

Albert Chou:

This is it IT Visionaries, your number one source for actionable insights and exclusive interviews with CIOs, CTOs, and CISOs, and many more. I'm your host, Albert Chou, a former CIO, former sales VP, and now podcast host.

Happy holidays. Listen, everyone. To close out this year, I want to revisit some of our favorite episodes and some of our favorite content from 2022. Next Tuesday, I'll be here with my friends and colleagues from Up Next in Commerce and Marketing Trends. That's right. I get to together with the CEO, the boss ladies, Stephanie Postles and Jeremy, some dude, to share some of the best, most unique perspectives we've heard all year long.

Then, on Thursday, we're going to be posting maybe my personal favorite episode. It's going to be me featuring my producer Jana. She's a superstar, a best of 2022 wrap-up. We'll talk about some of our favorite guests and favorite episodes. That's right. I do have favorites. Sorry, everybody else. It's okay. But you're going to hear from and hear clips from C-Suite leaders like Stefan Pretorius, WPP, Carrie SiuButt at SimpleHealth, Inna Braverman, who I think is changing the world at Eco Wave Power. Look forward to sharing that episode.

But today, we're going to revisit one of our favorite episodes of the year. I intro him all the time, and some people I might fanboy. But I just joked with Jana about it, like, "Listen, I don't actually have pictures of this guy in my house." I don't even have the book rework, although I have read it. I listen to the podcast every now and then. But David Heinemeier Hansson, and he's the CTO and co-founder of 37signals.

And the reason why I like this episode so much, one, it was an opportunity for me to meet someone that I have admired from afar for a while. I was introduced to David in Basecamp back in 2006 or so, maybe 2007, so back in the beginning of Basecamp. But really, it was more about how they work. And so, I didn't use project management software. I didn't need it. I didn't have a use for it. But it was more like how they worked. And it just fascinated me that they have figured out a way to be very thoughtful in what they do, build a great business with what they do, be very profitable with what they do, and seemingly also be able to enjoy their life outside of work.

And it's just something that I align to. And when you listen to this episode, I want you to listen carefully to some of the numbers that he talks about. For example, they have over 120,000 customers. They also only have 80 employees, 70 at the time that we recorded this. That's an absurd amount of support for a huge customer base. So how did they do that? How did they figure out a way to build software for that many customers, yet support it with only so many people?

I mean, they have apps on Android. They have apps on iTunes. They have native Mac OS apps. Their web app is obviously browser compatible with every single browser out there. They have many of the features that someone like other companies have spun off and created huge companies behind. And a lot of people have said like, "You guys could make more money if you did certain things." But I just like how he is and his partner Jason Fried, how they've stayed steadfast to their ethos. And they've kind of challenged everyone along the way like, "I don't know why you think we need to make more things. We're doing great." You know what I mean?

What is success? And I think it's something that I personally have challenged myself with all the time. What is success? Is it just monetary success? Is it personal time? What is success? And so to me, DHH is just an extremely successful person. I think he could easily pull a few levers to make him significantly more money, even though he is already fabulously wealthy. But he hasn't done it. And I just like the way he thinks about work and life. And so, it's an interesting lesson. It's also reaffirming to a lot of different things that I personally believe, such as valuing intuition over data. And they talk about it that they don't

really have that much data measurement inside their company, which is like blasphemy for most big companies. Most big companies have data analyst positions, data analysts one, two, three.

It's like all these people are trying to figure out what's going on. They're like, "Hey, if you can't figure out what's going on, maybe you're asking the wrong questions." And I thought that was pretty interesting that they don't really care. It was awesome and refreshing to hear. It's like they do things because they can prove it succeeds and they kind of trust their intuition a lot. And he talks about that which is also reminiscent.

And again, we all have confirmation bias. One of my favorite books is Blink by Malcolm Gladwell. And that book also talks about the ability that human intuition and knowledge that you've earned over time can be more powerful than a computer. And I think when I hear stories like that, it just gives me confidence in what I'm doing and that I'm trusting my gut a little bit. Now, have I reached the numbers that David has succeeded or like no, I'm not saying I'm in anywhere near his league if you're measuring life and success by money.

But it is interesting that he has embodied what I thoroughly believe is just enjoying your life, everything about your life, what you do, your work, your life outside of work. I feel like he's doing it. And I feel like I am too. So give this episode a listen. There's a reason why it's my favorite episode of the year. We look forward to hearing from you. And stay tuned for more episodes of IT Visionaries. David, welcome to the show.

David Heinemeier-Hansson:

Thanks for having me on.

Albert Chou:

Listen, you've written a book, rework. You are very vocal about what it takes to... what new work looks like. The one thing that I always tell people that I've never experienced your software product is like I always say like, "You guys as a company have to be probably the most forward thinking and well-planned companies out there." For the audience to know, the last we checked, you have over 120,000 customers. Is that accurate?

David Heinemeier-Hansson:

Yes. That's accounts. I mean, we've obviously millions of people using the software, but not millions of people paying for it.

Albert Chou:

There you go. 120,000 plus accounts customers on LinkedIn. There's only 130 something employees

David Heinemeier-Hansson:

And that number is actually, by the way, overstated by half. I think there's a bunch of people or a bunch of other companies called Basecamp. And people can just tag themselves with 37signals.

Albert Chou:

Oh wow.

David Heinemeier-Hansson:

So right now, we're actually only half that size, about 70 people.

Albert Chou:

Wow. Okay. Mind blown. So we're only at 78 headcount. So there has to be so much thought process that goes into what you guys build because you're supporting a huge customer layer with a small group. I want to start right away. For our audience who might not be familiar, tell us how you guys think about work and how that impacts what you choose to engineer and what you choose not to build in the products.

David Heinemeier-Hansson:

Yeah. It's actually interesting because you said, "You guys must be doing a lot of planning to make all this work." It's actually the opposite in many ways.

Albert Chou:

Really?

