

LE « CONTINUOUS » SPRINT 0 DU PROJET AGILE

Compte rendu d'une rencontre avec Jean-Pierre Vickoff

Martine.otter@adeli.org

Résumé :

Que reste-t-il des méthodes agiles 20 ans après la révélation de l'agile manifesto ? Comment répondent-elles actuellement aux besoins des entreprises ? Jean-Pierre Vickoff nous a exposé son point de vue le 13 mai 2019 lors d'une rencontre-débat ADELI.

Mots-clés :

Agile, SAFe, SCRUM, Backlog



Face à des frameworks agiles ne permettant pas d'appréhender totalement le besoin du client et d'exprimer ses exigences, comment « combler les trous dans la raquette » ?

JEAN-PIERRE VICKOFF, PIONNIER DE L'AGILITÉ

Le manifeste agile a été publié en 2001. Mais c'est bien avant, aux USA et au Canada, qu'il découvrit et mit en œuvre l'incrémental-itératif-adaptatif et le publia dans la Micro-Gazette de Décembre 1991 « Les environnements graphiques transforment nos méthodes ». Il en revient l'idée originale à la méthode RAD (Rapide Application Développement) de James Martin, la première des méthodes agiles - même si ce qualificatif n'existait pas encore avant 2001. De par ses nombreuses communications et propositions d'évolution, il en fut ensuite le pionnier et le promoteur francophone.

« RAD était une vraie proposition de développement rapide d'applications, la première des méthodes vraiment agiles. Sa souplesse vous permettait de jouer sur différents paramètres de planification comme variables d'ajustement. Son scope couvrait aussi bien la partie initialisation de la mise en œuvre méthodologique que le cadrage, le design de conception et la modélisation minimum. Ainsi sécurisée, la réalisation faisait suite lors d'une phase de construction structurée, extrême ou non. La dernière étape, le « CutOver » (ou finalisation) livrait l'application en site pilote puis en maintenance. RAD s'avérait ainsi une méthode complète en matière de développement de système d'information. »

Le manifeste agile en 2001 fit réagir Jean-Pierre qui imagina alors PUMA, une simple « Proposition pour l'Unification des Méthodes Agiles ». Cette initiative ne fut pas prise en considération par les concepteurs des méthodes de l'époque qui pensaient poursuivre la promotion de la leur. Ceci dit, moins de dix années plus tard, on constatait que toutes les autres approches réellement itératives avaient quasiment disparu. Il ne restait plus que l'amélioration continue et les sprints de SCRUM, ou les techniques de génie logiciel XP réservées à quelques développeurs.

Durant toute cette période, Jean-Pierre Vickoff a accompagné ADELI en communiquant sur l'évolution des approches Agiles dans la lettre d'ADELI. Nous l'avons reçu à plusieurs reprises lors des « Rencontres autour d'un verre ». Le 13 mai 2019, c'était donc la troisième rencontre animée par Jean-Pierre, venu nous présenter sa vision du Sprint 0.



MÉTHODES AGILES – HISTORIQUE ET CONSTAT

Son premier constat :

« Il y a trop de manques dans les méthodes agiles actuelles, au niveau du contrôle de la performance du projet, en regard de la nécessité d'adaptation imposant des changements éventuels du besoin qui ne peuvent être - dans l'absolu - gratuits ».

Jean-Pierre souligne alors la tendance à s'intéresser en priorité au produit, une manière de remettre en question ce qui se faisait avant où tout passait par le projet. Mais il n'en demeure pas moins, dit-il, qu'il y aura toujours nécessité de faire des projets pour créer des produits et que les ambitions de qualité ou d'alignement métier d'un produit sont souvent contradictoires avec les exigences de performance économique.

Un peu d'histoire ! Les méthodes cascades avaient leur raison d'être à l'époque des grandes chaînes batch. Puis, lorsque le client serveur et les environnements graphiques sont apparus dans les années 80, il fallait des méthodes plus légères. En 1991 James Martin développa donc la méthode RAD, relativement sommaire, mais suffisamment complète pour couvrir le scope des développements applicatifs.

À partir de Rad, Jennifer Stapleton, en Angleterre, créa en 1994 le consortium DSDM¹ (Dynamic systems development method). Jean-Pierre Vickoff, de son côté, en France, proposa RAD2 qui fut publié par le Gartner Group.

