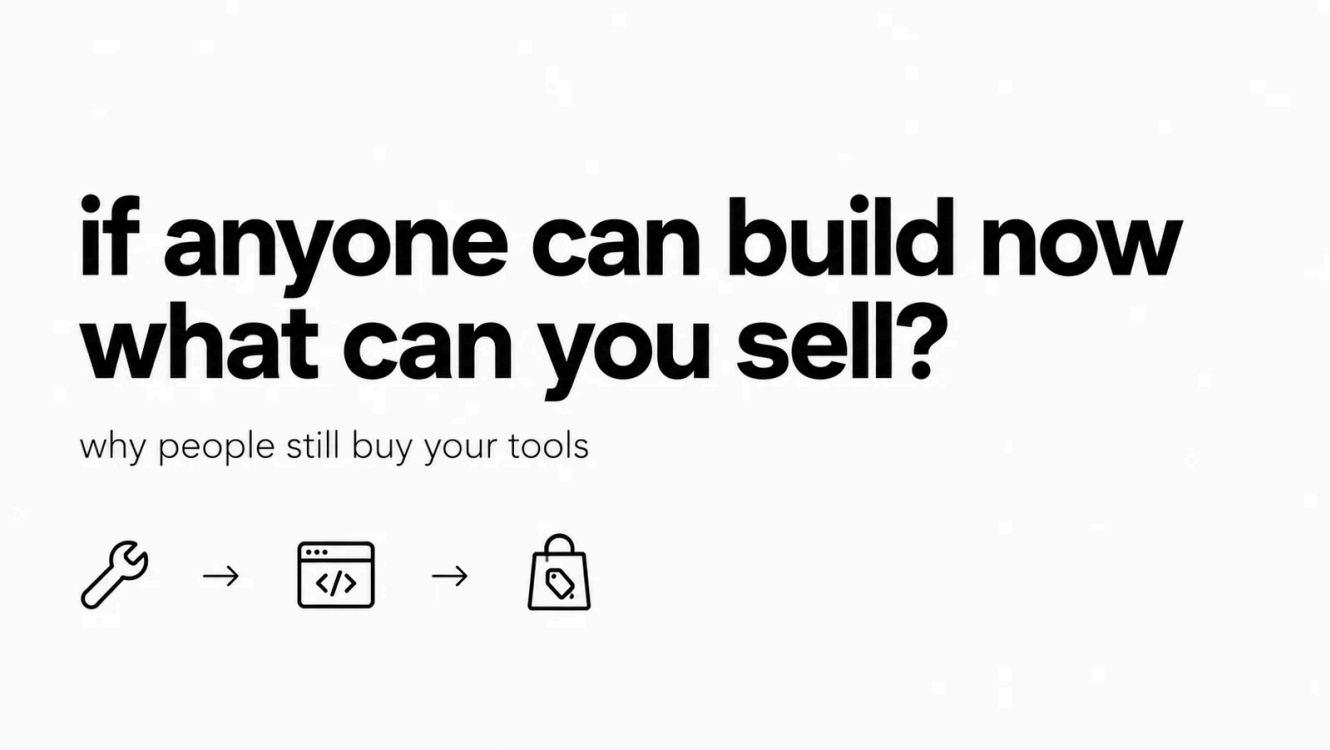
Si tout le monde peut construire maintenant, que pouvez-vous vraiment vendre ?

Pourquoi les gens achètent encore vos outils.



01 Les gens paient encore pour NE PAS construire eux-mêmes

02 Vendez le savoir accumulé, pas seulement le logiciel

03 La distribution devient encore plus importante

04 Vendez ce qui entoure l’outil

05 Les produits qui s’améliorent parce que d’autres les utilisent

06 Parfois, la réponse est littéralement : ne vendez pas de logiciel

07 Et évidemment, certains outils sont tout simplement foutus

Sept angles sur ce que les clients paient encore quand construire devient plus facile.

