback est celui des Utilisateurs dans leur environnement d'utilisation, il nous faut créer des moyens qui s'en approchent car la boucle serait bien trop longue la plupart du temps (c'est malheureusement ce qui se passe dans une approche classique).

Un feedback suffisamment concret permet d'améliorer la qualité de la solution.
Un feedback rapide améliore l'efficacité de la rectification.

Boucles de feedback dans l'Extreme Programming

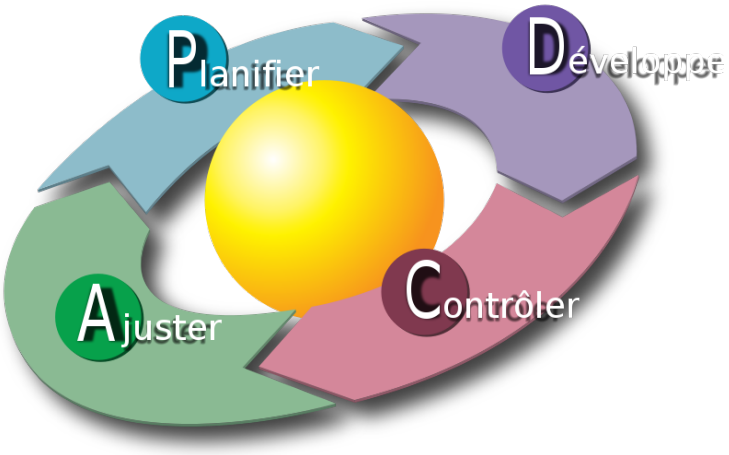
Troisième pilier : auto-amélioration continue

Le douzième principe du Manifeste agile pose l'importance de l'amélioration continue, plus précisément de l'auto-amélioration continue car c'est l'équipe qui est souveraine en ce domaine.

À intervalles réguliers, l'équipe réfléchit aux

> moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

C'est par exemple la session dite *rétrospective* dans Scrum. Ou plus généralement une adaptation de la fameuse "roue de Deming" emblématique du Lean.



Roue de Deming

Ce cyle PDCA (Plan Do Check Act) consiste à
- **Planifier** une proposition d'amélioration lors de la rétrospective de l'itération N
- **Mettre en oeuvre** (*Do*) cette proposition pendant l'itération N+1
- **Vérifier** les résultats de cette expérimentation (*Check* ou *Study*) en début de rétrospective N+1, à partir d'informations concrètes (mesures…)

- **Acter** : soit institutionaliser la proposition qui devient ainsi une pratique "applicable", soit la jeter car les résultats ne sont pas probants.

Ce harnais d'auto-amélioration favorise l'initiative car il minimise les risques : la période d'expérimentation est réduite.

En tant que Manager vous devriez parfois recevoir des demandes d'amélioration de l'équipe qui ne sont pas sous son contrôle : augmenter les capacités du serveur d'intégration, aménager les locaux…

Notez un point important. Dans une organisation classique militairement hiérarchisée, les personnes ne se donnent généralement pas la *permission* de changer. Il vous appartient de créer un climat de **confiance** qui autorisera des expérimentations. Nous en reparlerons dans le chapitre "Posture". Dans un tout premier temps, il vous semblera peut-être dérisoire de rappeler à l'équipe qu'elle a le droit de tenter telle ou telle amélioration. Il faut le faire.
Dans cet ordre d'idée je propose de collecter les propositions d'amélioration (par exemple en fin de rétrospective) sous la forme d'un tableau.

R & A	**R & non A**	**non R OU non A**
Horaire Std up	Jeter DC	Changer BD

R : "nous pouvons réaliser l'amélioration",
A : "nous avons l'autorisation de le faire".

La proposition d'amélioration "changer l'horaire du stand up le matin" est sous contrôle de l'équipe, elle peut la réaliser elle-même et est autorisée à le faire (ou du moins considère qu'elle l'est).
Par contre l'amélioration "jeter le Dossier de Conception" est réalisable par l'équipe mais elle considère (à tort ou à raison) qu'elle doit demander l'autorisation de le faire.
Enfin, changer de Base de Données n'est pas réalisable par l'équipe ou bien elle n'en a pas l'autorisation.