David Heinemeier-Hansson:

Our approach is one that is very short-term focused in terms of detail planning. We have a saying that planning is guessing. And we like to keep our guess range short such that we actually have a chance of hitting the mark, which means that we work in six-week cycles. And that is basically the extent of the time we try to plan in detail. Everything else beyond that is a big fuzzy cloud that we don't try to distill into something concrete until it's time again to plan a new cycle. We do this cool down thing two weeks between each cycle where we take time to recalibrate, figure out what we want to do next, evaluate what we've done so far. And then, we make the call then. We don't have roadmaps of any kind. We never have had a roadmap of any kind.

Albert Chou:

Wow.

David Heinemeier-Hansson:

We don't really know when we're starting a new product development. Everything basically runs on that six-week clock cycle that we have encoded in our development methodology called Shape Up, which actually is also, in some ways, a product. We like to create all these byproducts from making things like Basecamp & HEY, which are things like software. We've created Ruby on Rails, the open source framework that we just give away.

And then, we've written a bunch of books. And we've also detailed our software development methodology called Shape up at basecamp.com/shapeup. And that details that clock frequency, the six-week cycle where we try to be pretty detailed about what we want to work on there. But there's even actually a point about that. And I'll get to a second. And then, we have these two weeks of cool down. And then, that's basically it.

There isn't a, "Oh, what are we going to do this quarter? What are we going to do this half of the year? What does the next two, three, four years look like?" We have literally never done any kind of planning in that regard. And that's 20 years of operation. I've been working with Jason for 20 years in Basecamp. Our main product was launched in 2004, in the beginning of 2004, so over 18 years old, or 18 years, we've been building software in this fashion, which when we tell people that this is how we do it, a lot of

people as you sort of alluded to will go like, "Wait. What? You're how many people supporting how many customers, and how many products and you don't plan? How does that even work?"

And I think the reality of it is that basically no one is very good at planning. And if they try to do it anyway, they end up being stuck with decisions that they made when they had less information, then when it was actually required. If we had tried to map out, for example, what is our feature roadmap going to look like for the next year, for the next two years, how are we going to make those decisions with the most current information? It's very hard to do. So we try to keep things free, not have long backlogs.

Backlog is a dirty word at Basecamp. We basically believe that long lists don't get done, and no one's motivated when they face a backlog that pretends to know the defeat with the future six, 12, 18 or further months out.

Now, some of that is perhaps related to the industry we're in and the tools that we work with. We work with the web and so forth. But you know what? Our scenario looks like most people's scenario these days. It's different from when we got started when there were still a bunch of people writing software that had to go on CDs and in boxes and on shelves, and so on. So I think, in some ways, a lot of companies are still stuck a little bit in that mentality of long-term planning, the long-term roadmap and all these other things lining up. And what we've just found in our history is that not only is that not helpful. It's actively harmful.

So none of that jazz going on at Basecamp, just the focus on the very near future. That's the only thing we can make an educated guess about. And the fact is, even that, even trying to estimate or we don't even do that either, which I can get into. But even trying to just figure out what you're going to do for the next six to eight weeks is actually surprisingly difficult. Software is not something that yields well to planning. There's a lot of spice and creativity in it. And this is one of those reasons where I actually don't even like the word engineering, don't use it about the programmers we have. We just go with programmers because I don't think of myself as a software engineer, even though I've written everything from Ruby on Rails to the start of all our major products. So anyway, that's sort of like a quick whirlwind tour of some of the weird thoughts we have about developing software at 37signals.

Albert Chou:

Listen, that's why we wanted to have you on the show because it's such a contrarian thought process of how to implement and run a technical organization. We had to get your... I loved your comment about long lists don't get done. I can't tell you, I worked in software myself. We supported 1400 customers, many of them in the enterprise. The amount of feature requests we had was so unfathomable.

And to your point, if it didn't make the top three, it probably never got... It just didn't get done no matter what. It just didn't get done because someone else would insert something with priority. I want to ask you this one question about what you said about the short-term view, right? You guys, you mentioned 37signals operates with a short-term view, six-week view. Did you always know that to be true or did you at one point have long-term planning? Did you work in a long-term planning capacity, or was this a hypothesis you started with and just kept two?

David Heinemeier-Hansson:

Well, we sort of started with even shorter term planning, to be honest, in many ways. But we tried a lot of different things over the years. We tried several months. I think, at one point, maybe, we were running three or maybe even four months cycles, if you will, that we were trying to say like, "Oh, this is the unit of time we want to plan for."

And we kept moving it back and finding that we liked that experience better. And then, when we hit six weeks, it was sort of this magic moment for us that it was just enough time that there was a version of everything that we wanted to do basically in terms of new feature development that could be done in that time. And that's one of those key things about the cycle period being so short, is that it serves as a boundary for your appetite to do everything under the sun.

And the fact is the most software end up being crappy not because you built too little but because you built too much. And we try to be very conscientious about the fact that if we give ourselves, and we've been building software for 20 years, very popular software. But even us in our situation, knowing everything that we know, if we give ourselves months and months to work on something, it'll come out too much. We need those constraints. Every human needs constraints and boundaries to let the creativity really flow in ways that come up with something succinct, something that really cuts through, something that hits just the spot, not too much.

And for us, that magic frame has been six weeks. There's basically nothing that we work on in terms of the established products that cannot be done in some version in six weeks. And that's also the key part here. When we plan our work, we plan it at a very loose sketch level based. There's no big documents, no big design documents about how things should work. Jason, my business partner, works with the designers at a fat marker level of detail, fat marker, so fat that you can't even write a specific word on a button that makes sense. You can just do squiggles.

And when you do fat marker designing, you end up at a coarsely designed flow that you want to get done, that yields very well to the particulars on the ground when people actually start building it. And that's really the key, this belief that even people with decades long experience working in the same domain with the same tools and techniques, they can't envision exactly how a great feature is built. They can do a fuzzy sketch. That's about the best.

