

Source : <https://www.lean.org/balle/DisplayObject.cfm?o=4983>

Cher Gemba Coach,

Nous avons un nouveau patron qui ne connaît pas le Lean et qui demande à notre équipe Lean nous mettre plutôt à « l'Agile ». Des conseils?

Dites à votre patron que vous faites déjà de l'Agile – il n'y a pas vraiment de différence entre Agile et Lean. Le patron n'a pas toujours raison, mais il reste le patron. Une règle d'or de la psychologie est que les gens réduisent les problèmes complexes à la partie qu'ils pensent pouvoir résoudre. Un apport important du Lean est qu'il permet faire des hypothèses sur la performance (actuelle et future) d'une zone, en fonction des problèmes que le patron prend en charge, de la façon dont il les cadre et dont il s'y attaque (avec les gens ou bien « fais comme je te dis ou casse-toi »), et de quel genre de solution ils préfèrent.

La vraie question est la suivante : qu'est-ce que vous perdez en abandonnant le Lean et en adoptant l'Agile ? Tout dépend de la manière dont vous pratiquez le Lean. Dans certains cas, vous pourriez faire un pas un avant.

J'étais récemment sur le Gemba dans un service informatique et j'ai eu exactement cette discussion avec le manager Agile. Ils ne veulent pas entendre parler de l'approche Lean de l'entreprise car ils la trouvent trop rigide – ils pensent que dans leur environnement complexe et en évolution rapide, les standards et les processus standards ne peuvent s'appliquer – et je ne suis pas nécessairement en désaccord avec eux.

« Si vous avez utilisé les techniques Kanbans du Lean pour établir un flux d'informations plus étroit entre les concepteurs et les développeurs, alors l'Agile sera probablement un pas en arrière. »

D'un autre côté, ils ont admis avoir de longs délais de livraison, un important carnet de livraison, et beaucoup de retouches même s'ils pratiquaient l'Agile depuis des années, ce qui signifie essentiellement que :

- Ils définissent le travail à travers des « stories »: ils définissent des fonctionnalités avec des « user personas », spécifient ce que ce « user » aimerait faire pour obtenir cela et le décomposent en éléments de travail – stories « épiques » et stories « à coder ».
- Ils travaillent en petits lots : ils livrent le travail une fois par jour ou une fois toutes les deux semaines et essaient de tester le code avant la livraison.
- Ils travaillent en équipe : tous les matins, ils ont un « stand-up meeting » pour discuter du travail à faire et des problèmes qu'ils rencontrent.

Et ça marche ! Ils livrent.

Qu'est-ce que le Lean pourrait bien leur apporter ?

Des Points de Départ Différents

La partie plus compliquée. L'Agile est une victoire incontestable par rapport à la planification d'un système entier et à des mois de code à livrer et à faire fonctionner à la fin. Mais cela reste un système qui organise l'équipe et s'arrête au codeur individuel et donc – le code. Il y a beaucoup de techniques Agiles pour examiner la qualité du code, mais jusqu'à présent, je ne les ai jamais vues vraiment bien appliquées.

À l'autre bout du spectre, les clients des équipes Agiles sont beaucoup plus heureux parce qu'ils voient la livraison continue de fonctionnalités, mais on n'est pas certain que le produit résultant soit plus apprécié ou plus utilisé qu'avec la méthode traditionnelle de Waterfall.

L'un des problèmes clés de la pensée Agile est que – d'après ce que j'ai pu observer – elle est très basée sur les fonctionnalités. Le logiciel qui en résulte a tendance à être chargé de fonctionnalités, dont aucune n'est vraiment utile, et ne correspond pas nécessairement au besoin des utilisateurs sur les quelques fonctionnalités clés dont ils ont vraiment besoin, sur des problèmes plus profonds, tels que la vitesse, les bugs et les glitches, et la sûreté des données.

Le point de départ du Lean est différent – non pas avec l'équipe, mais avec le développeur et le propriétaire du produit ou l'architecte (en termes Lean, l'ingénieur en chef).

Avec l'Agile, l'équipe regarde le tableau d'affichage des tickets, en prend un provenant de l'en-cours des stories et le pousse dans les files d'attente « à faire », « en cours », « en vérification », « fini » – il y a beaucoup de variations pour ces en-têtes, mais ils signifient pratiquement tous la même chose.

