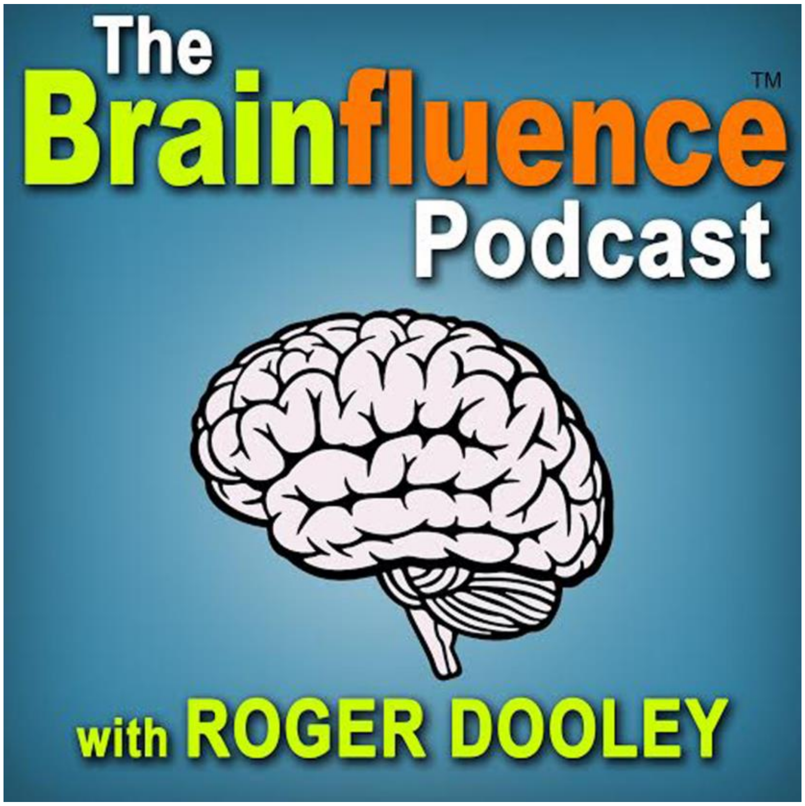
Ep #82 – How to Create and Launch a Software Product



Full Episode Transcript

With Your Host

Welcome to *The Brainfluence Podcast* with Roger Dooley, author, speaker and educator on neuromarketing and the psychology of persuasion. Every week, we talk with thought leaders that will help you improve your influence with factual evidence and concrete research. Introducing your host, Roger Dooley.

Roger Dooley: Welcome to *The Brainfluence Podcast*. I'm Roger Dooley. My guest this week is an entrepreneur, engineer, author, and speaker. She's a Duke grad and was the founding engineer at Mint where she helped build, launch, and scale the product until it was eventually acquired by Intuit. Following the acquisition, she went on to launch Femgineer, an education company for tech professionals and entrepreneurs who want to learn how to build software products and companies.

Her new book is *How to Transform Your Ideas into Software Products*. She's also the co-author of a not-yet released book *Present! The Techie's Guide to Engaging an Audience*. Welcome to the show Poornima Vijayashanker.

P. Vijayashanker: Thanks for having me, Roger.

Roger Dooley: Poornima, when I saw the title of your book, which the full title is *How to Transform Your Ideas into Software Products: A Step-By-Step Guide for Validating Your Ideas and Bringing Them to Life*, I got a little bit worried. Are you suggesting that anyone can have an idea and turn it into a software product?

P. Vijayashanker: Definitely. That's exactly what the title is meant to convey.

Roger Dooley: Okay, well, how is that? It seems like for many folks creating software would be somewhere akin to translating the Rosetta Stone. How should we interpret the title and the theory you advance in the book?

P. Vijayashanker: Sure, I’ll put it this way. There are a few different classifications. When we talk about software, there’s definitely the web-based software. For example, like Mint, where it is a software application and then you host it in the cloud. So that’s one type of software product you could build.

Another type of software product is an ecommerce store. If you are selling anything, books, clothes, whatever it is you’d like, you need to be able to build an online presence and that also takes a level of software. There’s a lot of great off-the-shelf tools you can use but as you scale, you’ll need to understand how to manage the software that basically powers your business.