La question inconfortable

Pourquoi paieraient-ils chaque mois ?

Cette question devient un peu inconfortable. Disons que vous construisez un petit SaaS aujourd'hui, un tableau de bord interne, un workflow avec de l'IA, un truc d'analyse de niche. Il y a quelques années, on pouvait dire « bon, le client ne peut pas construire ça lui-même » et c'était un argument assez raisonnable. Maintenant... on en est sûrs ?

Quelqu'un qui se débrouille raisonnablement avec la technologie peut ouvrir ChatGPT, Lovable, Replit, Cursor, n8n, etc. et obtenir quelque chose qui fait 70% du travail en un week-end. Ce ne sera pas exactement votre produit, probablement pas aussi joli non plus, mais pour son propre usage ça peut suffire. Alors pourquoi vous paie-t-il chaque mois ? Je trouve que pas mal d'idées de SaaS deviennent plus fragiles quand on reste vraiment un moment avec cette question, surtout comparées à il y a seulement 2 ans.

Si le tout est essentiellement une interface autour d'une API, je serais un peu nerveux. On peut le décrire en un paragraphe et obtenir une version approximative avant le dîner. Ça ne veut pas automatiquement dire qu'il n'y a pas de business, mais « je sais faire ça et vous non » vous aide de moins en moins.

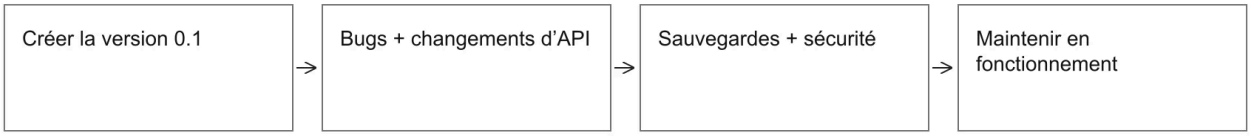
Commencez ici

Décrivez votre produit en un paragraphe. Puis listez le travail dont un client devrait s'occuper en construisant sa propre version.

01

Les gens paient encore pour NE PAS construire eux-mêmes

Les gens paient pour ne pas avoir à gérer la maintenance.



La première version n'est que le début. Le client prend aussi en charge tout ce qu'il faut pour la maintenir en fonctionnement.

Techniquement, je peux construire mon propre CRM. Aussi réparer mon évier, cuisiner tous mes repas, héberger un serveur de messagerie, coudre mes vêtements. Je ne fais pas la plupart de ces choses. Pouvoir créer quelque chose ne veut pas dire que j'ai envie d'un truc de plus à entretenir, et je pense qu'on l'oublie quand on compare le prix d'un abonnement à un week-end de code.

Après le week-end, il y a l'installation, les bugs, une API qui change, l'authentification qui casse sans raison, les sauvegardes, la sécurité, ce fichier bizarre que quelqu'un importe six mois plus tard. Puis votre vrai workflow change et il faut retourner dans ce truc que vous vous souvenez à peine d'avoir construit. La version 0.1 devient facile. La garder utile pour soi pendant les deux années suivantes paraît nettement moins amusant.

Donc oui, les gens paieront encore pour éviter tout ça. Mais il faut vraiment leur faciliter la vie. Si votre produit me donne plus de maux de tête que ma version faite en vibe coding, bon, qu'est-ce que je fais ici.

02

Vendez le savoir accumulé, pas seulement le logiciel

Ce qui est difficile à copier peut être ce que vous avez appris.



L’interface visible repose sur le workflow et le savoir acquis grâce à l’utilisation réelle.