L'eXtrem Programming apparut en 1999, avec comme caractéristique principale, dixit Jean-Pierre, d'être justement extrême, c'est-à-dire d'exiger un haut niveau de compétences des développeurs.

En février 2001, le manifeste agile définissait 4 grandes valeurs :

- individus et interactions plutôt que processus et outils ;
- fonctionnalités opérationnelles plutôt que documentation exhaustive ;
- collaboration avec le client plutôt que contractualisation des relations ;
- acceptation du changement plutôt que conformité aux plans.

Ces valeurs se déclinaient en 12 principes² à respecter sans concession par toute approche se recommandant de l'agilité.

Novembre 2001 : publication de Scrum par Schwaber et Beedle. Cette méthode reposait sur :

- une organisation de projet avec d'un côté le métier, de l'autre l'équipe de développement et au milieu un Scrum master, dans un rôle d'animateur-facilitateur ;
- la nécessité d'organiser régulièrement certaines réunions (Daily stand-up et de fin d'itération).

En fait, absolument rien de nouveau sur ces plans, car ces techniques faisaient déjà intégralement partie de la méthode RAD.

Le seul paradigme que la méthode Scrum bouleversa vraiment fut la notion de projet, et par là même du rôle de chef de projet qui, avec le RAD, devait être « consensuelle » mais qui disparut alors brutalement. On pourra aussi accorder à Scrum une généralisation de la notion de rétrospective, c'est à dire le conseil de s'arrêter à fréquence fixe pour observer son travail et rechercher la possibilité d'en améliorer le processus. En résumé, Scrum n'était en fait qu'un framework d'amélioration continue basé sur la Roue de Deming.

Qu'est-ce qui manquait alors aux approches agiles ? Tout simplement l'expression des exigences qui n'intéressait pas non plus les créateurs d'XP, occupés à refaire la paye de Chrysler à iso fonctionnalités, donc sans nécessité d'exprimer de nouveaux besoins.

¹ <https://www.agilebusiness.org/what-is-dsdm>

² https://fr.wikipedia.org/wiki/Manifeste_agile



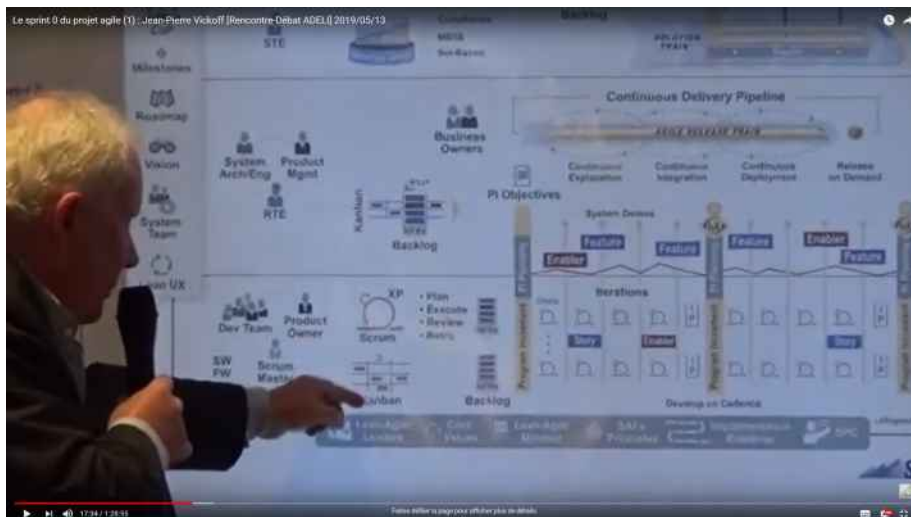
Après de nombreuses altercations sur Wikipédia, il est aujourd'hui admis que Scrum se limite à un framework, c'est-à-dire, au sens strict, à un simple cadre de travail. Il ne vous précise pas comment se réunissent les éléments de la description applicative au moment où on va les mettre en œuvre. La démarche Scrum démarre avec le principe d'un « backlog » dans lequel les acteurs du développement doivent déclarer les éléments à produire, et il ne s'intéresse pas non plus à la production de la chose. D'un autre côté, cela représente une certaine force parce que ce framework ne s'applique pas qu'au logiciel. Il est possible, par exemple, d'utiliser Scrum pour faire du déploiement ou des projets d'organisation.

