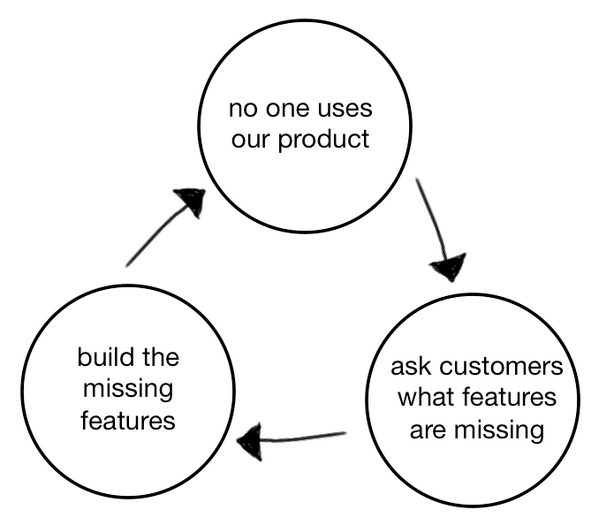
Product Death Cycle

Why users don’t use it?

Internet Research

07.21.15



The hardest part of any new product launch is the beginning, when it’s not quite working, and you’re iterating and molding the experience to fix it. It may be the hardest phase, but it’s also the most fun.

**The Product Death Cycle**

All of this was on my mind when I saw a great tweet from about a year ago, on the Product Death Cycle, when things go wrong. David Bland, a management consultant based in San Francisco.

A year ago when I saw this, I retweeted this diagram right away, and a year later, it’s hit 1,400+ RTs overall. This diagram has resonated with a ton of people because sadly, we’ve seen this Product Death Cycle happen many times. We’ve maybe even fallen into it ourselves – it’s all too easy. I’ve written about this phase before, in [After the Techcrunch bump: Life in the Trough of Sorrow](http://andrewchen.co/after-the-techcrunch-bump-life-in-the-trough-of-sorrow/" \t "_blank).  As well as some [thoughts](http://andrewchen.co/2011-blogging-roadmap-zero-to-productmarket-fit/" \t "_blank) and [strategies](http://andrewchen.co/ttpmf-time-to-product-market-fit/" \t "_blank) related to getting to product/market fit sooner rather than later.

Let’s talk about each step of this cycle, why it happens, and present a list of questions/provocations that might allow us to escape.

#### 1) No one uses our product

The natural state of any new product is that no one’s using it.  So that’s not a problem in itself. However, the way you react to this problem is what causes the Product Death Cycle.  
“Only way for a startup to be successful is to build what people want or need” – Paul Graham”

#### 2) Ask customers what features are missing

One of the big early mistakes is to be completely user-led rather than having a product vision. This manifests itself in asking users “What features are missing?”

Here are the problems with this approach:

* Users who love your product now may not represent the much larger market of non-users who’ve never experienced it. So the feedback you get might be skewed towards a niche group, and the features they suggest may not be mainstream
* User research is great for coming up with design problems, but you can’t expect users to come up with their own design solutions. That’s your job! They may be stuck in a certain paradigm and won’t have the tools/skills to come up with their own solutions. Faster horses and all that
* “What features are missing?” assumes that just adding more features will somehow fix the problem. But there are many, many other reasons why your product may not be working- maybe the pricing is wrong. Or it’s not being marketed well, the activation is broken, or the positioning sucks, etc.