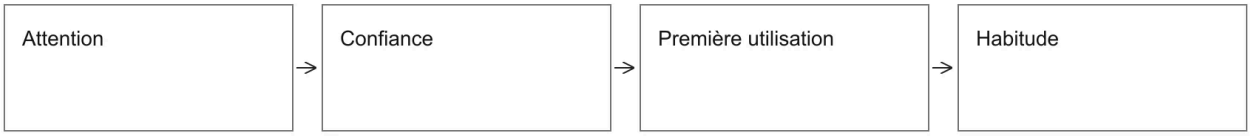
Je repense souvent à la différence entre un outil qui fait techniquement quelque chose et un autre qui gère depuis quelques années de vraies personnes faisant cette chose. Ils peuvent sembler presque identiques. On colle un truc, on reçoit une réponse, jolie interface. Sauf que l’un connaît déjà les fichiers affreux que les gens importent, les exceptions absentes de la spécification, ce que le client veut vraiment dire quand il demande ce résultat.

Il y a souvent un tas de règles moches là-dedans, des modèles que quelqu’un n’arrête pas de corriger, des prompts testés sur de vrais exemples, des détails du secteur auxquels vous ne penseriez même pas à demander. Vous pouvez copier le workflow principal le samedi. Vous ne récupérez pas avec les 3 dernières années de « ah oui, ça casse quand ils importent CE type de fichier ». Franchement, ce savoir ennuyeux commence à paraître assez précieux.

03

La distribution devient encore plus importante

Du code moins cher ne crée pas un produit de confiance.



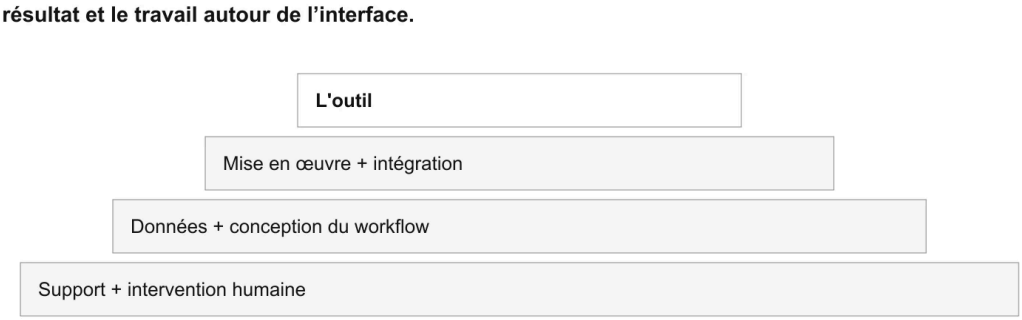
Construire un produit ne crée pas automatiquement l’attention, la confiance ou l’habitude de l’utiliser.

Et certains produits assez médiocres survivront probablement juste parce que les gens les connaissent déjà. Si j’ai besoin d’un outil aujourd’hui et qu’il en existe un avec 40 000 utilisateurs, des avis, de la documentation, des intégrations, quelqu’un à qui écrire quand ça casse, je peux simplement l’acheter. J’ai d’autres choses à faire. Le fait que je pourrais construire une alternative n’a même pas besoin d’entrer dans ma décision.

Faire produire le code coûte moins cher maintenant, faire en sorte qu’on vous remarque et qu’on vous fasse confiance paraît toujours assez difficile. Puis il y a l’habitude d’ouvrir le même outil chaque matin, d’avoir ses affaires dedans, de savoir où tout se trouve. Un produit très simple avec tout ça peut valoir bien plus qu’un produit impressionnant dont personne ne connaît l’existence. Un peu déprimant pour les ingénieurs peut-être, mais voilà.

04

Vendez ce qui entoure l’outil



L’outil est la partie visible. Le travail qui l’entoure peut représenter une grande part de la valeur que paient les clients.