Lorsque vous regardez le tableau, certains tickets avancent, d'autres traînent. L'équipe rééchelonne les tickets en fonction de ce qui a été réalisé et de ce qui a planté. La plupart du temps, le manager arbitre et a le dernier mot sur ce qui doit être fait et sur ce qui doit être reporté.

Le Lean commence avec le développeur. Le flux de tickets doit se faire au poste de travail : voici ce sur quoi vous devez travailler ensuite, et ensuite, et c'est tout – comme un cuisinier dans une cuisine recevant les commandes des tables du restaurant.

L'intérêt d'amener la file d'attente au poste de travail est de rendre la personne autonome : elle n'a pas besoin d'une équipe ou d'un patron pour savoir quoi faire. Ils peuvent travailler seuls ET appeler à l'aide s'ils rencontrent un problème.

La raison pour laquelle le travail s'accumule sur les bureaux des gens, c'est que lorsqu'ils rencontrent une difficulté, ils la mettent de côté et commencent quelque chose d'autre – ce qui a du sens. S'ils se heurtent à un deuxième problème, ils mettent une deuxième tâche de côté et ainsi de suite. Nous le faisons tous. Certaines difficultés se résolvent facilement – obtenir de quelqu'un une information manquante. D'autres traîneront et s'aggraveront parce qu'ils sont la partie cachée de l'iceberg de problèmes plus profonds.

Intégrer la Qualité

Le Lean ne fonctionne pas sans l'Andon – la hiérarchie est là pour être une chaîne d'aide. Lorsqu'une personne rencontre un problème, au lieu de mettre de côté et de travailler sur autre chose, le management répond présent et résout le problème, plutôt que de passer à une autre tâche. Ce n'est jamais facile, mais c'est ainsi que nous découvrons les vrais problèmes dès le début et les résolvons.

Cette chose simple mais difficile est la clé de l'intégration de la qualité : en fin de compte, le produit sera beaucoup plus robuste que si nous continuions à travailler, nous poussions tous les problèmes vers l'aval et si nous espérions pouvoir les résoudre tous par magie lors de l'intégration – voyez ce qui se passe lorsque les managers essaient cela avec la conception des avions. Aïe.

Et nous arrivons ici à un malentendu majeur au sujet du Lean. L'Andon ne consiste pas seulement à aider les développeurs et les former à faire des choses quand nous le pouvons. L'Andon vise à comprendre pourquoi le développeur rencontre un problème dans un premier temps et où nous nous sommes trompés dans l'instruction de travail.

L'autre terme de l'équation est l'architecte (en termes Agiles, l'ingénieur en chef en Lean), la personne qui garde une vision globale de :

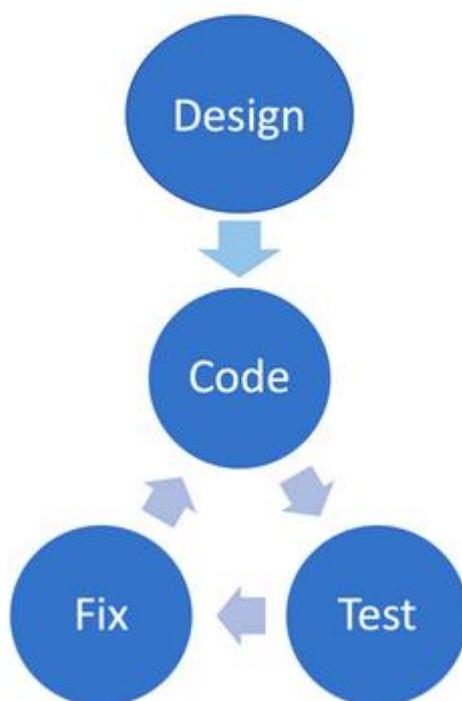
Bénéfices pour les clients :
fonctionnalités-- technologies -- intégration dans l'ensemble du système -- coût total.

Il faut que quelqu'un s'occupe de concevoir le travail. En Lean, l'ingénierie conçoit le travail dans les moindres détails et utilise ensuite les problèmes de production pour affiner cette conception. Dans l'Agile, le travail est conçu avec un coup de pinceau plus large et les développeurs trouvent un moyen de le faire fonctionner.

