

PERCEVOIR LE FUTUR

AGILITÉ



IMAGINER

RÉALISER

MESURER

Xebia

SOFTWARE DEVELOPMENT **DONE RIGHT**

Les TechTrends sont l'expression de notre savoir faire ; forgé sur le terrain, auprès de nos clients dans le cadre des projets que nous menons avec eux.

Fruit d'un travail collaboratif de nos consultants, vous y trouverez, nous l'espérons, les nouvelles tendances technologiques et méthodologiques ainsi que l'état de l'art de notre profession.

Nous tentons, dans le cadre de ces publications, de vous dispenser des conseils directement opérationnels afin de vous guider dans les décisions stratégiques que vous avez à prendre.

Distribués à plusieurs milliers d'exemplaires tous les ans, la collection des TechTrends s'étoffe régulièrement de nouveaux ouvrages.

Luc Legardeur, Président



IMAGINER



RÉALISER



MESURER

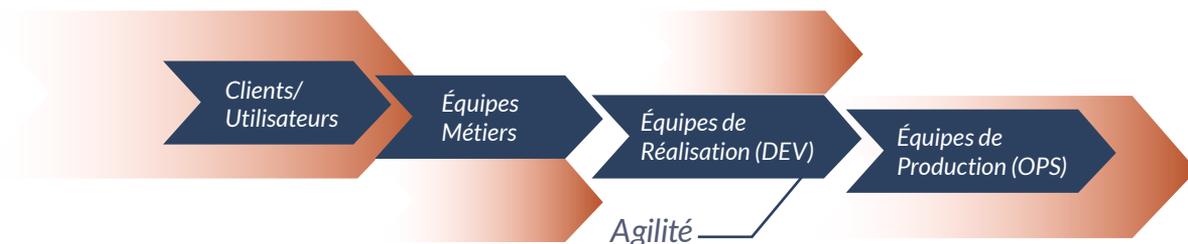
L'AGILITÉ EN MUTATION

Le virus de l'agilité

De l'eau a coulé sous les ponts depuis les premières tentatives d'équipes Scrum en France au début des années 2000. À cette époque, les équipes de développement cherchaient à transformer les chefs de projet technique en Scrum Masters, les MOA en Product Owner. Aujourd'hui, la culture de l'agilité est ancrée dans les mœurs, les PO et facilitateurs sont recrutés pour ces métiers. Le résultat est flagrant dans nos organisations : le management visuel, l'esprit d'équipe, la démarche d'amélioration continue sont maintenant des acquis pour de nombreuses équipes.

Les méthodes agiles sont arrivées à maturité et les équipes sont potentiellement efficaces. Potentiellement seulement, car le travail accompli jusqu'ici n'a optimisé qu'un maillon. Le développement n'est plus le goulot d'étranglement de la chaîne de valeur de la production logicielle.

L'étape suivante consiste à considérer le processus de production logicielle dans son ensemble. Nous constatons une propagation de l'agilité aux interfaces des équipes de réalisation : vers le métier et vers les équipes de production.



Processus de production logicielle

Concernant les équipes métiers, l'agilité a apporté de grands chamboulements, parfois incompris. **Les départements métiers réalisent de plus en plus l'intérêt de travailler avec des équipes agiles.** Après une phase d'adaptation, parfois un peu difficile, ils comprennent et adoptent ce nouveau fonctionnement : travailler par itération, obtenir du feedback, prioriser les demandes en fonction de leur valeur business, limiter le nombre de sujets en parallèle.

Les besoins des utilisateurs évoluent de plus en plus rapidement, le métier les structures donc en petits lots d'évolutions qualitatives. Il faut livrer souvent sans gêner l'utilisateur : c'est là que la partie en aval du développement doit être considérée. Alors que les équipes de production cherchent à sécuriser, les développeurs sont avides de nouvelles technologies et de changements permanents. **Ces frères ennemis de toujours apprennent maintenant à travailler main dans la main. Il s'agit du mouvement DevOps** (le TechTrends #2 était consacré à ce mouvement) qui prône la **collaboration** et le **partage d'outils et de méthodes** pour s'adapter et améliorer les délais de livraison.

Des méthodes qui trouvent un écho au delà des frontières du développement logiciel

L'adoption de l'agilité par des équipes qui travaillent au contact d'équipes agiles est nécessaire et assez évidente. Il faut cependant noter une autre forme d'adoption qui tend à prouver que l'agilité répond à un besoin. Il existe une tendance non négligeable, qui représente une petite partie des demandes de formations agiles : **de plus en plus de personnes se forment à Scrum et à Kanban avec la volonté d'appliquer les méthodes à leur métier, parfois très loin de l'univers du développement.**

En effet, la cohésion d'une équipe Scrum, la régularité des points de synchronisation, la chasse au gaspillage du temps sont des problèmes qui parlent en dehors des frontières de nos métiers. Il est fréquent que l'on entende : « **nous aussi on veut faire du Scrum !** ».

Renouvellement de l'ADN

L'agilité a vécu plusieurs vagues d'adoption significatives depuis une quinzaine d'années. La crise économique fût le moteur principal des 2 dernières vagues :

- en 2008, sur fond d'offshore pour les plus grosses organisations,
- en 2011, sur fond de réinternalisation des développements.

En 2014, l'adoption est poussée par une accélération technologique sans précédent : infrastructures virtuelles disponibles rapidement, analyse de gigantesques volumes de données, distribution massive sur les App Stores, marketing viral, etc. Beaucoup peuvent se lancer sur des marchés peu exploités avec un investissement minime et juger très rapidement de leur chance de succès. Des réussites fulgurantes, à l'instar de AirBnB, peuvent fortement bousculer des secteurs aussi bien établis et verrouillés que pouvait l'être l'hôtellerie.

La conséquence première de cette accélération : le terme agile paraît quelque peu galvaudé. Les organisations ayant adopté l'agilité voilà quelques années cherchent **de nouveaux axes de progression**. Les organisations n'ayant pas encore sauté le pas ne se contentent pas de vouloir être agiles. Toutes cherchent à **combinaisonner l'agilité à des axes d'optimisation tels que agile et réactif, agile et prédictible, agile et productif, agile et collaboratif, agile et global, agile et compétitif, etc.**



Il existe une tendance non négligeable, qui représente une petite partie des demandes de formations agiles : de plus en plus de personnes se forment à Scrum et à Kanban avec la volonté d'appliquer les méthodes à leur métier, parfois très loin de l'univers du développement.



Ce besoin d'aller au delà du simple terme agile rejaille sur les méthodes historiques. Principalement, les méthodes les plus populaires que sont Scrum et XP ne se suffisent plus à elles-mêmes. Popularisées par leur simplicité et la marge de manœuvre qu'elles laissent à l'adaptation, elles souffrent aujourd'hui de ce manque de précision dans des situations avec de fortes contraintes : budget/délais faibles, organisations lourdes, besoins instables, utilisateurs volatiles, business models imprévisibles, etc. **Nous observons un fourmillement d'outils de pratiques ou de méthodes qui s'avèrent indispensables pour compléter les approches en place et pérenniser l'agilité de votre organisation.** Que ce soit Lean Startup, Kanban, DevOps et bien d'autres encore, toutes viennent bousculer le status quo de l'agilité et se rendent incontournables. Elles donnent même naissance à **de nouveaux hybrides, tel que ScrumBan**, mariage de Scrum et Kanban.

Ainsi, l'agilité s'intègre petit à petit dans votre culture d'entreprise. De nouveaux métiers apparaissent, de nouveaux processus de gestion de produit remplacent les anciens, de nouvelles pratiques agiles bousculent le quotidien de vos équipes. Les enjeux aujourd'hui consistent à aller encore plus loin **en intégrant l'agilité dans l'imagination** même du produit et à fournir des indicateurs permettant une meilleure visibilité sur vos projets. Pour ce faire, forts d'une dizaine d'années d'expérience dans ce domaine, nous partageons avec vous dans ce TechTrends notre vision de l'agilité et de ses éléments clés au travers des 3 piliers du développement d'un produit : **imaginer, réaliser et mesurer.**

“ *Que ce soit Lean Startup, Kanban, DevOps et bien d'autres encore, toutes viennent bousculer le status quo de l'agilité et se rendent incontournables.* ”



IMAGINER

Les métiers de MOA / MOE laissent progressivement la place à celui de Product Owner dont il ne faut pas sous-estimer l'importance.

En effet, une simple utilisation des méthodes traditionnelles agiles telles que XP ou Scrum n'est aujourd'hui plus suffisante pour produire le bon produit, au bon moment. Votre PO doit inévitablement voir plus loin, imaginer des solutions encore plus innovantes. Pour ce faire, donnez-lui la marge de manœuvre nécessaire pour s'adapter aux changements et misez sur de nouvelles approches comme le Lean Startup et le Design Thinking. Cela vous permettra de vous tourner plus facilement vers une gestion de produit agile.

La professionnalisation du métier de Product Owner

Un peu d'histoire : client sur site et Product Owner

XP (eXtreme Programming) a été la première méthode agile à être mise en œuvre en France et reste très proche de Scrum sur bien des plans. XP s'appuie sur un rôle alors nouveau, le client sur site, qui :

- Porte la vision du produit à fabriquer.
- Est la voix du client / des utilisateurs.
- Priorise le travail de l'équipe de développement.
- S'appuie sur des User Stories, qui sont une rupture dans l'expression de besoins par rapport à des spécifications fonctionnelles.
- Est au contact de l'équipe pour réduire au maximum la boucle de feedback entre le développement et la ligne produit.

Client sur site et Product Owner sont extrêmement proches et des personnes en France ont assumé ce rôle dès le début des années 2000. Après plus d'une dizaine d'années, il semble logique que le rôle arrive à maturité.

Arrêtons de parler de projet : la culture produit est partout en 2014

Le modèle en silos MOA / MOE est dépassé et, de plus, purement français. Il n'a pas cours dans les petites structures : l'équipe de développement de produits de qualité y est minimale. Le Product Owner est au contact des utilisateurs (ou des clients) et de son équipe de développement. Les Startups ne gèrent pas un projet, elles développent un produit qui vise à satisfaire ses utilisateurs.

Ainsi, créer ou instaurer des postes de Product Owner, c'est assumer de replacer l'émergence de produits au centre des investissements. Vos POs ne se contentent pas de décrire des besoins, ils doivent avoir l'obsession de délivrer des fonctionnalités de grande qualité qui raviront vos utilisateurs. **Un bon produit n'est pas un patchwork de fonctionnalités maladroitement emboîtées les unes aux autres. C'est un produit aux choix forts,** qui ne contient peut-être que très peu de fonctionnalités, mais celles qui rendent le meilleur service à vos utilisateurs. Il doit être plaisant à utiliser et ergonomique. Montrez que votre produit est de grande qualité et avance : le « mode projet » passera au second plan.

D'autre part, **vos utilisateurs ont maintenant, même inconsciemment, une culture du produit.** Ils sont de plus en plus exigeants : qui peut se permettre de proposer des applications professionnelles de moindre qualité que celles disponibles sur tous les stores d'applications ?

