

De l'entreprise traditionnelle à l'entreprise logicielle

par

■ Yves Caseau ■

Head of Digital and Innovation – AXA Group

Membre de l'Académie des technologies

En bref

Le digital n'est désormais plus le pré carré des bâtisseurs de cathédrales que furent les informaticiens des dernières décennies. Son irruption dans tous les secteurs de la vie, la puissance des réseaux, l'évolution des clients ont fait émerger une nouvelle culture, celle des big data et des *big players*, Google, Apple, Amazon et consorts, qui ont compris les premiers que la rapidité n'était pas incompatible avec l'excellence. Le temps est venu des systèmes jetables, des processus de fabrication automatisés, du *machine learning*, des plateformes et du *cloud*, tous nécessitant un code noble, structuré et élégant. Les nouveaux programmeurs, au sein de *cross-functional teams*, inventent aujourd'hui, sous nos yeux, l'entreprise logicielle.

Compte rendu rédigé par Pascal Lefebvre

L'Association des Amis de l'École de Paris du management et Cap Digital organisent des débats et en diffusent des comptes rendus, les idées restant de la seule responsabilité de leurs auteurs. Ils peuvent également diffuser les commentaires que suscitent ces documents.

Un monde de compétition numérique

Pour les entreprises, la première révolution est d'être désormais en contact permanent avec des clients détenant un pouvoir qui a quitté la force de production pour passer entre leurs mains. Les marchés sont devenus, en quelque sorte, des conversations et, dans ce monde nouveau, les entreprises doivent écouter leurs clients, détecter leurs besoins et les opportunités de les satisfaire.

La deuxième révolution est que les clients sont devenus les architectes de leur propre expérience. On ne cherche plus tant à leur vendre un bien ou un service qu'à produire une expérience dans laquelle ils tiennent le rôle principal.

Une troisième révolution vient s'y ajouter: les entreprises doivent passer de plans stratégiques, qu'il s'agissait d'exécuter, à une forme d'agilité stratégique. Tout l'art du dirigeant va alors être de maintenir le cap dans cet environnement mouvant où le client change constamment d'attentes.

En conséquence, il n'est désormais plus possible d'être un acteur efficace sans avoir une stratégie logicielle devenue le cœur de la production de valeur numérique. Ce que l'entreprise sait faire ou ne pas faire, ce qu'elle achète, comment elle joue avec l'autre, etc., vont déterminer les écosystèmes auxquels elle participe. Son agilité stratégique va donc être déterminée, pour une large part, par ses stratégies logicielles.

Un autre point frappant est que cet effort logiciel a profondément changé en quinze ans. Il y a trente ans, il était noble et passionnant de faire du logiciel, puis sont arrivés l'*outsourcing* du code et des unités d'œuvre réalisées en Inde qui ont, pour un temps, radicalement changé nos façons de faire. Mais nous revenons à notre point de départ, car un code qui change sans arrêt doit désormais être beau, partageable et montrable. Si, comme c'est le cas chez Google, un module sur deux doit être changé chaque mois et que votre code n'est pas noble, structuré et élégant, vous ne serez pas en mesure de devenir un acteur dominant. En d'autres termes, si l'objet logiciel change de façon continue, la façon de faire devient tout aussi importante que ce que l'on construit. Le temps des systèmes pérennes est désormais révolu, mais lorsque l'on est dans des systèmes jetables, tous les processus de fabrication deviennent absolument fondamentaux et s'ils ne sont pas extrêmement automatisés et de très haute qualité, nul ne sera capable de se maintenir dans la course.

L'apprentissage organisationnel permanent

Puisque les objets sont en perpétuel renouvellement, les technologies, les outils et les éléments que l'on assemble le sont également. Pour l'entreprise, la grande ambition de l'apprentissage organisationnel permanent, plus qu'une évidence intellectuelle, est devenue une nécessité impérieuse.

Nous sommes désormais passés de la *black box* à la *white box*, voire à la *transparent box*. Cette nouvelle façon de faire du code implique de le réassembler en permanence. Le code source, que l'on avait longtemps caché, redevient la matière première. On fabrique désormais de grands systèmes avec 90% de code déjà créé par ailleurs, et de code d'assemblage pour le reste. Malgré les efforts très importants réalisés pour faire de l'assemblage automatique, le monde digital, tel qu'il est aujourd'hui dans la Silicon Valley, est essentiellement constitué de scripts Python et d'assemblages en tous sens. Cela signe donc le retour à des usages et une culture logicielle tels que j'ai personnellement pu les connaître et les pratiquer lors de mes débuts dans ce métier.

