Stanford eCorner

Unlearn Your MBA (Entire Talk)

David Heinemeier Hansson, *37signals*

January 20, 2010

Video URL: http://ecorner.stanford.edu/videos/2351/Unlearn-Your-MBA-Entire-Talk

David Heineimeier Hansson, the creator of Ruby on Rails and partner at 37signals in Chicago, says that planning is guessing, and for a start-up, the focus must be on today and not on tomorrow. He argues that constraints--fiscal, temporal, or otherwise--drive innovation and effective problem-solving. The most important thing, Hansson believes, is to make a dent in the universe with your company.

Transcript

David Heinemeier Hansson is the person who created Ruby on Rails and he's a serial global entrepreneur. You've had a chance to look at his bio on the Web and he'll tell you more about himself. But right now, we'd like to welcome. David, here at Stanford, we're really delighted to have you with us David. Thank you. Thank you. So, just a brief bio on myself. I'm David Heinemeier Hansson, I'm originally from Denmark. I moved to the US about four years ago and I'm a partner in a software company called 37signals. 37signals makes simple collaboration tools.

We've been around for 10 years. And even though we're often lumped in with the rest of the Web 2.0 world, there's sort of a quirk in our setup is that, we've actually been profitable for the duration of the company's lifespan. So, we're a little bit of a misfit at times, but that's sort of in part what I want to talk about here. The other part is, I'm the creator for a Web framework called Ruby on Rails that I've heard some of you guys have been using. We use that for all our products. It's free. It's open source and it's gotten quite popular and we give it away for free. And so, that's sort of the tech side of what I do, but tonight I'm mostly going to talk about the business side of what I do in terms of 37signals. And I want to talk about just, I think I have three or four main topics of things I have observations on currently. And we're going to have a lengthy duration at the end for Q&A

So, do get your questions ready for that. But first, I sort of want to set the stage for what we're going to talk about. Right now of course, all the headlines are bad news. There's a terrible economy. Unemployment is record high. Nobody is hiring and corporations are reeling and how are you going to get a job? And all of these sort of concerns, I would imagine would be something, you'd be thinking about if you're still in college. But I think this sort of, the bad news part of it doesn't need to be so bad. In fact, I think the bad news part is probably going to be the best thing that ever happened to you. I think right now is by far the most opportune time to come out of college. One of the smaller key benefits I think is that you don't have to write a bull resume.

You don't have to come up with all the standard exaggerations, half-thruths, and embellishments that college students usually have to put on their resume. when they come up with no work experience and want to join the workforce. And this is not only for resumes targeted at getting a job. I treat this as more of a broad sense of a resume, as also being stuff like business plans which we'll discuss a little later. I think that's actually excellent. I think it's excellent that you come out in a market where the first thing you have to do is not try to impress another professor. Because I think people who have been in college, I was in Copenhagen Business School for three years getting my bachelors, come out with sort of a warped sense of what matters. And what matters when you come out and you want to start a business is definitely not the professor. And we'll talk a little bit about that. What does matter though I think and what's really important about coming out now is that because you can't just get a standard, boring, middle-management job at some consultancy somewhere.

You actually have to go out and make something. You have to put a dent in the universe. So, you get this unique opportunity because you don't have to safe alternative but there's nothing else right now for you to do really than to make a splash. Start a new business on your own. Do something for yourself. Don't just become another management consultant at Accenture or something. OK. So, that's sort of the stage. So, you want to make a dent in the universe. And you're coming out and how are you going to do that? Just because now is a good time to start a business, doesn't mean it's easy to start a business.

In fact, it's always hard. There's not going to be any surefire way to instant success. There's not going to be any get rich quick schemes that's going to work for you. But before you can even get started, I think the most important thing for you to realize is that you have to unlearn your MBA. And I'm treating MBA here as a sort of a general grab bag for business school management theories and I think that's probably as kind as I can put it. I spent three years at Copenhagen Business School. I will probably say that, according to my estimation 96.7% of the time was completely wasted. It has nothing to do with what I actually do today and it has no impact on what I actually work with everyday. In fact, I came out slightly damaged. I came out with a head that been soaking in management theory for three years and it was actually a little off.

It was not very well-suited for the real world of just building a product, pleasing customers and making profits as a business because that's really not what you learn. And you just have to just readjust and recalibrate when you come out of school to that reality that nobody cares about a 20 page report on five forces. It just doesn't matter. There's none of your customers that's going to think, "Oh. Well, did you do your five forces for this setup? No. All right, then we're not going to buy your product." It just doesn't matter. So, all of these tools that you've learned are only for you. They're not going to impress anybody else when you start your own business. And what you learn is that when you're starting your own business and all businesses start small, is that none of it is relevant. When you have two people, three people, five people, nobody is going to do all the heavy-handed management theory on it.

It's just irrelevant because it's not targeted to your size. That might be fine for a management retreat at IBM's second level management somewhere in Florida. Like that's going to impress the rest of the group I'm sure, but it's not going to impress any of your customers. So, you have to first get out of that rut. And I suggest that you start in the key spot which is your writing. When you come out of business school, your writing sucks because your writing has been targeted at pleasing professors who want to see you demonstrate theories and write a certain amount of work, like papers are 20 pages or how many characters that is. That is, at least how it was for me at Copenhagen Business School. We have certain requirements. There's no best, special bonus point for writing a one page paper. There is no.