Aujourd'hui la problématique reste la même : les grandes entreprises se débrouillent comme elles le peuvent pour définir tout cela ainsi que l'ensemble des autres contraintes, ce que l'on appelle le cadrage. Vu sous un angle plus général, il s'agit de définir les conditions acceptables pour démarrer le projet et remplir le backlog. À partir de ce point on peut lancer le pilotage des Sprints de Scrum, qui, n'ayant pas de technique de génie logiciel, devra s'appuyer sur eXtreme Programming (XP) dans le meilleur des cas ou sur une méthode maison.

Depuis, les grandes entreprises sont confrontées à des préoccupations plus importantes : comment développer de grands projets, ce que l'on appelle passer à l'échelle ? Dès 2007, Jean-Pierre Vickoff formalise dans PUMA, devenu « Processus Urbanisant les Méthodes Agiles », une architecture simplifiée d'entreprise Agile et une vision de l'organisation des développements conséquents ou complexes. Le Forrester Research, spécialiste du domaine, l'analysera et la considérera dans son rapport 2013, comme l'une avec SAFe (*Scaled agile framework*) des deux approches d'adaptation Agile au niveau de l'entreprise. En 2008, avec PUMA Essentiel, Jean-Pierre propose une simplification de l'ensemble Puma dans un Framework de pratiques agilisées encapsulant Scrum et XP, comme le fait depuis SAFe, imaginé par Dean Leffingwell en 2011, avec une version « SAFe Essentiel ».

Quelques mots sur SAFe

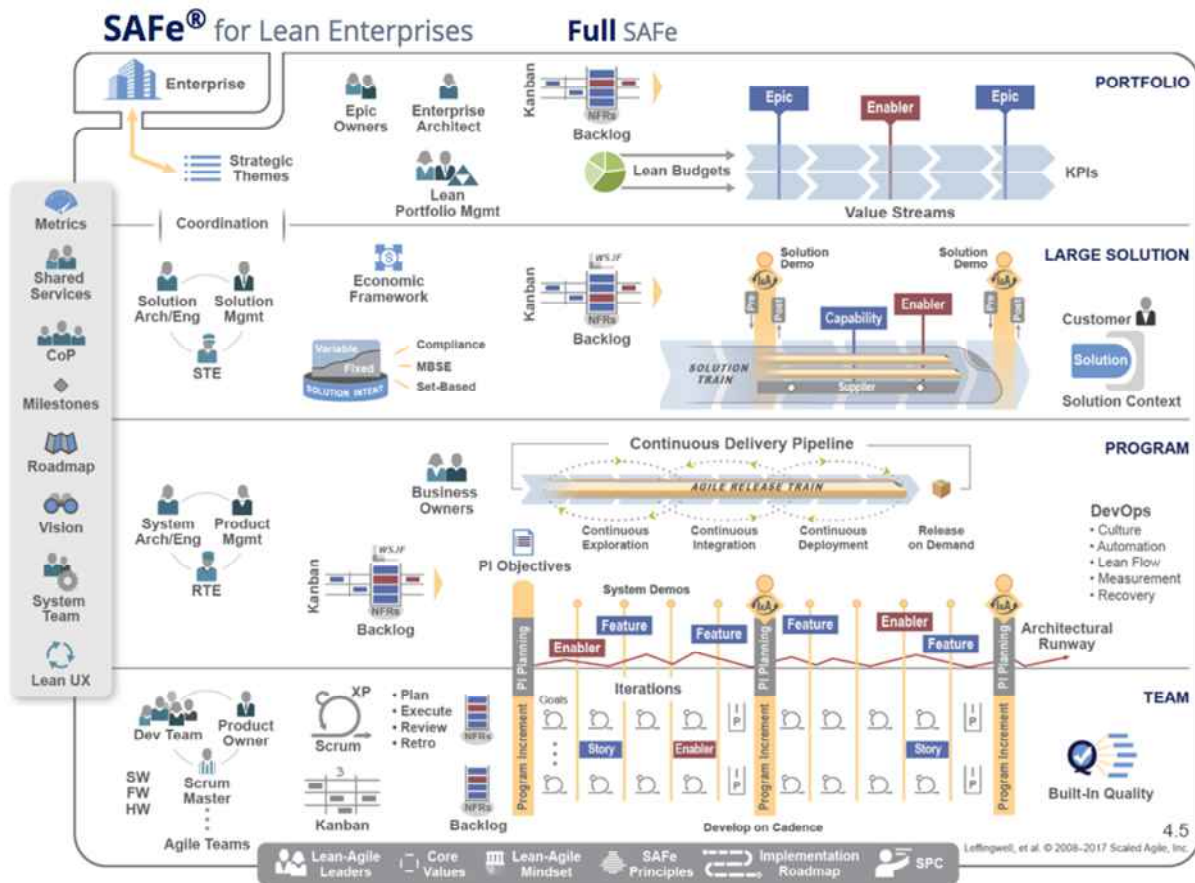
Il existe de nombreux frameworks Lean de « montée à l'échelle », mais le plus puissant s'avérant demeurer SAFe, Jean-Pierre Vickoff nous le présente de façon synthétique.



