Rashmi Mohan: This is ACM ByteCast, a podcast series from the Association for Computing Machinery, the world's largest educational and scientific computing society. We talk to researchers, practitioners, and innovators who are at the intersection of computing research and practice. They share their experiences, the lessons they've learned and their own visions for the future of computing. I am your host, Rashmi Mohan. The power of open source and the significant benefits, it is brought to the world of software development is unquestionable, creativity, innovation, and collaboration, and a genuine desire to share knowledge is what powers this movement.

Our next guest knows a thing or two about that. David Heinemeier Hansson is the creator of the very popular open source web framework, Ruby on Rails, that is used by some of the most well-known technology companies. He's an entrepreneur and co-founder and CTO of the technology company, Basecamp and a prolific author of multiple bestselling books on building and running a successful business. He also lives life on the fast lane as a race car driver, and is the 2014 winner of Le Mans, the world's biggest endurance race. David, welcome to ACM ByteCast.

David Heinemeie...: Thanks for having me.

Rashmi Mohan: We are super excited to be talking to you today, and I'd like to lead with a question that I ask all my guests, which is, if you could please introduce yourself and talk about what you're doing currently, as well as give us some insight into what drew you into the field of computing.

David Heinemeie...: Sure. So I have been working on open source software for almost the past 20 years or so, Ruby on Rails is at the center of that, and it really all started when I discovered Ruby back in the early two thousands. I discovered a programming language that had already been around for almost a decade by the time I looked upon it, but it was not very popular in the west at the time. I think one of the reasons, perhaps, it wasn't that popular was that, there weren't so many tools to use that wonderful programming language for things like web development.

So when I picked up Ruby and started using it to create Basecamp, the project management application, that is the name of our company and that I still work on to this day, I realized that I needed to build these things myself. So I built a bunch of open source tooling to go around the Ruby language to make it uniquely suited to solve the problem of web applications, such as connecting to a database or generating HTML or sending emails or any of the many other things you need to do when you're making a web application.

And by the end of that journey, I realized that I had built an entire toolkit and it only came natural that I should release that toolkit to the world as open source, because I was the beneficiary of so much open source, not just Ruby, but plenty of other packages like the database, MySQL, and the web server, Apache and Linux servers and so forth. So, it wasn't even within the realm of my purview that I would think, "Hey, we should try to commercialize this." No, of course let's make it open source, let's get hopefully more people to use Ruby because of these tools. That's what I've spent the past two decades or so focusing my career on really on the Basecamp commercial application and the Ruby on Rails open source framework and community.

Rashmi Mohan: It's lovely, I think what you're saying is really I think what really is the crux of the open source movement, which is really how do you pay it forward? Right? We have beneficiaries of so much great work that's happened before us, and how do we give back to that community? But one question I did have, David is, I mean, oftentimes, especially because you created Ruby on Rails for yourself really, right, to aid you in being more efficient and then you thought about releasing it to others so that they could use it as well. But there's a certain level of maybe polish or perfection that one thinks is necessary when you're releasing software to others, as opposed to using it yourself. Did you feel like you had to make that leap too? Or was that just the plan from the very start?

David Heinemeie...: I was very conscious about that leap. And one of the reasons I was conscious about that was at the time I felt I saw a lot of good open source software, not really having the impact that it could have had if it was not just polished in the terms of being well documented perhaps, or what else have you, but also marketed really. In the early two thousands, it was my impression that a lot of people in the open source community saw with great skepticism upon the idea of marketing that you could have a good looking website that you could do things like screencasts to show up your technology.

I didn't have any of those reservations whatsoever, I did have a burning desire to use Rails to create an interest in Ruby, the programming language. And what was the point of me going through all of the polish required to release the open source software if it was then just going to essentially go nowhere because no one had heard about it and no one were excited about it. So, I approached that problem as a marketing problem, as a product launch issue in a way that I think at the time at least was somewhat novel.

The original screencast for Ruby on Rails was a 15 minute video of me building a web block, really showing how to use it, not precooking anything, just going straight into the meat of it, and I think that had a tremendous impact on the people who eventually ended up trying Ruby on Rails and many of them staying and building applications with it, but also in the industry at large, that if you look upon many open source applications and frameworks, these days they are just as slick as a commercial product would be. In some cases they're even more slick and better marketed. And I think a lot of that came from perhaps some of the things we did in the Ruby on Rails community to spur that on and just be serious about adoption being a measure of success in itself.