And then, the people on the ground who's actually doing the work, they need a huge amount of leeway to make the detailed decisions in the moment that fits with the tools that you have, we like to say, cuts with the grain of the materials. The HTML, the CSS, the JavaScript, the Ruby, all the tools that we use cut with the grain to come up with those solutions where you get a lot for a little.

Albert Chou:

Yeah. The amount of freedom I guess that you've described, because I'm picturing if I'm a software engineer or I'm a developer working with you, I know you said you don't like the word. But I don't know what else the word you use.

David Heinemeier-Hansson:

Programmer is the one is we use.

Albert Chou:

Programmer.

David Heinemeier-Hansson:

But you can use whichever.

Albert Chou:

So I'm a programmer. It sounds like I have a long runway. So if I'm the master of my domain and you guys have a fat marker flow, well, that means there's a lot of details that are purposefully, it sounds like left untold so that I fill in the blanks. Is that accurate?

David Heinemeier-Hansson:

Well, the thing is at 37signals, we work usually on every feature in pairs. There's one designer. And then, there's one programmer. It's rare that we will just leave a programmer alone to do something on their own accord because that's not where the good tradeoffs come from. A lot of software actually looks the way it is and works the way it does because programmers were left too much to their own devices, and they came up with something that was just easy for them to do but was not a result of a trade-off between how does this work, what is the flow, and what's easy to implement. You can't actually trust programmers to just go off on their own in a coarse grain sketch and come up with something that's also great for users.

You need this pairing. You need this trading of concessions between the designer and a programmer to arrive at something great. So that's what we do. When we do this fat marker sketch, usually Jason, there's someone else that does the shaping, as we like to call it, of the pitch, of the feature that they're working on. That's then handed over to a product designer, which by the way, 37signals a product designer doesn't just mean draw things in Photoshop or and Figma, whatever. They actually make things in HTML, CSS and sometimes a fair bit of JavaScript and in some occasions even a dash of a Ruby.

So they work in the concrete materials. They work in when we actually build the user interfaces with. And then, they work together with a developer that then makes that come true. And there's often that sort of, as we say, the trading of concessions, a designer will come with an idea for how something should work. And then, the program will go like, "Oh yeah, that looks great." But could we change this one thing because if we did, it'd be so much easier to implement. I could shave a week off. And then, we could spend that week on something else.

And sometimes, the designer just goes like, "Yeah. Yeah, totally." I mean, I just drew that because whatever. It doesn't matter at all. And then, other times, they'll go like, "No, no, no. This is absolutely the crucial part of this feature. If you remove this one interaction, the whole thing is for nothing." So it's in that interplay between the designer and the programmer that really great trade-offs are made. And when you make something under those conditions, you can arrive at a great version of almost anything you want to build in a maximum of six weeks.

Albert Chou:

When you were developing this system, you and Jason have been working in this capacity for such a long time. Was there ever a, I guess, a desire or interest in seeing how other companies or other ways were being done, or was it just that the results were so good that you just kept building on this culture and saying, "This is how we're going to continue to do it?"

David Heinemeier-Hansson:

Well, both Jason and I were blessed with some absolutely horrendous experience working for other people. So we got to see quite a few different ways to build software before we started working together. And I think that really was just instrumental in us figuring out to do things this way because I worked at a number of software companies before I joined up with Jason. And we started building things together.

And those experiences were just seared not just on my mind but on my soul. And a lot of the motivation to both want to be my own boss at 37signals and persevere through it was that, you know what, I don't want to go back to working for other people because I've tried that and I didn't think they had great ideas when it came to some of this software building stuff. And there's a certain amount of arrogance and hubris that go into wanting to be an entrepreneur. And I certainly had plenty of it, and so did Jason.

And we went like, "Do you know what? On most things, we're going to start from first principles." I am a student of software development. I'm a student of software methodologies. And I've read and examined a huge variety of other approaches to software development and studied other organizations. And obviously, in 20 years in software, I heard plenty of stories and observed plenty of outcomes through other ways of approaching things.

And some of them had great ideas. And many of them had absolutely horrendously terrible, no good, very bad ideas that caused not only software to be late, to be buggy, to be user hostile, but to be an absolute drag to work on. And if there's anything that I've been motivated by in the 20 years of building my own company together with Jason is that I would want to go to work.

Albert Chou:

Yeah. You guys talk about that.

David Heinemeier-Hansson:

And I am a programmer at heart. So I want to go to work. And I want a program in a way where I smile as I do it. This is one of the reasons why I was so motivated to build Ruby on Rails because I like to build my own tools. We like to build our own tools at Basecamp. We like to build our own methodology. We like to build our own organizational processes. And a lot of them ended up looking kind of funky, weird, out of the ordinary, not the industry playbook exactly for that reason that both Jason and I, as executives at this company, we are still deeply involved in the work, sometimes to a fault.

But if I don't program... If not on a weekly basis, that at the very, very least on a biweekly basis, I go crazy. I am not a happy camper at all if I do not touch the particulars of the code, preferably my own code. But in a pinch, I can at least work with someone else on their code. But I absolutely have to be in the materials. And Jason is the same way. He has to be in the design. He has to be in the copywriting. We have to be in what we're working on. And in that regard, I think we are quite different from most other executives who run companies the size of ours, at least if you consider the number of customers that we have and the amount of revenue that we've generated.

Albert Chou:

Yeah. It's pretty crazy.

David Heinemeier-Hansson:

If not even also just the number of employees we have, even if we're a relatively small company and that produces different outcomes. You get a different kind of company if it's led by a designer and a programmer than if it's led by someone who used to do design and programming perhaps many moons ago and are not quite in touch with that stuff anymore and are focused on other things.

Perhaps, they're focused on pleasing their investors or other financial targets and goals. And do you know what? It's great if you run a good business. We always want to run a good business at 37signals. We've been profitable since day one. We own the business or the profits. And large slice go to us. I'm

very keen of that, and I have a deep aesthetic of traction to running a good business. But it's not the be all end all.

