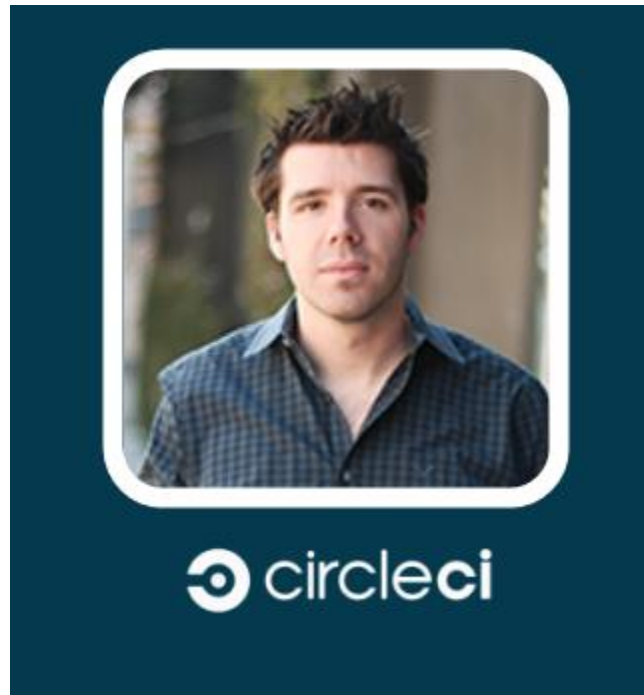


Podcast With Paul Biggar



Today's Podcast Highlights

- [1.46 - Software teams use continuous integration to make sure that they move quickly]
- [3.33 - Doing things in small sections, and testing them, and validating them]
- [4.23 - Shipping 10,000 changes once a year or 1,000 changes every few months, which is a much, much less safe]
- [5.04 - Waterfall is this derogatory name applied for how people used to design software in the 70s]
- [6.32 - There's a problem in how governments and large corporations design software]
- [7.53 - Continuous feedback and continuous contact with your clients, building small amounts, validating that it's the right thing, and then continuing on]
- [8.54 - I spent a year thinking, "If I was in charge, I would do this differently"]
- [9.44 - Try to figure out what exactly did the market need, would people even buy this, and to figure out what sort of version of the software needed to be built for the customers that were out there]
- [11.07 - We were very fortunate with the product, that the market was ready for it]
- [11.30 - We had a lot of validation early on that we were building the right thing]
- [11.55 - I didn't really have any idea how to fund raise]
- [12.26 - I moved to San Francisco, and I built up a bit of a network, people who were at the

same stage, a little bit further on]

[12.52 - Networking involves introductions from entrepreneurs and those entrepreneurs vet you, in effect, and spend a little bit of their social capital in making that introduction to the investors]

[14.07 - When we went out to the early investors, we didn't have any proof from other investors that this was investable]

[16.16 - Selecting an investor consisted of looking at what the investors would bring to the table, and what kind of expertise they could offer, and how they could guide us]

[17.19 - We didn't actually pick the best offer. We picked the second best offer]

[18.24 - One of the things that really worked for us was not really focusing on marketing that much]

[21.26 - They try it and then they have a wonderful experience and then they tell their friends]

[22.04 - Understand your customer, and understand your market, and have a good long-term vision of how pricing takes you to the kind of revenues that you're looking for]

[24.42 - You have to really look at what is the kind of philosophy of how your company is going to grow]

[26.51 - I think that the most important thing is really being able to show to either the customers or to investors, depending on which track you're going, that you just have an amazing product]

Podcast Transcript:

Paul Clifford: Hi there, software entrepreneurs and welcome to the [Disruptware](#) Podcast. This is the show for entrepreneurs who are either just starting out, or those who already have a software company and are looking for techniques and ideas for massively scaling their business. The way we do that is to interview experts in the market who are already running their software company. Whatever level they are at, they have great content to share and we dig deep to get factual experiences that you can put into action right now. Let's get on with our guest today.

Okay. On today's show I want to introduce someone called Paul Biggar, who is the founder of CircleCI. This is a San Francisco startup. It's all about something called continuous integration and continuous delivery. They've just closed a six million dollar series A funding round and have got a great story to tell. Let's get on with the show and introduce you to Paul.

Paul Biggar: Hey, Paul. How are you?