Not necessarily like who's beating who, who's at the top of what stack, but using adoption to get to escape velocity, basically bootstrapping communities such that they're self-sustaining, if you launch something into the world, particularly something like a web frame, where, again, you have five other people using it, that's a little difficult to make a self-sustaining community like that. But if you have thousands of people using it, it's quite easy to become sustainable, you don't need all the programmers in the world. In fact, it is far better when we don't have all the programmers in the world using the same tools we need at diverse ecosystem of tools and frameworks and programming languages to come up with and explore the best ideas, but we do need some base.

So, I was really thrilled to see that we reached that escape velocity with Ruby on Rails very quickly. In fact, dramatically fast, I'd say, compared to most of the other stuff that had been launched around at the same time and within, I think, less than a year, we had such a lively community to which that the ongoing success of the Ruby on Rails framework and people using it was virtually guaranteed. That's the interesting thing about technology. We talk about how fast it moves, but also in many ways, it doesn't, there's still COBOL programmers around, there's still J2E programmers around, there's programming and programs around from every single major programming environment that has reached escape velocity.

So, I think that's a very worthy goal without getting sucked into it, that everything needs to be a popularity contest or that you're devastated when you're no longer at the top of the hype pyramid, because that stuff also doesn't last forever. I mean, the thing with Ruby on Rails is that we've been around for 20 years now. We've been through multiple cycles of up and down in enthusiasm and hype, but because of that escape velocity being reached early on, it hasn't really necessarily had the greatest impact in terms of, is this something you can use to build wonderful, great modern web applications?

Rashmi Mohan: Kudos to you for even recognizing that, because I think a lot of times as engineers, we feel like our code or our work should speak for itself, recognizing the fact that you need momentum for something like this to add value, not necessarily like you said, for the popularity or for the fact that it is the adoption numbers, but really just in terms of, to enhance the product itself, it really does need to have that kind of momentum. I'm actually more curious as to ... I know that your background, you did a business background, if I'm not mistaken. Right? It was your early education in computing as well?

David Heinemeie...: No. I've been interested in computers as a general hobby since I was a kid, I got my first computer, an Amstrad. What is it? 464 or is it 646? I forgot, when I was six years old, and I tried many times over the years to learn programming because I was interested in computer games and video games and making them. I never really succeeded in that, I didn't truly learn how to program until I was almost 20 years old, and that was really due to the internet. Then around the same time I enrolled in Copenhagen Business School for a joint degree in ... they called it computer science, but it was more information technology and then business.

So, it was a joint degree where we focused both on how to build information systems and learning a little bit of programming, and also learning the basics of business administration, how to run a successful business, marketing was obviously a part of that. I've always had those two aspects of my career that I was interested in technology, I was interested in building the technology, but I was also interested in making a business out of that technology and I was interested in marketing that technology.

So I think that kind of joint degree where you're merging the lessons of business with the lessons of, if not outright computer science, then at least information technology is a pretty good pairing.

Rashmi Mohan: It certainly is, and it seems like, I mean, you started your work on your next flagship innovation Basecamp in 2004, and it was a radical and unique idea for its time. Would you be able to tell us a little bit more about how that came about, how did that opportunity arise, what made you take it?

David Heinemeie...: Yes. So when we started working on Basecamp, we weren't even intending this to be a product. At the time I was working with the company which was called 37signals back then, and we were working for clients as a client services company, 37signals would do design and I as an individual contractor would do programming at the time in PHP.

And we quickly realized that if you're trying to make a substantial project happen, particularly with clients over email alone, it is bound to become a mess. It's bound to be that something is dropped on the floor, something slips through the cracks, and all of a sudden you either look like a fool or even worse, you miss a deliverable or you make the client mad and we thought, "Do you know what? There's got to be a better way, there's got to be a better way than trying to do all of this over email and sending versions of the files and so and so forth."

So we thought, "Hey, what we're doing here is creating web public, couldn't we just create a web application that made this a little easier, one place to gather all the loose threads of this project, one place to put all the to-dos of the project that we have to deal with, all the files and all the messages and so forth?" So we started working on that actually in 2003, and within a few months, I think actually within a month and a half, we had like a basic prototype that we started using with our own clients.