I would absolutely trade a wonderful business opportunity away if it meant I had to do work I didn't want to do. So in that way, we are kind of, I don't know, you could call it selfish or myopic or whatever. But it's been a recipe where it has allowed us to focus on the things we do really well to double down on those things, to become absolute masters at that stuff and then, share. That's the other thing.

I mean, we get to build a weird company using weird software in a weird way. But then, we can tell other people about it. And sometimes, some of that weirdness rubs off, if not outright on the entire industry. Then, at least on individual companies and participants in that industry who then go around and say like, "Oh, this was great. This was not at all it was we worked before." And now, we're working this way. For example, now we've adopted Shape Up. Now, we work in these six-week cycles. We do this fat marker pitching. And wow, it is so much better than being a feature factory, pulling off the endless backlog in Jira that's just driving everyone freaking crazy. Right?

Albert Chou:

Yeah.

David Heinemeier-Hansson:

And the same thing with Ruby on Rails on a technical level. The number of companies who've started using Ruby on Rails, this is a system I've been building also for what, no, almost 20 years now. And seeing things like Shopify or GitHub or others have tremendous success with it is-

Albert Chou:

Massive, massive companies built on Ruby. Yeah.

David Heinemeier-Hansson:

Yeah. It is hugely gratifying. So those are some of the payoffs that we get from doing it from this first principle way from we get from doing it in this sort of weird way. Doesn't mean we're always right. And it certainly means that we're not always right for all the people all the time. No one is. But at least, it's different. And I think this is one of the things that always just inspires me. If I can make someone think, I don't need to change their mind to be exactly like mine. But I really do like to make them ponder, "Huh. So the way we're doing things, there are other ways of doing it. And those ways are producing great outcomes. Let's have a look again. Can we learn something? Can we pick something up?"

And in some of those cases, they'll pick up Basecamp and actually use that to run the project as you've done at your companies. They might pick up Ruby on Rails. They might pick up Shape Up. They might pick up any of these things, or at least even if they do none of those things, we will just have maybe given someone a smile because they read an essay that was off camper and maybe picked up one of the books, something like that.

Albert Chou:

Well, one of the things that you and Jason had talked about back in the early days when I was building my... I was part of a software startup that eventually got acquired. But there was a part where we diverted from your roadmap. And we diverted from your roadmap. And I got to feel all the pain you warned about one. And that point was, "Hey..." And I'm paraphrasing. But there was a point where you

and your team decided, "Hey, for the fee, we're going to charge a small business fee." At the time, I don't remember what it is. But it's a \$100 a month now.

But the reasoning was interesting. The reasoning was because as you get bigger customers, like you said, you're chasing the dollar, those bigger customers have an outweighed impact on your gross revenues. And so, when they say, "Do this," you kind of feel obligated. And for sure, there was that moment in time where when you're living it, you don't know that's happening. And then, someone maybe points the light of what's happening to you and you realize, "Wow, we are feeling just an inordinate amount of pain building this software," because we are developing all these features for our biggest customers that do not seem to be driving any type of incremental value to the rest of them. And that was happening to us. And it hurt. Did you learn that through experience or is that something you guys kind of made a bet on from the very beginning and said you were going to stick to this because that's... The one thing that for anyone who's never used a Base... or a 37signals product is it's probably more known for what it doesn't have than it does have which is crazy because you'd be like, "Oh, then, why is it useful?"

It's like, because it only provides what you need to get the job done. And that was the pitch to me when I remember when I first heard. I was like, "What do you mean?" You know what I mean? And then I tried it. I was like, "Oh, okay. I get it." But was that a conscious thing or was that just more through your experience because I'm sure along the way, people were pulling on you, "Hey, David, if you just built this for me, I'd pay more. I'd pay more, man." I'm sure you heard it.

David Heinemeier-Hansson:

Yes. No, absolutely. Well, we used to talk about this a little bit more and we labeled it Opinionated Software. And what we were building was opinionated software. It took a stance. It still takes a stance. Hey, our most recent product is incredibly opinionated in ways that some prospective customers don't just find like, "I don't want to use that." They find it actively offensive. They are mad that it does the things it does the things it does and the way it does, and that it doesn't do the things that they really wanted to do.

Albert Chou:

They could just not subscribe.

David Heinemeier-Hansson:

But we were simply... We were born in that forge. And part of that was like why were we starting this company? Why were we running this company? And I think number one thing for both Jason and I was independence. We wanted to do all the things that no one would give us permission to do. And that level of affiliation or attraction to freedom drove a whole host of decisions.

And very early on, as you say, as soon as we started having customers, those customers started having not just suggestions but demands. I need you to do this thing. I mean, some people ask nicely, and some people don't. And there are plenty of the people who don't. And they will demand, "I am paying you \$20 a month. You better do everything I say right now. Otherwise, you will lose my \$20." Okay. I mean, that would be sad. And I hope you find a piece of software that works for you. But it is so much easier to say that when it's \$20 or \$100 a month as it is right now, than if it's like \$20,000 a month, right?

Albert Chou:

That's right.

David Heinemeier-Hansson:

And particularly in the early times, it is so easy to fall under consulting [inaudible 00:28:13] capture, where you have a few big customers. You feel like they're too big to lose. Therefore, you must dance to whatever tune that they're playing. And then, you end up driving your software on the request of a few large customers. We never wanted to do that. We wanted to drive our software on behalf of every customer, both present and perspective.

And that's really the key insight that we found was so important for our success was that we were not just building for the customers we currently had. We're also building for the ones we wanted to have. And there's tension there. The customers that you will hear the most from, the people who have the most strenuous demands are you super users, the ones who figured everything out and want more configuration, want more options, want more features. The ones you never hear from are the prospective customers, the one who will only sign up for your product if it's easy to use, if it's easy to get others to use with them. And they won't tell you if it doesn't work.

