

Lecture 7 - How to Build Products Users Love (Kevin Hale)

47 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=sz_LgBAGYy0&list=PL5Q_LEF6ZVKATY_CT1K7QFNF2TIDHCE-1&index=7](https://www.youtube.com/watch?v=sz_LgBAGYy0&list=PL5Q_LEF6ZVKATY_CT1K7QFNF2TIDHCE-1&index=7)
https://www.youtube.com/watch?v=sz_LgBAGYy0&list=PL5Q_Lef6zVkaTY_cT1k7qFNF2TidHCe-1&index=7

SUMMARY

The talk focuses on creating products that users love by fostering deep emotional connections and prioritizing user experience. The speaker emphasizes the importance of understanding user relationships, both in acquiring new users and retaining existing ones, through effective customer support and product design.

- Growth is driven by conversion rates and churn; both need to be managed for success.*
- Building products that users love involves creating memorable first interactions and maintaining long-term relationships.*
- Companies should prioritize user experience and emotional engagement over mere functionality.*
- Customer support should be integrated into the development process to improve product quality and user satisfaction.*
- Wufoo's success was attributed to its focus on creating a delightful user experience and strong customer relationships.*
- The concept of "Support Driven Development" encourages developers to engage directly with users to understand their needs better.*
- Simple, thoughtful design elements can enhance user engagement and satisfaction.*
- Regular communication and appreciation for users, such as handwritten thank-you notes, can foster loyalty and community.*

All right. So when I talk about making products users love what I mean specifically is like how do we make things that has a passionate user base that our users are unconditionally wanting it to be successful both on the products that we build, but also the companies behind them. We're gonna go over tons of information. Try not to take too many notes, mostly just try to listen. I'll post the link to the slides on my Twitter account and on that link, there will be a way for you to annotate the slide and you can ask me questions.

And so if we don't get to them, I'll answer them after the

talk. So, you guys have been. Listening to and hearing a lot about growth over the last several weeks, and to me I feel like growth is usually fairly simple. It's the interaction between two sort of concepts or variables. Conversion rate and churn. Right? And the gap between those two things pretty much indicate how fast you're gonna grow. And most people, especially business type people, tend to look at this interaction in terms of a very calculated in a mathematical sort of way.

And today, I sort of want to talk about these things at a more human scale. All right, cuz at start up when you're interacting with your users, you have a fairly intimate interaction that you have in the early stages. And so, I think there's a different way of looking at this stuff in terms of how we build our products. And we'll look at a lot of different examples of that. And how it's executed well. My philosophy behind a lot of things that I teach startups is the best way to sort of get to a billion dollars is to focus on the values that help you get that first dollar, to acquire that first user.

If you sort of get that right everything else will sort of take care of itself. It's a sort of faith thing. So I came to be a partner at YC by way of being alumni. I went to the program in winter of 2006, so it's the second ever program and I built a product called Wufoo. Wufoo's an online forum builder. It helps you create contact forums and online surveys and simple payment forms. It's basically a database that looks like it's designed by Fisher Price.

What's. Interesting, though, is that because it was fairly easy to use, we've had customers from every industry, market, and vertical you can think of, including a, a majority of the Fortune 500 companies out there. Ran the company for five years, and then we were acquired by SurveyMonkey in 2013. And at the time, we were a very interesting acquisition. We were only a team of ten

people at the time. And, while we acquired funding out here in Silicon Valley through Y Combinator we actually ran the company from Florida.

We had no office. Everyone worked from home. And we're an interesting outlier. So, each dot here represents a start up that was, that exited through IPO or acquisition. And we're this outlier to the left. The bottom is the funding amount they took. And the vertical axis is, the valuation of the company at the time. To sum it up, the average startup rates as about \$25 million. And they return to their investors about 676%. Wufoo raised about \$118 thousand total.

And our return to our investors is about 29,000%. >> So a lot of people were very interested. And are sort of like. What makes. Wufoo a little bit different or how do we run the company very differently, and a lot of it was focused on product. We weren't interested in building software that, I guess that just people wanted to use. Right. That reminding you that you worked in a cubicle, cuz it was database app at its sort of core.

We wanted a product that people wanted to love. That people wanted to have a relationship with, and we were actually very fanatical about how we approached. This idea. At the point where it's almost sort of sort of sciency sort of way. So what we said was like Okay. What's interesting about start-ups in terms of us wanting to create things that people love is that, love and unconditional sort of feelings are things that are difficult for us to do in sort of real life.

And at start-ups we have to do it sort of at scale. So we've decided to do is just start of by just looking at, Okay, how does real relationships work in the real world, and how can we apply them to

sort of how we run our business and sort of build our product that way. So we'll go over basically these two metaphors, find new users as if we're trying to date them and existing users as if it were a successful marriage.