Jean-Pierre Vickoff (SPC 4.6) nous présente le cadre SAFe, où le niveau le plus bas, celui de l'équipe, fait entre autres apparaître Scrum, XP et Kanban. Malgré tout SAFe n'explique toujours pas comment obtenir l'expression des besoins et constituer le backlog initial même s'il se réfère dans ses niveaux supérieurs à la plupart des techniques, méthodes et frameworks Lean existants. On constatera aussi, et cela représente un changement majeur de réalité que le titre « Framework Agile for Lean Entreprises » a été transformé en « SAFe for Lean Entreprises ».



Par contre, selon Jean-Pierre, l'agilité « originale » demeure dans ce framework bien présente (ce qui n'est pas le cas de bien des autres propositions du même type) mais se limite plus ou moins à l'organisation du PI Planning de la couche « Program ».



CONTINUOUS SPRINT 0

Alors pourquoi un sprint 0 et pourquoi cette notion de « continuous » ? Reprenons : un projet SCRUM démarre avec un sprint 1 après que le backlog de produit se soit avéré suffisamment exhaustif et détaillé pour être estimable par l'équipe de projet (ce qui n'est pas toujours le cas). La proposition « Continuous Sprint 0 » se veut la première réponse structurée, détaillée et surtout Agilisée, à la question « Comment obtenir le backlog ? »

Jean-Pierre Vickoff se propose donc de combler ainsi le « trou dans la raquette » de l'expression des besoins. Pour obtenir ce résultat, il détaille une boîte à outils « complémentaires » à Scrum, SAFe, Kanban, XP et autres approches plus ou moins agiles). Cette proposition est constituée de 12 techniques agilisées (autant que faire se peut compte tenu de l'envergure des préoccupations à couvrir). Cet ensemble se compose de 4 groupes :

- Continuous Exploration,
- Continuous Design,
- Continuous Solution,
- Continuous Delivery.

Chaque groupe de connaissances additionnelles (un tryptique) comporte 3 éléments nommés « cartouches ». Le tout se présente comme un ensemble cohérent de pilotage de projet de développement. Il a été présenté globalement fin 2018 sur le Journaldunet.com. Son contenu est actuellement publié mensuellement par cartouche depuis janvier 2019. Début 2020, une synthèse et surtout l'orchestration de l'usage de l'ensemble en mode « continuos » seront communiquées.

Les communications sur JDN restent simples (un maximum de quelques pages) afin de préserver l'agilité lors de la mise en oeuvre. Par contre, il faut le comprendre, la première des cartouches se situe à la limite haute du périmètre d'un sprint 0 (Anticipation rationnelle du changement) et, les trois éléments du « continuous delivery » sont en dehors de son scope classique. Leurs présences se justifient par la nécessité de maîtriser les modifications liées au concept d'Adaptatif (amplitude, traçabilité) afin de conserver le contrôle du pilotage de la réalisation.