Then there are, of course, mobile applications and there are, I think, content. So there’s a lot of different ways in which you can build a software product. Now, I don’t go into the nitty-gritty details of how you would do every single one of these but instead what I talk about in the book is if you're thinking about building a product and that product is going to be software-based, you know, distributed through the internet, then you're going to want to understand what the steps are in terms of going from idea to finally launching it.

If you think about it, a lot of great entrepreneurs that we’ve seen over time aren’t necessarily ones who knew how to build the final product on their own. I don’t think Lee Iacocca decided that he was going to build a car from scratch and neither did Elon Musk. These people have teams in place. They need to understand the entire process and they bring on experts. So really what this book is doing is giving that process in particular when it comes to developing software products.

Roger Dooley: Right. I would guess that many of our listeners, many of whom in fact are entrepreneurs, have had some of kind of a software idea, perhaps a pet peeve where they're looking for an app and nothing on the market is really doing what they want. Or they’re coping with a software product that seems needlessly complex and hard to use and they think there might be a better way.

P. Vijayashanker: Yeah.

Roger Dooley: How would they go about getting started? Saying, “Okay, I have a little bit of time and a little bit of resources.” How would they get started in trying to address that problem?

P. Vijayashanker: Yeah, I actually talk about that at the beginning of the book where I say depending on the stage you're at, let’s say you’re in that idea phase. You might be the type of person who likes to take an existing product and as you mentioned, refine it, because maybe it’s really cumbersome or it just needs to have a better user experience. So you might take the approach of refining a product that’s already on the market. That’s totally one approach.

Another is you might decide that you want to be more on the cutting edge and develop the infrastructure for a new product. If you're familiar with companies like Twilio or even companies like Salesforce, where they now have bought Heroku. A lot of them are building the infrastructure to support more software products. So that might be a direction that you go in.

Or you might just decide you have an existing business. Let’s say for example, again, you're selling clothing and you now instead of want to have a brick and mortar store, you decide you want to have an online store or an ecommerce store. Again, you can go through the process of deciding how you want to build that out. What I talk about is in that first phase, regardless of which direction you're headed in or the type of idea it is, you’ve got to start by focusing in on your idea.

A lot of times people hold themselves back because they either are worried that their idea isn’t good enough or they're worried that they have too many ideas and they're really not sure how to whittle it down or they might have gotten farther along than most but now they’ve hit a sticky point and they're just not sure what to do next. How to actually go about taking the idea and building it into a product.

So I definitely provide all the steps, despite whichever one of those camps that you're in, but that’s the starting phase. It’s let’s figure out how to hone in on one particular idea. Let’s not have you worry about whether it’s good enough or it’s innovative enough. Let’s instead walk through some steps to seeing if the market, aside from you, but the market itself has a demand or has interest. Because a lot of times, we get wrapped up in our own desires or our own problems that we forget we need to check if other people have a similar problem and would be willing to pay for a solution.

Roger Dooley: It certainly seems like today the tools are getting somewhat better for folks who are not necessarily highly technical to be able to do many of their own things. Not perhaps create an original software app but you talk about things like setting up a store or perhaps setting up a website with some particular kind of functionality. Now with platforms like WordPress with a million plugins, you can, really, by putting together some building blocks create a relatively complex functionality on a website.

P. Vijayashanker: Yeah, that’s definitely true. In fact, I talk about a number of off-the-shelf tools that you can use and one of the exercises, because the book has a series of exercises, one of them is just to start generating some buzz and to put up a simple landing page to get people to come and serve as a beta tester or serve as an early customer. So, yeah, there’s certainly a lot that you can do on your own.

Even if you decide to build something, you don’t always have to start from scratch. One of the topics that I cover is the types of tools that are out there that you can piece together to get where you need to go. Get through those initial phases of attracting customers, building a small, rough prototype out before you have engineering talent who can build a more robust solution.