And we thought, "Do you know what? This is pretty good, we've only been working on this for a month and a half part-time and it's already clear that this is so much better than trying to organize projects exclusively over email." And then it didn't take much longer than that for some friends in the industry to get wind of what we were doing. And we showed them what we had and they were like, "Can I buy this?" And we thought, "Well, sure, let's turn this into a product." So about halfway through the development cycle, we decided to turn this into a product and spend another few months polishing up the first version, and released it in February of 2004.

And when we released it, it was originally conceived as, "Hey, this would be a nice little side shift for us. It'd be a nice addition to the revenue coming in from consulting and may be if we could just make, I don't know, $4,000 a month off this in a year from now, that would be great. It could pay for essentially one full-time person to work on it." Well, we blew through that target in, I think, two weeks. And that's when we realized, "Oh, there's something here. There's something more than just a side project."

But then we spent the entire next year continuing in this dual mode, still doing client work and improving and enhancing and evangelizing Basecamp, and in about a year into it, we realized, "You know what? As a business, it's solid enough now that it can support the meager salaries of four people." So we switched over and became a full fledged software company at that point in 2005, which was also the year I packed my bags and moved from Copenhagen, Denmark to Chicago, Illinois, and took the next step in that adventure.

Rashmi Mohan: That's an amazing story, David. I was also sensing a theme here, which is, sounds like, products that solve your biggest problems are the ones that eventually make it to market. So, is that your strategy? Do you approach this with like, What is my biggest problem? How do I solve it? And if it's a problem for me, then maybe there's 10 other people that have the same problem."

David Heinemeie...: That is exactly our method and exclusively so. I think making software for other people where you're trying to do something on their behalf is very, very difficult, and I have the greatest respect for people who try to do that, but that is not me, it is not what we do at Basecamp. We make products for ourselves. When we are our own customer, it is very easy to discern whether what we're building is good or not, when the metric of good is, do we like it? Would we have paid for it?

So, that goes through my entire approach to building both open source software and products that trying to build it on behalf of other people, it's very difficult and I'm not good at that. But I am quite good at building software both projects and products that solve my problems. And exactly, as you say, the realization is, we are not unique. If we come up with a project management platform that we really like to use, you know what, probably a pretty good bet that there are others just like us that will enjoy it too. If we come up with email service that we really like to use as the latest thing we've launched with Hey.com, chances are good there are other people who just like us would like an alternative to Gmail.

And we follow that methodology if you want for two decades, and it's not like you can know up front, just because I'm building something for myself that I would really want that it's going to be this big blowout success. We built plenty of things over the years that we wanted that weren't a big success as say

Basecamp or Hey, but I think it's a pretty good method or less likely in my opinion, to end up with something that's just totally off the mark.

And of course, just because you build something for yourself, doesn't mean you don't listen. What we build for ourselves first and foremost is V1. The first version of the software is more or less exclusively built for us, and we try in fact not to listen too much to customers before there's something for them to react to. That's the old saying of, if you ask customers what they want, they want a faster horse, they don't want a car, they conceive of a car in that moment, they can conceive of a faster horse.

I've never been interested in building faster horses, I've been interested in building new things that do novel things that people hadn't even thought about that they wanted. If you to take, Hey, our new email service, this idea of the screener were all emails that are sent to you that are sent from people you haven't already said I want to hear from that person, by default does not go into your inbox. That is a weird feature in some ways, it's something I've never heard anyone outright request, and now it's one of the fundamental pillars of the Hey service. This is something they didn't know they wanted, they try it and they go, "Oh, I didn't know how I could live without this before."

Rashmi Mohan: I think what you say makes so much sense because I also think that there are certain metrics I think that you can absolutely blowout, especially when you're building it for yourself and also using it. I mean, we used to often use the term, eat your own dog food. Really, I think the bar on quality is so high. Not that you don't have a high quality bar for customers, but really if it's not something that you would use and find easy to use, then it's very hard to justify why you could charge somebody money for it.

So yeah, now I completely hear what you're saying, but it's so radically different, David, than how most businesses run. Right? I mean, if you think about it, I've tried my hand at entrepreneurship and I remember at one point, even talking to other folks who were starting up, the one thing that came up was yes, the problem is real for you and the 10 other people in your circles who maybe have a very similar view of life or have the same opportunities and the privileges that you do, but really, how do you "scale" from there? Right? How do you make this something that lots of people will want? I know your approach to entrepreneur is different. I heard a lot of different ideas when I was listening to some of the other talks that you've given. I'd love to hear more about that.