But that simplicity, that approach to just writing exactly what you need and nothing more is exactly what you need to succeed out there. Not what you need to succeed in here. And I have funny example of how bad things can go when your brain has become all mushy, with all this MBA bull. This is Domino CEO David Brandon talking a few days ago, "The weakness in our value chain with the customer was really in our core product" What? What does that even mean? Well, I tried to translate that into human. And what I came up with is, "Well, let's be honest. Our pizza used to suck. I'm sorry. I swear the new ones will be better." That's the kind of language that you can have when you're small business that's not trying to impress anybody but your customers. This is absolutely a key thing. I'd say the most important technique, the most important competitive advantage we have at 37signals is good human writing.

And you just don't have good human writing when you say [stuff] like this. And nobody's going to take you seriously because it's just meaningless stuff. So, try to get into that frame of mind that you're speaking to humans and nobody is going to be impressed by big words. That's definitely a key thing and the process of unlearning your MBA. I tell you the second biggest thing is probably realizing that planning is guessing. And all of those techniques that you've been taught are mostly about planning. Long term planning, strategic planning, tactical planning. All sorts of planning that leads to really, really funny things like start-ups that have five-year growth plans. What? You don't even have anything yet and you're theorizing about how it's going to look in year five? How do you know? You don't. That sort of long term planning is fine again for big stable corporations.

Like if you're running sector five of McDonald's in Northern Illinois, I'm sure you can apply a five year plan to how many burgers you need in Year 3 because it's a pretty stable business. When you're starting your own business, especially if you're trying to do anything new, if you're trying to do anything in a new industry or if you're trying to tear up existing industries, or come up with something else that's not already there, you have no clue. And the sooner you realize that the better because planning is not just guessing, it's harmful guessing because it's a waste of time. All the time you spent doing your five year plans, you could worry about tomorrow. That's usually the timeline you need to worry about. 37signals has been around for 10 years. Do you know how long we usually worry about? Two weeks. Sometimes when it gets really crazy, we worry about 2 months. We absolutely do not worry about what next year is going to look like because it has very little impact on what we do today. So, realize that most planning is guessing.

And that's putting it 'friendly'. The guessing usually leads to a whole lot of other crap around it that comes up and wastes your time and sucks your energy away from worrying about what you really need to be worrying about which is today and tomorrow and next week. All right. And final thing of that is realizing that most decisions in a small business are incredibly temporary. You are not Intel. You are not building a billion dollar factory in India that's going to be done in five years. Those guys, that's not a temporary decision. You better be right about the market and whatever they need to worry about, to build a factory in five years somewhere that's going to cost a billion dollars. When you're starting out on your own, with your own ideas, with your own sweat and blood, the vast majority of all decisions are temporary. And what matters most is starting.

It doesn't really matter what the decision is as long as there is one. Just get going. OK. So, you're probably going to ignore all of these and you're not going to be unlearning your MBA. So, let's take that as the assumption. And I'll try to save you from one other thing which is venture capital. Venture capital is a time bomb. And realize first of all I'm talking about here is I'm talking from my experiences. I build software. Again, I don't build factories in India that cost a billion dollars.

I don't have all this infrastructure cost. I don't need any of all that ramp up. I'm talking about software, Web 2.0, blah blah blah blah. In this industry, venture capital is absolutely a time bomb and it's probably the most harmful thing you can do to a new business, is to accept venture capital upfront and I'll tell you why. At first, it sounds like it's a really good idea. If I didn't take venture capital, well, I wouldn't have any money and we would only be us three people and we would actually like, "How are we going to live?" Good question. Now, if you take venture capital, you got millions. You probably got a runway of years. You don't have to worry about tomorrow. You can worry about three years from now because tomorrow doesn't really matter.

You have $5 million in the bank. Who cares? Even better. It's not your money. You get to spend other people's money for a sustained period of time and the worst that can happen is that you waste five years of your life. That's pretty bad. Wasting five years on your life on something that doesn't work because the time bomb blew up. That's a lot worse than worrying about how you're going to make rent next week, in my mind. So, the tradeoff is sort of, are you going to have the pain now when you actually have to prove that your idea is good enough that it can be profitable and that it can be self sustainable? Or are you going to have the pain five years from now when you realize that the time bomb blew up. Now, the venture capitals own your company and they've just pushed you out and you have nothing to show for five years of effort. That's sort of the worst case scenario.

But it happens a lot more often than you'd think. I've been talking to a bunch of entrepreneurs in the Web 2.0 space. A lot of them took venture capital. I have not meet a single one who wouldn't say he would do it differently if he had to start out with today. But there's not a lot of information out there about that because the problem is once you take your series A round of funding, you're an addict. You're going to need that next fix. You're going to need series B right now. "We need some bridge capital to get it over just to the other side". And when you're in that situation and you're an addict, it's a pretty bad idea to piss off your dealer by saying, "I should never have gotten into this in the first place. I'm so sorry." No, doesn't really work.