Les deuxièmes et troisièmes colonnes sont à considérer par le Manager. Si, pour une raison quelconque, il décide de ne pas donner suite, il sera alors opportun de s'en expliquer. Dans tous les cas, communiquer sur l'avancement de ces actions renforcera la confiance.

Quatrième pilier : auto-organisation

Aux débuts de l'industrialisation, les Ouvriers étaient bien souvent des paysans illétrés descendus de leurs campagnes pour venir travailler dans des usines. Une organisation quasi-militaire, hyper hiérarchisée, paliait à cette situation, d'autant plus que les moyens de communication étaient rustiques.
Les personnes ont changé. La nature du travail aussi. L'éducation, Internet… Les Ouvriers (par exemple les Dévelop-

peurs de logiciels) savent lire… Et écrire des programmes[13].

Une équipe agile s'auto-organise comme une équipe de rugby sur le terrain. De même qu'un joueur s'adapte à la situation, par exemple en décidant de se placer d'une certaine façon dans une mélée ouverte, de même un Développeur répond de façon appropriée à la situation de l'équipe, en particulier lors du *stand-up meeting* quotidien. L'objectif de chacun est de terminer l'implémentation de besoins, de les livrer rapidement afin d'apporter de la valeur aux Utilisateurs. Autrement dit, terminer des tâches est un moyen qui permet d'atteindre l'objectif : terminer les stories. Le Développeur s'adapte "en temps réel" par rapport à cet objectif.

J'ai parfois tendance, en tant que membre de la communauté agile, à "réinventer l'eau chaude". D'innombrables travaux ont été entrepris depuis des décennies sur les sujets qui nous intéressent au premier plan : auto-organisation, délégation, empirisme, transformation d'organisation… Et nous avons parfois tendance à les ignorer. Pire, nous nous approprions parfois des découvertes que d'autres ont faites bien des années avant nous.

Nous venons de parcourir ce qui constitue à mon sens les piliers d'une approche agile : centré valeur, plus de

13. De là à dire que ce sont des "lettrés"… Les études scientifiques manquent cruellement de sciences humaines. Si seulement un Développeur faisait aussi ses "humanités", un minimum de psychologie, sociologie, épistémologie… Ou simplement lire les écrivains français du XIXème siècle.

feedback concret et rapide, auto-amélioration et auto-organisation.

Une approche minimaliste

Continuons notre exploration en territoire agile.
Voici un aspect essentiel de l'agile, parfois sous-estimé voire ignoré. Contrairement aux approches habituelles, l'agile est *minimaliste*. Autrement dit, ce que vous apporte une méthode telle que l'Extreme Programming ou un cadre tel que Scrum est considéré comme **nécessaire** - et peut-être pas suffisant - dans votre situation. Pour avoir longuement travaillé dans des contextes certifiés (pharmaceutique, aéronautique), je peux dire qu'il faut parfois rajouter des pratiques, des documments à la base agile.
Une autre situation est la configuration des acteurs : entre apporteurs d'ordre et réalisateurs d'ordre, modes projet et maintenance, un produit est dans les mains d'acteurs très divers, l'unité de lieu et de temps est illusoire. La continuité d'équipe, pratique agile, est souvent impossible.

L'agile étant minimaliste, toute la question est alors de déterminer ce qu'il convient de rajouter ou pas… Et comment le rajouter. Un "plan de validation" trouve naturellement sa justification : son absence est rhédibitoire pour la mise en exploitation du produit. À l'inverse, un "dossier de conception" de plusieurs centaines de pages est peut-être, encore aujourd'hui, un héritage de la glorieuse époque où nous codions à partir de cartes perforées.

Comment faire ?
Autorisez-vous à garder telle ou telle pratique, tel document.
Et challengez-le !
En quoi ce dossier a-t-il été vraiment utile ? Quelles sont les pages, parmi les centaines, qui sont vraiment lues, par qui ?
Notez tout de suite cette pratique de Management agile :
Jetez !