So when it comes to dating, lot of the stuff that we uncovered had to do with first impressions. all, all of you often talk about your relationships in terms of the origin story. If I asked you to tell me about the first kiss, how you sort of met, how you proposed, these are the things that we say over and over and over again. They're basically the word-of-mouth stories of our relationships, and they're the same kinds of things that we do with companies.

Human beings are relationship manufacturing creatures. We cannot help, but create and anthropomorphize the things we interact with over and over again. So, whether it's the cars we drive, the clothes we wear, the tools and software's we use, we eventually prescribe characteristics to it. A personality. And we expect it to behave a certain way. And that's how we sort of interact with it. Now first impressions are important for the starting of any relationship, because it's the one we tell over and over again.

Right? And there's something special about how we regard that origin story. I'll give an example. If you're on a first date with somebody, and you're having a nice dinner, and you catch them picking their nose. You are probably not gonna have another date with them. Right, but if you're married to someone for about 20 to 30 years, and you catch on the Barca Lounger digging for gold right? You don't immediately like call your lawyer, right, and then say like we have a problem here.

I need to start drawing up papers for divorce. You shrug your shoulders and say at least he has a heart of gold. So solving about. First time interactions means that the threshold is so much lower in terms of pass fail. So, in software, and for most products in internet software that we use, like, first impressions are pretty obvious. And they're the things that you see a lot of companies sort of pay attention to in terms of what they send their marketing people to work on.

My argument for people who are very good at product is they discover so many other first moments, and they make those something memorable. Right, the first email you ever get from a piece of software. What happens when you first log in? The links, the advertising, the very first time you interact with customer support. All of those are opportunities to seduce. So how did we think about. Sort of like making first moments on there, and we actually took this concept from the Japanese.

They actually have two words for how to describe things when you're finished with them in terms of saying like is this a quality item. And the two words of quality are atarimae hinshitsu and miryokuteki hinshitsu. And the first one means, taken for granted quality, basically functionality. And the last one sort of means, enchanting quality, right? Take for example a pen, right? Something has miryokuteki, right, if the weight of the pen, the way the ink flows out of it, the way it's viewed by the people reading the handwriting from the pen is pleasurable both to the user of the pen and the people who experience the byproducts of it, right, taking it to the sort of next level.

Start with some examples. So this is Wufoo's Login link, and it has a dinosaur on it, which I think is awesome. But if you hover over it, the spec has the added

benefit of having a tool tip that doesn't explain like how to login or what it does, but basically, rawr. And. What we noticed about this, like in early usability studies as like this put a smile on people's faces. Like, hands down. Right? Universally. And I think a lot of times when we are assessing products, we never think about, like, hey what is the emotion on the person's face when they interact with this?

This is Vimeo's log in page this is actually a couple integrations ago. It's one I find to be the most beautiful, but. It lets you know that when you're starting out on this journey with Vimeo, that this is gonna be something different. They do this all over the app. If you search for the word fart, as you scroll up and down it makes fart noises as you do this. Right? There's something different, like this site interacts with you.

>> Mm-hm.
>> It's a little bit magical, it's a little bit different. And it's something that you wanna talk about. You don't have to always do it with design. This is a sign up form for Cork which used to be a social network for people who loved to drink wine. On it, it says email address, it's also your sign-in and has to be legit. First name, what your mom calls you, last name, what your Army buddies call you.

Password, something you'll remember, but hard to guess, password confirmation, think it a, type it again. Think of it as a test. It's literally a poem as you fill out the form. Right? And this is a kind of like thing where you like, oh, I like the people behind this. I, I, I'm gonna enjoy this experience. Now what does it say when you fill out form like this? Right. On Yahoo, about what the personality. Of the site it's gonna be.

And what's disappointing to me is like, Yahoo forces every product and service under them to use this exact same login form. Flickr, I had thought, had one of the best sort of call to actions. It was, get in there. Right? This is Heroku's. Sign up page. I think this is an older version. But, what's remarkable about it is that, what you start getting a feel for is like, oh, scaling up. My sort of server, and back end services is as easy as sort of dragging up and down, different sort of knobs and levers.

It's gonna be beautifully used, and it looks fairly easy to scale. Since we're in a room full of computer science people, I think you'll appreciate this. This is Chocolat. This is a code editor, and they only have one call to action. When the time limit is up, they say, everything in terms of all the pieces are exactly the same except we change the font. To comic sans. And what they're basically saying is like hey we know who our users are, who our real customers are.

They're gonna be the people who care about this. This is Hurl, this is a website for checking http requests, and sometimes the places where you get errors are opportunities for first moments, you had a four of four this is what you get it, when we need help oftentimes what we do is where we create like really beautiful mark, marketing materials. But when you actually need, like, documentation we sort of like skimp out on sort of design features.

And this is a point that, like, you see happen over and over again. A company that gets this right is MailChimp. And what they did was they redesigned all of their help guides so that they looked like magazine covers. And overnight, basically, readership goes up on all these features. And customer support for these things that sort of help people optimize emails goes down. Speaking of documentation,

stripe. What's interesting about an API company is that there is no UX.