Intégrez le Product Owner comme un poste à part entière

Le marché de l'emploi se fait l'écho de l'adoption progressive du rôle en tant que métier

Dès lors que le rôle est adopté par les grandes organisations, il apparaît alors comme un métier à part entière. Cela commence à se ressentir dans les offres d'emploi. Ces postes sont sans ambiguïté des postes de Product Owner : le titre ne fait pas allusion à la MOA et les compétences recherchées exhibent clairement le contexte agile (on y parle de Backlog, de User Stories, de critères d'acceptation, d'être garant de la vision du produit, etc.).

Un profil encore rare et aux multiples compétences

Le rôle du Product Owner est essentiel à la réussite des applications que vous créez. C'est une personne qui doit être capable de parler à différentes populations.

Ainsi, même s'il n'a pas à être technique, il doit pouvoir comprendre le travail quotidien des ingénieurs de développement. La sensibilité aux bonnes pratiques de développement et à la qualité logicielle permet également de promouvoir l'importance de celles-ci tout au long du développement. Parallèlement à cela, il doit avoir une sensibilité à l'expérience utilisateur, au design, bref, être capable de parler avec des profils User eXperience (UX). **Il doit aussi capter les réels besoins de ses utilisateurs, rarement exprimés naturellement.**

Enfin, **il doit savoir parler à ses managers**, pour communiquer correctement sur l'avancement du produit et défendre le travail de l'équipe.

Toutes ces qualités mises bout à bout font émerger la nécessité de confier le poste à **une personne aux compétences rares.**

Donnez à vos Product Owners les moyens de réussir

Vous pratiquez déjà une méthode agile ? Si tel est le cas, vous devez savoir mettre votre ou vos Product Owner(s) en situation de réussite. Pour ce faire, il convient d'éviter les anti-patterns liés au rôle.

Un bon Product Owner est **légitime**, il a un pouvoir de décision suffisant sur son périmètre fonctionnel pour être en mesure d'effectuer rapidement les bons choix, de manière autonome. **Forcez-vous à la co-localisation** : le Product Owner est au contact de l'équipe, c'est comme cela que vous obtiendrez les meilleurs résultats.

De manière bienveillante, **vous devez faire confiance au Product Owner** : laissez lui les rênes sur le périmètre que vous lui confiez. En contrepartie, demandez lui clairement de la transparence sur l'avancement du travail de l'équipe dont il fait partie intégrante.

Ouvrez-vous à des pratiques, outils et disciplines périphériques

Les outils du Product Owner sont encore trop rarement maîtrisés

Nous savons qu'il est très difficile d'assurer une bonne qualité des logiciels en faisant l'impasse sur les bonnes pratiques d'ingénierie. De la même manière, il est très compliqué de s'engager sur le développement du bon produit sans maîtriser les bonnes pratiques qui devraient être classiques à l'heure actuelle.

Malheureusement, nous constatons tous les jours que ce n'est pas encore le cas. La boîte à outils minimale du Product Owner est pourtant légère. À des degrés divers, selon l'étendue du périmètre qui lui est confié, un Product Owner digne de ce nom se doit de les maîtriser.

- Les User Stories sont encore bien souvent des spécifications déguisées : trop verbeuses, pas assez granulaires, fastidieuses à écrire et à comprendre. Une User Story doit être le support strictement nécessaire et suffisant pour décrire le travail à faire. Plus qu'un contrat, c'est **une invitation à la discussion** entre le Product Owner et les développeurs.
- Un bon wireframe vaut bien des phrases et pourtant certains Product Owners ne font pas l'effort d'en faire.
- **Pratiquer la méthode du BDD** permet de franchir un cap de qualité pour les User Stories et leur développement mais son adoption n'est que naissante.

- **Les personas** ne sont pas systématiques. Ils offrent pourtant un format simple et efficace qui permet de se forcer à s'identifier aux utilisateurs.
- **La Story Map** est incontournable pour s'assurer facilement de la bonne priorisation d'un backlog fourni, pour autant elle n'est bien souvent pas maîtrisée.
- Mettre en place et maintenir **une roadmap réaliste et flexible** devrait être le B A BA. D'expérience, il reste du chemin à parcourir.



La boîte à outils du Product Owner

Les applications que vous concevez requièrent un profil global

Un Product Owner, ne peut plus se contenter de la seule approche agile. Même si ces méthodes ont révolutionné le développement de logiciel, la production de produits de qualité n'est plus un luxe. Il faut dorénavant innover toujours plus et toujours plus vite. Pour ce faire, il faut savoir s'adapter, prendre des risques, apprendre de ses erreurs et surtout apprendre à moindre coût pour pouvoir avancer dans l'inconnu d'un marché complexe. **En clair, après avoir rendu le développement agile, il faut maintenant rendre le management de produit agile.**

Agilité dans l'imagination du produit

Afin d'être toujours plus pro-actifs sur le marché, les Startups et les métiers soumis à une forte concurrence plébiscitent deux nouvelles approches agiles axées sur le produit : le **Lean Startup** et le **Design Thinking**. Ces approches s'intègrent naturellement avec des méthodes agiles lorsque les besoins sont très instables ou que le marché est incertain.

Adoptez la méthode du Lean Startup

L'objectif du Lean Startup, dans la continuité des méthodes agiles, est de réduire fortement le cycle de développement d'un produit. Le fait d'obtenir rapidement des feedbacks des clients permet de limiter les risques d'échecs. De cette façon, les entreprises, en particulier les Startups, peuvent concevoir leurs produits ou services pour répondre aux besoins de leurs clients sans qu'un important budget de démarrage ou de coûteux lancements de produits ne soient engagés.

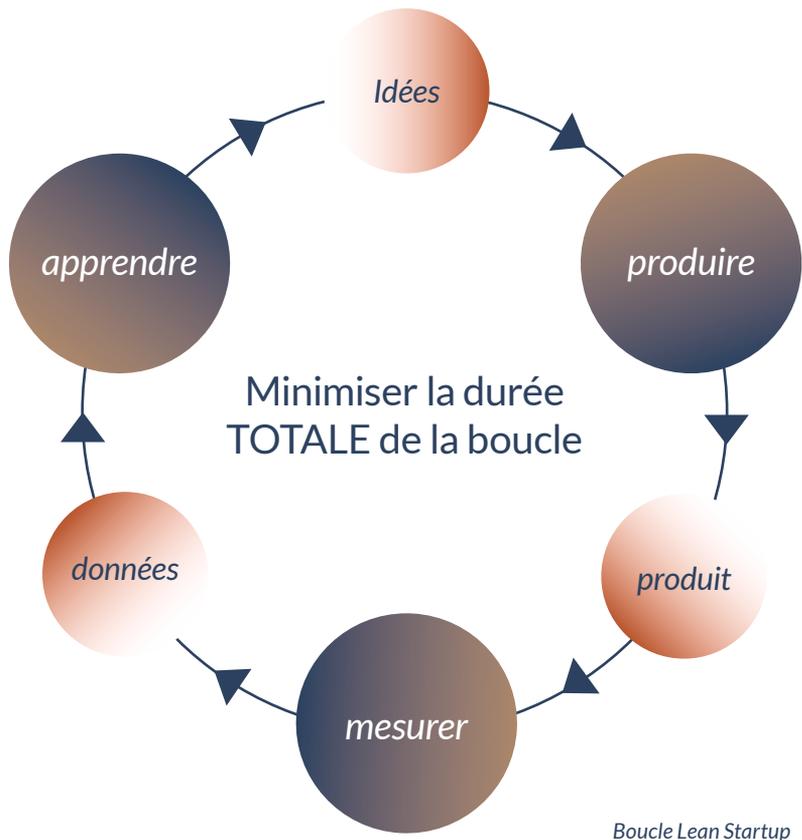
Cette méthode remet en cause les méthodes traditionnelles fondées sur des mythes comme :

- « Nous savons ce que le client veut ».
- « Nous pouvons prédire les revenus ».
- « Avancer selon le plan c'est progresser ».

À l'instar des préceptes du Lean Management, la **philosophie Lean Startup cherche à éliminer les gaspillages et accroître la création de valeur**. L'objectif est d'obtenir les meilleures chances de réussite sans engager un financement important, sans Business Plan finalisé ou produit parfait. Eric Ries, créateur de l'approche, estime que les retours des clients au cours du développement du produit font partie intégrante du processus Lean Startup. **Il faut veiller à ne pas investir trop de temps à concevoir des fonctions ou des services que les utilisateurs ne veulent potentiellement pas.**

L'approche repose principalement sur :

- la validation des hypothèses,
- des expérimentations scientifiques,
- un apprentissage constant,
- des déploiements en production les plus fréquents possible.



Le Lean Startup est une méthode d'amélioration continue du produit en 3 étapes :

- **Produire** : tout produit est un ensemble d'hypothèses qu'il faut tester à l'aide d'un MVP ou Minimum Viable Product. Ce MVP est la version la plus simple d'un nouveau produit. Il permet à une équipe d'apprendre à moindre coût par la validation des clients.
- **Mesurer** : la validation des hypothèses se fait de façon scientifique avec des méthodes aussi bien qualitatives (interview des clients) que quantitatives (A/B testing).
- **Apprendre** : l'objectif est d'apprendre quelles fonctionnalités apportent de la valeur, au contraire, celles qui créent du gaspillage. Mais pas seulement ! La méthode permet aussi de tester le marketing du produit ou encore son Business Model. Si les indicateurs sont bons, c'est le signe qu'il faut persévérer et continuer d'améliorer l'existant. Au contraire, s'ils sont mauvais, il faut alors savoir pivoter (changer son segment client, son Business Model, etc.).

Mettez l'expérience utilisateur au centre de vos décisions avec le Design Thinking

Historiquement, le design est souvent considéré comme l'une des dernières étapes au sein du processus de développement logiciel. En réalité, s'il était abordé plus tôt, il pourrait mener à des innovations au delà du périmètre de l'esthétisme. En effet, un produit à succès n'est pas le premier à arriver sur le marché mais plutôt le premier qui nous intéresse de façon émotionnelle et fonctionnelle. Ce constat va avoir de plus en plus d'importance dans le futur au vu de l'abondance des services. **L'utilisateur ne cherche plus simplement un ensemble de fonctionnalités mais une expérience**, qui correspond à la combinaison complexe de fonctionnalités bien pensées, de services cohérents, d'image de marque et d'émotions. **Le Design Thinking permet d'imaginer ces expériences.**

Il s'agit d'une méthodologie **centrée sur l'humain** qui englobe le spectre complet **des activités d'innovation**. L'innovation dans le cadre du Design Thinking rompt le mythe de l'inventeur de génie solitaire. L'approche se base sur **le travail collaboratif de l'équipe**, elle se veut extrêmement **itérative** et donne le **droit à l'erreur** afin d'apprendre rapidement. On distingue 3 espaces :