I mean, you can do sort of simulated things. You can try to cancellation studies and so on and so forth. But you as a software builder really have to internalize that and be their primary champion because the existing customers, they'll be their own champion any day after week 24/7. You have to be the champion of all the customers who have not yet signed up for your product. And we found that to be imminently compatible with our number one mission and goal to be running this company independently. This is why we never took venture capital.

This is why we never got put on someone else's financial calendar in terms of milestones we had to hit or whatever else that could drive the kind of products that we were using in a way we didn't sort of inherently wanted them to go. And now, that's the other thing, is we've only been building software that we've been using. And that is also sometimes unique. I mean sometimes, and thankfully there are people out there who will build software just for other people. They don't use it. They're trying to understand what a customer will want and then built on top of that understanding.

And I have the utmost respect for people like that because that is very, very difficult, or at least it's very, very difficult to build good software that way. If you can only evaluate whether you've done a good job or not through the eyes of others, man, that is a very long round trip. For us, we are building Basecamp for ourselves. We were building HEY for ourselves. We were building Highrise for ourselves like Campfire, and all the other products we've worked on for the past 20 years. We were building them for ourselves first.

And not only that, we were so selfish that version one and everything we've ever built, we purposely built exclusively for ourselves. We always build the first version of any piece of software we make just for us, no perspective features that we think maybe converts a little better for some people who warned us. No, no, no. None of that. B1 is for us. That means we put out an opinionated piece of software. It doesn't do all the things. It does a bunch of things that people have never even thought that they wanted or needed.

Anyone who signs up for our email product at hey.com will probably never have thought about the fact that having a screener, having a way to prevent anyone who have your email address from being able to invade your attention on their schedule and even cause a notification is just an absolutely bizarre and terrible way to run your communication through email. And you can do it a different way. And HEY has this concept of a screener, which is incredibly novel.

No one would've asked for that, right? It's the old saying of Henry Ford that the customer wants a faster horse. If you want to build a car that's on you. You have to come up with that vision. You have to put

into the market. You have to develop the demand yourself, sell people on the things they didn't think they needed. So that's the kind of stuff that really inspires us to build those super strong B1s.

And then, once you have the B1 in the market, yes, listen to the customers. Don't work on their request. Work on their behalf and spread it widely. You don't do something just because one big customer asks you to. You do something because you think, "You know what? This would be great for 10,000 customers."

Albert Chou:

So this plays very nicely into what you said earlier, which is this idea, if we go on a six-week sprint, we're going to develop and engineer our product the way we want. And then, we're going to do this two-week cool down. And I'm assuming during these two weeks, you're evaluating what you've done or how customers are using it and so on because I know someone's thinking right now, that's not enough time. There's not enough time to figure out what people are doing, what's happening. So there has to be like this. I think what's the most unique about you and your team is the thoughtfulness that goes into your practice.

I know you said you even think about these in six-week sprints. But you're making decisions in just two weeks. That seems there's definitely companies that plan for a year, like you said. You know what I mean?

David Heinemeier-Hansson:

Yes.

Albert Chou:

They're planning forever, or they're collecting user data like I need to see more. Right? You guys sound like you're making intuitive jumps. So give me an idea or give our audience an idea of how much data do I really need, and because you hit on it already, which is people will not provide any data to what they actually want? You already hit it, what they want, what they'll say. If you ask them show you data, it's probably the same as asking them what they want. They'll just want to improve what they're currently doing. They don't actually know what they need in the future.

David Heinemeier-Hansson:

Right. Exactly. And do you know what? The data won't tell you either.

Albert Chou:

That's right.

David Heinemeier-Hansson:

We are not a data-driven organization in that regard. We are an organization driven on intuition and experience. And those two things are sometimes very hard to quantify. But they allow you to make those kind of leaps that the lack of data keeps many organizations from making. There are a lot of organizations who've been enthralled with this sort of Google motto of, do you know what, we have to pick a shade of blue. Let's try 129 different versions of it, and we'll AB the whole thing to the moon, and we'll pick the one, whatever that does best which is a great way of getting these local optimus, local maximization that does not consider the entire picture.

We have a bunch of features, in Basecamp, for example, that are not highly used. If we take one of them, for example, Hill Charts, so Hill Chart is this really novel approach we came up to how to both get a broad bird's eye view of progress on the project for the people working on it, but also to communicate it to others. It has this bimodal setup of you have things that are going up the hill, things you're still figuring out and things that are going down the hill. And you have this way of capturing this fuzzy feeling of where are we with the project such that you can eliminate all these questions of like, "Oh, so where are we on that?"

Well, go look at the Hill Chart. Anyway, I think it's an incredibly strong conceptual model. It's something we've actually used a ton at 37signals. But it is a bit of a hard sell. But it's an aspirational feature. If we were just looking at usage data and we saw like, "Oh, well we can't get 50% of Basecamp users to use health charts. It's a failure. We should just drop it, right?" We would've missed that aspect of it. It's an inspirational feature. You have it because it's part of the appeal of the whole product is that we're striving to be something more than just like, "Oh, I don't know, working on the backlog." Even if that's what we're doing right now. People need aspiration in their lives. You look at this in all sorts of products where they get this totally a Rolex deep ocean watch can go down to, I don't know, whatever, 4,000 meters. Do you know how many buyers of the deep ocean go down to 4,000 meters?

Albert Chou:

No.

David Heinemeier-Hansson:

Zero. Absolutely zero. It has literally been probably tested once in somewhere. But there's an aspirational. I could do that with this product or a Bugatti Veyron. They'll do 400 kilometers an hour. How many people are doing 400 kilometers an hour in their Bugatti Veyron? Zero. Absolutely zero. Or if they ever do it, they do it once, right?

Albert Chou:

Yeah.

David Heinemeier-Hansson:

This is something we understand in many forms of product development. This is something that software developers, I think, don't quite understand. But this is something we've internalized dearly. So we try actually to actively counteract the data-driven approach. No. That's not to say that we never look at data. Sometimes, the data will give you answers. But you have to be damn sure that you're asking the right question. And that is often the much harder task is to figuring out what is the right question. And is that question one that can be answered by data?