So, big warning label on venture capital. And I was sort of hinting at that before. I think the key problem with venture capital is first of all, you're spending other people's money. When you're spending other people's money, you care much, much less about said money. When you're spending your own money, you care very deeply about this. When we're hiring a new person at 37signals, I'm spending my money. I'm going to get a lesser dividends cheque at the end of the year. Do you think I care then about which person I'm going to hire? Do you think I'm just going to hire some bozo just to fill up the org chart or to impress somebody else, the press? I don't know. No. That's not worth 120 grand.

I guarantee you that. Secondly, when it's your own money, you want to make more of it faster which usually means that you can't just put out a product, a product that doesn't have a price. You can't just sit idly by "Oh, we got two million eyeballs after two years. It's really cool." No. Nobody cares. Do you think your banker cares? Do you think you can pay up your credit card with two million eyeballs? No, you can't. You need to have an actual product that has an actual price so you can make actual profits. And being forced to get to that point sooner is the best thing for a business. The worst thing for business is when you have no constraints, when you just have as long as the timeline as those five million will carry you. You're going to make a lot of terrible, terrible decisions.

And you're going to waste a lot of time on stuff that just doesn't matter because you don't have that sense of urgency. The urgency you get from spending your own money and earning your own money is the most powerful driving force for an entrepreneur. So, get there. Secondly or thirdly, I don't know where we're at. The VC route is that small is a beginning. Small is a stepping stone. Small is what you go through to get to the major leagues. Small is inconsequential. It's sort of a path to the billion dollar business. And everything is designed around that pipeline.

Most VCs who invest are not going to be happy with the business that throws off a million dollars a year in cash flow, if that's sort of the scope of the business. That's not interesting. They need an exit that's 5, 10, 15 times, whatever it is, preferably a hundred million plus. "All right, we'll take 20 million, whatever." The fact of the matter is a million dollars is a lot of money when it goes straight into your bank account. So, if you can build a small business that only makes you a few million dollars a year, you're doing pretty damn well. Now, what do you think your odds are of building a billion dollar business versus the odds of building a $3 million a year business? This one, the billion dollar business is about the same as going into a casino and putting it all on Red 5. "All right, roll it." Trying to get a $2 million a year business, a $1 million a year business, half a million dollar a year business, it's more akin to being a fairly skilled poker player and going to a table. You have a chance. It might still involve luck. It might still involve unknowns but at least you have a fighting chance.

You're not just relying on the buy-out lottery. You're not just relying on the fact that the Google gods will favor you and buy you for X million dollars. You're actually playing for skill and not just random timing. So, I think you're much better off trying to build a small business. And the fact is that most big businesses started as small businesses. There are a few business that go from nothing to everything in 18 months. But the problem is that there's just enough that we believe the story. There's just enough YouTubes and whatever else that gets that trajectory. Which is also just enough that all the VCs get all sharky and want get in on that action, like of course, who would want to invest $50,000 and get $200 million back? So, they try. And when they try, it sort of becomes that hit-driven business where, "All right, let's fund these 20 guys.

If just one of them makes it, awesome. The other 19, collateral damage. That's fine. If it didn't work out or blow up their time bomb will reach its durational blow up. No skin off our game." Skin off all those 19 guys game. Skin of all the game in time that they lost and they wasted and the missing dent that they didn't put into the universe. Finally, there's tragic misconception that if you start a new business by yourself, you have to be workaholic. You have to work 60, 80, 100 hours a week because that's the only way you can succeed. No. We didn't do that.

Our main product, our flagship product Basecamp was build off me spending 10 hours per week while still in school. And that is today, by far still our main revenue generator and made us millions and millions of dollars. Based off me, 10 hours per week, a few designers as well for six months. This 80 hours a week bull, because the problem is when you work 80 hours a week, you're just a rabbit. You're just running around trying to do everything you need to do to get your business going, without realizing what it is that actually matters. And usually about 5 to 10% of the effort you actually put in matters. When you put it 80 hours of really hard labor, it's the 3, 5, 10 hours of actual stuff that ends up mattering. The rest of it's just the headless chicken running around. Being a workaholic is by no means either a guarantee of success or a requirement. I think there's just a lot of misconception out there about how hard you must work.

I'm sure this was true in the industrial age when you were hammering on stuff. If you hammered for another five hours, you would produce another 10 widgets and you can sell those widgets. If you're working in stuff like software or anything else that doesn't require your personal labor in manufacturing widgets which is basically none. None of this stuff matters. The great idea matters. The less execution matters. And all of these great ideas and less execution comes from a well-rested mind. It sorts of goes into the trophy that a lot of people bear. I only sleep four hours a night. And you hear a lot of entrepreneurs bragging about how little they sleep which is about the stupidest thing you can do.

Like I'm bragging about how incompetent my performance is going to be when I can't stay awake tomorrow for this thing. Or a programmer, how many bugs I'm going to introduce in the system tomorrow? Like, what? That's a direct correlation. When you're not well rested, your mind is not working at peak performance. You're going to produce [worthless stuff]. And it's just not a factor of 10% [worthless stuff]. It's not like if you sleep half, your stuff is going to be half as good. No, it's going to be 100th as good. So, that doesn't work. The fact is, when you restrict yourself, when you put up constraints for yourself, when you say, "I'm only going to work 40 hours a week on this." That's going to be the hard limit. I'm going to work 30 hours a week or as I had when we're building Basecamp, "I'm going to work 10 hours a week." You have no choice but to underdo your competition.