The UX is actually just documentation, right? And there's opportunities even in documentation. Sort of the enchant and amaze. So one of the things that I love about them is their examples are wonderful. But if you're actually, like, sort of logging into the app, one of the things that is a super pain for most people, when you're dealing with most people's APIs, is, like, grabbing your API credentials and keys. And what Stripe does is it says oh, if you're logged into the app.

We automatically put your API credentials into the examples, so you only have to copy paste once when trying to learn their API. When Wufoo wanted to launch the third version of our API we realized like, Okay, that finally this is good enough that we want people to sort of build on top of it. We were trying to figure out, like, how do we launch this out to the world, that sort of has our personality behind it.

Because a lot of people, they usually do things like a programming API contest, and they give out iPads and iPhones. And it makes you look like everyone else. And so, in our company one weird value to have it's a quirk of us, is that the co-founders are big medieval nuts. And we take everyone out to Medieval Times every single year, on the anniversary of the founding of the company. And so, we said we have to do something in that flavor.

And so we contacted the guys at armor.com, we said can you forward us a custom battle-axe. And what we said was if you win our programming contest you would win one. The result is like people wanted to talk about this. It's something that people

wanted to work on cuz they wanted to be able to say it like I'm programming. For a weapon. And what's cool is we had over 25 different applications created for us of quality and quantity that we could not have paid for on the budget and the sort of time that we had for this.

We got things like an iPhone app and Android and WordPress plugins, right? And all because what we did was we changed how people wanna talk about the origin story of how they're interacting with one of our services. And go like all day long of going over these examples. But I'm gonna short cut this by saying, you should just subscribe a little bit of details. It's just basically tons of screen shots of software that's just doing it right.

That shows that they're being conscientious of the user and the customers. When it comes to long term relationships or marriages, the only research that we ended up having to read was the stuff that was done by John Gottman. He's been featured in This American Life, in Malcolm Gladwell's books. He's a marriage researcher up in Seattle. And he has an interesting parlor trick that he can do. He can watch a video tape of a couple fighting about some issue for 15 minutes.

And predict with an 85% accuracy rate whether that couple will be together or not, or divorced in four years. If he increases that video up to an hour. And asks them to also talk about their hopes and dreams. That prediction rating goes up to 94%. They showed these same video tapes to marriage counselors, successfully married couples, sociologists, psychiatrists, priests, et cetera and they can't predict with random chance whether people are gonna be together or not. So John Gauntman understands something fundamental about.

How relationships work in the long term. And that basically how we fight, even in a short term period can indicate sort of the whole system and what it's gonna look like. And one of the surprising things he discovered, it's not that successfully married people don't fight at all. It turns out everybody fights. And we all fight about the exact same things money, kids, sex, time and others. And others are things like jealousy and in-laws. To bring this around, right, you can actually attribute every single one of these to problems that you see in customer support when you're building out your products.

Right, so. This costs too much. I'm having problems with a credit card. If you're building a service that helps people deal with their clients they're very sensitive about anything happening with that. Performance. How long you're up and how fast. Others are I said jealousy and in-laws. Right, so that's competition and partnerships. So anything weird happening there, people are gonna write to you about. And the reason I like to think about this in terms of customer support is that in every ones that are processing of like a conversion funnel, customer support is the thing that happens in between every one of these steps.

It's the reason why people don't make it further down there. It's the thing that prevents conversion from happening. Now, as we were thinking through all these ideas and as we were building up the company, we realized that there's a big problem about how everyone sort of starts their company or build up their sort of engineering teams. And that is that there's a broken feedback loop there. People are divorced from the consequences of their actions. And this is the result of actually the natural evolution of how most companies get founded, especially by technical co-founders, right.

Before launch it is a time of bliss, nirvana, and opportunity, right? Nothing that you do is wrong, right? By your hand, which you feel is like god, everything that you write, every line of code feels perfect, right, and is ingenious to you. The thing that happens is after launch reality sort of sets in and then all these other tasks sort of come into place that we have to deal with. Now what technical co-founders wanna do is get back to that initial state and so what we often do, and what we often see is that companies start siloing off all these other things that actually is what makes a start up or a company sort of real, right?

And have other people do them. To, in our minds, these other tasks are inferior. Right? And we have other people in the company do them. And so for us, what we're trying to figure out is how do we change software development so that we inject some values that we don't talk about enough? Responsibility, accountability, humility, modesty. Right, and we called this, like a lot of other people, we had an acronym, Support Driven Development, and it's very similar to TDD or other agile practice.

It's a way of creating high quality software, but it's super simple. You don't need like, a SCRUM, you don't need a bunch of post-it notes, all you have to do is make everyone do customer support. And what you end up having is you fix the feedback loop, right? The people who build the software are the ones supporting it and you get all these sort of nice benefits as a result. So one of them is, support responsible developers and designers and people who build the stuff, they give the very best support.