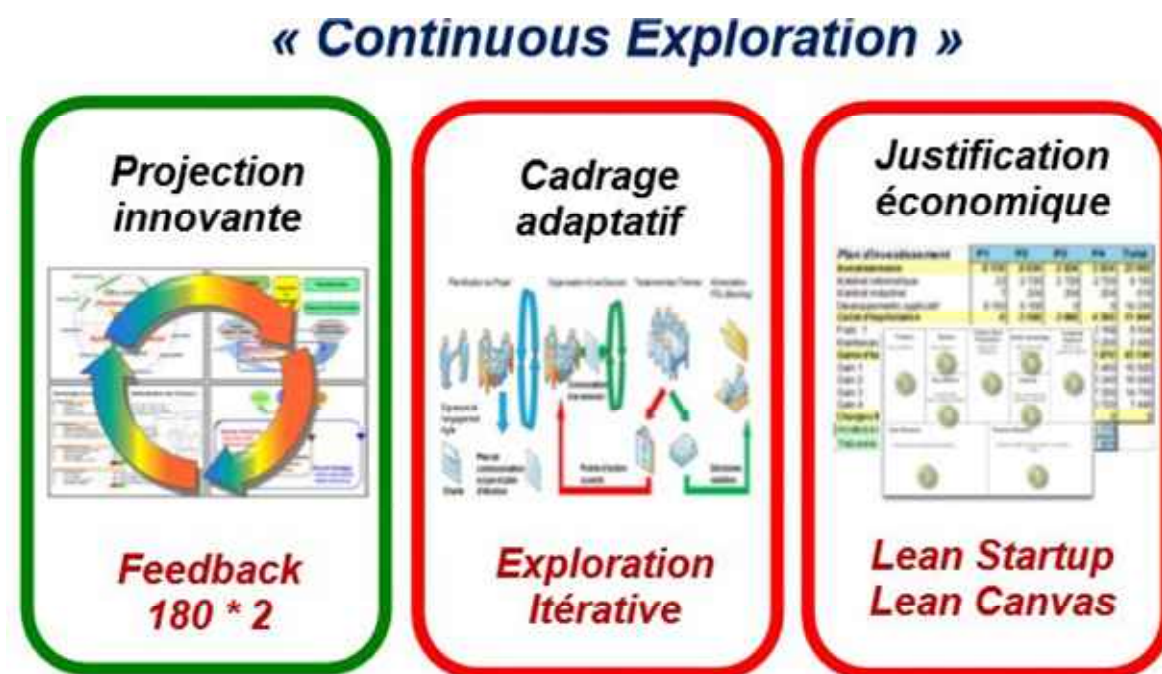
<https://www.journaldunet.com/solutions/expert/70066/methodes-agiles----00---continuons-sprint-0---une-simple--chek-list.shtml>

Pour Jean-Pierre Vickoff, l'essentiel est la continuité de la démarche d'exploration et de mise en oeuvre (dans le sens organisé par SAFe lors du dernier Sprint d'exploration de son « planning de program »).

Les nombreuses questions posées par les participants n'ont pas permis de rentrer dans le détail des 12 cartouches. Nous en présentons donc ci-après uniquement la liste, vous laissant les découvrir progressivement dans les articles parus ou à paraître du journaldunet.

Continuous Exploration

Objectif : permettre à l'organisation apprenante de promouvoir un mode coopératif de projection rationnelle s'appuyant sur l'engagement de ses ressources humaines dans une captation des événements conditionnant l'évolution de sa transformation.



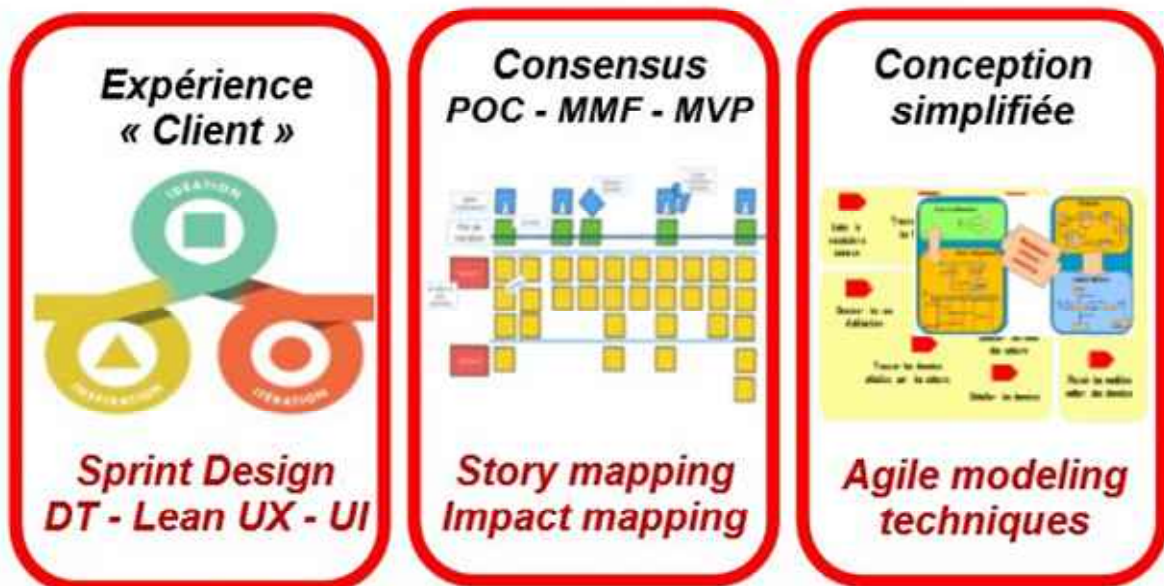
<https://www.journaldunet.com/solutions/expert/70555/01---anticipation-rationnelle-du-changement.shtml>