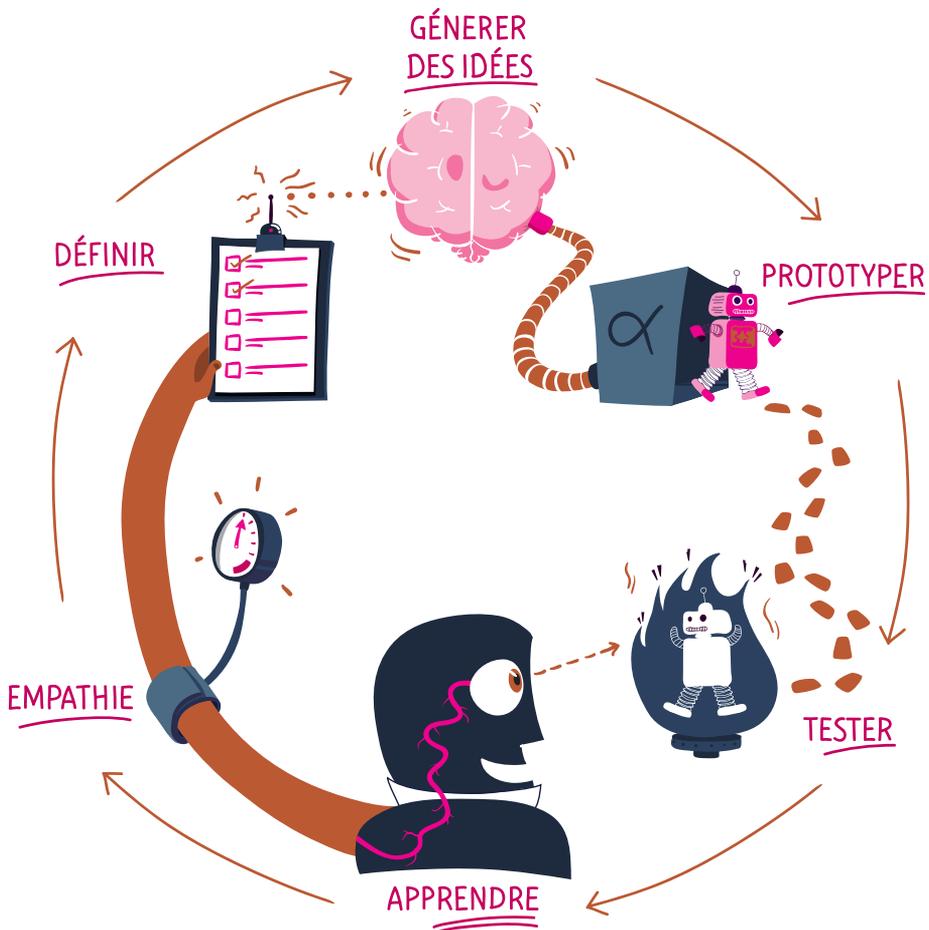
- **Inspiration** : quel est le problème du métier ? Où est l'opportunité ? Quel est le besoin des utilisateurs ? Il faut savoir impliquer différentes disciplines dès le début.
- **Idéation** : processus de génération, de prototypage et de tests d'idées. Il faut mettre l'utilisateur au premier plan et savoir communiquer sur ces expérimentations.
- **Implémentation** : exécution de la vision jusqu'au marché (en aidant le marketing à concevoir une stratégie de communication et en plaidant la cause de cette vision auprès du métier).

“ *L'innovation dans le cadre du Design Thinking rompt le mythe de l'inventeur de génie solitaire.* ”

La Design School de Stanford représente le Design Thinking en une série d'étapes :

- **Faire preuve d'empathie** : observer les utilisateurs en situation réelle, s'immerger dans leur contexte.
- **Définir le problème** à résoudre selon un point de vue précis.
- **Générer des idées** qui peuvent être des alternatives radicales de conception.
- **Créer**, dans la mesure du possible, **des prototypes** avec lesquels les gens peuvent interagir.
- **Tester les solutions** afin d'obtenir les avis et les réactions des utilisateurs.

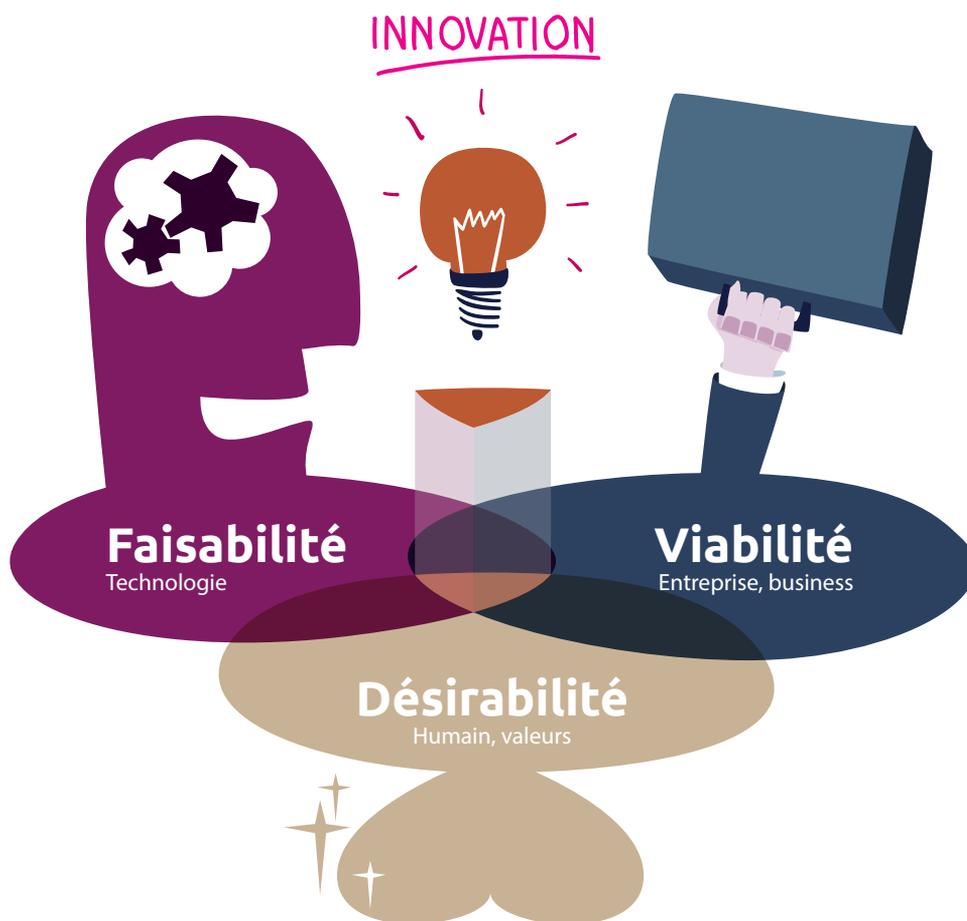
Nous y ajoutons une dernière étape qui est celle de l'apprentissage. Celle-ci nous permet de créer une boucle et de repartir sur des tests utilisateurs qui vont permettre de préciser le problème, de générer de nouvelles idées et donc un nouveau prototype qui va pouvoir être testé et ainsi de suite.



Boucle Design Thinking

Pour conclure, le Design Thinking est extrêmement intéressant pour le management de produit car il permet :

- d'explorer plus d'idées,
- de créer une innovation qui reflète mieux ce que veulent les utilisateurs,
- de faire émerger les besoins ou problèmes profonds des utilisateurs,
- d'encourager les équipes à faire des expérimentations rapides,
- de chercher des opportunités de co-crée avec les clients.



L'innovation créée par le Design Thinking

Le Design Thinking a pour objectif de réunir le côté désirable du produit auprès du client, sa faisabilité technique mais aussi sa viabilité business afin de créer l'innovation.

Créez une gestion de produit agile

Le Lean Startup et le Design Thinking ont beaucoup de points communs. Ces deux méthodologies ont pour objectif de **mieux comprendre le besoin des utilisateurs**. Tester des hypothèses afin d'apprendre au contact direct des utilisateurs, de façon itérative et le plus rapidement possible permet de créer le bon produit. Ainsi, rendre votre management de produit plus agile va vous permettre de **vous adapter efficacement aux changements rapides du marché, d'appréhender l'inconnu d'un environnement** de plus en plus complexe, **voire d'être plus innovant** que vos concurrents.

Nous avons présenté en particulier ces deux approches mais n'hésitez pas à vous inspirer d'autres méthodes comme la Frugal Innovation, le Prototyping ou encore l'effectuation.

TAKE AWAY AGILE TRENDS



IMAGINER

- Créer une équipe soudée autour d'un pilier central qu'est le PO
- Donner aux POs les moyens logistiques / humains de réussir
- Faire confiance à vos POs dans la prise de décision
- Mettre l'utilisateur au centre des réflexions et de toute prise de décision relative au produit



RÉALISER

Avec de bonnes méthodes pour imaginer votre produit, la réalisation constitue toujours un challenge afin de livrer vite et juste,

d'autant plus si le produit en question doit s'insérer dans un SI complexe et présenter de nombreuses fonctionnalités. Voyons ici trois approches pouvant répondre à ces problématiques : la méthode Kanban, l'organisation en Feature Teams et la transformation agile à grande échelle.

L'émergence de Kanban

Après le culte du sprint, le culte du flux

Dans un contexte de recherche d'alternatives ou de compléments aux méthodes agiles classiques, il faut noter **la popularité croissante de la méthode Kanban**. Encore moins normatif que Scrum ou XP, Kanban se base sur **5 principes simples et ne définit aucun rôle, activité ou livrable**. Il se combine donc facilement avec ses prédécesseurs Scrum ou XP qui peuvent servir de point de départ, mais peut également être utilisé seul sur n'importe quel processus existant. Ceci étant, nous constatons de plus en plus des phénomènes de bascule vers Kanban dans le prolongement d'une démarche Scrum qui s'essouffle.

Kanban se base sur le concept suivant : **la réalisation est organisée en flux tiré**, c'est à dire que chaque étape de réalisation est cadencée par la performance de l'étape suivante. Par exemple, on ne produit pas plus pour le code que ce que l'on peut tester. Il s'agit alors **de définir des limites de travail en cours** pour chacune des étapes. Lorsque la limite est atteinte, on ne cherchera pas à produire plus. Ceci permet de conserver **une fluidité optimale dans le processus de réalisation**. Nous pouvons comparer ceci à la vitesse à laquelle nous pouvons rouler sur une autoroute : afin de rouler à une vitesse proche de la limitation et donc diminuer le temps de parcours, il faut limiter le nombre de véhicules sur les voies. Kanban ne cherche pas à optimiser uniquement le nombre de véhicules à l'arrivée mais également les temps de parcours et tout ceci au moindre coût.

La méthode permet également **une dynamique d'amélioration** basée sur la mesure stricte de la performance du processus et l'analyse des résultats. Par exemple, si les limites sont trop basses, les temps de réalisation seront très rapides mais le nombre d'éléments livrés trop faible. Kanban met également **un accent très fort sur le management visuel** qui permet de faciliter la prise de conscience de la situation et les décisions nécessaires à l'optimisation du processus.

Adoptez Kanban pour booster votre agilité

Nous constatons que Kanban permet d'aller un cran plus loin que Scrum dans l'efficacité des pratiques agiles en rompant le carcan du sacro-saint Sprint fixe. Loin de provoquer un retour en arrière, **Kanban bouscule indéniablement le status quo**. Des pratiques agiles historiques tombent en désuétude : l'estimation n'est plus aussi importante et fait place à **la prédictibilité** (ce point sera abordé plus en détail dans la partie « Mesurer »), le planning s'efface au profit **d'une priorisation régulière**, le découpage en tâches peut être évité par la création de **User Stories de tailles standardisées**.

La méthode permet également d’embrasser plus largement la chaîne de production logicielle, là où la méthode Scrum se focalise sur un périmètre plus restreint.