Now we're not the first person to think of this. Paul English was a big proponent of this at Kayak, and what he did was install

a red customer support phone line in the middle of the engineering floor. And they were just regular customer support calls. And people would ask him often times, why would you pay engineers \$120,000 or more to do something that you can pay other people a fraction of to handle in like a call center?

And he says, well, after the second or third time that phone rings and the engineer gets the same problem they stop what they're doing, they fix the bug, and we stop getting phone calls about it. It, it's a way of having Q-A in a sort of nice elegant solution. Now, John Gotman talks about the reason that we often break up with one another as due to four major causes and their warning signs. He calls them the four horsemen, right?

Criticism, contempt, defensiveness, and stonewalling. Now, criticism is basically people starting to focus, not just on the specific issue at hand, but on the overarching issues. Like, you never, right, listen to your users or you never think about us all the time. Right? Contempt is when someone is purposely trying to insult somebody. Defensiveness is not trying to take accountability trying to make excuses for the actions.

And stonewalling is basically shutting down. Stonewalling, to John Gotman, is, is one of the worst things that we can do in a relationship. Hold up. And oftentimes, you know, we don't worry much about this in customer se, criticism and contempt. Right? Defensiveness, you see this all the time, all the times in companies, especially as they get older. But stonewalling, this is something I see happen with start-ups all the time. You get a bunch of customer support sort of coming in and you just think I don't need to answer it.

I don't need to respond. Right?

And that act of just not even getting back to them is one of the worst things you can do. And it's probably some of the biggest causes of churn in the early stages of start-ups. This is how support worked out with Wufoo. When we were acquired we had about 500,000 users on the system, 5 million people used Wufoo forms and reports, whether they knew it or not, and all those people got support from the same ten people, and usually it was only one person dedicated support a day, or any shift.

Results in about 400 issues a week. That's about 800 emails. But a response time from 9 a.m. to 9 p.m. was between seven to 12 minutes. Right? And from 9 p.m. to midnight it was an hour. And on the weekend it would be no longer than 24 hours. And we carried this up all the way up to the scale. What a lot of people forget about, and often talk about with Airbnb, is how, like oh, they did this interesting thing where they had, went up to New York and offered, like, professional photographers, and the founders would go out there and actually take pictures of the people's apartments to help them sell more.

Focusing on the stories around conversion. What most people don't realize is a lot of times when I saw Joe in the early days of Airbnb, he had a phone, sort of head set stuck to his head all the time, because he was doing phone support nonstop. Churn is a story we don't like to talk about often, all the time. Airbnb's sort of growth really started picking up once they figured out how to match capacity to the demand, or the phone calls that they were getting into their support system.

At Wufoo we actually constantly did experiments around support, because we're so obsessed with it. One experiment we did was,

we heard Kathy Sierra do a talk about there's a disconnect between the motions that we have when we need help, and sort of. The content and the reactions we get from people when we get help from them, especially online. Because they just don't see all those nonverbal cues. So she said unless there's face recognition on the web, we're just always going to be disconnected from our users.

Our feeling was, like, well we're not face recognition experts but I think there's another way of getting empathy. So, as form builders, we added a drop-down and what we said was like, hey, what's your emotional state? And our hypothesis was that no one was gonna fill this out. We basically thought, oh okay, you know what the, this is gonna be pretty a lame experiment, but we'll see how it sort of goes. And it turned out the Emotional State drop-down field was filled out 75.8% of the time.

The browser type drop-down filled just in comparison was filled out 78.1% of the time. All right? So people were basically telling us, for my technical support issue how I feel about this problem is just as important as, like, all the technical details you need to sorta figure out how to debug it. Now we didn't prioritize things or triage things by emotion, right, and for the most part, people didn't game the system. One of the interesting byproducts of it was that we noticed that people started being nicer to us in the customer support.

It was something sort of subconscious. We just were thinking like, wow, users are so much better now. What's going on? And we went back and looked at the data and we did some text analysis and we realized is that, oh, when it comes to only communicating with people over written words, like email, there's only three ways that you

show strong emotions, right? Exclamation marks curse words, and all caps. And sure enough on all three of those metrics, they've gone down.

In sort of the way people were talking to us in the customer support. Once people had a simple outlet for their emotions, right, people would be a lot more rational, and it made our jobs a lot more pleasant as a result. The other byproduct that is awesome is that you actually build better software when you do this. Far better software. This is actually backed up by tons of research. Jared Spool, a user interface engineer, which is sort of the big players in this space says like, there's a direct correlation to how much time we spend directly exposed to users and how good our designs sort of get.

He says it has to come in this specific way. It has to be direct exposure, right? It can't be something where someone generates a report or through a graph, you have to be interacting with them somewhat real time. It has to be a minimum of every six weeks. And it has to be for at least two hours. Otherwise your software will get worse over time. Our developers, our people who are on, on Wufoo were getting exposed to our users four to eight hours every single week.