Paul Clifford: Yeah. Very well, thanks. Very well. I've been looking at your business and I'm really excited to get you on the show. For the benefit of our audience who are both software entrepreneurs, but also technical people, perhaps you can just give an overview of what your solution is.

Paul Biggar: Sure. The CI in [CircleCI](#) is, as I say, continuous integration. Basically, software teams use continuous integration to make sure that they move quickly, and that their code is always tested as soon as it gets written, and that the entire team is on the same page as to the state of their code base. It integrates so that the second part of CI is continuous delivery. That's where you extend your CI to continuously deliver your software to your customers by deploying it every time you write new code and every time the test passed.

Paul Clifford: Brilliant. Now, just to put that in context of today's development world. Obviously, I know how important that is. But from sort of a bird's eye view, continuous integration stems from the whole concept of, whenever you're building technology nowadays, it's important to build it in small chunks and to deliver in small chunks. Right?

Paul Biggar: Right.

Paul Clifford: This concept comes from things like Lean and Kaizen, which originally were invented from the manufacturing industry. In fact, Toyota was the leader in this whole idea of building everything in small batches. The reason you do that is because you see problems really, really early on in the whole development process.

Paul Biggar: Right. Continuous integration is kind of the technology version, or the software version, of continuous and version intended. It's constantly making sure, doing things in small sections, and testing them, and validating them. It's a key part of the Lean startup. It's central to the whole Agile movement. What you get to do, basically, is, instead of batching up hundreds of things which may affect each other in weird ways, and where if something breaks, you're not going to know the source of them, or you're not going to know the cause of the breakage, you basically validate each thing one step at a time.

The relationship to continuous delivery there is particularly good. A lot of people are kind of confused about why you would think that it's unsafe to ship their code to their customers 10 times a day. The opposite of that of course is shipping 10,000 changes once a year or 1,000 changes every few months, which is a much, much less safe thing to do, because there could be so many things that interact in unforeseen ways when they finally get in front of people.

Paul Clifford: Right. Exactly. Just think. If the healthcare website in the US ...

Paul Biggar: Right.

Paul Clifford: If that followed your methodology, they probably wouldn't be in the mess they're in today. Right?

Paul Biggar: Right. There's an interesting piece about this in the Washington Post, actually, where they talked about how healthcare.gov had followed, basically, a strict waterfall model. Waterfall is this derogatory name applied for how people used to design software in the 70s, which is first we will design it, and then we will turn that into a spec. Then the engineers will turn that spec, will follow the spec and literally crank out working perfect code that follows the spec exactly, because the spec anticipated everything that might possibly go wrong. Then we'll ship it to customers.

It's pure madness and it's exactly what healthcare.gov did. It's the antithesis of the whole kind of Agile movement. It's exactly why continuous integration and this model of moving quickly and continually testing what you're doing is so important.

Paul Clifford: Yeah. When I read all about that, I just thought that it was completely amazing that, in today's world, where we've kind of learned the lessons of huge projects, especially for local governments, but even in the private sector historically. Whenever you talked to software...I remember like 10 years ago, whenever you talked about big software projects, they were guaranteed to be late; they were guaranteed to be buggy; they were guaranteed to be 10 times over budget. Software projects got a bad name because of that and because of the whole waterfall method, which, of course, is all we knew about then. Right?

Paul Biggar: Right. I think that there's a problem in how governments and large corporations design software, which is that they believe that it's easy to just, "We'll just spec it out and it will just work." Or the idea that you can do a fixed-cost bid and give it to the lowest bidder, or something like that. I think that these kind of cost overruns are the inevitable results of the process that got them there, where there isn't people who are experienced in building software this way or something like that involved.

It would be like building a bridge and not having an architect and an engineer being involved in deciding who's actually going to build your bridge. But that's how software contracts are awarded today in governments and large businesses, and it's fundamentally flawed. I don't want to sound like I'm an Agile zealot,

because I'm really not. One of the places that Agile came out of was these consultants saying that this model is screwed up, and that there is a different way of doing it, there is a better way of doing it. It involves continuous feedback and continuous contact with your clients, building small amounts, validating that it's the right thing, and then continuing on.

Paul Clifford: Brilliant. Now that we got all the technical stuff out of the way, tell me a bit about yourself. What did you do before starting this project or this company?

Paul Biggar: Just before I started CircleCI, I was working as an engineer at Mozilla. I was a compiler engineer and I worked on the Firefox javascript engine.