Le passage à grande échelle est également facilité avec Kanban, car l’approche permet de construire des organisations complexes avec un kit de base assez simple. Finalement, Kanban est un peu « **le Kapla de l’agilité** ».

Feature teams

Il était une fois le modèle Spotify

Depuis la publication de Henrick Kniberg et Anders Ivarsson en Novembre 2012, **le modèle Spotify est devenu « LE » modèle organisationnel à la mode : toutes les compétences réunies dans une même équipe - la feature team** - pour réaliser un besoin fonctionnel de bout en bout. Cela vous fait rêver ? En tout cas, nos clients nous sollicitent de plus en plus pour ce type d’accompagnement. Parfois, la décision d’appliquer le modèle Spotify est même déjà prise avant notre intervention.

Pourquoi passer en Feature Teams ?

Dans la majorité des transformations agiles que nous accompagnons, nous coachons des équipes, parfois déjà en Scrum, fragmentées par couches techniques (le CMS, le Front du site, etc.) ou par disciplines (designers, testeurs, développeurs, etc.). Nous appelons cela **une organisation par composant**. **Héritage du cycle en V ? Peu importe.**

“ *Le modèle Spotify est devenu « LE » modèle organisationnel à la mode : toutes les compétences sont réunies dans une même équipe : la Feature Team.* ”

Imaginons un site eCommerce grand public multi-canal (web, tablette et mobile), alimenté par un système d'information centralisé pour la partie client, un CMS pour la gestion de contenu et présentant l'information avec une ergonomie moderne. Ce produit vous fait penser à quelque chose ? Et bien voici à quoi va ressembler l'organisation en mode composant des équipes de réalisation :

- une équipe marketing,
- une équipe de designers,
- une équipe de développement de la couche de services s'interfaçant avec le SI (souvent en Java),
- une équipe de développement de la partie CMS (Drupal, WordPress, Joomla!, etc.),
- une équipe de développement du Front du site Internet (souvent en HTML5 et JavaScript),
- une équipe de développement de la partie mobile (iOS et Android),
- une équipe de tests,
- une équipe qui s'occupe de l'infrastructure et des déploiements,
- une équipe de chefs de projets pour piloter le tout.

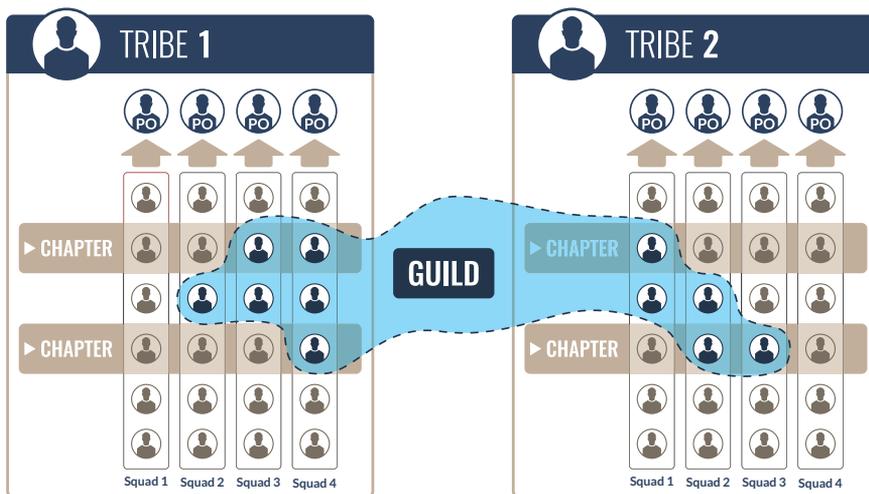
Avec cette organisation, comment va se dérouler la réalisation du produit ? Que l'on soit en agile ou non, la phase initiale va consister à spécifier et planifier afin d'identifier les adhérences entre toutes ces équipes et le staffing qui en découle. Qu'entend-on par adhérence ? C'est une dépendance entre plusieurs équipes, un point de rencontre, une livraison, un jalon qui, s'il n'est pas respecté, va provoquer un blocage ou un ralentissement de l'activité de réalisation du produit. Bien entendu, cette planification n'étant qu'une estimation, fausse par définition, ces adhérences seront très compliquées, voire impossibles à respecter. Par exemple, le contrat d'interfaces des services livrés ne correspondra pas aux spécifications et le branchement « des tuyaux » ne fonctionnera pas, les maquettes d'interface ne contiendront pas tous les cas d'affichage et des éléments graphiques ne seront pas chartés. **Un phénomène va alors se mettre en place : des allers-retours entre équipes empêchant toute prédictibilité sur la date de fin de réalisation.** Les chefs de projet vont donc passer leur temps à gérer des problèmes, arbitrer, replanifier et mettre en place des solutions de contournement afin que les équipes staffées soient alimentées.

C'est là où les Feature Teams prennent tout leur sens : **éviter ce gaspillage avec des équipes autonomes et auto-organisées ayant toutes les compétences en interne pour réaliser un besoin fonctionnel de bout en bout.** Dans notre exemple de site eCommerce, imaginez une équipe responsable du tunnel d'achat web, allant du SI jusqu'au Front en passant par le CMS, composée d'un Product Owner issu du marketing, un designer, 4 développeurs pluri-disciplinaires Java/Drupal/JavaScript et un testeur.

Imaginez également des communautés de pratiques, sorte d'instances où tous les développeurs Java, tous les Product Owners ou tous les designers se rencontrent entre eux et prennent le temps de discuter des bonnes pratiques afin d'homogénéiser le produit et d'augmenter la qualité.

C'est ça les Feature Teams ! Toute personne ayant vécu les difficultés inhérentes à la réalisation d'un produit comme notre site eCommerce sera sensible aux évidentes promesses envoyées par ce modèle organisationnel. C'est pour cela que le modèle a autant de succès. Les produits sont devenus tellement gros, les SI tellement complexes, les besoins d'interactions et de réactivité tellement importants que **le pilotage d'organisation en mode composant est devenu ingérable**.

Le modèle Spotify est une implémentation en Feature Teams. Dans les grandes lignes, la publication de Kniberg définit le vocabulaire d'une **organisation matricielle** (Squads, Tribes, Chapters et Guilds) et explique comment les adhérences entre les équipes ont été réduites au minimum par la **stratégie de découpe fonctionnelle des équipes**. C'était le « vrai » premier retour d'expérience concernant la mise en place de Feature Teams à grande échelle. C'est pour cela que le modèle Spotify est devenu la référence.



Malheureusement, la publication ne dit pas tout. Elle n'aborde pas les sujets très vite soulevés lors de l'implémentation du modèle tels que les instances et la fréquence des rencontres, les différentes stratégies de découpe des équipes, les stratégies de livraison, les pratiques d'ingénierie à mettre en place, etc. Et on ne sait absolument pas comment implémenter concrètement le modèle chez soi.

Bref, à la fin de la lecture, nous restons sur notre faim.

Etes-vous prêt pour le modèle Spotify ?

Après plusieurs transformations, nous lançons un avertissement à toute personne l'envisageant : **attention à l'engouement autour du modèle Spotify, c'est un modèle organisationnel très compliqué à implémenter.** Les impacts humains et techniques sont largement sous-estimés. Le chantier est risqué, avec un coût d'entrée élevé. De plus, il existe une multitude de solutions d'implémentation. Malgré les gains que sa mise en place peut apporter, ce n'est pas la réponse à tous les problèmes. **Le modèle n'est pas applicable dans certains contextes ou certaines cultures d'entreprise.** Cette transformation n'est pas magique et comme l'agilité, les Feature Teams ne sont pas une fin en soi, il ne faut pas les idéaliser.

C'est pour cette raison que nous préconisons de prendre le temps d'étudier en détail la faisabilité et d'estimer le coût du passage. La formalisation des sources d'insatisfaction dans l'organisation actuelle, des objectifs à atteindre, de la dette technique à régler, etc. permet de savoir si la mise en place d'une telle organisation correspond vraiment au besoin. Il y a peut-être d'autres axes d'amélioration à considérer avant.

Pour vous aider à qualifier votre éligibilité au modèle, voici les points à considérer avant de prendre la décision de transformer l'organisation.

Pourquoi voulez-vous passer en Feature Teams ?

La question est assez basique mais reste sans doute la plus importante : **revenez au besoin.** Les interviews constituent un très bon outil. Vous pourrez alors identifier d'éventuels décalages avec votre perception des problèmes. Dans la mesure du possible, soyez factuel et évitez les ressentis. D'autres solutions, plus pertinentes, plus rapides ou moins coûteuses pourront potentiellement émerger.

Dans tous les cas, la formalisation permettra de se remémorer **le pourquoi de la transformation** dans les passages difficiles : les sources d'insatisfaction dans l'organisation actuelle ainsi que les objectifs à atteindre sont des « must have ».

Quels sont vos produits / vos canaux et comment sont organisées vos équipes ?

Qu'est-ce qu'un produit et un canal ? Bien souvent, lorsque l'on pose la question à nos clients, ils nous répondent qu'ils n'ont qu'un seul produit. Par contre, ce produit est décliné sur plusieurs canaux : web, mobile et tablette. Dans ce cas, il n'y a effectivement qu'un seul produit d'un point de vue marketing. Par contre, il y en a au moins 5 d'un point de vue informatique (site Internet, mobile Iphone, mobile Android, tablette Iphone, tablette Android, etc).

Le nombre de produits informatiques a un impact direct sur le choix de la stratégie de découpe des équipes.

Toujours dans l'optique de choisir l'axe de découpe, l'organisation actuelle de vos équipes permet d'affiner la vision de la transformation. Les équipes de développement sont-elles organisées par couches techniques (équipe CMS, équipe Front, etc.) ? Par fonctionnalité ? Par canal ?

Le problème sous-jacent est de savoir où se trouvent les équipes, s'il y a des adhérences entre elles et comment elles sont gérées. Par exemple, le modèle Spotify préconise que chaque équipe (Squad) et communauté de pratiques (Chapter) d'une tribu soient **colocalisées**.

La vision des produits et de l'organisation de vos équipes vous permet de dresser un **état des lieux indispensable pour choisir la stratégie de découpe**. À partir de là, en fonction de ce que vous avez identifié, plusieurs possibilités s'offrent à vous. Il faut néanmoins toujours garder en tête que l'autonomie d'une équipe (nous entendons par là sa capacité à réaliser le besoin fonctionnel de bout en bout) reste le critère le plus important : **idéalement, chaque équipe est autonome**.

La stratégie de découpe dépend du contexte. Ce choix est primordial pour la réussite de la transformation. Une fois identifiée, il faut constituer les équipes et répartir les personnes. C'est ici que la résistance au changement est la plus forte. Les personnes ont l'habitude de travailler ensemble. Cette transformation change fortement les habitudes. Un suivi personnalisé, une communication permanente et une transparence absolue sont obligatoires.

 *Il faut néanmoins toujours garder en tête que l'autonomie d'une équipe (nous entendons par là sa capacité à réaliser le besoin fonctionnel de bout en bout) reste le critère le plus important : idéalement, chaque équipe est autonome.* 

Quelle est votre culture d'entreprise / comment communiquer ?