I think a lot of organizations work the other way around. Which kind of questions can we pose against this data? Those are the ones we'll ask. Yeah. Well, I'm sorry. This is how we get the software that we have today. A bunch of user hostile stuff that's full of micro optimizations. This is why we had websites that are just pop up of sign up for my newsletter. There's more in your card. Here's 15% off. They're all individually AB tested breaks that lead you on a road to hell.

Albert Chou:

Listen, I like the way you framed that because it does... When I have that same frame of thought, which is so we'll use for example, our show right now. But people have said, "Well, we need to do more

surveys. We need to do more surveys." But the surveys will only tell you what the people who choose to answer the survey will tell you. The real question we want to know is why don't people listen? So we have a million downloads plus a year. But who is not listening? Why have they not selected us to listen to? That's the question. And like you just said, there is no data on that because it's not possible to collect. We don't know why. We would have to make a guess. So I always tell our team. We have to make guesses. If we want to raise our ratings, we got to guess. That doesn't mean no data to support it.

David Heinemeier-Hansson:

Yeah. And this is the thing where, I mean, I like the word guess because it's honest. We're not sure. But it also sometimes undersells it a little bit. You're not just guessing blindly. You're guessing on the basis of intuition.

Albert Chou:

That's true.

David Heinemeier-Hansson:

Why are you good at what you do? Well, because you've developed an intuition for what's right. Why are we good at making software? Well, because we've kind of been doing it for 30 years each. And even before that, that we had an intuition that was developed. And you can trust that. And I think this is one of these errors of getting captured by data. And I think you can get captured by data. You can get captured by your customers, existing customers, not perspective ones.

You can get captured by your audience. And in this data-spewing world that we have today where literally, there are faucets of data just open, spewing out everywhere. It seems like that that is the logical, rational thing to do. To be data driven is a compliment in most circles. I don't actually think it's a compliment, not as an overarching strategy for how you run product. Now, you need to be intuition driven. That intuition should be informed by some data some of the time to some of the questions where that make sense. But I've found that there are few of the important questions that can be answered by data. And if you try to push it too hard, then, you end up in a bad place.

And the problem here is that relying on intuition means relying on someone who's then possibly to blame, which means that it relies on literal courage. In the face of the unknown, do you dare go forth with your best ideas for how things should work? And if those ideas come up flat and don't turn out, what happens then? In some organizations, maybe you get your head chopped off. In our organization, we accept that we will be wrong. This intuition will take us down blind alley sometimes. And when we hit a blind alley, we say, "Oh, well, that's the price of having an intuition." Let's turn back and go the other way. And that's fine. Most decisions most of the times are reversible, particularly when it comes to product development. And it's the only way you can develop this kind of intuition is by making calls. Many of them moving on and keeping up the pace. This is the other problem with data. It takes too damn long.

Albert Chou:

It's expensive to collect too.

David Heinemeier-Hansson:

To collect the right data. It's expensive-

Albert Chou:

Expensive to collect.

David Heinemeier-Hansson:

... to collect. It's complicated to actually get right. You really need people who can figure it out and da, da, da, da. If we had to base every decision we make at 37signals on data, we would never make even a hundredth of the decisions that we're making. And we need to be making decisions all the time and moving on because this is how we can keep the pace. This is how we can deliver features with one designer, one programmer in six weeks.

Albert Chou:

Yeah. When I saw the total team, there's an Android app. There's an iPhone app. You guys are available on every platform. It was bananas, the thing about it, because I've seen this done. The other thing that is really fascinating about your company and specifically you, now, I'm going right to you David, is this dedication towards writing as a form of communicating and establishing the thought, purpose, vision, whatever you might want to say for what to do next, you've clearly invested quite a bit of time.

Very few technical leaders think writing or they don't appear to be writing. I don't know. It's not like a strength of theirs. And they don't seem to cultivate and develop it. I got to ask you. Were you always a good writer or is this something you've figured out, like, "Hey, if I get better at this, this is going to help my..." Especially since your team was among the first to have hybrid or remote or I believe companies. How did you make the decision to invest in your writing? How'd you get better, or were you just naturally pretty good at it already?

David Heinemeier-Hansson:

No. No. I think there are very few natural born super writers in this world. But what I was good at quite early was I was a good reader. I read a lot. And I read the best there was. And I was just obsessed when reading about software development in particular, organizational structure. It's actually only sort of in the last few years I've even started reading nonfiction because I was just so... or sorry, started reading fiction because I was so obsessed with reading nonfiction.

I was so obsessed with learning about my craft, learning about my industry, learning about all these things, psychology and all the adjacent areas that are relevant to running the business. And then, I got inspired. I got inspired by reading great writers, great technical writers in particular. And I thought like, "Do you know what? I'd love to be one of them." And then, I started writing. And I've literally been writing for the web for, what is that, 26 years now or something like that? I must have written literally thousands and thousands if not tens of thousands of entries on the web in large or small form.

And eventually if you do that, you'll probably become pretty good at it. Now, it's not an automatic process, and there's a repeated training that does nothing. You need to do dedicated training, intentional training where you're trying to push yourself and get better. And it also just helped that I love writing. It is such a joyous activity to me that it's not something I'm forcing myself to. This is one of the reasons why I fell in love with programming.

It was because it felt like writing, at least at the level of abstraction that I'm interested in working at with the kinds of IT systems that I work on and with the kind of programming language I work with. Ruby just fit like a glove for my brain in that regard. But writing to me is absolutely crucial if you want to be not just a good writer but a good thinker. Clear writing is clear thinking. And this is one of those sometimes

uncomfortable connections because you might think, "Well, I'm not good at all about writing. But I'm a very strong thinker."