You are not going to out-effort Microsoft or Google. You're not going to be able to put in more programming hours than they are. Thus, you don't go head-to-head with them because there's no need. It just doesn't work. It doesn't work to go head-to-head with somebody who has a million X more than you do. You are going to lose. So, the sooner you realize this, the sooner you realize this, you realize that the constraints are your friend because the constraints force you to do way less and different than your competition. You can't just build a me-too product on 10 hours a week. That's not going to work. We were not going to build a better Word.

That was just not going to work. We would have to pick a new category, a new type of product, something that combines something else, something that was vastly different. Different enough that we could do it in the time that we have and it would be simpler than everything else. Well, here's the wonderful thing. Consumers generally love that. The number one feedback we have, whenever we do customer service and we've been doing this for five years is that "We love how simple your product is. We love how easy it is to get started. We love the fact that it doesn't have a 300 page manual, that we have to wade through. We love the fact that there are no training classes. We love all the things that come from doing less.

We love all the things that comes from underdoing your competition, having fewer features than your competition." So, that's a good thing. And finally, realize that no matter what you do, you're not going to be an overnight success. Nobody's an overnight success. Most overnight successes you see, have been working at it for 10 years. And that's exactly how it was for us. 37signals was not an overnight success. Basecamp was not an overnight success. And then all of a sudden something blows up and it looks like these guys just got started on it. Which is why this whole thing of accelerating your growth and getting venture capital so you can go faster and quicker is just a charade. It doesn't work like that.

Companies take a long time to get to something good. So, you can either take those 5 years or 10 years or however long it is going to take for you to take to get to somewhere good at a reasonable, sustainable pace where you end up actually owning your company in the end, or you can strap that time bomb on your back and see if that's going to make you run faster and if you're going to be happier in the end. I think you're not. Thank you. Thank you David. As most of you know, I'm Steve Blank. I teach in Stanford in MS&E278 the Spirit of Entrepreneurship. It's a class that wraps around these ETL lectures and allows students to have access to great speakers like David. And actually after this session, we go back and analyze what the speaker said, where they actually came from and what were their motivations when they started a company. And today we're honored that David is going to walk over the class with us and for those you in the class will get to have David with us.

And so, we'll get to ask him the real questions there. But David just a segue. I understand the first thing you did was actually a gaming website. Was that true? Sure. That's how I originally got into programming. I used to do game reviews, computer games, console games. Were you a hardcore gamer? I was a pretty hardcore gamer, I was. And I was just interested in getting this stuff out there and going through the traditional way of getting your review published in a magazine was very hard and it relied on the graces of other people. I didn't like that too much. So, instead the Internet (this was '97) was just popping up or I was getting aware of it.

I was like, "This is a perfect spot to do that." So, I started publishing a number of gaming websites. And to do those gaming websites, I had to teach myself programming. So, that's how I gone into that. And did you think of it as a business or a passion at the time? Nope. At that time, I was still blissfully ignorant. In large part because I sort had VC money strapped on to the back of me. Not in the sense of my specific things but this was still dotcom days and I made it into an incubator who for mysterious reasons decided to pay me a princely monthly wage for writing game reviews with no absolutely idea of how they're going to make money out of this. But at the time, I didn't care. It wasn't my money. So, I got to play games and write reviews and teach myself programming.

So, it seemed like a sweet deal at the time. And you think that experience affected your view of where you are today? Absolutely. First of all, just realizing that that stuff doesn't last forever. It lasted I think 18 months and then they realized, "Hey. We don't have a clue are we going to make any money here." And the bank account is almost empty of all the VC money we got in. You're fired." I didn't get that far because I actually quit a few months before that, sort of seeing that that was the road it was going to take. And then I joined another software company where I got the best seven months of education in my life because this was the worst software company I had ever worked for. They made every single mistake in the book. And that is by far the best way to learn. It doesn't occur here in Silicon Valley.

Right. I'm sure. So, I actually say for people who have sort of aspiration of becoming an entrepreneur, finding a [terrible] company and forcing yourself to work there for six months is awesome motivation and you'll learn so much about how not to do things. That's actually a great insight. So, deconstructing that, what did you actually take from that experience specifically? Sure. So, a lot of it was based on terrible management. Here was a couple of guys who've been running this company and they haven't worked for somebody else. So, that was definitely a key point identified. When people haven't worked for other people, they tend to be terrible managers which means that going straight from college into management is a terrible idea. You actually have to work for somebody else otherwise you can't empathize.

You can't put yourself in employees' shoes. And if you can't do that, you can't build a working environment that's worth working at. And if you can't do that, you can't build a company that's worth working for. So, you felt the culture was pretty bad. It was terrible. It was just full of bad decisions. Got it. And if you were the boss of at that place, what would you have done different? Well, I will probably not have done anything differently if I had been in their shoes. And that's sort of why I took away from that is that it's so much about the environment. It's not because any of us are special or magical, it's because of the experiences we've been put through.