Durant la transformation, les personnes vont être inquiètes. Vous allez sans doute les changer d'équipe, de bureau, bouleverser leurs habitudes, leur demander de communiquer différemment, d'appliquer de nouvelles pratiques à la fois techniques et organisationnelles. Pour certaines personnes, il n'y aura pas de problème. Pour d'autres, cela peut être très mal vécu. Bref, **si vous n'avez jamais été confronté à ce que l'on appelle la résistance au changement, vous allez comprendre !**

Comment procéder pour que la transformation se fasse le plus en douceur possible avec un maximum de gens moteurs et sans en perdre en route ?

Il faut communiquer ! Nous ne vous parlons pas d'une annonce isolée par email, mais **d'un vrai plan de communication**. Pour réussir, vous aurez besoin de communiquer en permanence, à différents niveaux, de différentes manières, d'être transparent, d'expliquer, de rassurer, de donner de la visibilité sur les résultats obtenus et de rappeler pourquoi la transformation était nécessaire (premier point ci-dessus).

Un plan de communication se construit et se prépare à l'avance. Soyez prudent, le manque de communication peut causer l'échec de la transformation.

Comment gérez-vous la réalisation du / des produit(s) ?

Pour pouvoir piloter la production d'une organisation en Feature Teams, **il est nécessaire de maîtriser l'avancement fonctionnel de son produit**. Vous devez identifier l'équipe en charge de trouver les prochaines fonctionnalités, attribuer les responsabilités d'arbitrage et de priorisation, définir les premiers indicateurs de gestion de produit, organiser vos backlogs ou encore rédiger les Users Stories.

Autant de prérequis, qui nécessitent une certaine maturité dans la gestion de produit en agile, mais sans lesquelles une organisation passera d'une production maîtrisée à une activité subie.

“ *Il faut communiquer ! Nous ne vous parlons pas d'une annonce isolée par email, mais d'un vrai plan de communication. Pour réussir, vous aurez besoin de communiquer en permanence, à différents niveaux, de différentes manières, d'être transparent, d'expliquer, de rassurer, de donner de la visibilité sur l'avancement, les résultats obtenus et de rappeler pourquoi la transformation était nécessaire.* ”

Comment se déroule la mise en production ?

Un des challenges majeurs d'une organisation en Feature Teams est le **déploiement**. En effet, que se passe-t-il quand plusieurs équipes souhaitent mettre en production plusieurs pans fonctionnels d'un même produit en même temps ? Chaque équipe a peut-être testé sa fonctionnalité de son côté, mais comment va réagir l'application face à une intégration fonctionnelle de masse touchant toutes les couches applicatives ?

Si les bonnes pratiques d'ingénierie n'ont pas été mises en place (gestion des branches, Toggle Feature Pattern, automatisation de plusieurs typologies de tests, etc.), votre seule option est le cycle en V avec des équipes dédiées aux tests, aux déploiements, avec des allers-retours interminables entre les équipes qui annulent tout espoir de « vraie » prédictibilité.

Est-ce vraiment l'objectif d'une transformation en Feature Teams ?

La connaissance du processus de mise en production permet de mesurer le gap à franchir (sorte de dette) avant de pouvoir obtenir tous les bénéfices de la transformation. Dans certains contextes, la dette est tellement importante qu'il vaut mieux attendre de l'avoir résorbée avant de songer à tout changement organisationnel.

Comment garantir la cohérence de votre produit ?

Dans une organisation en Feature Teams, les équipes (Squads) sont idéalement autonomes. **Le risque est de créer de véritables silos** en coupant toute communication entre équipes. Or, ces équipes peuvent travailler sur les mêmes couches techniques et réaliser des fonctionnalités équivalentes. **Sans communication, chaque équipe va travailler « à sa manière »** en faisant ses propres choix techniques. Le résultat sera un produit techniquement hétérogène, difficile à maintenir et à faire évoluer.

Afin de couvrir ce risque, le modèle Spotify propose **des communautés de pratiques** (Chapters et Guilds).

Un autre challenge consiste donc à **identifier les bonnes communautés de pratiques** à mettre en place. Quel est le besoin de synchronisation et de partage d'information entre les équipes ?

Le premier axe d'identification est **la technique** (CMS, Front, etc.). Généralement, nous préconisons une communauté par couche technique au début de la transformation. En venant d'une organisation en mode composant, l'effet est alors double. Les personnes issues de leurs anciennes équipes continuent à travailler ensemble. L'acceptation du changement en est facilitée.

Une fois les communautés identifiées, **la question du cérémonial se pose**. À quelle fréquence les communautés doivent-elles se réunir ? Combien de temps ? Que se disent-elles durant les réunions ?

Enfin, pour que la communauté vive, il faut un animateur / facilitateur, une sorte de Scrum Master de communauté. Il faut donc l'identifier. Généralement, ces leaders de communautés sont élus pour **leurs compétences techniques et leur savoir être**.

Go ou NoGo ?

Maintenant que vous vous êtes posé les bonnes questions, que vous avez une meilleure visibilité sur votre future transformation, que vous avez toutes les cartes en main, il reste une dernière question, LA question : **Go ou NoGo ?**

Vous l'avez compris, une transformation en Feature Teams apporte de gros avantages mais ne se prend pas à la légère. La décision vous appartient.

 *La meilleure démarche consiste à afficher vos résultats, vous allez ainsi générer de la curiosité autour de vous et la curiosité est un outil très puissant.* 

Transformations agiles à grande échelle

Derrière ce terme pompeux, l'idée est de permettre à l'organisation dans son ensemble de pivoter vers l'agilité (nous parlons ici de programmes impactants plus de 450 collaborateurs). Tout comme pour les Feature Teams, nous vous conseillons d'être attentifs principalement à 3 aspects de la transformation : **le choix des projets qui vont participer au programme, la culture de votre entreprise et la conduite du changement**.

Ne sélectionnez pas vos projets

Dans le cadre d'une transformation globale, nous avons souvent vu apparaître le besoin de définir des critères d'éligibilité de projets pour un accompagnement agile. Toutes les tentatives que nous avons pu voir ont tout au mieux été remises au placard après quelque temps, au pire ont abouti à des usines à gaz qui ne permettaient pas d'identifier les bons projets.

Plutôt que de définir une grille d'éligibilité, nous vous recommandons **une checklist d'éléments facilitateurs**. Ainsi, nous mesurons les efforts à fournir et les actions à mettre en œuvre pour permettre une réussite de cette transformation plutôt que l'éligibilité d'un projet ou d'une organisation à passer à l'agilité. En effet, **la réussite des projets n'est pas liée aux outils mais aux hommes**. Même dans le meilleur contexte possible un projet (agile ou non) peut échouer par la faute de ses acteurs (problèmes politiques, relationnels, résistance au changement, etc.). À l'inverse, il y a des success stories comme celle de Wikispeed qui montrent que dans des contextes a priori difficiles, des équipes extrêmement agiles peuvent réussir des projets.

Attention donc à la grille qui se transforme en outil validant la réussite d'un projet. **À l'inverse, la checklist d'éléments facilitateurs ne garantit pas la réussite du projet mais recense ce qui en facilitera la réalisation**. Une telle checklist est aussi très intéressante pour la phase de coaching : par exemple, l'absence d'équipe colocalisée ne doit pas interdire à un projet un fonctionnement agile mais constitue un point d'attention fort pour le coach (ou l'équipe) sur la communication au sein du projet. Cette absence de plateau projet peut même nécessiter la mise en place de processus plus lourds afin de palier à ce manque de communication (outillage et formalisme de la communication).

Le facteur de réussite le plus important d'une transformation est de **partir de l'existant et d'adapter ce dernier**. Il est donc essentiel de démarrer à partir de ce que vous pouvez contrôler. Au contraire, si vous essayez d'imposer les choses, la transformation risque d'échouer. La meilleure démarche consiste à **afficher vos résultats**, vous allez ainsi générer de la curiosité autour de vous et **la curiosité est un outil très puissant**. Nous l'observons systématiquement lors de nos missions, les personnes (non accompagnées) viennent nous interroger et leurs désirs d'en savoir plus grossissent au point de vouloir être accompagnées à leur tour.

Façonnez votre culture Agile

Une transformation à grande échelle représente une opération complexe qui nécessite **un changement culturel**.

Le « **Culture Hacking** » est une solution car il permet de faciliter l'adoption de l'agilité à l'échelle d'une organisation. Robert Richmann, l'auteur du livre « The Culture Blueprint », s'occupe de la culture d'entreprise chez Zappos, leader mondial du eCommerce de chaussures. Zappos est un cas d'école, elle génère énormément d'argent tout en entretenant un niveau de satisfaction élevé chez ses employés. Zappos montre volontiers des vidéos de ses salariés dansant déguisés en plein milieu d'un Open Space (<http://www.zappostv.magnify.net/>).

L'objectif du « Culture Hacking » est de **trouver un sujet « fâcheux » afin de le modifier et le rendre plus efficace**. L'exemple type est le changement de format de réunion. Les réunions constituent dans de nombreuses entreprises une perte de temps non négligeable. Définir un ensemble de règles pour une réunion réussie permet de hacker la culture et de faire évoluer les pratiques.

Le changement de culture induit **un changement d'état d'esprit des individus**. Il s'agit là précisément de toute la difficulté d'une mise à l'échelle. Comment évaluer l'état d'esprit, la capacité d'adoption, le désir ou au contraire le refus des équipes ? La littérature peut vous donner quelques pistes :

- Dave Logan dans son livre « Tribal Leadership » nous apporte une réponse en introduisant cinq stades de profils. **Il met à disposition les principaux leviers permettant de passer d'un stade à un autre**. La simple existence de ces stades permet une prise de conscience. Le « tribal leadership » révèle l'importance de définir des valeurs fondamentales pour l'entreprise. Dans cette démarche le manager devient un leader, il est en charge de faire respecter les valeurs fondamentales, quitte à faire des concessions. Dans le cas contraire, il perdrait la confiance et régresserait d'un stade.
- **Le manager doit donc changer de posture**, Jurgen Appelo apporte un framework de méthodes dans son approche Management 3.0. Il doit apprendre à maîtriser les motivations intrinsèques des individus, à leur faire confiance et à trouver leur niveau raisonnable de délégation. Il est également en charge de développer les compétences des membres de sa tribu. A ce titre, lors de la phase de transformation à grande échelle, il est de sa responsabilité de mettre en place les moyens nécessaires à la réussite de cette transformation, comme, par exemple, un plan de formations.

Là encore, il faut partir de l'existant et procéder pas à pas afin de façonner la culture de votre entreprise, **l'objectif étant l'adoption des principes agiles comme partie intégrante de votre culture**. La transformation sera ainsi grandement facilitée.

 *L'objectif du « Culture Hacking » est de trouver un sujet « fâcheux » afin de le modifier et le rendre plus efficace. L'exemple type est le changement de format de réunion. Les réunions constituent dans de nombreuses entreprises à une perte de temps non négligeable. Définir un ensemble de règles pour une réunion réussie permet de hacker la culture et de faire évoluer les pratiques.* 

Guidez votre transformation

Le pivot de votre culture représente une étape cruciale. L'adoption de cette nouvelle culture implique, tout comme pour les Feature Teams, des changements plus ou moins importants au sein de votre organisation. Ici encore, **il faut gérer la conduite de ces changements : communiquer** auprès de vos équipes, **impliquer** chaque acteur dans cette transformation, **être transparent** et rassurer.

