Stanford eCorner

Building the Minimum Viable Product

Eric Ries, *Author*

September 30, 2009

Video URL: http://ecorner.stanford.edu/videos/2295/Building-the-Minimum-Viable-Product

In the successful build-out of a lean startup, entrepreneur and author Eric Ries suggests adapting the rules of creating the minimum viable product. Rather than getting the product "right"; or employing the "release early, release often" philosophy of soliciting customer feedback for product development, the road to minimum viable product fuses the two ideas. The challenge for the startup is to figure out the smallest amount of product features and capabilities necessary for release, and then to slowly add more functionality as needed. For most entrepreneurs, says Ries, that means that any product will be about one-eigth as robust as they would like it to be. Ship it skinny, and add more features later - if your early adopter customers even notice what they're doing without.

Transcript

Let me say a word about minimum viable product. I know people will have heard of this phrase at least a little bit. The idea here is we want to kind of -- most startups are torn between these two different approaches to building product. One, which I call maximizing chance of success, says "Look, we only got one chance at this so let's get it right." Right? That's what I talked about in startup number one. We're going to ship it when it's right and that actually is perfectly rational. If you only have one shot, you want to take the best shot you can and build the most perfect product you can. The issue is, of course, you know, you can spend, I don't know, say five years of stealth R&D building a product you think customers want and then discover to your chagrin that they don't. So the other possible extreme approach is to say, "Well, let's just do 'release early, release often'." People have heard that phrase. And this is look, we'll just throw whatever crap we have out there and then we'll hear what customers say and we'll do whatever they say. But the issue there is if you show a product to three customers, you get 30 opinions, and now what do you do? So minimum viable product is kind of a synthesis of those two possible extremes.

We want to figure out what's the minimum set of features necessary to engage with those early evangelists to start the learning feedback loop going. And so, sometimes, I get the best question, "Well, how do you know if you've found the minimum minimum viable product?" And from a theoretical point of view, this is quite challenging. You could make a really interesting argument that any given feature is absolutely 100% necessary to learn. But the good news is, there's no reason to deal with this theoretical issue because if you're like me and like every entrepreneur I know, what you think the minimum viable product is, is way too big, probably two orders of magnitude too many features. I'm not exaggerating. So the easy formula for finding out what the minimum viable product is, is take what you think it is right now and cut it in half and do that two more times and ship it back. And I know, I know, customers are going to absolutely hate that thing. It's only one-eighth as big as you thought it should be, right? And that's fine. If you ship that and customers say, "You moron, how could you have shipped without having features X, Y and Z?" - the things that were all going to be on your road map anyway. You can say, "Good idea, good point," and then go build features X, Y and Z.

But you may be surprised, well, of course, not you, but maybe you know a friend who would be surprised to ship a product, as I did and nobody cares. They don't say, "You idiot, it should have features X, Y and Z." The worst fate of any shipping of any product is that nobody cares. You don't get any feedback at all. Right? That's what most features or most products do. They're just dead weight. So what we want to do is try to eliminate those and ship without them. Of course, that's because visionary customers can fill in the gaps. Right? Early adopters can be very forgiving of missing features. They see the vision and you can be in dialogue with them going through that learning feedback loop. Here are the kind of the reasons why people don't do the minimum viable product; I'll just try to address them really quickly: First, the fear of the false negatives.

So I ship my minimum viable product. If it had just that one extra feature, customers would have loved it; but because it didn't, of course, they hated it. So, duh, why would I bother shipping something I know customers will hate? And there's nothing wrong with that reasoning. All I want you to do is ship anyway so maybe customers will love it even though it doesn't have the feature X, and then you can kind of go on and be very successful. And if you're wrong and I really do need feature X, then you can just build feature X. If you want to do minimum viable product, you have to be prepared to iterate. And so you have to have the courage to say, "Yeah, we'll ship something, get negative feedback and respond."