Dans la pensée Lean, nous mettons l'accent sur les nouvelles fonctionnalités que nous apportons au marché et nous nous assurons qu'elles ne nuisent pas à la performance globale du produit, et qu'elles sont solidement conçues avec des standards afin que les équipes de production sachent ce qu'elles doivent faire. Les caractéristiques innovantes ou l'utilisation d'une technologie innovante sont employées avec parcimonie, dans des projets qui font l'objet d'une surveillance intense.

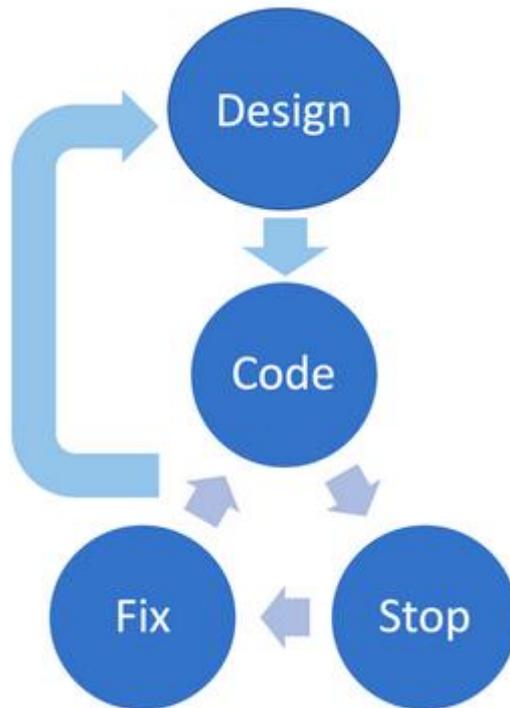
Le développement Agile a tendance à être beaucoup plus laxiste sur la production du code, et les architectes ont tendance à voir des progrès en termes d'ajout de fonctionnalités pour satisfaire davantage les besoins des utilisateurs (attention toutefois – il s'agit d'une généralité, il n'existe pas deux architectes qui ont le même point de vue sur cette question).

C'est une différence fondamentale qui explique une grande partie des discussions Agile/Lean.



En Lean, le mécanisme de rétroaction vers la conception n'est pas la réaction de l'utilisateur au produit complet une fois qu'il est mis à disposition, mais au processus même de stop-and-fix (à condition que les concepteurs sont réellement intéressés).

En Lean, la production elle-même est la méthode de test de la conception.



Les ateliers Kaizen sont vraiment intéressants en Lean lorsque nous abordons des problèmes de design profonds qui peuvent ensuite être expliqués aux ingénieurs. Les explications ne seront utiles que si le concepteur a une connaissance approfondie des avantages pour le client, des fonctionnalités et de la technologie nécessaire pour leur fournir facilement le bon niveau de qualité, à un coût raisonnable. Pas facile.

Au bout du compte, les techniques Lean ont pour but d'aider les gens à voir au-delà des frontières de leur travail fonctionnel – l'objectif est une meilleure collaboration tout au long de la chaîne et la suppression des coûts – gaspillage – des lacunes dans la coopération (rien de plus inutile qu'un produit non utilisé par les utilisateurs).

Malheureusement, de nombreux programmes Lean sont exclusivement axés sur l'amélioration de la productivité de chaque équipe et passent complètement à côté de l'objectif global de « vendre un, fabriquer un – pour s'assurer que vous continuez à en vendre un ».

Pour répondre à votre question, cela dépend vraiment du type de programme Lean que vous menez. Si l'accent a été mis sur le travail standardisé et les processus standardisés, comme certains essaient de le faire, l'Agile sera un pas en avant. Cela détendra l'atmosphère et vous empêchera de forcer les clients et le personnel à se conformer à vos processus arbitraires. D'un autre côté, si vous avez utilisé les techniques Kanban du Lean pour établir un flux d'informations plus étroit entre les concepteurs et les développeurs, alors l'Agile sera probablement un pas en arrière, qui permettra aux managers intermédiaires de reprendre le contrôle du qui-fait-quoi et de ne plus examiner le code des développeurs?

Les questions clés sont les suivantes : Qu'essayez-vous de faire ? Où cherchez-vous des gains ? Quel type de Lean pratiquez-vous actuellement ? Quel genre d'Agile votre nouveau patron souhaite-t-il que vous fassiez ? Je n'ai pas de conseil : tout dépend !

Traduit de l'américain par Nicolas Villemain, Marc-Antoine Guichard et François Lopez