Paul Clifford: All right.

Paul Biggar: The reason that I started this, that I had been using Mozilla's continuous integration software. I use it basically every day. Every time I wrote some code, I tried to push that code out. It has to run through the continuous integration software, and it just wasn't a great product. It was built by people whose job it was to keep this huge infrastructure of 2,000 computers up, and not people who worried about the day-to-day problems that I had. I spent a year thinking, "If I was in charge, I would do this differently." Then when I decided to leave Mozilla and start another company, this was the idea that had been in my head for the last year.

Paul Clifford: Basically, it came from you being at the sharp end thinking there's got to be a better way of doing this.

Paul Biggar: Right. It's kind of reflected a lot in the product and in the business. It's a very developer-focused business, a business by developers, for developers.

Paul Clifford: Right. Okay. How did you get started? Did you go out to get funding, or did you do it from your own pocket?

Paul Biggar: I was still working full-time, and I was building on the side and trying to talk to customers. It's a very Lean startup approach to do customer development upfront, try to figure out what exactly did the market need, would people even buy this, and to figure out what sort of version of the software needed to be built for the customers that were out there. While I was doing this, I pitched a friend who owns a software business. Allen, my co-founder, who I didn't know at the time, pitched the same guy the same week, and was pitching exactly the same product.

We started working together. It wasn't until maybe two months after that, when we were just starting to get our first customers, that we got our first investor. It was someone who put in about \$50,000. Before that, and a lot after that as well, we were working off savings, until we raised our first round, which was in December 2012. That was one and a half million.

Paul Clifford: You've got your partner together, you've formed the company, and then you managed to get an angel investor?

Paul Biggar: Yeah, exactly. An angel investor of 50K and that took us through the early uncertain parts of the business, the ones where we were trying to get product market fit. We were very fortunate with the product, that the market was ready for it. We had product market fit incredibly early, and I see lots of entrepreneurs who spend years trying to get product market fit. So I know how fortunate we were.

We started in September 2011, had our first customers around January, and had our first paying customers in April of 2012. We had a lot of validation early on that we were building the right thing. We had to adopt the product in a lot of ways based on that feedback, but it was a very customer- focused process.

Paul Clifford: When you got the 1.8 million, how did you go out to the market, the investor community, to actually get interest in that?

Paul Biggar: I didn't really have any idea how to fund raise. When I started, I had done Y Combinator the year before. It doesn't quite teach you how to fundraise, but it gives you sort of an idea. I had a rough overview, but I didn't really know how to apply it, and we hadn't done very well when I was in [Y Combinator](#) with a previous company with our fund raising.

What I did was, I started talking to a lot of entrepreneurs that I knew. I moved to San Francisco, and I built up a bit of a network, people who were at the same stage, a little bit further on, a little bit behind ... just friends. I went out and I sat down with all of them and said, "Look, here's what we're trying to do."

They asked me lots of questions. They told me how exactly one goes about fundraising. Basically the idea is, you get introductions from entrepreneurs. Those entrepreneurs vet you, in effect, and spend a little bit of their social capital in making that introduction to the investors. When I was talking to all these entrepreneurs, I hadn't realized that I was actually doing the exact right thing, convincing those people to believe in us.

We didn't convince everyone to believe in us. Some people thought it was silly or wasn't going to be a big business. Some people saw a big vision there, and introduced us to investors. That kind of continued. Investors introduced us to other investors, or had us talk to some other entrepreneurs to vet us, and those entrepreneurs introduced us on. That went on from the end of August until mid-November, when we had a handful of term sheets. We picked from them and closed in December.

Paul Clifford: So you kind of like developed a network and then built upon that. You developed a network of entrepreneurs. They introduced you to some investors, and then you built on that, and almost a continuous cycle process. Right?

Paul Biggar: It very much was. When we went out to the early investors, we didn't have any proof from other investors that this was investable. No one had yet said, "I am putting my money behind these guys." That's very important for investors. I think it's fairly straightforward to see why.

Then, as we got further on ... Let's say we had 100K in commitments ... Then we went to talk to some of the bigger investors. They knew some of the investors who had committed already. There's a lot of back channeling, investors talking amongst themselves. Having that early proof is very important for the larger guys. It kind of sparkles, and it continues.