Osez remettre en cause vos pratiques et processus, documents, reporting… De quand datent-ils ? Qu'induisent-ils en termes de croyances, principes ? Leur justification est-elle toujours d'actualité ? Avez-vous conscience du prix que vous payez pour maintenir ce *status quo* ?

Vous pouvez considérer la rédaction d'un document (et plus généralement les pratiques actuelles) à l'aune des piliers que j'évoquais précédemment.
Quelle valeur ajoutée,
quel feedback concret et rapide sur le document,
quelle auto-amélioration,
et auto-organisation dans sa rédaction ?

Les collèges agiles

Un point particulier à comprendre en agile : les typologies de rôles, ce que j'appelle "collèges". Il en exite trois.
Autrement dit, quels sont les joueurs ?

Le collège “Besoin”

Ce groupe réunit les personnes qui représentent et/ou constituent le besoin. Il possède normalement un “porte-parole” (*The Voice of Customer*) : le “Client” dans l’Extreme Programming ou bien le “Product Owner” dans Scrum.
Ses activités régaliennes sont :
- l’expression de besoins, à différents niveaux,
- la planification dont l’objectif est d’optimiser la valeur acquise, voir dans cet ordre d’idée la stratégie *Weighted Shortest Job First* proposée par le cadre SAFe[14].
- l’acceptation (ou pas…) du produit **au fur et à mesure de son développement**.
Au-delà de ces activités directement liées au développement, ce collège est concerné par la conduite du changement (changements induits par le nouveau produit), la formation des Utilisateurs, l’alignement entre fonctionnalités et thèmes stratégiques de l’organisation…

Nous trouvons dans ce collège besoin (ou “métier”) :
- Client ou Product Owner,
- Utilisateurs “pilotes”,
- Experts métiers (métier des Utilisateurs),
- Analystes fonctionnels,
- Testeurs fonctionnels,

14. Cette stratégie consiste à prioriser en fonction d’un ratio *valeur / effort* décroissant. Voir à ce sujet http://scaledagileframework.com/wsjf/. La valeur est le *Cost of Delay* constitué de trois composantes : la business value, la criticité par rapport au temps, la diminution de risques et les Opportunités.

- Stratèges de l'organisation (marketing…),
- Managers côté Utilisateurs,
- …

En que Manager, vous êtes peut-être membre d'un tel collège, vous êtes alors partie prenante du produit et avez votre mot à dire quant à la stratégie, au budget…

Le collège "Solution"

Ce sont de façon générale les "Développeurs" c'est-à-dire les personnes qui, d'une manière ou d'une autre, participent à la solution, la développent en regard du besoin.

Leurs principales responsabilités sont :
- la réalisation de la solution (conception, implémentation),
- confirmer la faisabilité, fournir des ordres de grandeur de l'effort nécessaire,
- conseiller le collège métier,
- veiller à la *qualité intrinsèque* (maintenabilité en particulier) de la solution.

Dans le domaine du développement de logiciels, ce collège "solution" est constitué de :
- Développeurs,
- Développeurs séniors légitimes pour élaborer les squelettes de solutions (appelés parfois "architectes"),
- Experts techniques (bases de données, ergonomie, design graphique, temps réel…),
- Testeurs techniques (tests d'intégration…),
- …

Le collège “Soutien”

En tant que Manager vous faites partie de ce collège. Comme le précise le cinquième principe du Manifeste, il s’agit de *soutenir* les équipes opérationnelles, par exemple en supprimant les obstacles qui se présentent à elles.

Mais vous n’êtes pas seul dans ce collège.
- Coaches,
- Outillage,
- Qualité et Méthode,
- …
offrent - si nécessaire - un *support* aux Opérationnels.