David Heinemeie...: I think one of the fundamental differences in our approach as Basecamp over the two decades that we've been building the business, is that we aren't interested in being the best in the grand sense of the most market share or the biggest company or the highest head count or the most revenue or any of these other traditional business metrics that when you just look at business as an optimization problem, and you are, "I want to be the best in email." That to a lot of people means the biggest or perhaps the most profitable, or at least the one with the most market share.

And that definition of best never really appealed to me, and I think that that is a relatively radical approach as a commercial capitalist company. Then we're not trying to squeeze every last drop out of the ... I was going to say onion, but I guess it's a lemon, that we are building something that we want to be proud of, and we have confidence that if we do that, we can build a good enough business around that.

I think that's what we've proven to my satisfaction, particularly over these past two decades, is that you can totally do that. Yeah, we didn't turn into Gmail or something else, and so what? We ran and have run and continue to run a wonderful place that makes great software, that the people who are making it really enjoy making, because we have such a focus on the mastery of the craft that is not just about making a product that someone will buy, it's also about making a product that we will be proud of. This idea that the back of the box needs to be as pretty as upfront is an engineering ethos.

I remember hearing Steve Jobs and Wozniak talk about laying out the circuit board inside the early max and making sure that the circuit actually looked good, even though no one could see it. And I thought that's just such an inspiring notion, and that's a huge reason as to why I'm still interested in doing this work after 20 years, that it's not just about how can we make the most money, but it's just as much about how can we increase our own mastery and increase the satisfaction that there is, and just building really good stuff, and then accepting the trade off that sometimes that gets in the way of the business part of it.

We've put other constraints on ourselves over the years in terms of some of our ethics of how we want to distribute our software, for example, we've had a long standing feud and fight with the big App Store monopolist like Apple and Google about their monopolist behavior and abuses. We had a fight when we launched Hey.com last year, almost got shut down by Apple before we'd barely gotten off the ground. And many of those fights that we picked, again, they don't really make sense in a pure business analysis.

If you were going to make the cost benefit analysis on these things, you'd go like, "Do you know what? The risks are too high here? Let's not do that, let's just bow our head and do as we're told by these platforms that have more power, and again, because we built this business in such a way that we answer to essentially ourselves, Jason and I, we answer to our customers, we answer to our employees, but we don't answer to investors per se, for example, there aren't anyone who can come in and simply dictate, "Hey, you're not doing this fast enough, you're not maximizing this enough. And then say, you can't do it." I feel like we have an obligation to do some of the things that no one would ever get permission to do.

Rashmi Mohan: I love your ideas on entrepreneurship and how you break it down basically to make it more accessible, right, reachable to a larger part of the population. Especially like what you were just talking about, you started your business without raising external capital that probably helps in keeping the pressure down in terms of catering to a certain market, but really building something for the joy of it, and like you said, the mastery of it.

Any other, either myths you'd like to dispel about starting your own business or any advice that you might give, because we have a lot of listeners also who are earlier in their careers and oftentimes many of us think about, "Yeah, I have a great idea, I'd love to go and start it." But there are barriers that hold us back and maybe even just looking at the rate of success, being so low and success also in quotes. I think

David Heinemeie...: That's perhaps a good place to start. What is your definition of success? If your definition success is, "Can I raise capital? Can I satisfy that capital to the point that I get to become a unicorn or whatever we call it these days, the measure of success?" Yeah, the answer are very low. That's just facts.

But there are many other ways to build businesses. Our own example of building Basecamp on the side, not taking any risk at all, funding it through consulting revenues as a side project and then not switching over until it was doing well enough to pay our salaries, is a way of approaching entrepreneurship with so much less risk as so much higher odds that yeah, outcome might not be as spectacular as the one in a thousand or one in a 10,000 hits that come out of that venture capital pipeline. But so what?

I think many of the greatest joys of starting your own business, they aren't about that, they're about being in charge of your own destiny. And if you're in charge of your own destiny with four employees or eight employees or 12 employees, in many ways, that is far better than not being in charge of your own destiny, even if you get to below $30 million or a hundred million dollars, adding as many people as fast as you possibly can, that all comes from interrogating. What are we trying to do here? Why are you starting a business? What is your end goal?