If you had not been put through the experience of working for somebody else, you're just going to make bad decisions. It's simply just lacking that training. And from that startup, where did you go? I went actually to Copenhagen Business School. I saw that that ship was going to sink too. This was just at the end of dotcom phase and these guys were also venture capital funded and who weren't making any enough money to pay the bills. So, I decided that, "Hey. Let's go to business school and see how that goes for three years." Interesting. And then you went out and did Ruby on Rails by yourself? So let's see, I always get the timing mixed up but sometimes during my business school stuff I started collaborating with 37signals as sort of a way to work, next to the schooling. And we started doing some client projects together and then we started on Basecamp. And it was the Basecamp project that was the birth of Ruby on Rails.

And what did you like about collaborating with 37signals that you didn't like with your past experience? What was it that made you actually...? So, I think a fair amount again was environment. I was still in Copenhagen, Denmark. The interested company was in was in Chicago. We had a seven hour time zone gap. So, what did that mean? Was that the distance you need to be managed? That's a great distance. Absolutely. I still work from home to this day. And I think a lot of it comes from the fact that they couldn't call me into pointless meetings. We couldn't waste a lot of time just sitting around and shooting the [breeze]. We had to actually do work because that was the only way we really communicated.

We communicated by showing work. So, they'd have some new designs and I would implement those designs and they would say, "All right, that's good" or "That's bad. Let's do it again." And from that we took as a company, actually, a ton of lessons out of how valuable constraints are, how valuable it was to have me seven time zones away so you couldn't over-collaborate. And that sort of fed into how the structure of the company is today. Most of the company, or about half of the company, is still in Chicago. Most of those people in Chicago even work from home and the rest of the company is spread out all over the rest of the world. And so, now you're a partner at 37signals rather than just a collaborator. You got invited to be part of the company. Yes. And so, the 37signals that you were collaborator for and the 37signals today, does it have the same goals or has it evolved or? I think it has very much have the same goals.

We try constantly to take it back to that. I think actually, most companies start out with the right ideas, the right philosophy and then they lose their way. I think that's the easiest way to destroy a company is by losing your way. So, we constantly try to go back to "Hey, how did we do this when were three people and didn't have time for bull? When we actually just have to get stuff done all the time." It was just a different environment and still our main source of inspiration. Got it. And the last question and this one, I might get what you said wrong so correct me, which I'm sure you will, is your comment about a small business versus what I'll call a scalable startup which runs by design to be a billion dollar company. I think those two are the same thing. You think they're the same. Well, that's what I want to ask. Do you believe 37signals also has the possibility of being a billion dollar company by growing this way? Absolutely.

OK. Can you expand on ...? Absolutely. Yep. I think that there is no inherent correlation between revenues and employee count. There's no inherent correlation between number of stupid policies and revenue growth. But yet it seems like there is because a lot of big companies work in really silly ways. And people think that, "Well, I guess I got to be that. I guess I got to do the meeting thing and hire the business development guy", and all this other chrome that most businesses, in my mind, do not need but think they need to get there. I think that these two things can be absolutely disconnected. We're serving, over three million people have been using our products.

Wow. We're 15 people. There's no way you can get that without being scalable. And I think this is exactly the point. I think exactly the point is you should build something scalable and scalable in my mind means that there's not a direct correlation. It's not a one to one. You don't add another $500,000 in revenue and then you have to hire two guys. And then you add another $500,000 in revenue and then you have to hire another two guys. That to my opinion is not a scalable business. I want it to be disconnected.

I want to be able to add another $5 million dollars in revenue and not hire anybody. And that's how we're trying to design our business, that there doesn't have to be a correlation between these things. There doesn't have to be a correlation between the structure and your revenues. Amazing business model. And I actually lied, I do have one more question. Sure. You said, three million people are using the products in some way. And that number is a year and a half old. I don't even know what it is today. Do you share your approximate revenues? No.

OK. But are you able to fly first class anywhere? If I choose to but I don't. Good. Just remember that. Because again, who's money am I spending? My money. If I'm flying first class, I just spent my money on sitting in a slightly nicer chair for two hours for $2,000. No. Great. So, we know you're not buying your own plane. No.

No. Not yet. All right. Well, that's something else. At some time, you have to enjoy your sports which I think is totally fine but I think to run a successful business, you have to have the mentality that every dollar is your own. Wonderful. All right. I've taken up too much time. Let's open it up for questions from the rest of the class. And David, you get to pick.

Hey. I'm Greg. I'm a PhD student at Stanford. And I use Ruby on Rails on my website. Actually David, can you repeat the, after the event? Yeah. It's a marketplace at Stanford called supost.com and I just love it. Active record, the immigration agent is just amazing. I just want you to tell more of your story of how you actually built the Ruby on Rails framework. Sure. So, the question is, you like Ruby on Rails a lot, how's the story? So, the story is, in the summer of 2003, we were starting the work on Basecamp.

Basecamp is our first product. And at that time, it wasn't even a product. It was an internal tool. We were just realizing that we were being really disorganized with clients, and all these emails back and forth were getting lost, and who has the newest version of the file, and all these sort of difficult things you go through when you try to manage a project over email. So, we were going to build our own tool for our own needs with no aspirations of turning it into a product. And that was the wonderful switch. Until then I've been working for clients. And clients usually have their very particular, silly ideas of what tools you must use based on something they heard, that their nephew use. Not based on rationality at all. So, they would mandate, "All right, you have to build it in Java.

