

Cher Gemba Coach,

Votre précédent post a soulevé des questions de fond pour moi et mon équipe de développement logiciel. Je vous en remercie. Une de ces questions est : quels sont nos « points critiques du procédé » ? Comment, en tant que managers Lean pouvons-nous favoriser une réflexion collective sur le sujet ? Quels comportements avoir pour faire en sorte que « de meilleures personnes travaillent mieux ensemble pour créer de meilleurs processus ? »

Merci pour votre commentaire, cela fait plaisir. Il y a deux aspects à cette question. En développement logiciel, le « point critique du procédé » est tout simplement le code.

En production, un point critique de procédé est quelque-chose de répétable. On pourrait penser que dans le cas de la fixation d'une poignée sur la porte d'une voiture, chaque poignée est fixée de la même manière sur chaque porte de la ligne. C'est vrai en partie, mais il existe des variations : comment cela se passe-t-il si plusieurs modèles différents sont assemblés sur la même ligne – les poignées se fixent certainement de manière similaire, mais avec de légères variations. Et la qualité réside précisément dans cette différence.

Il peut aussi arriver qu'il y ait des variations entre les opérateurs, par exemple si un jour vous avez un homme grand, alors que d'habitude c'est une dame petite. Leurs gestes seront probablement différents : leur ligne de vision, leur angle d'attaque ne seront pas les mêmes. Au bout du compte, le point critique du procédé sera différent car la poignée ne sera pas présentée exactement de la même manière à son emplacement sur la porte.

En termes Lean, la qualité et la flexibilité découlent du fait que nous considérons qu'il est de notre responsabilité (en fait, cela doit être notre obsession) de bien comprendre comment la poignée est fixée sur la porte : comment elle est présentée, comment les fixations sont mises en place, avec quels outils, comment l'équipier comprend la tâche et les gestes à réaliser.

Cela dit, bien évidemment, certains points critiques de procédé sont plus délicats que d'autres. Par exemple, il n'y a pas cinquante manières de visser une vis, et cela varie peu d'une voiture ou d'un opérateur à un autre. Au bout du compte, la vis est serrée et remplit sa fonction.

Mais certains sont très sensibles au contexte – imaginez-vous comment placer un pare-brise sur une carrosserie sans rien rayer.

Mon expérience du développement logiciel est que tout d'abord, tout le monde suppose que chaque bout de code est unique (pourquoi en serait-il ainsi?) et qu'il faut laisser chaque développeur faire tranquillement son boulot (comme s'il savait exactement quoi faire).

Ces deux hypothèses sont exactes dans une large mesure – au bout du compte, le logiciel est le résultat de ce que les développeurs ont fait, et a globalement la performance qu'on attend de lui, quitte à nécessiter un peu de débogage. On considère que c'est normal, inévitable, et que c'est le prix à payer en développement logiciel.

Mais est-ce vrai?

Je ne sais pas sur quel type de logiciel vous travaillez, mais j'aimerais vous parler d'un Gemba sur lequel j'étais récemment – un développement d'appli web chez Theodo, un leader dans ce domaine.

La vitesse à laquelle une page se charge sur votre ordinateur ou votre téléphone est un élément critique de performance face à vos concurrents. Intuitivement, je me dis que si la page met plus de 5 secondes à se charger, vous êtes dans les 25% les plus lents. Les utilisateurs souhaitent un temps de 2 secondes, et abandonnent au bout de 3 secondes. Si votre page met moins d'une seconde à charger, vous êtes au top du métier. [Amazon prétend que pour chaque seconde de délai, le taux de conversion baisse de 7%](#)

A l'évidence, la manière de coder est importante. Intuitivement encore, j'imagine que le temps de chargement de la page dépend du temps nécessaire à la génération des fichiers (temps de traitement par les serveurs, optimisation de la base de données, ...), de la bande passante de votre connexion (ça, on n'y peut pas grand-chose), de la taille des fichiers et de la rapidité de votre navigateur.

Je vais maintenant m'aventurer sur des concepts que je ne maîtrise pas vraiment: on me dit que la manière de coder en Java (c'est quoi ce truc ?) a un très fort impact sur la rapidité de chargement de la page.

Et comme on laisse chaque développeur coder son script Java dans son coin, ceux qui sont bons codent des pages qui se chargent vite, et les autres créent des pages lentes.

Une manière Lean d'aborder le problème serait de soutenir le team leader pour qu'il devienne un expert du lien entre codage Java et rapidité de chargement de la page – et c'est là que se situe le point critique du procédé.