And what we've tried to do over the past two decades is to put out an alternate vision of what that could be because in tech, it is such a monoculture when it comes to, how do you start a business? How are you supposed to do it? What does success look like? It's incredibly dominated by this venture capital San Francisco Bay Area set of role models and ideals. And there are a set of role models and ideals that people around the world are looking at, even if they don't really apply to them, because those conditions are not available there. But do you have 10 hours a week to pour into something?

A lot of people could say yes to that, not everyone, but a lot of people could. That's how much time I spent building Basecamp on the side while going to school and doing some consulting as well, all within 40 hour work week. So, you can start slow, and we did too, even when Basecamp was "success" and we could pay our salaries. We were after I think it took maybe four, five years before we were even, what, eight or nine people. We grew very slowly, and within our scope of low odds, how could we advance what we're doing, get better at what we're doing, provide better products, but not take on all this risk?

And I think if you look at that as an option, it becomes a whole lot less daunting. When you dismantle the idea that entrepreneurship is about taking maximum risk. You hear this all the time. "Oh, I was running up five credit cards and I was getting millions of dollars from VCs and thinking about mortgaging my house." Okay, yeah. That does not sound like something I think most people would look at and go like, "Yeah, that's a reasonable amount of risk. I'd love to do that." So, putting out this alternate vision that there are other ways to start businesses far less risky ways of starting a business, and then also putting out the testimony that if you start a business that way, and you have a modicum of success, you can have a wonderful life in many ways, a better life.

One of the things that continuously boggles my mind is how American entrepreneurs in particular are so proud of bragging of how long hours they work. That really came to head for me when Marissa Mayer in an interview, as the CEO of Yahoo was bragging about how she worked 120 hours several times in her career. And she could do that because she was strategic with her bathroom breaks. And you hear those things, and clearly for some people that sounds inspirational, for me, it sounds absolutely dystopian.

There are slaves in the history of civilization who've had it better than what that appears to sound like. At least if it's factually true, which of course it isn't in the many, many ways. People who brag about working endless hours often don't actually work that hard for that many hours, but they think of themselves as doing so, and I think we just need to dismantle that that's good, that that's a bar that you have to clear that unless you're willing to work 80 or a 100 hour weeks, you don't have what it takes. Bullshit.

Rashmi Mohan: That's a very, very refreshing perspective, David, I know you're a prolific author of business books like Rework and doesn't have to be crazy. I was looking at some of the highlights of the books, and I was very fascinated by the philosophies that you share. Right? Namely, when that said, you need less than you think, let your customers outgrow you, how to grow a sustainable and calm business. I thought they were ... I mean, there were so many others, I just picked a handful that really, really stood out.

I mean, how do you think these have helped you in your career and how does one incorporate these into our lives? Because you're right, I think we're in some cases, especially now with the pandemic and all in our work and life, just blending into this blur, I think it would be amazing if some of us started to think about these a little bit more in terms of not impacting our careers and our goals, but actually weaving it into our lives.

David Heinemeie...: Yeah, that's part of the reason why we wrote it all down. These are hard one in the sense of, they took a long time to arrive at these conclusions, because many of them are derived from first principles. Instead of thinking like, "Oh, how is it that you're supposed to start a business? And how is it that you're supposed to run a business?" Jason and I questioned a lot of the fundamental assumptions that are ingrained in the business ideology and particularly the tech business ideology of the United States. And we went like, "Do you know what? Does this make sense? Does this carry its weight? Is this helpful to us?"

And for many of those examinations, the answer was a resounding no, that the way most people or at least many people do things was not something we thought was a good thing to copy. So, we've just been working our way through and iterating upon and sometimes changing our mind about some of these topics, but always trying to address them from like, "Let's not assume that just what is has to be, we could come up with something else." That is one of the wonderful things about starting and operating a company is that you get to set new rules. If you don't like the rules that came in in the book of business that you've been handed at other companies.

And I think for me, in particular, I spent about four years working for other people in the late nineties, early 2000s, and it was a wonderful education in many way, a far more valuable education than the one I got from the Copenhagen Business School about how not to run a business. I took a lot of fantastic negative lessons away from that experience, and I swore that if I got the chance to set the agenda in my own business one day, I was going to do things differently. I was going to approach things differently. I think it's safe to say that we've done just that. And part of the satisfying thing of doing that is then to tell other people about it, at least give them a seed of inspiration.