And what it does is it changes the way you sort of build software. Jared Spool has another way of talking about how we build products. Right? Let's imagine that this represents all the knowledge needed to sort of use your app on a spectrum. Right? This is like no knowledge. Right? And this is all the knowledge needed. Right? And these two lines are pretty much your interactions with users what you're trying to get them to. This is currently where their knowledge point is and this is the target knowledge point that you're trying to get them to, to understand

that to use your app.

The gap between those is called the knowledge gap, Jared Spool called it, Spool calls. And what's interesting about this is there's only two ways, right, to sort of fix this. That gap represents how intuitive your app is, right? You either get the user to increase their knowledge or you decrease the amount of knowledge that's needed to use your application. And oftentimes, as engineers and people who build and work on products, we think let's add new features.

And new features only means let's increase the knowledge gap. So for us we actually focused a lot on the other sort of direction. And so what that meant we spent a lot of time, 30% of our engineering time was spent on internal tools to help with our customer support stuff. But oftentimes it was spent helping people help themselves. Things like frequently asked questions, tool tips. Things like, if you just click the help link, right, instead of taking you to the generic help sort of documentation page, you go to the specific page where you're looking at is going to be most, sort of, appropriate for what you're working on.

We redesigned our documentation over and over again, AB tested it constantly. One iteration of our documentation reduced customer support by 30% overnight. It's one of those things where, like, overnight, all the people that work on the product immediately had 30% less work to do. Now what happens if you have everyone work on customer support constantly, and thinking about it in terms of a remarkable way? Well, I talked a lot about, in the very beginning growth is a function of conversion and churn.

This is Wufoo's growth curve for the first five years. Right? What's interesting is that we paid no money on advertising, on marketing. All of it was done by word of mouth growth. Right? And the interaction between, like, new users and downgrades are this. It's so slight what it takes, that gap, making that sort of work. And what a lot of people keep forgetting is that there's almost no difference between an increase in conversion rate, 1% increase and a 1% decrease churn.

They do exactly the same thing to your growth. However, the ladder is actually much easier to do. It's much cheaper to do, in your apps, and a lot of the times we neglect this, neglect this to way far along. Right? And we usually have our B team works on these sort of projects and services. This is actually not the graph that we track most of the time at Wufoo, it's not even the one I'm proud of.

This is the one I'm proud of. Cuz even though we have this sort of nice, awesome curve of growth, this is what, how loud is this scale? Keep the company small, have an awesome culture. And that required doing a lot of these things to help people sort of do what they need. So John Gotman noticed there was a different type of behavior for relationships and why people divorced. Basically there would be some subset of people who would stay together 10, 15 years and then all of a sudden they divorce and there was none of the other indicators which sort of show that this is what was gonna happen.

And I was looking through the data and I realized oh, there's no passion, there's no fire between these people, right? When it comes to

relationships they kinda follow the second law of thermodynamics, right? In a closed energy system things tend to run down so you have to constantly be putting energy and effort back into it. Now the way a lot of people sort of think about showing people that I care about you in products and in companies is to do things like let's have a blog, right?

Lets' have a newsletter? The thing is, we look at these rates and basically it was such a small percentage of our active users that it was like, most of our users have no idea all the awesome stuff that we're doing for them. So we built a new tool. We called it the Wufoo system, and what it allowed us to do was just time stamp every new feature that we're building for users and then every time they would log in, we would look at the difference between their log in time or last log in time and the new features that were implemented.

And when you had this message show up, hey since you've been gone here's all the awesome stuff that Wufoo did for you. Hands down this was the most talked about feature I've ever had every time I went out to talk to users. Right they'd say like dude I love that since you've be, since you've gone thing even though I pay the same amount every single month you guys are doing something for me almost every week and it's totally awesome and makes me feel, I'm getting maximum value.

The other thing that we did in addition to having everyone support the people that paid their pay check is have them say thank you. And this was a large part due to us injecting sort of humility and modesty in to sort of the equation. Every single Friday we would get together and we'd just write simple hand written thank you cards to our users. And I know there's tons of people who would not be sort of excited about doing this, but it was a ritual that made sort of all the difference in terms of, like, having a team that was

very tightly neat, tightly knit, also.

And working on stuff that they really cared about. They always constantly knew what the mission was for, and why we sort of did what we did. These aren't fancy thank you cards, right? They're just simple, like handwritten stuff on an index card. We threw in a sticker, and slapped on a dinosaur on the front of it. And, what's interesting is we started this practice as a result of the early days of starting Wufoo. Chris, Ron, and I were talking, and we're trying to figure out, what are we gonna do to sort of show users that we appreciate them around Christmas.

And he, Chris came up with this idea where he said hey guys, so a couple years ago my mom, like, made me write thank you notes to all of my relatives. For my Christmas gifts. And I didn't really like to do it but the following year all my presents were super good. So, I think we should try this for our business and see how it goes. So, that first year we wrote handwritten Christmas cards to all of our users that first year.