Un modèle, en 8 étapes, permettant une transition couronnée de succès, est proposé par **John Kotter** :

- créer un sentiment d'urgence,
- former une coalition,
- développer une vision et une stratégie,
- communiquer cette vision,
- responsabiliser les collaborateurs pour une action à grande échelle,
- démontrer des résultats positifs à court terme,
- consolider et bâtir sur les premiers résultats pour accélérer les changements,
- ancrer les nouvelles pratiques dans la culture de l'entreprise.

Nous nous appuyons sur ce modèle pour accompagner nos clients.

Kanban apporte également une réponse dans la visibilité d'une transformation à grande échelle : **le management visuel** qu'apporte Kanban est scalable. Il est donc possible de suivre l'avancement des équipes transformées au travers d'un Portfolio Kanban. L'avantage est double :

- il permet de limiter les demandes en cours de traitement, un principe fondamental dans un système Kanban,
- il donne accès à l'information, à tous et à tout moment, ce qui permet de garantir la transparence auprès des managers.

Il n'existe pas de solution miracle pour réussir une transformation à grande échelle. Nous vous avons présenté ici quelques pistes pour y parvenir. Il faut néanmoins retenir que **chaque organisation doit trouver ses propres solutions au vue du contexte, des aspirations des équipes, etc.**

La clé consiste à toujours faire un état des lieux et partir de ce qui existe.

TAKE AWAY AGILE TRENDS



RÉALISER

- Passer en Feature Teams pour apporter **plus de réactivité et de souplesse** dans une organisation à plusieurs équipes
- **Peser soigneusement le pour et le contre d'un passage en Feature Teams** car la transformation engendre des coûts humains, techniques et financiers élevés et demande une grande maturité agile
- **Une transformation agile globale doit passer par un changement culturel** : contrôler la convergence entre volonté managériale et adoption au sein des équipes de développement
- **Mesurer l'effort** plutôt que l'éligibilité d'un projet à réussir cette transformation
- **Miser sur la viralité** de la transformation agile pour accélérer l'adoption



MESURER

Vous avez identifié le bon produit, votre organisation a (ou est en train) de pivoter vers l'agilité et votre produit avance, vous êtes confiants et fier du travail accompli.

Aujourd'hui, vous participez au comité de direction de votre entreprise et vous exposez cette réussite. Votre enthousiasme est contagieux mais un de vos collègues vous pose alors les questions qui fâchent : « Quand le produit sera-t-il prêt ? » et « As-tu des chiffres à nous montrer pour étayer tes propos ? ». Il s'agit ici de prédictibilité et des indicateurs permettant un suivi factuel et objectif de la réalisation du produit. Explications.

Intégrez la notion de Prédicibilité

Les limites inhérentes à l'estimation

De tout temps, les Hommes ont voulu prédire l'avenir, que ce soit les Oracles du temps des guerres grecques, Nostradamus ou même Paco Rabanne.

Le monde de l'informatique ne déroge pas à la règle.

Tout projet s'accompagne d'une estimation de la charge prévisionnelle, revue régulièrement, avec plus ou moins de succès. L'approche la plus répandue est la fameuse estimation en jours/homme du reste à faire, qui s'apparente souvent à une estimation « au doigt mouillé ».

Les projets agiles se doivent également de proposer des estimations. Le mode d'estimation le plus courant en agile reste la mesure de la vélocité, c'est à dire le nombre de points réalisés dans un Sprint correspondant à l'ensemble des fonctionnalités ayant été développées et validées par le Product Owner.

Qu'il s'agisse des méthodes dites « classiques » ou agiles, **toute estimation est néanmoins fortement soumise à interprétation et peut varier d'une personne à l'autre** selon son expérience, son humeur ou sa connaissance fonctionnelle du sujet à traiter. Rendre les estimations plus proches de la réalité (toujours de manière relative) requiert un effort supplémentaire dont le ROI n'est souvent pas viable.

Ces imprécisions et défauts des estimations ont récemment profité à Kanban et ScrumBan qui émergent aujourd'hui, portés par **leur capacité à produire des métriques apportant une visibilité forte sur l'état de santé d'un projet tout en limitant l'effort de départ sur les estimations.**

Les avantages de la prédictibilité

Une des métriques les plus intéressantes reste sans conteste **la prédictibilité.**

La prédictibilité, a contrario des méthodes d'estimation classique en jours/homme ou Story points, se base sur **des mesures factuelles** (sous forme de moyennes) **du temps nécessaire à la réalisation d'un élément fonctionnel d'un produit.**

Imaginez que vous deviez estimer le temps nécessaire pour parcourir votre trajet habituel domicile-lieu de travail. Le premier reflexe serait de donner une valeur un peu arbitraire, tenant compte de votre humeur du moment (fatigué, en forme, malade), de la météo ou des perturbations de transport. Cette valeur reste valable à un instant donné,

mais n'est pas représentative du temps moyen habituel de votre trajet. Par contre, mesurez chaque jour votre temps de trajet, sur 2 semaines à 1 mois, avec tous ses aléas et vous obtenez un délai de parcours moyen proche de la réalité, car tenant compte aussi bien des parcours "normaux" et de ceux où vous rencontrez des obstacles.

La prédictibilité expliquée

La prédictibilité peut se mesurer via deux métriques, le **Cycle Time** et le **Débit des Cartes**.

LE CYCLE TIME - DÉFINITION

Le Cycle Time - aussi appelé « Temps de Cycle » - permet de mesurer **la durée nécessaire à la réalisation d'une fonctionnalité**, en partant des travaux d'ingénierie jusqu'à sa validation.

- La date de démarrage du Cycle Time est la date de passage vers les activités d'ingénierie.
- La date de fin est la date à laquelle la fonctionnalité a été validée, selon la « Définition de Terminé » (Definition of Done).

LE DÉBIT DE CARTES - DÉFINITION

Le débit représente **le nombre de fonctionnalités** (ou Cartes Kanban, représentant les Users Stories et les anomalies) **ayant été réalisées, sur une période définie** (une semaine idéalement).

Il est possible de distinguer le débit en fonction du type de Cartes à traiter, comme les Users Stories ou les anomalies par exemple, mais l'objectif principal de cette mesure est d'identifier la capacité du système à produire de la valeur Métier ou réduire la dette technique.

CARTE KANBAN - DÉFINITION

Une Carte kanban est un élément de travail dans la vie d'un produit. Il s'agit d'une User Story (au sens Scrum), une Technical Story, une anomalie, un travail fait par une autre équipe en adhérence ou un ticket de changement type JIRA.

La prédictibilité par le Cycle Time

La prédictibilité par le Cycle Time consiste à **projeter la capacité réelle de l'équipe afin d'estimer une date de livraison du produit.**

La mesure de la prédictibilité par le Cycle Time est un élément important dans l'estimation du temps nécessaire à la réalisation d'une Carte Kanban, une aide à la priorisation et à la prise de décisions. L'objectif reste de permettre à l'équipe Kanban de tenir ses engagements.

L'estimation va tenir compte, non pas d'une estimation du reste à faire, mais d'une projection de ce qui a déjà été produit par l'équipe de développement. À cette prédictibilité par le Cycle Time va s'ajouter l'impact sur votre reste à faire des anomalies produites par les Stories réalisées. En effet, toute nouvelle anomalie à corriger provoque une perturbation de votre système Kanban et a un impact sur la capacité de votre équipe projet à tenir ses engagements.

La prédictibilité va donc intégrer le temps moyen de correction des anomalies restantes, mais également effectuer une projection du nombre potentiel d'anomalies pouvant être produites par User Story. Cette dernière métrique est obtenue en calculant le ratio moyen d'anomalie(s) produite(s) pour chaque Story réalisée.

Les pièges de la prédictibilité par le Cycle Time

Il peut néanmoins arriver que la mesure de la prédictibilité par le Cycle Time soit faussée, pour les raisons suivantes :

- Mesure des Cycle Times réalisés de façon non rigoureuse (oubli de cartes, renseignement de la mauvaise date de début, etc.) ;
- Cartes Kanban bloquées en raison de problèmes techniques ou d'adhérences externes ;
- Tout autre blocage non imputable à l'équipe de développement.

Pour palier à ces désagréments, la prédictibilité peut également être mesurée au travers du débit des cartes.

 *La prédictibilité va donc intégrer le temps moyen de correction des anomalies restantes, mais également effectuer une projection du nombre potentiel d'anomalies pouvant être produites par User Story. Cette dernière métrique est obtenue en calculant le ratio moyen d'anomalie(s) produite(s) pour chaque Story réalisée.* 

Le débit des cartes

Comme la mesure de la prédictibilité par le Cycle Time, la mesure de la prédictibilité par le débit nous offre une indication assez fiable de la date prévisionnelle d'atterrissage d'un projet, car **elle tient compte d'un état d'avancement factuel.**

Le calcul de base est simple et tient compte des éléments suivants :

- Le débit des cartes Kanban ;
- Le nombre de Users Stories restant à réaliser ;
- Le nombre d'anomalies restant à corriger.
- Le nombre potentiel d'anomalie(s) pouvant être produite(s) par User Story restante, basé sur le ratio moyen d'anomalie(s) produite(s) pour une User Story réalisée.

Idéalement, le débit est calculé sur une semaine (5 jours ouvrés) et nous fournit donc le nombre de semaines nécessaires afin de réaliser l'ensemble de notre Backlog de Produit.

Jusque là rien de compliqué, sauf que le débit est mesuré chaque semaine en fonction de la capacité de l'équipe à ce moment. Qu'en est-il de la fiabilité du débit lorsque cette capacité vient à baisser ou augmenter ?

Afin de pallier à ce risque, **la charge de l'équipe ayant permis de mesurer le débit, ainsi que la charge de l'équipe sur laquelle doit se baser la prédictibilité sont prises en compte.**



La recette de la prédictibilité par le Cycle Time

Ingrédients principaux :

- Temps moyen de réalisation d'une Story ;
- Temps moyen de correction d'une Anomalie ;
- Nombre moyen d'anomalie(s) détectée(s) par Story réalisée ;
- Nombre de Stories restant à réaliser ;
- Nombre d'anomalies restant à corriger ;
- Charge de l'équipe de développement (en équivalent temps plein)

Temps de préparation : 2 à 3 semaines, le temps de récupérer suffisamment de données.

La recette par le Cycle Time

Etape 1 : Calcul du temps de réalisation des Stories restantes

- **Faire** la somme de tous les temps de réalisation (Cycle Time) des Stories ayant été déclarées « terminées » et en calculer la moyenne. Vous obtenez le Cycle Time Moyen d'une Story.
- **Multiplier** le Cycle Time moyen d'une Story par le nombre de Stories restant à réaliser.
- **Réserver.**

Etape 2 : Calcul du temps de correction des anomalies restantes

- **Faire** la somme de tous les temps de correction (Cycle Time) des anomalies ayant été détectées sur les Stories terminées. En faire la moyenne. Vous obtenez le Cycle Time moyen d'une anomalie.
- **Multiplier** le Cycle Time moyen d'une anomalie par le nombre d'anomalies restant à corriger.
- **Réserver.**

Etape 3 : Calcul de l'estimation des anomalies susceptibles de se produire sur les Stories restantes

- **Calculer** le nombre moyen d'anomalie(s) créée(s) par Story réalisée (ratio). Pour ce faire, prendre le nombre d'anomalie(s) détectée(s) sur les Stories validées et le diviser par le nombre de Stories « Terminées ».
- **Prendre** le nombre de Stories restant à réaliser, le multiplier par le ratio moyen des anomalies.
- **Multiplier** le résultat obtenu par le Cycle Time moyen d'une anomalie.
- **Réserver.**

Etape 4 : Le dressage

- **Additionner** les résultats obtenus sur les étapes 1 à 3.
- **Diviser** le tout par la charge de l'équipe de développement.
- Le résultat obtenu donne le nombre de jours nécessaires à la réalisation des Stories et anomalies restantes.