You have to build it in PHP, blah blah blah blah." At that time, Ruby, the program language was not a very well known program language in the west. But it was awesome. I'd looked at it, I'd tried it, and I'm not trying to impress anybody here. I'm just trying to make a better working environment for me, I should use Ruby. So, I started using Ruby for this project and Basecamp turned from project into product and halfway through the development, I realized that I've been building a ton of stuff, to build Basecamp on top of Ruby, that was reusable. I built stuff to talk to the database, and stuff to generate HTML, and all these tools that together was the package of Rails. And I was just looking at all the tools that I've used to get to that point, the Apache Web server, MySQL database, Linux, all these free open source software, and I think, "I'd be an asshole, if I didn't share this. I'm having a lot more fun doing Web development with the stuff I built, if I don't share this and just keep it to myself, that doesn't seem fair." I've gotten so much out of the open source community, keeping it to myself seems like a selfish, stupid decision. So, I decided not to do that and released Ruby on Rails in summer of 2004. But all of it was based on actual needs.

I didn't sit down and just read a lot of theory about how to design a Web framework and just sit down and try to figure out what somebody somewhere might need someday. I was just trying to get this product done in 10 hours per week. And that just wasn't happening with Java or PHP. I couldn't see how to make that fit. So, I needed something that was vastly more productive and vastly more interesting for me to work on. So, all of these constraints pushed on the design of Ruby on Rails, and to my mind is a large part of what it is today. It would have been totally different if it had been a committee group at Microsoft. Let's think about what's the new best way of designing Web applications. It just doesn't happen that way. Products and projects are sort of a result of the environment that they're created [in].

This is why in the beginning of 37signals, we got asked a ton, "Aren't you afraid that Microsoft is going to come in and steal your business? Like they're going to make a better Basecamp?" No. Because what is Microsoft going to do if they want to compete with Basecamp? They're going to say, "Awesome. These guys are just five guys in a garage. We're going to put 30 people on a team and they're going to have two years to do it and they're going to have unlimited resources and they're going to build the best thing ever?" That best thing ever is going to have no correlation to what it is that we're doing. The type of products you build when you have 30 people for two years is in a different galaxy from the kind of product you build when you have three people, spare time, for six months. All right. So, a lot of people say that 37signals is successful because of the things you built besides the core product, like Ruby on Rails, micro blog, and the book "Getting Real". Do you think that's a good strategy for others to emulate, to sort of try and do stuff like build a reputation on other things you create besides the core business that you charge for? Yes. And the fact is, that when you're starting out, and again, the assumption here, you're working for your own money, you cannot outspend. You can't outspend a development.

You can't outspend on marketing either. The only way we know of how to do this is Kathy Sierra which is somebody you always should get familiar with. She wrote a wonderful blog called Creating Passionate Users. She's on Twitter, look her up. She has this wonderful saying of you can either outspend or you can out-teach. We're trying to out-teach because that's what we can do with our resources. We can share the inner workings of how the company runs, how we're building our products, what technology we're using, and we can use all of these things to build an audience. And that's I think a key part of the success of the company is that we're trying to build an audience. We're not just trying to have customers. We're not just trying to push something on people: "Buy this product.

Buy this product. Buy this product." We're trying to build an audience that might not need our product today. We have plenty of readers of our weblog or users of Ruby on Rails that doesn't pay us a dime today. At some point, these people will either need it themselves or they will recommend it to other people. I think all sustainable businesses are build by word-of-mouth, in the end. And the way to get to great word-of-mouth is, you can try the outspend approach on marketing and that might work for some. It's very expensive. It's not going to work without the VC capital and that has its own time-bomb issues. So, what can work for you is to try to out-teach. And this is one of those things where it's just not about money.

It's just about sharing what it is that you're doing, being a good writer, getting people interested and impassioned about what's going on. And people constantly saying, "Well, I'm working in some other business that's not interesting." Bull. Every business is interesting. There's always somebody who cares. There's always somebody who cares about your niche. If you said, "Who's going to care about how HTML or CSS looks like?" Well, that sounds like a stupid question but there's plenty of people who do. Plenty of people who care. You said that for your time at Copenhagen Business School, that 96.7% of your time was basically wasted. So, can you tell us what you found about the remaining 3.3% that you had? Yes. So, the question is, I said that I wasted the vast majority of my time in business school.

What did I actually take away that was worthwhile? One lesson which I could have gotten somewhere else, actually I just heard it on "The Wire", that HBO showed the other day is "Buy for a dollar, sell for two." There's a key lesson that you have to sell for more than what you're buying for. And it seems like a lot of people in the Web 2.0 space still haven't conceptualized that idea. So, that idea of just the marketplace of capitalism, of people being selfish bastards, that that works, that was probably sort of a key insight that I took away from it. That you just have to appeal to people's own self-interest. And that works in terms of running your company, and getting customers, and all of these things. I think capitalism in terms of people assigning value to what you do by paying you money, I didn't have that narrowed down before I came to business school. It's very elementary but sometimes the simplest things are the hardest to learn. So, I took that away. Let's see, what else? I think in some ways, just being well read has some sense of usefulness in terms of it's easy to shoot other people's bull down when you are intimately familiar with what it is. So, I think that was definitely helpful.