<https://www.journaldunet.com/solutions/expert/70768/02---exploration-et-cadrage-iteratifs-adaptatifs.shtml>

Continuous Design

Objectif : concevoir une vision produit minimum modélisée de façon agile.

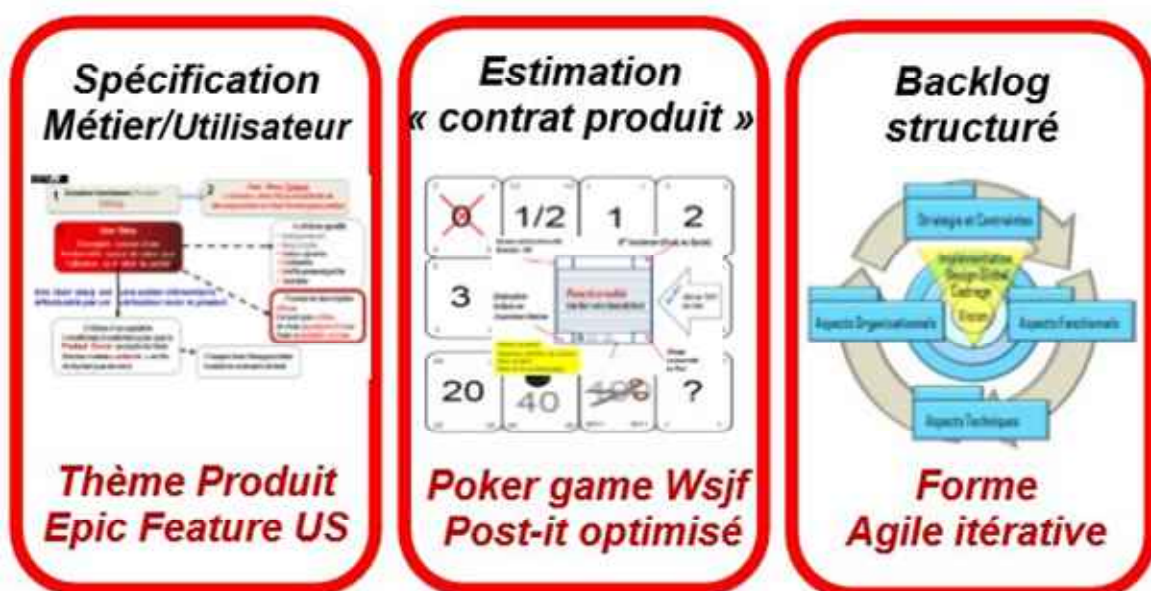
« Continuous Design »



Continuous Solution

Objectif : transformer la vision en exigences de fonctionnalités à produire.