Second year rolls around, and we have too many customers, like, and it's still just the three founders. And we were going like we're kinda screwed, I don't know what we're gonna do. And we read a book called the ultimate question, and in it he talks about hey, just focus on your most profitable users. And just tend them and take care of them, then it'll work out. So we're like, all right, that, that makes sense, that's scalable.

So that year we only write to our highest paying customers. And the January rolls around that second year and one of our long time loyal users writes us and he's basically like, hey guys I, I really loved that Christmas card you sent

me the first year and I just wanted you to know I haven't received my second card yet and I'm just looking forward to it. I know you didn't forget about me. Thanks a lot. So we're like, fuck, because the best way to sort of exceed expectations is not to send any to begin with, so we were like sort of in this conundrum.

And what we decided, after thinking about it for a while, is that we need to stop doing it, you know, just one time a year. It needs to be something that's part of the culture. Happens every, every, every sort of week, even. And even though we'll never catch up to all of our customers. Just the practice of doing it will make all the difference. I talked a lot about a bunch of like lovey-dovey stuff and sort of like touchy-feely things that I think a lot of engineers don't like to think about too often.

And so I'll end on some sort of hard business data or research. There's an article that was put out by the Harvard Business Review several years ago by Michael Treacy and Fred Wiersema and in it they talk about the discipline of market leaders. They say there's only three ways that you achieve market dominance and depending on how you want to achieve that market dominance you have to organize your company in a very specific way. Best price, best product and best overall solution.

If you want to be the best price out there you focus on logistics. A Wal-Mart, an Amazon. If you want to be the best product out there you focus on R&D. Apple's usually a quintessential example of that. Best overall solution is about being customer intimate. And this is the path that you see followed by luxury brands and hospitality industry. What I love about this path towards market dominance is that the third one is the only one that everyone can do at any stage of their company.

Requires almost no money to get started with. Usually just co. Requires a little bit of humility and some manners. And as a result you can achieve the success as any other people in sort of your market. That's all I got. Thank you very much. Yeah, let's take some questions if you guys have any. Right in the back there. >> Building products that users love? You might have multiple different types of users.

How do you build one product that all users love? Maybe there is a feature that one really likes but detracts value from one that. >> All right.

So what do you do when you have a product with lots of different type of users, right? Some users will love one thing and another will, will another. And I agree, there's a interesting fine line for that. What I always, usually tell people, is focus on the people who are the most passionate, especially in the early stages.

Right?

Whoever's, whatever niche it's gonna be, that's who I focus on completely. Things that a lot of different projects did. I think Ben Silverman of Pinterest started off with a designer bloggers, right? Curtail your thing for them and eventually you'll figure out sort of universal values that will appeal to a lot of other people. So, just start one at a time. And. The, a lot of the examples that you see up there, a lot of people make the mistake is like, oh, I'll just make my app funny.

But, humor is like really difficult to do. Right?

What you wanna shoot for is something sort of witty. And, quite honestly, you have to get functionality right. So like the Japanese quality. If you don't have a on there, right, don't try to do anything witty, right, cuz it will backfire on you. So hands down, our number one focus is make it as easy to use as possible for and anything else on top was

polish. Right here.

>> So so everybody says that to focus on your product. I'm also good at that. I love to do a project and I love to make it the best. But we are to that certain point that we are focused on our product but we don't get like constantly right? Sorry. So second thing so how much we should focus on product? But because we should do now marketing. We should get somebody customers and like and like start talking to customers but when you are too focused on your product.

Like users online have them. Right, so what exactly do you guys mean when you are saying like focus only on your product and give the best product? >> Okay. So the question sort of is how do we balance this sort of thing where we wanna be obsessed with working on product. Yet. But all the other skills, and sort of tasks that are needed by a company, like marketing and branding and all that stuff, and how, how we sort of balance that.

And the thing is, like, it starts off as you juggling, like, tons of things constantly in the air. The thing is, if you're working on products, like, you should also always have this flip side as when you're talking to users. Right?

And for us inside of Wufoo, the way we got people to talk to users is they just did customer support. And they got to see firsthand, right away. Whether that feature sucked or not, and also impacted everyone else in the company, because everyone had a customer support shift.

So you have this sort of social incentive to sort of make everything work. And so, like I said, there should be no point where you're only focused on product. You should always have time

where you work on product, and then you see sort of what users say, say to you. And you should always have this virtual, like, feedback loop on there. So be careful when you don't have that. Usually what ends up happening, if you're lucky.

In terms of marketing and sales, like, usually my feeling is like, you have to spend money on marketing and advertising, all this stuff. It's usually a tax you pay because you haven't made your product remarkable. Right? Word of mouth growth is the easiest kind of growth, and it's how a lot of the great companies sort of grow. So figure out how to wait, how to like, have a story that people want to tell. About your product.