Si votre Cycle Time est de mauvaise qualité, remplacez-le par le débit des cartes Kanban.

La recette de la prédictibilité par le débit de cartes

Ingrédients principaux :

- Nombre moyen de cartes (Story, anomalie) réalisées par semaine ;
- Nombre de Stories restant à réaliser ;
- Nombre d'anomalie(s) restant à corriger ;
- Nombre moyen d'anomalie(s) détectée(s) par Story réalisée.

La recette par le débit

Etape 1 : Calculer le débit moyen des cartes par semaine

- **Prendre** le nombre total de cartes (Story, anomalie) terminées depuis le début du projet.
- **Diviser** ce total par le nombre de jours ouvrés depuis le début du projet.
- **Ramener** le résultat obtenu par semaine (multipliez par 5). Vous obtenez le débit moyen de cartes par semaine.
- **Réserver**

Etape 2 : Estimer les anomalies susceptibles de se produire sur les Stories restantes

- **Calculer** le nombre moyen d'anomalie(s) créée(s) par Story réalisée (ratio). Pour ce faire, prendre le nombre d'anomalie(s) détectée(s) sur les Stories validées et le diviser par le nombre de Stories « Terminées ».
- **Prendre** le nombre de Stories restant à réaliser, le multiplier par le ratio moyen des anomalies.
- **L'additionner** au nombre d'Anomalie(s) restant à corriger
- **Réserver.**

Etape 3 : Le Dressage

- **Prendre** le résultat de l'étape 2.
- **L'additionner** au nombre de Stories restant à réaliser
- **Diviser** ce résultat par le débit moyen de carte calculé à l'étape 1.
- **Vous obtenez** le nombre de semaines nécessaires pour terminer les cartes restantes.

Conclusion

La prédictibilité par le Cycle Time est un outil puissant dans **la mesure de la capacité d'une équipe à tenir ses promesses**. Elle est une **aide forte à la prise de décisions**. Celle-ci se doit néanmoins d'être utilisée **avec discernement** et de tenir compte des perturbations pouvant influencer sur son résultat.

Tout comme le Cycle Time, le débit des cartes est un élément puissant et fiable dans **la mesure de la prédictibilité d'un projet**. Il se base avant tout sur un état des lieux factuel des activités réalisées. Le résultat obtenu reste proche de celui mesuré via le Cycle Time, mais attention, un retour d'expérience portant sur plusieurs projets IT montre un écart pouvant aller jusqu'à 6 jours.

L'essentiel reste donc de mesurer, non pas une date d'atterrissage au jour près, mais une tendance de la performance du système Kanban et **la capacité de l'équipe à tenir ses engagements**.

Indicateurs

« C'est bien beau de nous faire rêver avec l'agilité, mais comment je sais si mon projet se porte bien ? ». Cette phrase, nous l'avons tous entendue un jour, concernant l'agilité, lorsque nous participions à un Comité de Pilotage ou à un Comité de Direction.

Les DSI, après avoir rêvé devant les bénéfices de l'agilité, ont maintenant besoins de résultats tangibles et pérennes auxquels les Burndown Chart et la vélocité ne peuvent plus répondre.

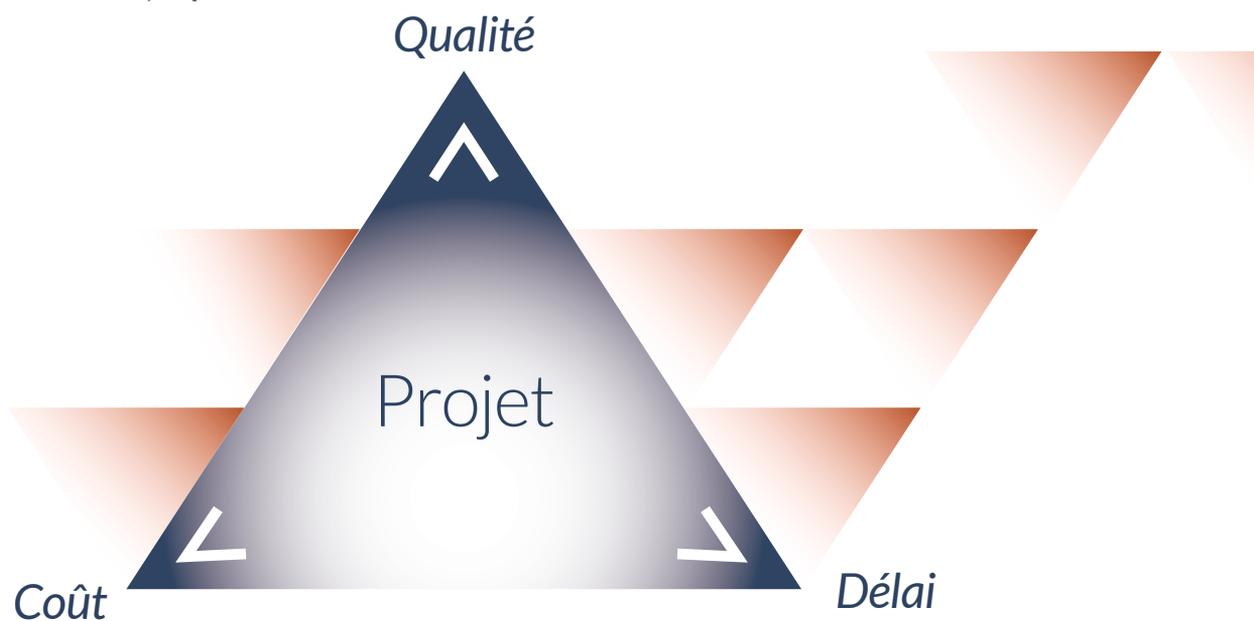
Cela rappelle l'époque des débuts de l'informatique en France où les « bidouilleurs » étaient les Rois, ensuite le besoin de professionnalisation a amené la mise en place de normes et pratiques de gestion de projets qui ont produit des effets positifs, dans un premier temps, avant de basculer vers trop de formalisme.

Aujourd'hui, c'est au tour de l'agilité d'évoluer et de donner aux managers et aux équipes agiles les éléments de visibilité nécessaires au suivi de la performance et au pilotage des projets. **Cette visibilité est apportée par des KPI** (Key Performance Indicator), adaptés aux besoins des équipes projet, ainsi qu'à ceux du management. L'objectif des KPI est de :

- évaluer,
- diagnostiquer,
- communiquer,
- informer,
- motiver,
- améliorer en continu le processus.

Le « Triangle du Fou » des projets classiques

Tout projet dit « classique » se doit de tenir compte des trois notions fondamentales, qui sont le **Coût**, la **Qualité** et le **Délai**.



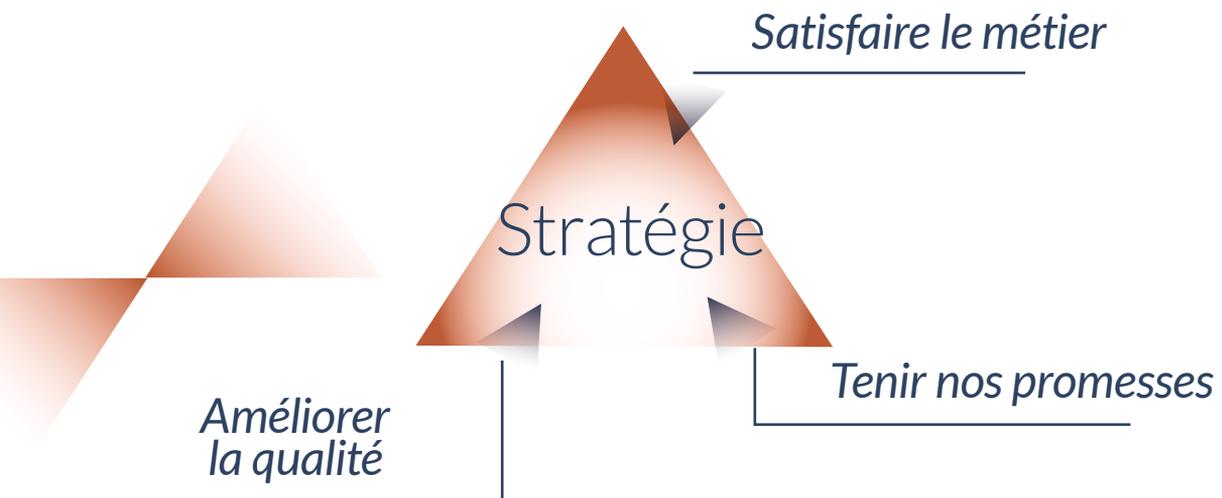
Ces trois notions sont inter-dépendantes et sont rarement, voire jamais, mises en place ensemble. Au démarrage du projet ou lors des arbitrages, les choix suivants sont possibles :

- **Option 1** : Un projet rapide à faible coût >> Le risque portera sur une mauvaise qualité du produit final ;

- **Option 2** : Un projet rapide et de bonne qualité >> Le risque portera sur un coût élevé des développements et des phases de testing ;
- **Option 3** : Un projet de qualité à un coût faible >> Le délai de réalisation sera potentiellement plus long que prévu et portera un risque sur le respect du Time To Market.