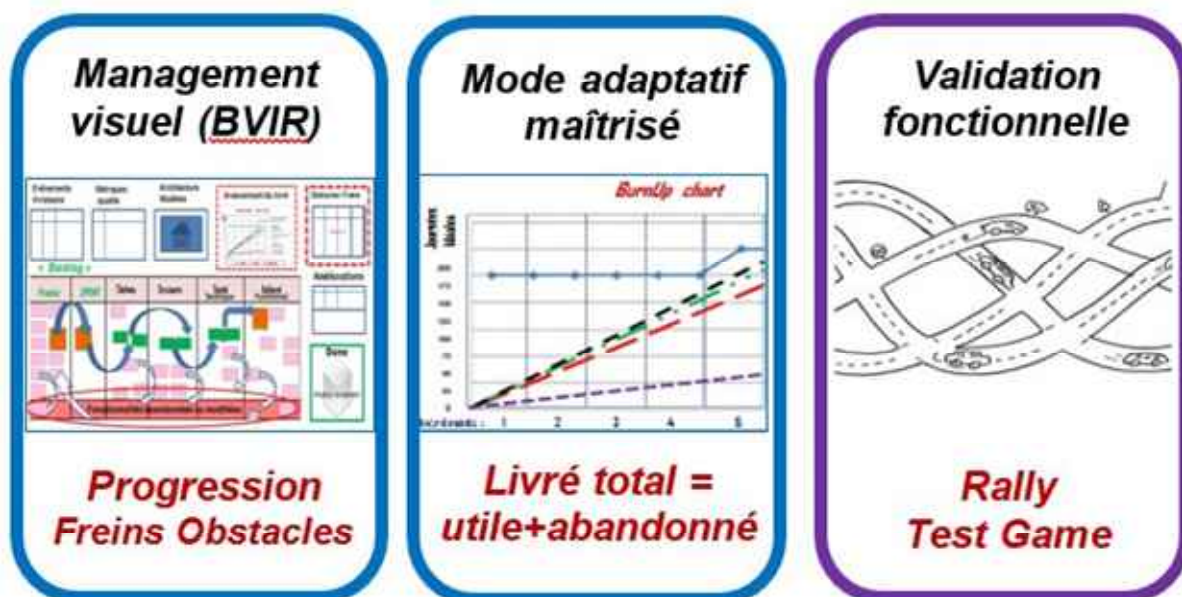
« Continuous Solution »



Continuous Delivery

Objectif : disposer d'une mesure fiable de l'état de la construction d'une solution validée en permanence. Pour pouvoir accepter le changement il faut le contrôler.

« Continuous Delivery »



Note :

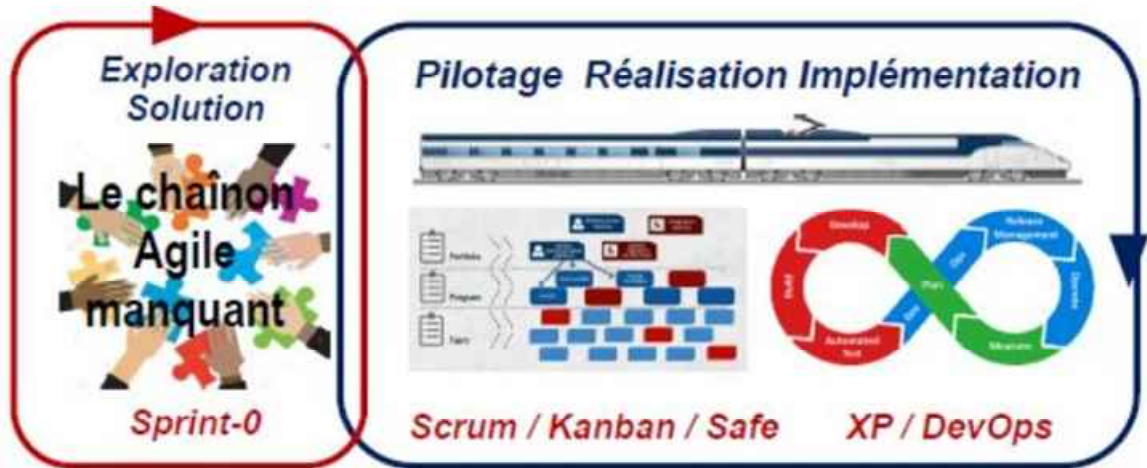
La complexité du détail de la réalisation ne sera jamais à formaliser autrement que lors de la phase de Construction, dans le code, sa documentation, ses tests. **Cette « spécification détaillée – codage – test – validation permanente » co-engageant simultanément métier et informaticiens, c'est cela l'Agilité, et rien d'autre !**

CONCLUSION AGILE

Une recommandation : même dans les cas les plus complexes, faire simple (mais pas simpliste) afin de rester agile.

continuous Sprint 0

pour Scrum, SAFe, ...



SPRINT 0 et au-delà : boîte à outils complémentaires