But yeah, there's nothing that jumps straight to mind, which I think itself is a problem. I'm sure there's plenty of other things that I did learn and did take away from it. But I was just not generally a big fan of it because it does not apply to what I do. The vast, vast majority of all these stuff does not apply to running a 15-people company. It might apply to running a hundred thousand but I'm not interested in that anyway. I would never want to run a hundred thousand people company. I think that'd be a terrible scaling thing, or basically when you scale. Actually what's kind of funny with our industry that there's a fair number of prominent case studies that get a lot attention for being big successes. And in our space, software as a service, I'm going to pick on somebody in particular here, salesforce. So, salesforce has gotten a ton of good press about, "Oh, how wonderful it is that they're moving all their software online and how innovative that business model is." Have you guys looked at their numbers? Have you guys looked at their margins? Have you guys looked at their PE? It's hysterically bad.

Do you know what their margins is for the last reported earnings? 5.9%. If you can only make 5.9% off something that you don't even make. It's software. Each marginal thing is free. And you've been at it for 10 years? You're doing it wrong. They might be a billion dollar company, and that's where all the attention comes from because people look at stupid things like revenues, as though they meant anything. Revenues doesn't mean anything. Profits mean something. And I think there are a few smart companies who've realized this like Apple for example. They, for a long time, got, "Why you guys only have 5% market share?" Who cares? If you're the guys making the most money, if you guys are taking the most profits up.

So, there's this awesome stat about the iPhone, how they only have 20%, 15% market share with like 50% profit share? That's exactly the kind of company that we want to build. And I encourage you guys to build too. Build companies that focus on profits. I don't even know where that came from, but... In the back. Earlier, you mentioned your disdain for venture capital. Do you feel the same way about angel capital? I think I heard that you guys took 20 from Jeff Bezos a few years ago. Yep. Can you talk about that a bit and what drove that decision? Sure. So, I talk about the Bezos thing.

When the first three or four years of running the business, we got about I think 35 to 40 venture capital firms coming knocking on our doors wanting to give us money. And we couldn't understand how to make that work because the company was profitable, more than profitable. So, what we were going to spend that money on? Other than buying a nicer office or hiring more people that we didn't really know what they were going to do. So, that didn't work. For the one thing that made the Bezos thing work was A, he wasn't venture capitalist. He didn't have the three to five year time bomb view. He had the "Let's build an awesome company in 20 years. Let's not sell this company. Let's not do all of those things." And secondly, we set it up in such a way that it's very un-VC-like. For us, it was not funding the operation.

For us, it was all money off the table. So, it was sort of a way to reduce risk for us. The company could blow up next year and we would have nothing to show for it. So, we took some risk off the table without giving up any control at all. Bezos doesn't have a board member seating. Like he would show anyway. The investment that he made in our company is like a rounding error on a coffee break for a guy that's worth $13 billion. So for him, it's more about just the, being interested and having association with it. And for us, it was about getting something off the table and having somebody like Bezos involved, that we can draw and have an adviser of sorts. So, absolutely not about, it wasn't a VC thing.

And about the second question, angel funding, how do you feel about that? I feel that it's a lesser evil of the two but also generally a bad idea. I think as far as you can, you should be spending your own money. And sometimes spending your own money means that you have to work on the side, too. When we were building Basecamp, basically the second act of 37signals, which is, the first act was doing Web design. The second act is that we're now a product company. Those two things don't have that strong of a correlation that's a business goes. When we were building that second act, we had a day job. 37signals had Web design clients come in to pay the bills. I was even going to school and I was doing other stuff on the side too. So, you don't always have to give up everything in order to start something.

Oftentimes, you're spending weekends or nights or something else until you can get something off the ground, is in my mind a much better way to go than trying to secure funding upfront. But the problem with that is that it requires that you actually do stuff. It doesn't work if you're the idea guy. So, I think with starting a company, being the idea guy doesn't work. There's no room in a new small company for an idea guy. You have to do stuff. You have to have utility value. You have to be able to build stuff, design stuff, market stuff, something that has direct value to the business, not just idea. And that's where I see a lot of this actually go wrong because people think they can get along just being idea guys or just being management. There's no management when we're three people.

There's no management when we're 15 people. We still don't have any managers hired in 37signals. Every single one of us are producers. I'm still a producer. I still write code. Jason still designs. We still do all the stuff and management is sort of the offshoot of, "Oh, yeah. Sometimes we have to deal with issues that they come up." Problem is when you have actual managers, whose sole job it is just to manage, they make up [stuff] to manage because you've got to fill an eight-hour day. And in the beginning, there's 40 minutes of management every three days. That's what you need for management.