La qualité n'est pas négociable

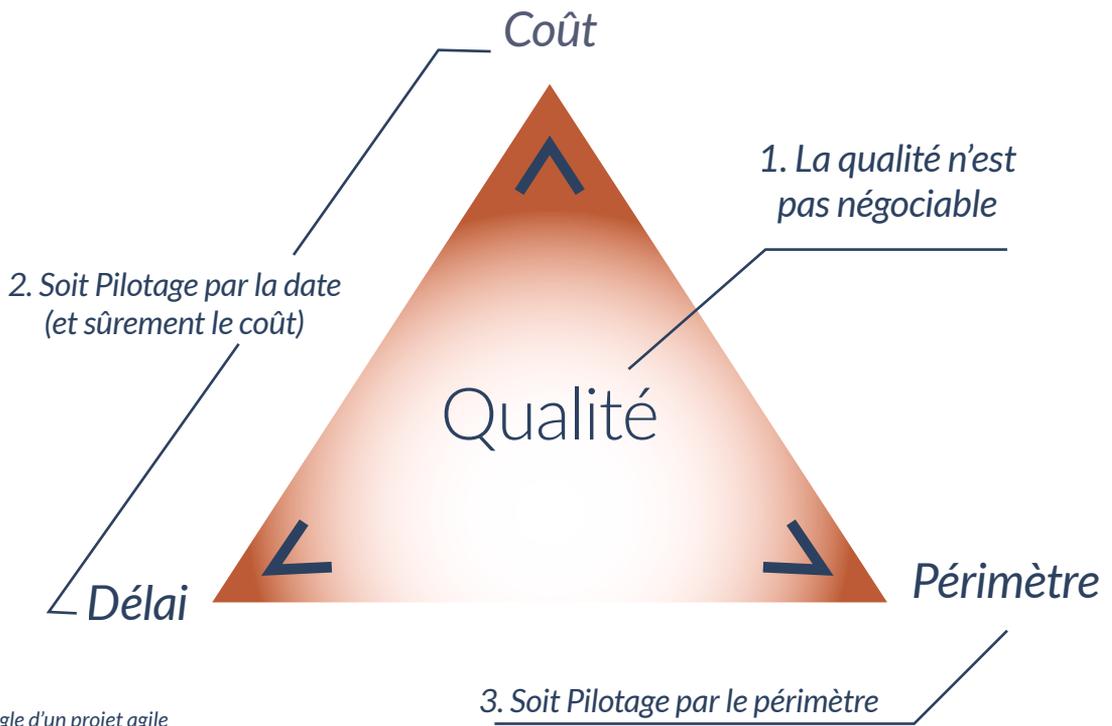
Sur un projet agile, la stratégie porte sur 3 éléments :



- **Satisfaire le métier** : l'objectif est de livrer un produit apportant une forte valeur métier ;
- **Améliorer la qualité** : livrer un produit « zéro défaut » ou tout du moins robuste et nécessitant peu d'interventions correctives en production ;
- **Tenir nos promesses** : il s'agit ici du respect du délai de livraison du produit au métier, ce que les équipes marketing nomment le TTM (Time To Market).

L'agilité ne négocie pas la qualité, qui est une composante essentielle de son ADN.

Sur les deux autres éléments, une matrice des compromis est mise en place afin de définir les choix de stratégie à appliquer sur votre projet.



Le résultat de la matrice des compromis va influencer les indicateurs (KPI) à suivre, qui seront toujours basés sur des éléments mesurables et factuels :

- Nombre de Users Stories réalisées ;
- Nombre d'anomalies corrigées ;
- Ratio du nombre d'anomalie(s) produite(s) par User Story réalisée ;
- Date de fin estimée du projet, tenant compte de la prédictibilité par le Cycle Time ou par le débit des cartes ;
- Etc.

Des KPI pour tous

Le nombre d'indicateurs sur un projet peut être extrêmement important et varié. Il est néanmoins indispensable d'identifier les indicateurs pertinents et d'en limiter la quantité sur un sujet précis. Là où un Scrum Master souhaitera mesurer le niveau de satisfaction de son équipe ou le niveau de qualité des fonctionnalités réalisées, un manager cherchera la maîtrise du coût ou du délai.

Le deuxième point important à aborder après la matrice de compromis est donc de **définir les KPI à suivre et pour quels profils.**

Ces KPI doivent être, non seulement limités en nombre, mais également représentés de façon synthétique, au travers d'un « Happy board ».



Date de mesure : 13 février 2014
Version : **Projet beta**
Projet/lot: **Tous projet/lots**

Stories Done	Anomalies par Story Done	Coût des Anomalies (jours)	Nombre de jours restants	Nombre de jours pour finir	Débit des cartes/semaine
65%	0,00	0,00	8	58	11,64
😊	😊				
Cycle Time Stories (jours)	Cycle Time anomalies (jours)	Lead Time Stories (jours)	Lead Time Anomalies (jours)	Maturité Équipe	Météo Équipe
6,21	0,00	19,4	0,00		

TAKE AWAY AGILE TRENDS



MESURER

- Déterminer votre Cycle Time et votre débit de carte, en tenant compte des possibles anomalies
- Définir vos besoins humains et budgétaires ainsi que le périmètre d'action afin de ne pas compromettre la qualité du produit final
- Etablir votre tableau des KPI représentatifs des objectifs finaux

CONCLUSION

Le mythe de l'entreprise agile

Les entreprises se transforment à différents niveaux. Au-delà des équipes de développement, les échelles changent, les méthodes évoluent, la gestion de produit se professionnalise, le pilotage s'adapte et devient pragmatique. Devenir une entreprise agile semble être le graal poursuivi par nombre d'organisations. C'est presque devenu un bon de commande, une sorte de garantie de faire mieux, plus vite, moins cher avec des collaborateurs épanouis, embarqués, pro-actifs et motivés.

On utilise souvent des indicateurs pour mesurer les taux d'utilisation des méthodes. C'est très utile pour piloter des programmes de transformation mais **qu'en est-il de la pérennité ? De la capacité à s'adapter en s'améliorant de façon continue ?**

Le véritable défi est là. Une entreprise ne devient pas du jour au lendemain l'égal de FAVI : sans hiérarchie ou presque et qui résiste admirablement à la mondialisation depuis 35 ans tout en affichant des résultats insolents dans le contexte de morosité économique actuel.

Alors qu'est-ce que c'est une entreprise agile ?

Le paradoxe est que celles qui le sont le plus ne se réclament pas spécifiquement de l'agilité. Chez Xebia, nous utilisons le terme de « **résilience** ».

Une entreprise agile est celle qui a la **capacité à se réorganiser pour surmonter les difficultés, à durer en s'adaptant à un contexte changeant.**

Une entreprise en résilience aura des collaborateurs autonomes, auto-organisés, acteurs et responsables, mais également des équipes pluridisciplinaires, où la transparence est radicale, l'erreur est permise pour promouvoir l'innovation. Le management est accessible, à l'écoute, au service des équipes, la gestion de produit embrasse l'incertitude. Mais surtout, elle peut s'or-

ganiser rapidement autour d'un obstacle, tous niveaux hiérarchiques et toutes compétences confondues, pour le grignoter en profondeur. Chez Xebia, nous avons mis en place un système de fondation.

Comment devenir une entreprise en résilience ?

Il n'y a pas de réponse magique ni de méthode sur étagère toute prête à appliquer. C'est un mythe ! **Chaque transformation est dépendante du contexte, de la taille et de la culture de l'entreprise.** La stratégie à appliquer sera construite au cas par cas.

Une tendance semble néanmoins se dégager : la taille. Le « nombre de Dunbar » stipule **qu'au dessus de 150 personnes, la confiance mutuelle et la communication ne suffisent plus à assurer le fonctionnement d'un groupe.** Au-delà, une organisation plus complexe, structurée, hiérarchique avec des règles précises est nécessaire... rendant donc la résilience plus compliquée.

La tendance qui semble apparaître est de **découper l'organisation suivant des tailles critiques.** Cette forme de découpe est différente de celles que nous avons connues : elle ne suit pas une logique de services par compétences mais bien des groupes autonomes orientés produit.

Les transformations agiles se complexifient. Il ne s'agit plus seulement de déployer des méthodes dans des équipes existantes. **Il s'agit de restructurer complètement des entreprises sur les plans organisationnels et culturels avec tous les impacts que cela engendre.** L'humain n'a jamais été autant au cœur des transformations.

GLOSSAIRE

SCRUM : Scrum est fondé sur la théorie de contrôle des processus empiriques : la connaissance s'acquiert par l'expérience et favorise la prise de décision basée sur ce qui est connu. Cette théorie s'appuie sur 3 piliers que sont l'inspection, la transparence et l'adaptation.

KANBAN : Kanban est un mode d'organisation en flux plutôt qu'en Sprint. Il implique deux principes : la limitation du travail en cours et une recherche de la perfection au moindre coût.

SCRUMBAN : Le ScrumBan utilise le caractère prescriptif de Scrum pour être agile et l'amélioration des processus de Kanban.

PRODUCT BACKLOG : Au coeur de la méthode Scrum, c'est une liste priorisée de besoins et demandes du client. Il peut contenir des Users Stories (« en tant que ..., je veux ..., afin de... ») et leurs critères d'acceptation.

USER STORY : Une User Story décrit une fonctionnalité qui aura de la valeur aux yeux d'un utilisateur ou d'un stakeholder du produit.

WIREFRAMES : Aussi appelé Storyboard, le Wireframe est une maquette orientée expérience utilisateur. Ses objectifs sont de définir les zones et l'ergonomie du produit, architecturer et hiérarchiser les contenus, et enfin définir les principes de navigation et le parcours utilisateur.

PERSONAS : Il s'agit d'une personne fictive représentant un groupe cible. Il décrit son profil, ses envies et ses besoins, en lien avec le produit.

BEHAVIOR DRIVEN DEVELOPMENT (BDD) : Le BDD est un langage métier qui permet de décrire des comportements sous forme de tests auto-suffisants. Celui-ci va permettre à l'expert métier, au PO et aux développeurs de se comprendre, grâce à l'utilisation d'un vocabulaire commun accessible à tous.

WIKISPEED : Entreprise américaine qui produit des prototypes de véhicules à faible consommation en s'inspirant des méthodes agiles de développement logiciel.

FAVI : Fonderie et Ateliers du VImeu. Cette entreprise utilise des méthodes de management révolutionnaires basées sur l'autonomie de ses employés.



TAKE AWAY AGILE TRENDS



IMAGINER

- Créer une équipe soudée autour d'un pilier central qu'est le PO
- Donner aux POs les moyens logistiques / humains de réussir et leur faire confiance dans la prise de décision
- Cultiver la curiosité des POs pour qu'ils apportent une nouvelle dimension à vos produits
- Mettre l'utilisateur au centre des réflexions et de toute prise de décision relative au produit



RÉALISER

- Passer en Feature Teams pour apporter plus de réactivité et de souplesse dans une organisation à plusieurs équipes
- Peser soigneusement le pour et le contre d'un passage en Feature Teams car la transformation engendre des coûts humains, techniques, et financiers élevés, et demande une grande maturité agile
- Une transformation agile globale doit passer par un changement culturel, contrôler la convergence entre volonté managériale et adoption au sein des équipes de développement
- Mesurer l'effort plutôt que l'éligibilité d'un projet à réussir cette transformation
- Miser sur la viralité de la transformation agile pour accélérer l'adoption



MESURER

- Déterminer votre cycle time et votre débit de carte, en tenant compte des possibles anomalies
- Définir vos besoins humains et budgétaires ainsi que le périmètre d'action afin de ne pas compromettre la qualité du produit final
- Etablir votre tableau des KPI représentatif des objectifs finaux



À lire et à relire



Les précédents numéros des TechTrends sont disponibles en téléchargement (pdf et epub) sur xebia.fr. Si vous souhaitez recevoir une version papier, nous vous invitons à envoyer un mail à : marketing@xebia.fr

Merci à

*Benjamin Moitié, Emmanuel Sciara,
Laurène Vol-Monnot, Christophe Heubès,
Luc Legardeur, Marina Tracco, Laetitia Janné,
Marine Krol, Chloé Desault, Cécile Andrieu,
Wendy Berton.*

Les auteurs



Clément
Rochas



Renaud
Chevalier



Yannick
Grenzinger



Gilles Mantel



Audrey
Pedro



Ludovic
Pérot



Anne
Beauchart



Couthaïer
Farfra



Gwénaél
Bonhommeau



Nicolas
Lochet



SOFTWARE DEVELOPMENT **DONE RIGHT**

Xebia France
156 bd Haussmann, 75008 Paris
+33 (0)1 53 89 99 99
info@xebia.fr

Toutes les informations sur :
xebia.fr