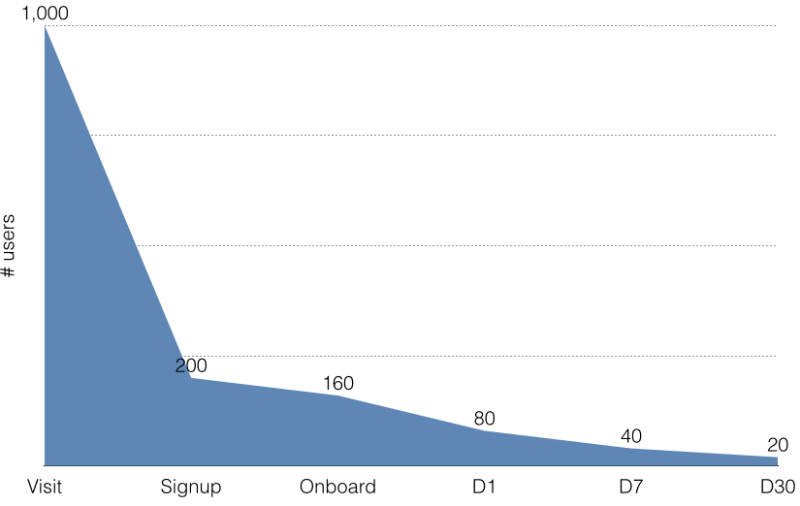
Ultimately, the solution to this is to ask- why are users not using the product? Where’s the bottleneck? In a world where [80% of daily active users are lost within 30 days](http://andrewchen.co/new-data-shows-why-losing-80-of-your-mobile-users-is-normal-and-that-the-best-apps-do-much-better/" \t "_blank), there’s a lot of reasons why users are bouncing before they even get into the “deep engagement features” you’ve built out. Asking engaged users what features they want won’t help much- instead you’ll likely get a laundry list of disorganized features that will push you towards your competitors.

One book recommendation on this topic: Harvard Business School professor Youngme Moon’s [book on competitive differentiation](http://www.amazon.com/Different-Escaping-Competitive-Youngme-Moon/dp/030746086X" \t "_blank) precisely describes the process in which customer research quickly leads to muddled differentiation, it’s worth reading.

#### 3) Build the missing features

The second jump in the Product Death Cycle is to take features that customers have suggested, and just build the missing features. However, this quickly falls into the [Next Feature Fallacy](http://andrewchen.co/the-next-feature-fallacy-the-fallacy-that-the-next-new-feature-will-suddenly-make-people-use-your-product/" \t "_blank), which is the mistaken belief that just adding one more new feature will suddenly make people want to use your whole product.

As I discussed in that essay, every product has an amazing dropoff of usage from when people first encounter it:



I’ve also published some real-life data at [Losing 80% of mobile users is normal](http://andrewchen.co/new-data-shows-why-losing-80-of-your-mobile-users-is-normal-and-that-the-best-apps-do-much-better/" \t "_blank). The point is, most interaction with a product happens in the first few visits. That’s where you can ask the user to setup for long-term retention and to present the user with a[magic moment](http://genius.com/Alex-schultz-lecture-6-growth-annotated/" \t "_blank). Building a bunch of “missing” features is unlikely to target the leakiest part of the user experience, which is in the onboarding. If the new features are meant to target the core experience, it’s important that they really improve the majority workflows within the UI, otherwise people won’t use them enough to change their engagement levels.

To break out of this part of the Product Death Cycle, ask yourself- is this enough of a change to influence the experience? Is it far enough “up the funnel” to impact the leakiest parts of the product experience? Is this just another cool feature that only a small % of users will experience?

### Breaking out

The Product Death Cycle is tricky because it’s driven by the right intentions: Listen to customers and build what they want. People in the Product Death Cycle naturally believe that they are doing the right things, but good intentions don’t translate to good traction. Instead, [ask “why?” over and over](https://en.wikipedia.org/wiki/5_Whys" \t "_blank), to understand the root cause for lack of growth.

The response to these root causes should consist of a large toolkit of responses- maybe marketing: Pricing, positioning, distribution, PR, content marketing, etc. Maybe it has to do with the strategy: Going high-end instead of low-end. Building a utilitarian product instead of a network-based one, or vice versa. The point is, the solution should be tailored to the root cause, rather than to be explicitly driven by the desire of every product team to build more product.

**Why users don't use the product?** It lies in the issue of the pain point. Is the app truly resolving their issue or just a bonus? The product death cycle happens when people just give superficial comments to include features into the app but they themselves won't be using it.

Another bad issue is when founders try to include as many features as they can to get more traction, but it causes a death spiral to the product which have too many options that no one can understand using it. UI/UX experts always tell me to KISS (keep it simple stupid) and ensure it is slick and easy to use.

Before even starting a revision in the product, understand first who are the active users and read into their behavioural pattern. If you read it right, you will get the 80/20 rule, where 80% of activity comes from 20% of the users. Ensure that the user base who gives the feedback are similar to that of the 20% and push features to serve this market. You may see lower growths of users but find engagement levels growing higher.

"One of the big early mistakes is to be completely user-led rather than having a product vision." - I've done this mistake over and over for my first start-up. Never again.