I pushed back a little bit on that. I think if you are a very strong thinker, you can certainly also learn just the trade of writing to a point where you can express yourself clearly. And if you are a muddled writer, there's probably a good change you're also a muddled thinker. And again, you can work on both of those things. There are some levels of innate talent. But very much of it is in the dedication that you put into it. But lots of people just aren't interested in putting in the work it requires to become really good at writing, which I think is just such a shame because in almost all the mains that we work in, anything adjacent to the creative world or the programming world or even the business world or writing is a superpower.

It allows you to communicate your message, your mission, your motivation to just vast audiences that I can publish something that is read by tens of thousands of people. And I find that just to be thrilling, to be able to communicate ideas clearly in this way and not just thrilling, incredibly effective. There is no way we could run our organization as productive as it is on as many platforms that it's on, as profitable as it's been with as few people without this absolute dedication to writing first and meetings last, I should say too. These things go a little bit hand in hand.

Sometimes, the people who don't writing very much, they like meetings. They like talking. And that has its place too. I don't want to poo-poo that. And sometimes, I wish perhaps I was better or at least could simply endure meetings to a higher degree. But unfortunately in some circumstances, almost allergic. And that means I'm not very good in that domain. So I ply my trade with the writing. But it is something I think is absolutely crucial.

I wish more people put more energy into it. It's one of those things that aligns so well with the best piece of advice I ever got from Jeff Bezos who bought a minority stake from Jason and I, not invested in the company but bought secondaries from Jason and I back in 2006. And we had a bunch of great free-ranging conversations with over the years. But the one thing that really sticks out that he told us early on was invest in the things that don't change. And he made the example from Amazon of saying, "Are customers ever going to wake up thinking, 'Oh, I wish my packages were delivered slower. I wish it took a week instead of two days.'" Absolutely never. So as he put it, all the money we invest in getting better and better and better at delivering packages faster, those are just things that don't change that investment is never going to waste.

And I think of the same way with writing. Every investment I make in writing, that is something that's going to pay dividends until I close my eyes literally and forever. And that seems like just these kinds of too good to pass up investments where I can invest a tremendous amount of energy in whatever, some hot new technology and then, "Okay, that has a lifespan of three years or five years, maybe." The rare cases, perhaps, the lucky cases are the stopping cases as I've had it with something like Ruby that last for 20 years. They're not that common in technology. Things do churn over. But you know what? Writing is forever.

Albert Chou:

Yeah. I learned that when I was earmarked very early on in my career to do leadership training. They talked about the number one thing that breaks apart companies or projects, they said... And you can guess all these things, oh, it's skill. It's resource. It's all these things. They actually told us it was an alignment.

Alignment is the ability for everyone to think about the same end objective and maybe the same way to get there. And your points about writing is exactly that. It's aligning many people to see the same thing

because I think that's kind of my hypothesis on why people have meetings is because they're trying to revalidate like this is what we're doing, or this is what we're not doing. It's all in an effort to get that alignment where we have to be thinking the same way.

When you were writing these ideas of how we're going to do the next thing because that's cool about your company. You're building new things every six weeks, I guess. And you've already kind of hinted at. It sounds like you're investing quite a bit of time into, I guess, declaring the next step before putting it to paper so that everyone on the team can say, "Okay. We understand where the company needs to go next." Is that accurate or do you also sprint through that?

David Heinemeier-Hansson:

Yes and no. I would say we spend a lot of time thinking about this and reflecting on it. But then, we try actually to generalize this further than just the people who happen to work at 37signals right now. This is why we've written so many public posts. This is why we've written so many books because we're also designing this company for prospective employees and for everyone else who can benefit from this. Right?

Albert Chou:

Yeah. I've read it.

David Heinemeier-Hansson:

Exactly. Right? The kind of ideas that we are trying to promote are ideas that I think are right both for the people who actually work at the company I'm a co-owner at, and also everyone else that we are in the same tribe. We're trying to figure out how to write good software.

And that problem is a very difficult one, which means it's a fun one. It's a kind of like chess. You can start developing software with surprisingly little knowledge about anything. And then, you can spend an absolute lifetime trying to master it, trying to master the techniques and the tactics and everything else there is to it.

So I kind of feel like it's a collective action problem that we shouldn't try to constrain just within a single company that's kind of actually lazy. It's also slow. We iterate faster on the best ideas if we have our eyeballs dedicated to them. Doesn't mean we will always disagree in all regards. But we actually put most of our writing and thinking out there. But at the same time, we also try to be brief. I mean if there's anything I hate, it's a 600-page tome about something that could have been a, if not a tweet, then a blog post. So this is why all our books are just collections of essays. There is almost no topic, individual scope topic that I cannot distill a good version of what I want to say about that in two pages.

Albert Chou:

Yeah. Listen. This has been an awesome conversation. With what you just said, I think that's probably why... I mean it'll sound like I'm fan boarding. But I guess I am. But I think for myself, I align very quickly to how you... Because I came across your writings first. I found myself aligning very quickly to how you guys thought, which is, "Hey, we want to dedicate ourselves to doing great things." We want to not be beholden to the dollar to pull us apart.

We don't think life is meant to go 24 hours a day of working. We want to work. We want to have family. We want to live life. And so, when I was thinking of all these things, I was like, "Man, I do align with that." And so when someone told me to try your software, I was like, "Ah, this is pretty easy to try." You know what I mean? That's kind of the story. So you've heard probably this story a hundred times. But

yes, you're absolutely true that the way you're writing is finding because you're opinionated software. Of course, those will people disagree with you. But those who do, they are going to try it out a use case.

David Heinemeier-Hansson:

And this is where it's all this big happy kumbaya circle. The good writing is a way to think clear. So it's helpful to myself. I always write to myself. I am the reader. I don't write on what I think other people should hear or should know. I write to myself usually to work out an idea. And then, once I've worked out that idea to remember how I worked it out and to remind myself of the conclusions that I've reached. I've find that the hardest things, especially the longer I've been on this earth, is it's not the hard thing. It's not learning new things. It's keeping the things that really matter top of mind.