Many of our books contain, I don't know, 60, 70, 80 essays, and it's not realistic that anyone anywhere can take all of those ideas and they can put them into action tomorrow. But you know what? Maybe there's 2, 3, 4, 5 of them that they can, and they can see that, "Oh wow. That made a big difference to me." For example, we have an essay in our 2010 book, Rework, that talks about meetings being toxic. I've heard from so many people who thought after reading that essay that like, "Oh, I hadn't even thought about that. I knew there was something keeping me from doing my best work, I didn't know that it was necessarily to be pinned on these meetings that kept puncturing my day, just that I would only get two hours here and an hour and a half there, but now it makes a whole lot more sense.

And finding a way to get these long stretches of uninterrupted time has really allowed me to flourish. And yes, because this is me. This is why we're sharing it. But we found these things that have worked for us, and I go like, "Do you know what? It'd be a damn shame if we kept that a secret." That's the same thing I felt about Ruby. When I started working on Rails and had the intention of releasing it, a lot of it was like, "Do you know what? This can't remain a secret. Ruby is too good. We need more people to know about this." So, evangelism of good ideas and good practices and good tools is something that is near and dear to my heart in a way that is almost reflective. I can't not do it.

Rashmi Mohan: And that's such an amazing, amazing feat. I mean, where obviously, I would say, the world is a beneficiary of that. And also what you were talking about is, there is no cookie cutter way of running your own business. And like you said, questioning some of the tenants that have always been around is probably the most liberating fact that you can actually take advantage of if you're starting up your own business.

One of the other things I know that you've spent significant parts of your career being a remote worker, and now that the whole world has come to this party, I know you have a book from 2013 about why remote working is the future. Do you feel like the last 18 or so months where all of us have been remote workers have just proven the ideas that you've observed over your career or do you feel that something else has changed or you had to change something to accommodate now the whole world being remote?

David Heinemeie...: Well, I'd say it's many things at once. The thing to start with perhaps is that, yes, we've all done remote work, but we've all done remote work during a pandemic, which is substantially different from doing remote work, not during a pandemic, and that's just an important market to have in mind, because what's been really interesting is that we've run this grand social experiment with literally hundreds of millions if not billions of people at the same time, and it's the kind of research that you would never otherwise have had the ability to do.

And the fact that we haven't had the ability to do that at such a vast scale, allowed a bunch of misconceptions to linger, a bunch of opposition to remote work as though it was even a question of whether it's possible that companies can operate this way. And now we have the empirical data, of course, it's possible for companies to operate this way. And this is what we've been screaming about for the past, well, more or less two decades, but culminating in that book from 2013, where we tried to make the broad argument outside a forcing agent, like a pandemic, that more companies should embrace remote work because it is great for a lot of employees who get the freedom to live with far greater flexibility in their day and where they want to live for the company because they get access to much better talent.

They don't have to restrict themselves to who can commute into their office, and these advantages were so great that I thought why aren't more people doing it? And it was just a testament to how set in our ways we can be and how habits are really hard to change. And all the justifications people will come up with as to why actually they shouldn't change anything at all. So here alone comes the pandemic and it blows all that away. It blows the idea that, "Oh, unless you're in front of a whiteboard, you can't be creative." Does anyone seriously think that no creativity has happened in the last year and a half? Of course, not. Right?

Like now that's actually a ludicrous proposition to advance, but it wasn't so three years ago, very serious people were arguing that you could not do creative work, you could not build or sustain a culture, you could not do all these things if you're working remotely, this was simply not possible, ignoring of course, the fact that plenty of companies had done just that and the entire open source community has done that for, not just two decades, but three or four. So in a way, it's nice that this has happened under unfortunate circumstances. I think it's fair to say. But when we come out, hopefully on the other side of this, the argument will be settled in a way that we could never have settled it without it.

Now, that doesn't mean that everyone has loved remote work during the pandemic, because there's a lot not to love about working remote during a pandemic, and I think there're subset of people who will go, "Absolutely, I want nothing to do with remote work, I want to get back into the office, 9:00 to 5:00, five days a week. But I think that's actually a relatively small minority, as least as all the surveys that I've seen, that the majority of people who've tried remote work, maybe not all of them want to do it all the time, but most people want to do at least some of the time. And that's the other thing.