Paul Clifford: The culmination of all that, then, because of one, you had a product that was already kind of validated because you had customers coming on board, you had ...

Paul Biggar: Exactly. We had around 50 customers at that point paying, and we had a couple of thousand dollars of monthly revenue. It was slightly validated, but it was a still very early stage.

Paul Clifford: Right, but enough to prove the concept works, that there's a need in the market. You're providing a solution that people are paying money for. It's obvious, then, that you just need the money really, I guess, to scale. Right?

Paul Biggar: Right. Yeah. We had to have a good story around what we were building and a story as well around how we were going to spend the money, how this was going to be a billion dollar business. That wasn't important for the small angels, but for the larger people, the people who were writing half million dollar checks. They wanted to see that there was a path to something that could be worth more than a hundred million dollars.

Paul Clifford: Right and they want to do more than just throw money. Also, I think it's important from your perspective, when you have that selection of term sheets on your desk. Maybe I should be asking you this, but what was the criteria? When you had several offers on the table, what was the criteria that you went through to make your decision?

Paul Biggar: A lot of it was looking at what the investors would bring to the table, and what kind of expertise they could offer, and how they could guide us. From our side it was also risky, you only get a small number of investors. There's a lot of factors on which investor you want to bring in. It includes: Do they have the expertise to help you? Have they done this before? Even such things as, how does this reflect on us choosing this investor? If you go and you choose kind of an unknown investor, then it reflects poorly on yourself for investors later down the line.

We definitely used the terms that we were offered as a sort of a first-pass filter that people who made sort of really low offers, or that sort of thing, obviously didn't see the value of the company, and also wouldn't be people that we'd want to be working with. Once you got within the sort of sensible range, we didn't actually pick the best offer. We picked the second best offer, because we felt that they would be more valuable to the company.

In particular, the investors we picked were Heroku's investors. They had grown a developer phasing company before. Heroku had already sold at the point that we were raising money. There was this company that had sold completely developer phasing consoles for over 200 million dollars. We had people who knew what was involved in that and would be able to guide us.

Paul Clifford: Where you're at today, how many paying customers have you got on board today?

Paul Biggar: We raised six million from DFJ in our series A financing. We announced that we had over 1,000 customers and over a million a year in revenue.

Paul Clifford: It's obviously really, really going well. Obviously, the product market fit is excellent. But how you doing your marketing?

Paul Biggar: It's funny, actually, because one of the things that really worked for us was not really focusing on marketing that much. It sounds ridiculous, but once we got to about 100 customers, the product sort of sold itself. Developers referred other developers to it. Like you can see it on Twitter, people say, "What should I use for CI?" and three of our customers reply. People who know that that person would

say, “Oh, you should definitely use CircleCI.” That started happening once we got to about 100 customers.

We did some marketing experiments. We did some content marketing, and writing blog posts, and did some ads, and that sort of thing. Some of it was kind of effective; but, really, when we sat down and looked at where all of our customers come from, it was all referrals from existing customers.

Paul Clifford: Which is obviously the best lead. Right?

Paul Biggar: It is, but I think it’s also indicative of how you build your business. We were focusing just a huge amount on the product and constantly making the product better in supporting new-use cases. CI is a very simple concept, but it’s actually a platform, and it’s very broad in how people use it. By spending all of that time on the product, it meant that it worked, and it meant that people came in and just had this kind of delightful experience, and were willing to refer to other people. As a result, we didn’t really spend that much time on marketing. We definitely did nothing systematic.

Paul Clifford: So what you’ve done, effectively, has hit that sweet spot so your customers are your evangelists.

Paul Biggar: Right.

Paul Clifford: Yeah.

Paul Biggar: We didn’t build a referral program to make this happen. It’s not that we’re against marketing at all. In fact, we’re kind of rebuilding some efforts at the moment. What we realized is, that developers don’t like to be marketed to; they don’t like to be sold to. You can still do things that are traditionally marketing, but you have to make it clear to developers that you’re doing it in a way that is authentic to them and that is really building around the customer base, instead of just hiring a VP of Marketing and hiring a VP of Sales and putting a million dollars into AdWords or more traditional marketing activities.

Paul Clifford: So something like content marketing would be ideal for that. Right? So where you’re delivering education ...

Paul Biggar: If you call it content marketing, your developers and customers are going to hate it.

