

# Le mythe du MVP

*Expliquons ce qu'un produit minimum viable (ce que les initiés appellent un MVP) devrait ou ne devrait pas être.*

Temps de lecture : minute

---

2 décembre 2019

Pour vous donner un peu de contexte, le MVP ne trouve pas ses origines dans le monde des startups mais il a gagné en popularité, permettant aux petites entreprises qui disposent de ressources limitées d'entrer rapidement sur le marché et de prouver leurs idées de produits.

L'objectif du MVP est de mettre sur le marché une version rudimentaire de votre produit que de vrais clients peuvent acheter et utiliser. Ensuite, vous éliminez les échecs, corrigez les points faibles et, surtout, déterminez où vos clients trouvent de la valeur et où ils ne le font pas.

Ne vous y méprenez pas, je suis un fervent partisan du concept de MVP. En fait, je me considère moi-même comme un vrai fanatique. Cela dit, j'ai bien vu le concept être galvaudé, en particulier ces dernières années, et je remarque que la définition de MVP a été légèrement modifiée. Cela devient une fin en soi. Et c'est dangereux.

Un MVP est en effet censé être la version la plus rapide du produit que vous pouvez commercialiser. Mais lorsque j'utilise le mot "plus rapide", cette réduction du délai de mise sur le marché est obtenue par une réduction de l'ensemble des fonctionnalités et non par une réduction de la qualité. Dernièrement, je vois le pendule s'éloigner trop souvent de la qualité.

Pour illustrer à quoi devrait ressembler un MVP, prenons un exemple

semi-fictif facile à suivre. Si nous construisions Uber à partir de rien, avant qu'aucune application n'apparaisse, la construction de l'infrastructure technique et opérationnelle nécessaire à la mise en place du service serait extrêmement lourde et d'un coût prohibitif. Lequel ? Je ne sais pas exactement. Mais ce serait cher, très cher. Supposons que nous aurions besoin de dizaines de millions de dollars pour prouver qu'Uber fonctionnerait comme prévu. Notre startup ne pourrait obtenir ce type de capitaux au démarrage à moins de pouvoir prouver que le modèle soit fonctionnel et que les gens finiraient par ne plus pouvoir s'en passer. C'est l'histoire de l'oeuf et de la poule.

Sauf si nous nous tournons vers un MVP.

Je ne sais pas si c'est ainsi que cela a été fait. Je n'étais pas là. Mais pour les besoins de notre exemple, disons que les choses pourraient être faites comme ceci : le MVP pourrait être une application simple avec un gros bouton à cliquer pour commander une course. La localisation du client est automatiquement faite grâce à son mobile. Ensuite, nous enverrions peut-être un e-mail contenant tous les détails de la course et du chauffeur à une équipe d'assistance 24h/24, 7j/7, peut-être même l'équipe fondatrice, qui appellerait ou enverrait un message à un groupe de conducteurs test (amis) et organiserait la course. Lorsque le trajet serait terminé, le paiement pourrait être effectué avec un paiement par carte, réalisé manuellement.

Dans ce faux exemple d'Uber, il est probable que le scénario de test critique consiste à localiser automatiquement le client. Si nous n'avons pas cela, notre version d'Uber ne mériterait pas d'exister. Nous pourrions embaucher manuellement les pilotes et déterminer comment nous les intégrerions au processus. Nous pourrions nous tromper en déterminant le prix avant d'accepter le trajet. Nous pourrions surpayer pour ajouter le paiement à l'application.

C'est la raison pour laquelle nous travaillerions sur ce MVP - pour tester la viabilité des aspects les plus importants de ce qui rendrait le produit digne du temps et des ressources nécessaires pour atteindre le milliard de dollars. Une fois que nous aurions des données qui diraient : *"Cela fonctionne. Les gens vont l'utiliser. C'est rentable. Nous pouvons évoluer sans chuter"*, nous pourrions affiner ces aspects importants, puis lever des fonds et aller au charbon. Pour ainsi dire.

Donc, c'est un MVP. C'est là que la définition devient floue.