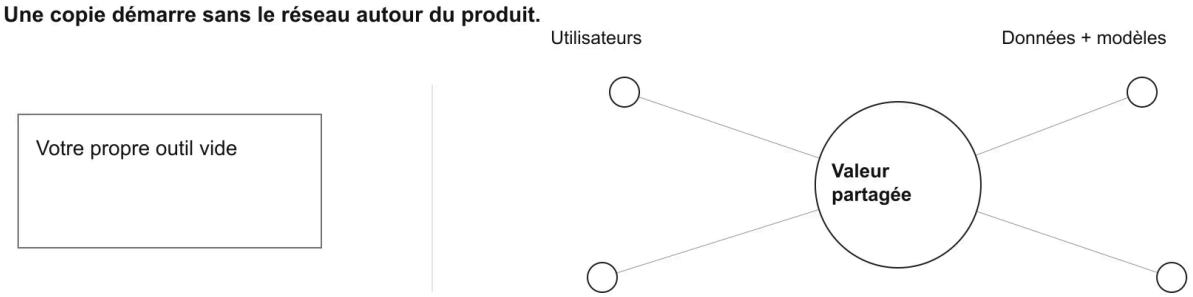
Pour certains produits, je pense que l’argent va se déplacer vers le travail autour du logiciel. Un client ne veut pas forcément un « outil d’IA qui résume les retours ». Il veut que les retours de cinq endroits soient réunis, nettoyés, classés selon les catégories de son équipe, et il veut quelqu’un qui réponde quand les résumés commencent à produire n’importe quoi.

Construire le système de résumé est assez faisable. Faire fonctionner tout ce bazar dans une vraie entreprise, puis le garder utile, c’est une autre quantité de travail. Vous pouvez facturer l’installation, l’adaptation aux processus, les intégrations, le support, la revue par un expert. La conformité réglementaire ou une personne qui prend le relais quand l’automatisation se perd peuvent aussi beaucoup compter, selon vos clients.

L’interface est juste la partie qu’ils voient. Je ne supposerais pas que c’est ce qu’ils paient réellement. Ces workflows « simples » peuvent vite devenir un tas d’exceptions, et je pense qu’il reste pas mal d’argent à gagner en gérant les morceaux pénibles pour quelqu’un d’autre.

05

Les produits qui s’améliorent parce que d’autres les utilisent



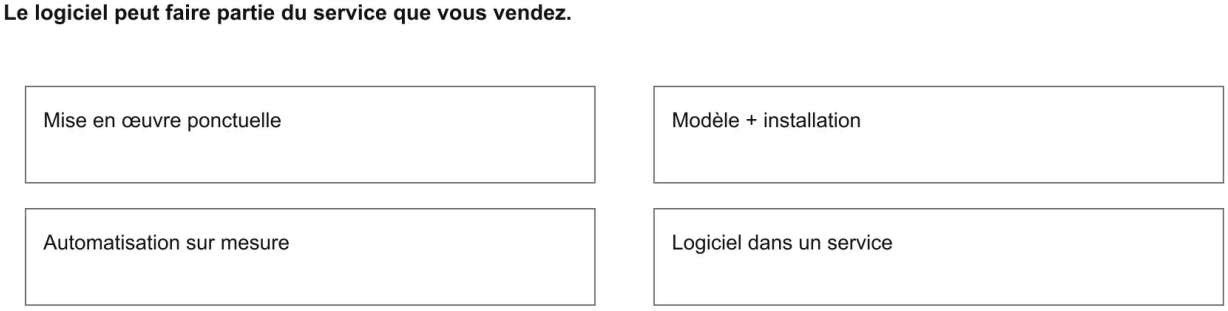
Une copie isolée peut reproduire le code. Elle ne reproduit pas instantanément un réseau utile de personnes et de contributions.

Bien sûr, construire sa propre version devient moins tentant si ce qui est utile vient de la présence d’autres personnes. Des modèles créés par les utilisateurs, les données qu’ils ont apportées, des intégrations, une réputation, des gens avec qui collaborer. Vous pouvez aussi construire votre propre marketplace vide, félicitations, maintenant il faut trouver tous ceux qui étaient censés être dedans.