But, yes. Blogging, saying things that are interesting. As a result of saying things that are interesting, people come to know about your company, and they come

to know about your products. Then they realize that they need this product, and they try it and then they have a wonderful experience and then they tell their friends.

Paul Clifford: When you're looking at your business model, you're selling, obviously, to development communities. You're selling a SaaS model. How did you decide on your pricing approach? I guess a common answer to that is, "Well, you just split test." I'd be interested to see how you decided what works and what doesn't work in terms of pricing methodologies.

Paul Biggar: I don't really think that you can split test pricing. I think that you really have to understand your customer, and understand your market, and have a good long-term vision of how pricing takes you to the kind of revenues that you're looking for.

One of the things about Circle is that the price point is very, very low. It's \$19 dollars for...that's kind of the starter package. A team of 4 could easily do that, and they're getting way, way more than \$19 worth of value. I think it would be fair to say that they're probably getting, \$1,000 of value. We charge them \$19. We could charge them \$50, or we could charge them \$200, or \$500, but, whether or not we do that is really based on, "What is our philosophy?"

Our philosophy is around getting adoption. It's around getting a lot of people to use us, getting the entire market to use us. We're really happy to have that low price point, even though we could probably have 3 times as much as revenue if we jacked our prices up. What we realize as well is that our large customers are going to be the ones who bring in the vast majority of the revenue.

It's kind of like a premium model that is actually paid for. Our small customers are the people who tell everyone else about it, and all of our big customers all start out as small customers. They'll sign up with one team or one individual and they'll put \$19 on a personal card. People who pay us \$10,000 a month started with that model.

Paul Clifford: By having a really low entry point, you're getting the individuals who are looking for the solution to take a risk almost on themselves and try it out. Then they fall in love with it. Right? They fall in love with it and become the evangelist.

Paul Biggar: Right. It's expensive for a team to try it. You need to try it. You need to validate that it's a good product. Then you need to invite the rest of your team, until you're spending some of your social capital, I guess, in how your team believes in what you say by recommending that people try us. Adding big road blocks in the

way ... You have to spend \$500 a month to even try it out ... That's kind of ridiculous.

I think the point that I'm really making is that there's no one right way to approach pricing. You have to really look at what is the kind of philosophy of how your company is going to grow. If we were trying to be like a bootstrapped company, we weren't growing, or we weren't taking VC, or something like that, we would probably have a very different pricing model, especially if we didn't have the ability to hire just a lot of engineers to be able to take on a lot of customers immediately and fix their problems.

Paul Clifford: Paul, as an entrepreneur as well, what do you look for in terms of your inspiration? Who do you look up to? Do you follow any particular blogs or read any books? Who do you sort of use as your guidelines for growth as an entrepreneur?

Paul Biggar: What I have found amazing is, really, mentors and other entrepreneurs I know who have been doing it, and who have sat down with me and given me their philosophy. That's the primary way that I've grown.

In terms of things that you can read, I think one of the best resources I've ever had for learning about product has been the Intercom blog, which is insideintercom.io, I think, or it could be .com. It's just an amazing resource for how to think about product, and how to think about delighting customers, and how to communicate to customers. All those sort of things has been amazing.

There's a lot of industry blogs that I read. Fred Wilson's blog has been very good...a couple of VCs. Ben Horowitz has had amazing stories and, obviously, I read everything that Paul Graham wrote.

Paul Clifford: Any parting words of wisdom, or inspiration, to my tribe of software entrepreneurs who are either bootstrapping themselves or looking to get funding?

Paul Biggar: I think that the most important thing is really being able to show to either the customers or to investors, depending on which track you're going, that you just have an amazing product. Really focus incredibly hard on the product. Make sure that you listen to your customers. Make sure you solve their problems. You don't need to do directly what they're telling you, but solve their problems and delight them. Without that base, it's very, very difficult to accomplish anything, but with that base, you can kind of build anything.

Paul Clifford: Paul, I really appreciate you coming on the show and taking time out of your busy schedule today.

Paul Biggar: I appreciate you having me on.

Paul Clifford: If you enjoyed the show, you can get the show notes from disruptware.com. If you are not a subscriber, and you're listening to this in the iTunes store, then please visit disruptware.com and sign up. That's it for this episode. Look out for next week's show. I'm Paul Clifford, and thanks for listening.

Are you ready to become a software entrepreneur?

Get everything you need at www.disruptware.com/.