

URL: <https://stvp.stanford.edu/clips/solving-a-world-of-product-problems>

Spotify Co-Founder Daniel Ek explains his approach to building products to solve problems, with KPCB venture capitalist Chi-Hua Chien. Ek credits his engineering background and the ability to test product iterations as key components in his approach. He also discusses the importance of streamlined user interfaces, the opportunity to leverage global platforms to reach audiences at scale, and being selective in addressing the right customer use cases.



Transcript

I think one of the really distinctive elements of this new generation of entrepreneurs compared to web 1.0 is the web 1.0 guys were primarily technologist.. They solved hard technology problems.. In this kind of social mobile world, the great entrepreneurs are product people, they focus on product because a lot of the infrastructure on technology has been solved.. So, when you come and think about solving a product problem, what are the principles that you use to - that you apply to build a product like Spotify, like Spotify Mobile? And then we will talk a little bit about platform.. Sure.. So, first to begin, like I actually started out as an engineer.. So and I think this is really, really important because as an engineer you kind of get taught the ways, you get taught the basic technologies and what's actually possible and how to construct solutions around that.. And I think - I like thinking solution-oriented, so I think that's helped me.. I would say today that I'm actually a pretty crappy programmer, so don't try me out on that one, but my way of looking at an interface is still by asking myself the question why is it there? What is it - what's the purpose of that interface? And I think a lot about what's the shortest way between point - from point A to point B? And how can you assume as much as possible for the user in that process for making that journey shorter? And that means that you quite often have to go through iteration of it, iteration of designs until you got that one.. And what's also so fantastic with today's platforms is we got this really rigid testing infrastructure, where we can actually test things..

So, at Spotify for instance, we, quite often if we are unsure ourselves about certain decisions we make, we quite often make three or four of them and test them at the same time and see how users are reacting to it.. And a few years back that infrastructure was really expensive to build, but now it kind of exists on can and a lot of people can try it.. So, I would just say, like, think about what it is you want to solve and be really crisp about that, especially if you're a team of people.. Like, what is this page designed to solve or what is this product - what's the ultimate solution here, and then think about what is the shortest way from point A to point B.. And I mean, there is many ways to think about that, like in a product like Spotify, the probably optimal way is if we knew everything about you, so that we can instead suggest music so that you can just get it and just press play and it would work and you'd just have perfect music.. It turns out that that problem is incredibly hard to solve from an engineering standpoint.. So we might not go there.. So where do we take it from there? How do we ratchet that down? Well, it turns out that it's probably good if you give us a song, so that we can then in turn recommend you more songs.. Or in your case if we can get you to search the song and then show you related songs and related artists, so that you can keep navigating and kind of go in deeper and deeper.. So I think a lot about what is the perfect and ideal solution to a problem and then you kind of ratchet it down from there..

So in the case of platform, which you guys just announced and platform is - well I'll let you talk about platform, but how would you talk - how would you think about that in terms of the problem that's being solved, and then what features you guys have put in place and what features you want developers to be building? Sure, so first I think we live in a world where we get access to two super platforms, Facebook and the iPhone and the Android ecosystem with smartphones.. And I think that's really interesting because if you think back five years ago, none of those really existed.. So five years ago to this date, Facebook kind of announced their platform.. Five years ago, I think it got on sale, there was no iPhone, for instance, that was June.. So if you think about that, in a very short period of time the world has changed, and you now have these amazing channels where you can get products out to people, hundreds of millions of people that are actively looking for consumption experiences, so that didn't exist.. So if we now take the Spotify platform, because what Spotify really has built is, we built this music platform.. And what we realized is music is something that's really, really broad, and means something, it's really

unique to you as an individual, and there's a lot of different use cases.. So the use cases could be you're going to a party, you want to figure out with your friends, what kind of songs you want to listen to, other use cases is like a lean-back radio type of experience, a third use case is lyrics which is really popular, especially in Asian countries, where you want to sing karaoke next to the songs out of tune, like me, or there's ways where just curation of content.. Rolling Stone does it for a certain type of audience, Pitchfork for a more kind of indy audience, so what we realized is, why don't we just, because there's so many problems with licensing and getting all these rights, why don't we instead for developers and entrepreneurs take away that hurdle, and just expose the music, and let people hack their own solutions on what they want to do with Spotify and basically crowd source the innovation.. And we thought that was like a really, really important step, and so far we've seen everything from games, music quiz type of games, to last week someone actually created a dating App where it compares people's music taste and thereby you're supposedly going to match better, I don't know whether that will work, by the way, but try it out..

So there's really all these types of use cases, which isn't like in Spotify's core interest.. We want to be the music object, and we want to make that available across the entire web and across all these different platforms, but we can't solve every single use case and that was the sort of core notion with the platform...