You do not need eight hours of management, which is how you get policies and all this other bull that crops up when people don't have anything to do. Idle managers are absolutely the worst. So, how do you make company decisions if there are no managers, or is it totally collaborative, or does one person just say, "Hey, we need to do this" and runs with it or? What's the process? Oftentimes, somebody brings up a good idea and we say, "That sounds good. Let's do that." There's not a formalized process. We don't have any board meetings to chart the direction of our strategy or something like that. We just have a chat room. Actually it's one of our own products. We call Campfire. If somebody has a good idea, they type it in and we say, "Let's do that" or "Let's not do that." There's not a whole lot of management going on. Yes Jason and I, Jason is my business partner, probably weigh in on more decisions than most because that's what we do.

But we also implement the stuff. When you disconnect those things entirely, the people who do stuff and the people who decide stuff, bad things usually happen. If you guys can create very good profit margins, why do you not want to build a hundred thousand people business because that'll create more opportunities for a hundred thousand people over here. Now, let's say a 15-people company are just building personal wealth. Would you agree? Yes. I am just building personal wealth. I think that's how the system works. To me, if I have a very big profit margin, the goal is to get it higher not lower. To me, just hiring another hundred thousand people is not going to bring wealth to anything. Our wealth, what we contribute to the system, is that we build cheap stuff.

We sell our software for very little compared to the vast majority of our competitors. And why can we sell our stuff for very little? Because our costs are very low. We have an entire different cost structure from somebody, say Salesforce. When you sign a key account manager to everything, you have to charge something very different from 15 people dealing with four million customers. So, I think building a scalable business is something entirely different. And the capitalism of making the system work is entirely different than charity. Charity is fine too. You can build up your personal wealth and then you can give it all away for free which... Or... "for free".

Makes sense. Bill Gates sort of like, "All right. That's cool." If you try to mix those two things, I don't think you're getting anything good out of that. Encourage those hundred thousand people who you would have hired to actually do [stuff] instead. Start their own companies. Quick question about ... Oh. We'll, go over here. What's your feeling for early entrepreneurs about the program like Y Combinator and Techstars and the incubators? I spoke at a startup school in 2008 which is connected to Y Combinator so I have to be at least a little bit nice. That's the disclaimer.

I don't think it's that bad. I think the stake that they take is not very large. It's not a controlling stake and so on. So, I think the key setup of it is not so bad. You get to be part of the community. It's like going to summer camp. That's fun and for some people, I think that's a good experience. What I don't like about it is the pipeline it sets up. The pipeline is absolutely set up in such a way we give you a little bit to get started and then you build a prototype and then you get VC. That's the part that I don't like.

If he was mainly about funding people with a little bit money to get started to build a real product that had real revenues and started being self-sufficient to have profits and so on, I think it be a lot better. So, it's not the model itself that's broken, it's the pipeline that's broken in my mind. Earlier I think you said that this is probably the most opportunistic or best time for people graduating from undergrad or graduate programs. And I was wondering at first if you could maybe just expand on that a little and then talk also about what kind of companies are out there now that you think are good examples of achieving success the way that you're trying to achieve success. Sure. So, two part question. What's good about starting right now and who's doing well. So, what's good about starting right now I think is people looking to, in many ways, reduce cost. A lot of people are coming to Highrise or Basecamp or other things like that because they don't want to pay their IT department a ton of money anymore. That they don't want to pay Microsoft licenses or they don't want to pay per seat at Salesforce.

So, if you have a drastically different cost structure than your competitors, you can charge drastically less. That alone I think is a big part of it, like why we're getting a bunch of business right now, why we had basically no dips during the recession. There was no recession for us because stuff moves around. So, all the stuff is not created equal. I think you get a huge advantage out of that. And I think people are more willing to try new stuff when [stuff] isn't working. When something is broken, we're losing money and so on, people get a lot more interested in trying new things. When it's the fat times and things are rolling and the quarterly numbers are good, let's not mess with a good thing here. Let's just buy the IBM. When it's not going so well and stuff has to change, that's when new companies and new products can make headway.

Second question was ... What other companies today... Oh, what other companies are doing well? ...are a model like yours. I think that the funny thing is that the company I look to that are doing well, rarely get any PR at all because the fact is most companies that run like us are smart. They duck. They don't talk about how much money they make. They don't want to attract any attention. The kind of people who want to attract attention is the people who are just announcing Series A funding because they have to get some PR going for Series B funding and all that stuff. And they have to get on TechCrunch because that's what matters to their investors. And all this stuff that just doesn't matter.

The people who are actually making money and have profitable business have their heads down and are trying to increase their profit margins. Some of these companies that I really like was Threadless. It's actually not even a software company per se. It's more of a community company. They create T-shirts. And they sold something like 40 million T-shirts last year. Unfortunately, they did just take VC money like a year and a half ago and I think that was a big mistake and it had very obvious consequences. The entire team is gone which is often what happens with these kinds of things, and release times, and after a year and a half, then you can go, and people do. So, those guys are doing really well. There's a company called Campaign Monitor which is sort of a mailing list shop.

They're doing really well. There's SurveyMonkey. I think they just got bought by somebody which ... Yeah. They sold it for $100 million I think and it was like 90% of their margins. Yeah. That was good. They actually still own a part of the company. Right. Yeah, I think that's awesome.

There's a bunch of companies like this but very few of them are making big splashes about it. David, thank you very much. Sure.