L'autre caractéristique du moment est que, puisque l'on fabrique très rapidement des choses qui n'ont qu'une faible durée de vie, il devient fondamental de ne pas avoir les coûts par ligne de programmes que nous connaissions il y a dix ans. Il faut donc chercher à les réduire de façon massive. Les logiciels *open source* vont ainsi faire apparaître des trésors de valeur car on a besoin de logiciels non seulement beaucoup plus nombreux qu'avant, mais aussi qui changent plus souvent et qui doivent être de très grande qualité.

Face à toutes ces contraintes, l'intérêt du logiciel *open source* est qu'il est le seul à être réalisé de façon incrémentale, avec beaucoup de *feed-back* et de *eyeballs*¹. La recette de fabrication tient d'une part à ce que vous allez trouver dans le code *open source* et, d'autre part, à ce que vous allez développer vous-même.

Par ailleurs, la seule façon de développer un logiciel évolutif et de grande qualité est qu'il soit soumis à un très grand nombre de testeurs. La productivité, en trente ans, en termes de nombre de lignes ou de défauts (*defects*) par heure, ne progresse que très lentement. Pour l'améliorer, cela impose donc de travailler autrement, c'est-à-dire en mode communautaire.

Un réel jeu culturel

Que font les compagnies de référence, les GAFAs² ou autres, tels Netflix ou Spotify? Ce qui frappe en premier lieu, c'est que toutes ont une culture fondée sur les données et les décisions factuelles. On mesure absolument tout afin d'entrer dans une démarche d'amélioration continue, facile à mettre en œuvre dès lors que l'on est dans des systèmes de logiciels que l'on renouvelle sans arrêt. Améliorer une cathédrale de façon continue n'est pas évident mais, avec un objet que l'on remplace tous les mois, cela devient possible.

Parmi tout ce que l'on mesure de façon obsessionnelle, le temps que passe le client à utiliser votre service figure en première place. Apprendre qu'il n'y a rien de plus précieux que le temps de ses clients serait une grande leçon pour les entreprises du CAC 40. Le genre de chose à absolument éviter, c'est de développer un site dans le seul but que des tâches auparavant dévolues au *back office* soient désormais réalisées par le client. Chacun a déjà vécu cette déplaisante expérience où, pour la dixième fois, il doit remplir le même formulaire d'informations pour un même service.

L'autre caractéristique de ce *new way of working* est l'intégration du processus de fabrication des logiciels. Si je passe mon temps à tout changer, les coûts de transaction avec des producteurs externes de logiciel deviennent trop importants. Les grands du web ont donc tous développé une maîtrise remarquable de leur création logicielle.

Le mot *plateforme* est également devenu un incontournable. Ce n'est pas un hasard car la plateforme a deux avantages. C'est un lieu de diversité, ce qui crée de la satisfaction client, mais c'est aussi un lieu qui génère de la valeur économique en maximisant la réutilisation. On pourrait donc considérer que c'est la solution d'une équation permettant de faire plus de choses en moins de temps, avec moins d'efforts. Cela implique alors d'aller enrôler de l'intelligence à l'extérieur, pour des raisons tant économiques que d'innovation. L'extraordinaire course des géants du web, d'une part sur les méthodes de l'intelligence artificielle et, d'autre part, sur l'exposition de ce qu'ils font, montre qu'ils sont bien conscients qu'à vouloir tout faire tout seuls, ils risquent de mauvaises surprises venant de la communauté ouverte.

Dans les années 2000, les grandes compagnies de télécommunications, comme Orange ou Verizon, avaient déjà la même tendance à développer des plateformes. Toutes ont eu la volonté d'en créer et d'y faire venir les développeurs, mais c'en sont d'autres qui ont marché, construites par des gens qui avaient une réelle culture de l'*open source*. Aujourd'hui, le challenge des entreprises françaises est d'internaliser cette culture, car la meilleure façon de réaliser un logiciel qui va donner l'envie à d'autres développeurs de venir jouer avec, c'est, d'abord, d'avoir soi-même passé beaucoup de temps sur la plateforme des autres. C'est un réel jeu culturel, du *give and take*, qui suppose de commencer par contribuer au travail des autres et d'abandonner la vision égocentrique si naturelle chez les acteurs dominants.

1. « *Beaucoup d'yeux qui voient le même code* » : l'*open source* conduit à une « revue de code » généralisée, ce qui apporte qualité et simplicité.

2. GAFAs désigne les actuels géants du Net américains : Google, Apple, Facebook, Amazon.