Trop souvent aujourd'hui, nous voyons des MVP qui tentent d'en faire trop, et qui en même temps font mal les choses. Cela crée de nombreux dégâts :

1 - Nous mettons des déchets sur le marché et créons immédiatement une expérience client déplorable. Les clients peuvent vivre avec un produit qui ne fait pas tout ce qu'ils veulent. Ils n'accepteront pas un produit dont la valeur ajoutée est mal voire pas du tout calibrée. Si le cas d'utilisation critique n'est pas solide, nous ne devrions pas publier de MVP.

2 - Nous sacrifions des éléments importants comme la sécurité et la confidentialité en pensant que nous n'en avons pas besoin pour le moment. Au-delà des points d'attention moraux et éthiques, c'est quelque chose qui tuera le produit et peut-être même l'entreprise si l'exposition est trop grande. Vous ne pouvez jamais mettre vos clients en danger, si minime soit-il. C'est l'équivalent de ne jamais créer une entreprise sans avocat car même si vous faites bien les choses, vous n'avez aucune idée du genre de problèmes qui vous attendent avant d'être au fond du trou.

3 - Nous avons échoué à sortir un MVP viable. Notre version devrait tout au plus consister en un test de quelques fonctionnalités, dont nous savons que les clients veulent. Si nos clients ignorent certaines

fonctionnalités, nous devons savoir pourquoi. Et si la raison est "cela n'a pas fonctionné", le test devient inutile. Cela devient également coûteux, puisque nous nous précipitons sur le marché pour re-tester notre hypothèse.

Dans notre exemple Uber, tous les éléments non automatisés devraient fonctionner correctement, à chaque fois. Nous devrions veiller à ce que le client arrive là où il devrait aller rapidement, en toute sécurité et à peu de frais, avec peu de maux de tête pour lui, ce qui signifierait beaucoup de maux de tête pour nous.

Maintenant, vous pouvez vous demander si Uber a évolué, en particulier dans les domaines de la sécurité des conducteurs et des voyageurs, de la situation économique des conducteurs et des réactions négatives de l'industrie et du gouvernement. Je ne vais pas défendre la méthode d'Uber mais permettez-moi d'ajouter quelques nuances. Bien que nous ne puissions pas mettre n'importe quoi sur le marché et espérer tout apprendre, nous ne pouvons pas non plus être à 100% à l'épreuve des balles. Notre MVP devrait échouer, surtout aux limites de performances ou de cas extrêmes. Plus nous passons de temps à bricoler et à assurer la qualité, à tester l'alpha, à superposer et à faire du lobbying, moins nous avons à apprendre et à itérer.

Je ne dis pas que c'est bien. Je dis que c'est comme ça.

Heureusement, c'est là que les données et les meilleures pratiques entrent en jeu. Nous ne devrions jamais être complètement à l'aise avec l'idée de commercialiser un nouveau produit ou une nouvelle version. Nous ne devrions jamais être surpris lorsque quelque chose ne va pas et nous ne devrions jamais être à plus de quelques clics de la recherche de la source et de l'arrêt du saignement.

Etre capable de changer et de peaufiner à la volée est la marque de

fabrique d'un bon démarrage. Les clients pourraient même nous dire que nous le faisons mal 90% du temps et bien seulement 10% mais si nous pouvons capitaliser sur ces 10%, nous pourrions faire quelque chose de bien. Notre MVP devrait pouvoir nous permettre d'effectuer ces changements rapidement et facilement. Cela signifie qu'un MVP ne devrait jamais manquer d'une boucle de rétroaction, d'une architecture facilement modifiable et d'un commutateur d'arrêt.

Un MVP nécessite un ensemble de fonctionnalités réduit, une haute qualité, une automatisation réduite et une flexibilité maximale. Une fois tout cela réuni, la seule chose qui nous empêche de nous lancer est la peur elle-même. Et dans une startup, la peur est une bonne chose.

---

Cet article est une traduction du post original de Joe Procopio sur Medium

Joe Procopio est un entrepreneur " multi-exit " et " multi-échecs ". Il développe actuellement Spiffy, une plateforme dédiée à la maintenance de véhicules " on-demand ". Auparavant, Joe a fondé et vendu Automated Insights à Vista Equity Partners et lancé puis vendu ExitEvent à Capitol Broadcasting Corporation. Vous pouvez obtenir des conseils auprès de Joe sur son site internet dédié.

---

Article écrit par Joe Procopio