Where they're the most interesting person at the dinner table, right? And then that person is your sales person, right? That person is your sales force for you. Right here. >> like, where do you find crystal clear customer or user need and the demand is there is the right solution. How do you communicate with engineering and designing team to make sure that because sometimes people in the team come up with ideas, and but still at the end of the day, how can you make a decision with where to go?

>> Oh, so how do you make a decision on product? And communicate that with your sort of engineering team when there's like lots of different directions to go? My feeling is that. So for us we just looked at support. It was really easy cuz you often just saw what are things that people are having the most amount of problems with? Or people asking all the time. You cannot help but get feature requests from people no matter like whatever opening or orifice you have in your product or app.

Like, people will like jam feature requests in there. So you're easily going to know sort of what they sort

of what. Your job as a product person and engineer is to not just do what they say, because that way, you'll just be a slave, is to figure out, sort of deeply, what are the reasons why underlying those things and sort of solve that deep underlying reason. The thing is that everyone wants to have a different way of.

To sort of go, then ultimately it comes down to, like, someone's gonna figure something out. But I also make the smallest version of each little idea. No longer than a week or two weeks to build it out there. And you can try them out and see sort of what works and don't work. I think it's dangerous to have multiple different product directions that requires lots of time to sort of figure out. Sam. >> Related to that can you tell the story of how the king for a day thing >> Yeah.

Okay. So so I don't like hackathons. I think they sort of suck in terms of ones done inside of companies. Because. You spend like 48 hours working on something really hard that you're sort of passionate about and 99% of them never make it to production right, and it's sort, sort of real like super sad. So for us we like flipped it on it's head and we came up with an idea that we called king for a day and it actually worked over the weekend.

But how it worked is someone randomly in the company got drawn and they got to be the king, and the king got to tell everyone else what to do on the product. So everything that was bothering them about Wufoo. About the customer support stuff, or some feature they really want to have built. They've got the engineering resources, the market resources, the advertising resources of everyone inside of the company, to make it sort of happen. And of course, we'd work with them to figure out like what can be actually done in 48 hours.

But we would do this one to
two times a year. And it was like a huge hit
and it was a boost to morale,
cuz what people most love. It is like working on things
where it's like, oh, I made a difference to the
app. Right? And so, for us, that's one
way that we would like sort of divide time for like
product direction. It's like some times the
people that work for you, they have a strong
opinion about where it, where it should go.

And it's a good way to sort
of democratize it a little bit, by rotating it
around. Yes. >> You said you guys all
work from home. Usually seems like a
nightmare. In that office, how do you
make that work? >> Okay, so we all work from
home. So, I will tell you this. We all still work within the
Tampa Bay area. We would allow anybody to work from any where but
usually. As we tried to recruit
them, they sort of meet our team, and they just
decide, okay we just want to come
and move here anyway.

Remote working, is
especially tricky, a lot of people like to
romanticize it, especially people, who are
like employees. But the thing is, An office
gives you a lot of, sort of, benefits, right, and
efficiencies that you now have to compensate for when
you remote working. But remote working also has
these other sort of efficiencies in place, for
example, I don't have to worry about
my employees losing two hours of their day to
commuting, for instance. So the biggest thing that we
had to do for remote working is to respect
people's time.

And so the way we had it set
up is we actually had a four and a half day work
week at Wufoo. Half day on Friday was for
all the meetings and stuff. We said like no business

deal, meetings, no talking with other outside parties. They all have to be done on Friday, on that half day. It couldn't be done in the middle of the week. And then also one day of everyone was already dedicated customer support.

So everyone in our company effectively only had three days each week to actually build or work on whatever they were doing. But I actually firmly believe that if you have three solid days, right, eight to ten hours, where you are only working on what you need to build, you can get a ton of shit done. And so what we said was, you have to respect everyone's time during that three day period, if they're in that three day period.

And what we came up with is a 15 minute rule and the way it worked is you can have a discussion like a chat or a phone call or whatever with someone but it could go no longer than 15 minutes so if you have some complicated issue that you couldn't figure out we'd say, at 15 minutes you are to immediately table that item. Right and have us discuss it on Friday and you'd move on to the next item on your list right, the enhanced productivity.

More often than not I, I would say 90% of the time the item never got brought up on Friday because usually what would happen is people would sleep on it and then they'd just magically say like, hey I found a solution or like hey that's not a big problem whatsoever. Because most problems inside of companies, they don't need to be solved in real time, or right away. The only things are like when the site is down, or payments aren't working.

Right?
Everything outside of that is basically kind of luxury, so focus on your primaries as

much as possible. And, as a result, our ten person team did far more than many, many other companies, as a result. But, it takes extra work to make remote working happen. We are an extremely disciplined sort of team. And I would have to say there's almost not many YC companies that actually have been able to replicate sort of what we do.