Attention toutefois à ne pas trop compartimenter non plus. Je me souviens d’une équipe dans laquelle le Product Owner était un ancien Développeur. À ce titre il connaissait parfaitement le produit, mieux que la plupart des membres du collège solution. Pourtant lors d’une session de Planning Poker, il lui fut reproché d’intervenir dans la conception d’une story…
Pas malin. Ne sacrifiez pas les bonnes volontés, les bonnes idées, sur l’autel des processus et autres définitions de rôles.

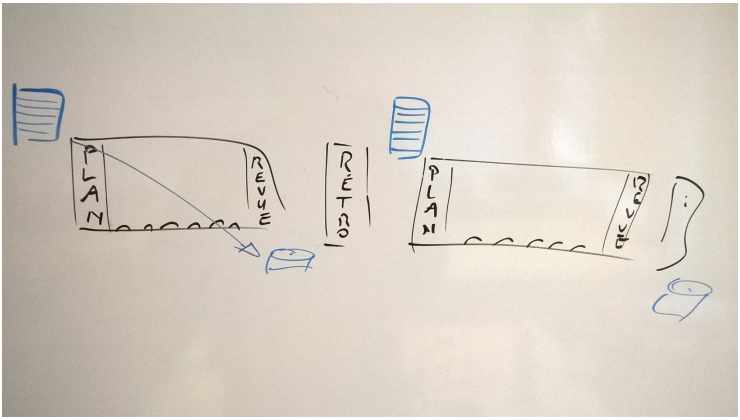
Veillez à susciter l’intelligence collective. C’est plus facile si les rôles et responsabilités sont non seulement clarifiés : explicités, exprimés. Nous en reparlerons à propos de *délégation.*

Équipe coeur, équipe élargie

Les personnes présentes “au quotidien” sur le développement forment l’équipe coeur. Celles-ci et les personnes qui interviennent ponctuellement constituent l’équipe complète.

Agile itératif

L’unité de temps du développement agile est le plus souvent l’*itération*. C’est une boite de temps destinée à provoquer plus de feedback. Aujourd’hui une itération agile intègre généralement des pratiques de Scrum (démonstration…), de l’Extreme Programming (User Story, planification par la valeur…), de Kanban (tableau Kanban, limiter le Work In Progress…). Tout cela regroupé sous le vocable passe-partout “Scrum” dans lequel une itération est un *sprint*. Quelle idée ! Nous sommes dans une endurance… Rythme soutenable ! Je parlerai donc d’itération. Si vous êtes puriste Scrumien, remplacez par *sprint*.



Agile itératif

Contrairement à une idée très répandue, **le rythme agile est celui des livraisons**, pas celui des itérations, nous verrons cela lors de notre prochaine rencontre. Attention donc : itération après itération, vous n'êtes pas forcément agile.

Revenons brièvement sur les étapes d'une itération, analysées à l'aune des piliers que nous venons d"évoquer.

Le *plan d'itération* consiste à optimiser la valeur produite. Une "astuce" consiste à envisager plusieurs plans et choisir celui qui maximise la valeur. Ce plan tient compte du feedback des itérations précédentes (vélocité, amélioration par feedback des besoins, des pratiques). Le contenu de l'itération est le résultat d'une collaboration entre Client et Développeurs. Le rythme itératif et court est la concrétisation de la quatrième valeur : répondre aux changements.

Le stand-up meeting, chaque jour, est typiquement la session d'auto-organisation de ceux qui font. L'objectif est de terminer des besoins, autrement dit, d'apporter de la valeur.
La revue est un espace d'amélioration du produit par feedback.
Enfin la rétrospective est l'espace d'amélioration des pratiques. Notez que c'est le même principe de feedback concret et rapide qui s'applique, au produit, aux pratiques.

Une autre idée fausse : l'objectif d'une itération est de fournir un *produit partiel potentiellement livrable*. Cette expression masque le rythme agile décrit ci-dessus. Surtout, elle oublie que l'objectif d'une itération est aussi d'obtenir une **nouvelle version du backlog**, constituée d'un nombre pertinent de demandes (généralement user stories) *prêtes* (dont le niveau de qualité est suffisant pour être planifiable dans une itération) :
- suffisamment pour ne pas être en famine,
- pas trop pour ne pas tomber dans le piège cyclenV-esque de la "grosse spécification du début" et donc prendre le risque de stocker des choses inutiles et/ou fausses.