Le team leader peut alors encourager le partage de pratiques entre développeurs, tenter de caractériser et d'isoler des problèmes typiques, demander aux gens de commencer à résoudre ces problèmes à leur façon, et finalement commencer à dégager d'éventuels standards. Du Kaizen donc !

Considérons maintenant la deuxième partie de votre question. Sur ce Gemba, la rapidité de chargement ne dépend pas seulement du code Java, mais de bien d'autres facteurs. Ces facteurs ne dépendent pas de l'équipe de développement Java – il peut s'agir de traitement de données, de la conception de la page, de sa taille, etc. ... Quand on observe cela de plus près, on se rend donc compte, après avoir nettoyé les erreurs les plus grossières, que toutes ces lignes de code qui ralentissent le chargement de la page ont leur utilité.

Livré à lui-même, chaque développeur trouve une solution particulière à chaque problème, un peu comme si on laissait chaque opérateur inventer sa propre manière de fixer la poignée de la porte. Mais si vous regardez en détail, vous vous apercevez qu'ils ont des manières différentes de surmonter de vraies difficultés, liées à la conception de la porte ou de la poignée, ou de sa fixation.

Ce qui se passe dans d'autres services loin du Gemba, la manière de transmettre le travail au système sans se préoccuper de son vrai destinataire, et de laisser les développeurs Java tout seuls face au problème de chargement de la page, tout cela peut avoir un impact disproportionné sur le temps de chargement.

Et le moment clé du travail d'équipe a lieu quand le PDG fait un Gemba Walk, va voir sur le terrain les efforts de Kaizen pour nettoyer les scripts Java, et qu'il regarde en détail les standards que ses équipes de développement sont en train de bâtir (le point critique de procédé et la manière dont les équipes l'appréhendent), pour ensuite demander à son codir – les responsables des équipes de codage – d'expliquer les problèmes qu'à l'évidence leurs équipes doivent affronter : pourquoi leur fournissez-vous un code fonctionnel qui soit si difficile à charger ?

La puissance des points critiques de procédé

Par exemple, chez Theodo, au lieu de s'attacher à résoudre les problèmes les plus évidents, l'équipe Kaizen pilotée par le Team Leader a exploré d'autres pistes: compression des fichiers, élimination de requêtes inutiles sur les serveurs, mise en priorité des requêtes utilisateurs (en chargeant en priorité le contenu le plus critique), et optimisation du code original par l'élimination de requêtes en double. Tout cela a nécessité une analyse fine du code et le traitement des problèmes un à un.

Le « moment Lean » s'est produit quand les responsables des fonctions primaires ont réalisé en observant le Kaizen (sous l'impulsion forte du PDG) que le travail amont qu'ils avaient réalisé était la cause des difficultés de l'équipe Java, et qu'une grande part de la non standardisation venait de problèmes de livraison entre des processus internes. De plus, une fois que ces problèmes ont été mieux compris, il a été plus facile de les partager entre services au sein de l'entreprise, afin que chacun comprenne à quoi prêter attention.

C'est comme se rendre compte que la poignée a été mal conçue, arrive avec des défauts de fabrication, que la porte n'a pas été conçue pour qu'il soit facile d'y fixer la poignée, que les outils utiliser pour fixer la poignée s'enrayent une fois sur dix Il est maintenant évident que les membres de l'équipe « poignée » ont dû s'accommoder de ce qu'ils avaient, ce qui est dangereux en termes de qualité et dispendieux en terme d'efficacité – mais ils sont bien obligés de faire avec ce qu'on leur donne...

Le Kaizen au point critique du procédé génère trois opportunités très puissantes :

- Amener les gens qui font le travail au quotidien à reconsidérer leurs manières de travailler, à réfléchir en profondeur à leur propres techniques, et à imaginer, en marge de l'analyse fine de problèmes précis, de nouvelles manière de faire qui soient bien plus performantes
- Amener les responsables des fonctions support et des processus amont à s'intéresser à ces idées, à comprendre que la plupart des équipes sont aux prises avec une matière première de piètre qualité, et que bien des problèmes devraient être traités dans les processus amont ou au sein des fonctions support.
- À l'occasion, le PDG peut découvrir au détour d'une suggestion d'une équipe locale une idée qui pourra avoir un impact sur l'ensemble de son activité. En effet, le fait de comprendre les liens entre les activités des différents services peut lui faire réaliser qu'un problème très local est en fait le symptôme d'un gaspillage endémique que personne n'avait encore détecté, et qu'il a un fort potentiel disruptif.

Et tout cela à partir du point critique de procédé !

Traduit de l'américain par François Lopez

Source du document: <https://www.lean.org/balle/DisplayObject.cfm?o=3593>