I think there's only two other companies in YC that sort of have the same sort of discipline working style. It takes more work in a very different fashion. Right? An office allows you to be a little bit lazier, right, in terms of all these things around productivity. Okay. Over here. >> Right. Just to go off that question as the leader of the team how did you manage to instill a company culture but and also the count, accountability the employees especially cuz.

Working space. >> Okay, how do we, how do we set up accountability for employees? As, as a manager. Alright, so, at Wufoo we were profitable nine months after launch, so, we had profit sharing, right? And so, it makes pretty simple and clear. It, it would be a multiple of whatever bonus pull that we sort of had, and the performance measures would be based on sort of, how they did in customer support. All right, on their duties there, and sort of what they said they were wanting to accomplish or do.

I don't like process, and I don't like lots of tools to help get people to be productive. So the only thing that we had for helping people manage, like, sort of their projects, is to-do lists. And that is, like, simple text files that we shared in our Dropbox account. Each person had their name on it, and you got to see every time somebody updated their to-do list. What we said is every single night just set everything that you did that day.

Right?

And then on Friday, we would just go over. Okay, this is what you said last week that you're gonna do. This is what you actually got done, or the sort of the problems at hand. And it's super simple, right? It creates this like nice written trail for how to sort of handle stuff, right? And I don't have to worry about managing them, right? They sorta set the tone for how they want to be sort of assessed.

And it makes it really simple. And for people who are excellent at what they do. Right? It works very, very well. And then when you actually have problems, it's very easy to fire people. I was fortunate that I never had to fire anyone at Wufoo, right? But we were able to correct a lot of people's behavior very, very quickly. Cuz we just kinda look at this, and it's like look, this is your pattern of behavior. You finish a fraction of the items on your list.

You do most of the items at the last second right before Friday. That's a problem, you've gotta manage your time better and this is evidence that you've provided to us. All we have to do is sort of describe it back to you and because everyone in the company sort of sees it, right, there's social pressure that's put into place that helps make it all sort of happen. Right here. >> How did you hire people that, you know, you felt would be able to work with in this kind of environment that's, that's >> So, how do you hire people that can work, remotely?

And then sort of work in this sort of fashion. So, pretty easily, you have them work on a side project for you. So you contract them out, and they have to work remotely. As such, usually, the projects I like to have them work on is about a

month long, right? I could do things much faster for a week. But usually get a good sense of, like, how well people sort of manage themselves, and work on things from a project like that.

So that was always the first assessment. Like we never did anything just by interviews. The other thing we had to, sort of, screen them for is their ability to do customer support. Because not any, every engineer sort of has the empathy skills to handle that stress. So sometimes I would have people write breakup letters to me, right, in an interview, just like hey, pretend you have to break up with me you have 15 minutes to write in on there and you can only write it by hand, what are you gonna say?

And so you get a good sense of sort of their writing skills, because like 90% of what you do in customer support is tell them bad news, like, oh, we don't support that feature, sorry that's not gonna work, or. It's not gonna be available. And so people have to have sort of tacked at that. >> How 'bout one more question? >> One question, right here the glasses. >> So, since Wiki has all these like tricks and experiments that have really helped the company.

Do you [INAUDIBLE] of ones that didn't work out? >> Have all these tricks and experiments to help the company, are there any ones that didn't work out? All right, I'll talk about one. So one of the things that we did early on to try to motivate ourselves was try to get, like we understood this idea of like crunch mode and that it's really bad for people. Like if you're doing the subscription business you need people to last for the long term, and video games a lot of times they like crunch people.

All, all the time. For like a specific deadline or have multiple sprints every two weeks and you have to shoot up to this deadline and it's like exhausting. And so, because, when it's happening it's like you might get an increase in productivity but the recovery period that you need for people is always greater, right, than the productivity you gain. And at a company where you need everyone doing customer support and being on their game and constantly put, pushed out features you don't have time for recovery.

So, we were thinking about, okay, we want to build like, a company vacation into how Wufoo sort of works to reward our users every single year. And we said okay, if the vacation is sort of built in there for the recovery, we could have one crunch period, right, before that vacation set up. And we'll just only do customer support that will just sort of scale with people. So. The way we did the very first crunch mode is that it was just between the three founders.

And we had each of us draw a ten-item to-do list. That would be fairly aggressive. And the first person to get through seven of their items would win. And the the last person to get through seven of their items. Would become what we called trippage. And trippage meant that you carried other person's luggage and got people drinks when you're on the company vacation. So we did that and during that period, everyone was like pretty excited about it and motivated, and only the winner got to choose the next company vacation the following year.

And then all the sudden Ryan had basically poorly estimated the items on his list. And he realized very quickly I'm going to fucking lose. And so he was just like I give up. And he just sort of stopped. So crunch mode turned out to be blah mode for him because he knew he was

gonna lose and became pretty demoralized. So as a result of doing that, we decided not to do it in that similar fashion anymore.

So, good idea that we like to talk about. But is one that we, we never did again. All right, guys. Thanks a lot. You can email me at kevin@