That is a very difficult thing. And it is something that requires repetition. And it requires further clarification on your thinking. And I can do all that stuff in my workshop. And then, I can put it out there. And it works for employees at the company. They now have an idea of how to align around how I see things or how adjacent you see things. It creates this implicit alignment without me having to do it on a one by one basis.

It works for prospective employees. And it totally works on prospective customers as well. We find our kindred spirits. And if you find yourselves as you just recalled reading something that I wrote or that Jason wrote or someone else at the company wrote that you resonate with, there's a very good chance that you will also resonate with the kind of software that we build because it's built by the same people. And writing software, there's a reason we call it that, right?

Albert Chou:

Yeah.

David Heinemeier-Hansson:

Writing software, because it is an expression of writing. It is an expression of your ideas, your opinions, your values. And if you find yourself aligned with the kind of stuff that we are writing in rework, yes. Basecamp is probably going to feel like something you would want to use. Hey, it's probably going to feel like the kind of email system that you would want to build.

And I love that. And I love the fact that it's not for everyone. I love the fact that we don't need to conquer the world. I've never cared about market share. We don't need to convince any public markets or investors or anything else like that. We simply just have to find a solid group of customers who can allow us to continue to profitably create great software, which is the thing we really love to do. And that's what we've been doing now for the 20 years that we've been at it. And I hope to do it for another 20 still.

Albert Chou:

David, it's been awesome having you on the show, sharing this mantra and philosophy that you live by. It's not just words like you guys actually live it. And it's awesome to hear. For anyone who hasn't checked it out yet, if you're not familiar, Basecamp is project management software. It is found at www.basecamp.com. And they also have an email product, which is at hey.com. This is pretty awesome having you here. But before you go, David, it is time for the lightning round. The lightning round is brought to us by Salesforce platform, the number one cloud platform for digital transformation of every experience. David, this is where we ask you questions about your life outside of work so our audience can get to know you a little bit better. Are you ready?

David Heinemeier-Hansson:

Shoot.

Albert Chou:

All right. You've won a Le Mans race. How did your body feel after that?

David Heinemeier-Hansson:

It felt sore and exhausted. It's a 24-hour race. But you're not just awake for 24 hours. I think, in most cases, you're awake for almost 40. That is a long time to be mostly awake the whole time but absolutely a thrill. One of the highlights of my, if you will, career outside the career in 2014 to win the 24 hours of Le Mans with Aston Martin.

Albert Chou:

What's your favorite car? We know you're a big car guy. What's your favorite car?

David Heinemeier-Hansson:

My favorite road car is the Pagani Zonda. It's a small coach building operation in Italy that makes these handcrafted cars, if you will, that I was so fortunate to let all the good wealth that we've accumulated at Basecamp, some of that went into a car, my favorite car in the world. I've had that car for 11 years. It's still my favorite car in the whole world. And I sometimes pinch myself when I see it in the garage and think like, "Wait. I have the coolest car in the world." That's amazing. And that's fun too. I mean, I think this is part of it that when the business works out well, then, you get to do some of those crazy things.

Albert Chou:

You are also very famously a family man. What do you like to do best with your family? What are some of the things that you enjoy?

David Heinemeier-Hansson:

Yeah. I have three boys with my lovely wife, Jamie. And yeah, I love spending time with them. I love playing video games with them. We played a bunch of different games. And I'm sure some people will tsk-tsk-tsk. But I love playing Fortnite with the kids, get them started early. I think I probably had the best four or five-year old Fortnite player anywhere in the world. That little man once got a 13 battle Royale, Victory Royale streak in a single day. I helped out on half of those wins. That was absolutely incredible. I love doing things like building Legos with them as well, just spending time together with them, particularly at those different ages. You get to see just little humans grow. It's truly amazing, one of the most satisfying experiences that I've ever had.

Albert Chou:

What's the most complicated Lego figure you've ever built?

David Heinemeier-Hansson:

Do you know what? The sad thing is that we really don't have the stamina for the big sets. I have the stamina to put Legos together for about an hour at the most. So we have a bunch of half-built sets. And then, the ones that are completed are usually the simple ones.

Albert Chou:

There you go. David, it's been awesome having you on the show. Thanks for sharing all about how you guys developed and build Basecamp. You've written about it a ton. But you really are... I mean, I think the reason why your books are so popular is because there's very few people in this industry that think like you. That's the fact, right? You're very contrarian in that regard. Thanks for sharing your life outside of work. I'm pretty hyped up to have you on the show. I thought it was interesting that you could definitely endure a Le Mans race, which is physically exhausting, right?

David Heinemeier-Hansson:

Yes.

Albert Chou:

But a Lego set really halfway there.

David Heinemeier-Hansson:

Absolutely.

Albert Chou:

David, it's awesome having you on IT Visionaries.

David Heinemeier-Hansson:

All right. Thank you so much. And if anyone want to read any of my writing, it's dhh.dk.

Albert Chou:

And for everyone out there listening, before I got to mission, it was using another project management software. We won't name it. But it's publicly traded. I want to tell you, it was the most complicated thing I've ever witnessed in my life. And the first thing I talked to the existing employees when I got there was like, "What is this supposed to do?" And then, they're like, "Oh, it's supposed to help us organize." And I just looked at it. And it's like watching people use it. It's like, "I don't think this is helping anybody." And I think there's more people ad... I call it admining the admin. Once your administration requires administration, you are down the death. You're going down the death hole. I said, "We spend more time logging stuff than doing the thing," which is making podcasts, like, "This sounds terrible. Let's kill it." I brought Basecamp in and said, "Let's simplify. We just need to know who needs to do this, and is it done? That's all I need to know."

David Heinemeier-Hansson:

You are speaking exactly actually to these core drivers that we hear over and over and over again that Basecamp is not a system that you have to have an administrative for. It's not something that you have to have a manual for. People can just figure it out, get their stuff done, and get back to work.