So, we've been doing remote work for 20 years at Basecamp. And when we say remote work, that's not the same thing as I work from home all the time. You can work remotely from a colocation office, you can work remotely from a coffee shop. You can work remotely while still being around other people without being in the same city or commuting into a shared office. So I think that that's the fundamental liberation that isn't remote work, that you can have people live and work from wherever they want to. And you can collaborate across continents, across countries, at Basecamp, we have people all over the world, mostly in the US, Canada and Europe, but still, that this is possible.

So, satisfying to see that this argument is settled, clearly there are people who still need some help figuring out how to do it well. I mean, they've been forced to do it, it doesn't mean they've been able to do it well, that's one of the reasons we wrote the book, was to encapsulate some of the lessons we had taken away from working like that for a very long time. It's not just about installing Zoom and Slack and then thinking, "Oh, we're ready to do remote." There's a bit more to it than that.

Rashmi Mohan: Absolutely. I think the point that you bring up, which is really, we've hired people, we've built relationships, we've build products over these past 18 months. So we have been able to do a lot of the things that we thought we couldn't earlier, but really the core of it is to give people the flexibility, the folks that want to go in, they should have that option, but there are a whole bunch of us who may not want to commute or want to still, and can still be effective being not always colocated with our colleagues.

And I'm hoping that post pandemic when we don't have all of the other constraints and strains on our health and other systems, we will continue to adopt this. I pardon the cliche, but to switch gears into another passion of yours, it's not often that I speak to a guest who has a parallel interest in something as exciting as yours. You are a race car driver, please take us through that journey.

David Heinemeie...: So, race car driving for me started quite late in life, I didn't get my driver's license until I was 25. I grew up in Copenhagen city of bikes where not only do you not need a car, they're also very expensive and I could neither afford one nor had a particular interest in one. But I always liked racing video games. And when I got my driver's license at 25, that was the same year I moved to the US, and two years later, a friend took me to a local racetrack at 27. I sat in and drove a race car for the first time and felt, whoa, this is a portal to another world, or it is a portal to a certain state of mind, perhaps I should say, to the state of mind of flow, which is this concept of totally being engrossed in an activity to the point that you lose the sense of place and time.

Some of my favorite moments in programming have had and involved those states of flow where you just you're so engrossed in the program activity, you look up and you go like, "Wow, how did two hours just pass by? This seemed like 10 minutes." And in race car driving, I found, again, pardon the pun a way to simply just turn the key and then boom, you're there. Where with programming, it was far more, if not accidental, then difficult to trigger reliably, but with risk car driving, I could very reliably trigger that state of flow. I simply had to get into a race car drive very fast, and then I would think of nothing else.

And that is just such a moment of bliss that not only is just something I've experienced, this is a well studied phenomenon that if you ask people, what are their happiest moments, they're usually giving your accounts of flow. Mihaly Csikszentmihalyi, I'm totally butchering his last name. I should really learn to pronounce it, wrote a book, essentially called just Flow. The document is this phenomenon that happiness is tied to these states of flow where you're trying to improve ever so slightly and move ever so slightly outside of your comfort zone, but it's still within reach. And that just produces these moments of bliss.

So, with race car driving, I found that I found a hobby that felt in many ways like an engineering problem, like an optimization problem, you're going around a racetrack and you get a verdict about every minute and a half to two minutes about whether you improved or not. And then you get, however many corners are under track, let's call them 15, you get 15 new chances to try something different. And that sense of tinkering to find the optimal most efficient and fastest line or approach to a corner or a track reminded me a lot about programming. And I thought that this was some of these aspects that in programming I thought was so interesting, and then I found that in totally different domain, a former physical domain.

And then it was also just a lot of fun. I mean, I challenge anyone to sit in a race car, drive around a track and not have a good time. The vast majority of people I've taken out on a track, they've had a good time. It's an overwhelming experience in some ways, particularly when you do it first few times, I would take people out on the track after I'd learnt how to drive properly, and I thought I was just doing a Sunday stroll and they were just petrified as though we were going to slide off and crash. There's just something very visceral about going very quickly in a metal box where if you do make a catastrophic mistake, it can turn out quite catastrophically, but that level of risk and danger is part of the [PL 00:41:38]

So, it's something I've pursued for a good 12 years or so. I also treated it as a learning exercise. By the time I started driving race cars, I had some time to think about how to learn and how to get better, primarily through programming and business, learning those crafts and getting better at them and realizing, "Hey, there are some patterns here that you can employ and you can employ them in any field." And then trying to employ those patterns with race car driving and seeing, "Oh, actually it works pretty well over here as well." That's the story of race car driving.