Questions et Actions

Ce paragraphe ponctue le chapitre en vous proposant des sujets de réflexion et des actions. Avant de nous séparer, jusqu'à notre prochaine rencontre, je vous engage vivement à répondre à ces questions. Vos recherches et

réflexions personnelles ont bien plus de valeur et d'effet que votre seule lecture !
Et, je vous en prie, ne faites pas comme moi ! J'ai lu d'innombrables ouvrages contenant des suggestions d'actions en me disant "Ah oui, évidemment !". Lectures suivies de… Zéro actions. Il est prouvé scientifiquement que cela ne sert pas à grand chose.
Agissez et souvenez-vous de ce que disait ma grand-mère : *un peu de quelque chose vaut mieux que beaucoup de rien.*

1. Quelle différence entre "compliqué" et "complexe" ?
J'ai utilisé le terme "complexe" pour préciser le terrain de jeu de l'agile. Ce terme a une définition précise qu'il est important de connaître en particulier pour mieux comprendre la nature de la réponse agile (empirique). Recherchez dans un dictionnaire la définition de ces mots.
Posez-vous la question : en quoi mon domaine est compliqué et/ou complexe ?
De ce point de vue, comment analyseriez-vous les réponses habituelles basées sur de nombreux processus, contrôles et documents ?
Considérez la pertinence (ou pas…) d'une approche *empirique* face à une situation complexe.

2.Sur quels principes est basé le cycle en V (cascade) ?
Si les différentes méthodes agiles proposent des valeurs ou principes[15], il en est de même pour le cycle en V.

15. Scrum est basé sur des "piliers" : Transparence, Inspection, Adaptation. http://www.scrumguides.org/. Une rétrospective dans Scrum consiste à inspecter et adapter les pratiques de l'équipe.

Si une croyance est (devenue) fausse, comment voulez-vous que les pratiques qui en découlent soient pertinentes ?

3. Quels sont vos critères de succès ?
Qu'est-ce qui vous fait dire, à la fin d'un projet ou bien lors d'une mise en exploitation, que là… C'est un sacré succès ! Recherchez ensuite, *dans votre propre expérience*, des situations qui ont permis d'atteindre tout ou partie de ce succès. Quelles étaient les pratiques, les relations, la logistique… ? C'est une pratique d'investigation positive. Le point important est de rechercher des exemples *concrets* et pas des facteurs de succès fantasmés décrits dans tel ou tel livre, y compris celui-ci !, ou méthode.

4. Lisez le Manifeste, les quatre valeurs et les douze principes.
Vous les trouverez sur le site du Manifeste agile[16].
En quoi les valeurs et principes de ce Manifeste corroborent (ou pas…) votre propre expérience ?
Allez… Je vous engage à lire aussi l'histoire du Manifeste (en anglais) accessible depuis la page d'accueil du site.

5. Comment décririez-vous votre métier de Manager ?
En quoi votre description est agile… Ou pas ?

6. Quelles sont, selon vous, les qualités d'un Manager agile ?
Concentrez-vous sur les qualités *spécifiques* au Management agile.

16. http://agilemanifesto.org/iso/fr/manifesto.html

7. Jetez ! Oui, là, tout de suite !

Une transformation agile consiste à faire autrement. Il ne s'agit donc pas de rajouter des rôles, des pratiques mais de *transformer*, transmuer ce qui est déjà là dans l'organisation.

Donc, lancez-vous tout de suite !

Quelle pratique, document, rôle (pas personne, rôle !) est-il temps de jeter ?

8. Quelle pratique simple vous inspire la définition de l'agile de ce chapitre ?

À l'issue de ce chapitre vous avez peut-être une idée de nouvelle pratique. Complexité, VUCA, empirique, valeur, auto-organisation, équilibres… Si vous deviez inventer une mise en pratique de l'une de ces facettes de l'agile, que diriez-vous ?