Ça dépend du produit, tout n’a pas un effet de réseau simplement parce qu’il y a des utilisateurs. Mais quand ces éléments partagés comptent vraiment, il manque beaucoup de choses à une copie personnelle. Même chose pour certains outils B2B déjà enfouis dans le fonctionnement d’une entreprise. Recréer les écrans peut être la partie facile de leur remplacement.

06

Parfois, la réponse est littéralement : ne vendez pas de logiciel



L’abonnement est une option. Vous pouvez aussi vendre la mise en œuvre, l’installation, l’automatisation et les services.

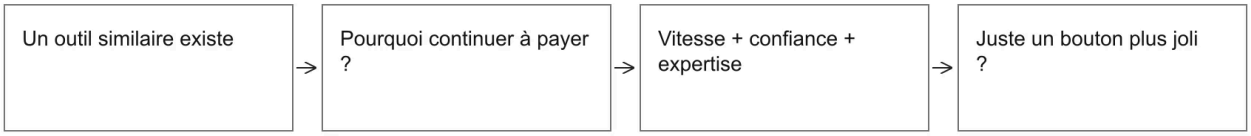
Je suis aussi moins convaincu que chaque chose utile doive devenir un SaaS par abonnement. Si vous faites des automatisations pour un type d’entreprise précis, on peut vous payer pour les construire et les installer, puis vous facturez les modifications quand elles deviennent nécessaires. Un modèle avec un peu d’aide pour le faire fonctionner peut suffire. Ou le logiciel est simplement quelque chose que vous utilisez pendant une mission de conseil, le client paie pour que le travail soit fait.

Je pense qu’on s’est un peu habitués à transformer chaque idée en « comment avoir des milliers de personnes qui paient 19 dollars par mois ». Mais si construire coûte bien moins cher, un truc très précis pour 20 clients peut aussi commencer à avoir du sens. Il faut toujours s’occuper d’eux et faire le travail évidemment, ça ne devient pas magiquement un revenu passif. J’ai juste l’impression qu’il y a davantage de place pour de petites entreprises qui n’auraient pas valu la peine d’être créées avant.

07

Et évidemment, certains outils sont tout simplement foutus

Un bouton plus joli est une faible raison de rester.



Testez la raison de rester quand le client comprend qu'il peut construire un outil similaire.

Il y a aussi des outils pour lesquels « pourquoi ne pas le construire moi-même ? » est une question assez juste. Des wrappers génériques, des tableaux de bord basiques, de petits utilitaires qui font une transformation, des générateurs de contenu très simples. Tout le monde reçoit plus ou moins le même workflow et il ne se passe pas grand-chose derrière. Pas de données particulières, pas d'années à apprendre les difficultés, pas de raison précise de faire davantage confiance à celui-ci qu'à un autre.

Les gens en achèteront encore certains parce que c'est pratique. Je ne dis pas que tous les outils simples sont morts. Je n'en construirais juste pas un en supposant que les clients ne pourront pas le faire eux-mêmes pendant les 5 prochaines années. Cette hypothèse paraît déjà assez fragile.

La question à garder en tête

Pourquoi quelqu’un continuerait-il à utiliser le vôtre ?

Donc quand je regarde une idée maintenant, « est-ce que je peux la construire ? » devient la partie moins intéressante. Généralement, probablement, oui. Ce qui m’intéresse davantage, c’est ce qui se passe quand un client comprend qu’il peut faire quelque chose de similaire lui-même. Il part, ou il regarde tout le reste dont vous vous occupez et décide qu’il préfère continuer à payer ?

Ça peut être les données, les gens, les intégrations, le fait que vous comprenez son problème. Franchement, ça peut simplement être que le vôtre fonctionne et que sa version à lui n’arrête pas de casser. C’est une raison tout à fait valable. Si tout ce que j’ai, c’est que mon bouton est plus joli, je passerais probablement encore un peu de temps sur l’idée avant de la construire.

Écrivez votre réponse

Les clients continuent de payer parce que...

Lisez l’article complet ↗