I am also though, to some extent kind of winding it down a little bit. I also believe in looking up every now and then and thinking, "Do you know what? Am I still doing the thing I most want to do?" And for me, part of the problem with race car driving as of late, is that it's no longer quite the turn key that it used to be. I don't get access to that flow state quite as easily, and it's not as repeatable. I've been in plenty of race cars in the last few years where I just couldn't access that. I was thinking about all sorts of other things when I was driving the car.

Do you know what? Then to me, it's not really worth it, the rush was this portal of flow. Anyway, sometimes you just need some time away too, and you get back to it and it's all good again, or other times it's just time to try something else.

Rashmi Mohan: Absolutely. I mean, it sounds exhilarating the way you describe it, and I love how you drew parallels between, in your case, the two worlds that meant so much to you, whether that was programming or race car driving. This has been an incredibly, incredibly exciting conversation, David. For our final bite, what are you most excited about, either in the field of software or racing or anything else? What is holding your passion today over the next say five years?

David Heinemeie...: Well, five years is too long of horizon for me, but right now I am very excited about some of the fundamental changes that's happening on the web. I feel like we've gone through at least a decade of a bit of a monoculture with JavaScript, if you wanted to build things for the web, and we're finally coming out on the other side of that, in part, because browsers have gotten a lot better. I'm putting a lot of those fundamental advances to good use in service of Rails 7, the next big release of the framework.

I've been working a bunch of things around these fundamental advances and how we can use those advances to radically simplify how we're going about our programming work. I've been building web applications for 25 years now, and in substantial ways, we've managed to regress over that time, where things just got more complicated for very little benefit. And sometimes it seems like there's nothing you can do to arrest that, and that's just simply how it is and you just have to get on board and there's just more and more and more and more to learn, and there's nothing we can do.

But I think that's a fatalistic approach to it. I think there are these moments where we can apply conceptual compression, where we can take these great big concepts that used to require total justification by specialists, and somehow we can just hit compress on them because we find a breakthrough in either our understanding of the problem or the technology we're applying to it. And then it's no longer a thing people need to worry about. I live for those moments. I live for the moments of conceptual compression.

And I think we are in a wonderful phase right now with conceptual compression when it comes to the web where we can go a different path where it's just not more and more complicated, but actually we're like, "Hey, we could turn back the clock a bit, we could make it as easy to build web application in 2021, as it was maybe in 2007, again, without romanticizing the past needlessly so, there were plenty of things that were wrong or difficult or whatever back then too." But just looking at the problem space that we have, seeing the accumulation of accidental complexity, realizing there's something we can do about that and compressing it just that we can get back to the essential complexity that is inherent in writing good software.

Don't we all want to do that work on problems that actually matter not fighting without tools, not fighting with all the indirection? Of course, we do. That has as my attention right now, Rails 7 is where my mind is and has been for a bit here. I hope we get it out this year, and some of these fundamental conceptual compressions can spread to other ecosystems as well as many of the conceptual compressions we've championed over the years through Rails have done. That's satisfying to see ideas spread outside of your own ecosystem.

And then for me to put that to use, building more great stuff at Basecamp Hey, I think we've only just gotten started with Hey.com. Email is a very big problem. And 25 years later here after I started using email, it's still a difficult problem, and we shouldn't just give our up and say like the best we could come up with was whatever Google launched in 2004.

Rashmi Mohan: Thank you. Thank you for opening our minds to all the fascinating ideas on the opportunities available for those of us that want to start businesses, that don't always have to play by the rules set many, many years before our us. Thank you for taking the time to speak with us at ACM ByteCast, David.

David Heinemeie...: It's been my pleasure. Thanks for having me on.

Rashmi Mohan: ACM ByteCast is a production of the Association for Computing Machinery's Practitioners Board. To learn more about ACM and its activities, visit acm.org. For more information about this and other episodes, please visit our website at learning.acm.org/ByteCast, that's learning.acm.org/B-Y-T-E-C-A-S-T.