Roger Dooley: Yeah, I think that point about testing the market with something as simple as a landing page or a signup page makes a huge amount of sense. I've certainly heard of successful entrepreneurs doing that but if you think that there’s a great market for Uber for dog sitting or something, simply putting up a signup page and seeing whether people actually sign up for information or not is a good test. If it’s out there and you’ve driven some traffic to it and the conversion rate is near zero, then chances are your idea is not going to succeed even if you code it out.

P. Vijayashanker: Exactly. That’s really what we’re trying to save people from doing. I think too often whether people are engineers or non-technical, they go down this path of, “I need to build something immediately, I need to get it out there.” They can spend a lot of man hours, a lot of money, doing this. Only to then discover that either the market was too small or the market wasn’t ready, or it just wasn’t clear what the product’s functionality had to offer in terms of benefits.

One of the main tenets of the book is to get to talk to customers initially, even if they're not ready to buy. Oftentimes, it’s better to go in and talk to customers before you have something because, one, you're not really seen as a salesperson. You're just sitting there in front of them with an idea. So they're certainly a lot more open to having a conversation with you versus you coming, sitting down, and saying “Hey I’m going to do a demo,” because immediately they think, “Oh my gosh, this person is trying to sell me something,” and their defenses go up.

It’s a lot easier to say, “Hey, I've got an idea and I just want to have a conversation to understand what your thoughts are around it and how you might approach a problem or the solutions that you're using.” It’s a great way to start building out that early adopters segment that you can then use to do further tests on your idea and on your product.

Roger Dooley: Poornima, say I've gotten to the point where I think I've got an idea. I've validated it somewhat either by conversations or some other feedback mechanism but I realize that I’m going to actually have to create some code or at least glue some pieces together that are beyond my capability of gluing together and I want to bring somebody on board who has the technical skills to do that. What should I be looking for? A technical co-founder? Should I be looking to hire somebody that understands the technology? What avenue would you use do you think?

P. Vijayashanker: So there’s a couple steps that I recommend people do before they get to the piece where they're ready to hire somebody or even bring somebody on to build. The first step is to actually go through and round up a set of customers. Some people who are actually willing to pay, even if it’s just a dollar, but they’ve basically handed you their credit card. So that’s the first.

The second is you’ve created a rough workflow of what the prototype will be like. Again, you’ve tested this on your customers. It can just be paper. I give examples of how to do that in the book. You want to work through that. The reason that I mentioned doing these two steps is because it actually naturally falls into recruiting somebody who’s technical because, one, engineers, regardless of where they are, whether they're in Silicon Valley or outside, get hit up a lot by folks saying, “Hey, I've got an idea. Will you help me build it?” So there’s no dearth of things to build for them.

What they want to understand is if they're going to spend their time, whether it’s time that you're paying them for, or that they're moonlighting, or for sweat equity, or whatever, they want to make sure that you're committed to your idea. So when you’ve taken the time as the entrepreneur to go out and talk to customers, get them to pay you money, and build a paper prototype, that shows a level of commitment to the engineer and it takes away a lot of the risk of them worrying about, “Oh my gosh, will this entrepreneur change their mind tomorrow? Will they actually go about launching it? Will we have people who are using it?” So when you take those initial steps, engineers are more receptive to wanting to work with you.

The second is it also provides a source of motivation because once they see you have people who are eager to use this product, then they’re eager to build it. They're like, “I’m going to build this, I’m going to launch it, and we’re going to have customers on day one.” That’s amazing because at the end of the day, they get a sense of pride from building and launching and hearing from customers that what they have put out there is beneficial. So that’s another key step.

I wouldn’t ignore those two steps. I wouldn’t jump right in and go find somebody technical because you're going to hear a lot of nos. You're going to hear people that tell you, “Sounds interesting, maybe you should go find somebody else.” Now, if you’ve done those steps, then it’s definitely time to go out and find someone and what I recommend is figuring out what your goals are. So if your goals are you want to bring somebody on full time because you really think that this is going to take off and you have the capital to make that happen, then you can go out and recruit an engineer, designer, and put a team together.

But if you're just not sure and you feel like you’ve got one more step to go, you want to build a small prototype. You want to put it out there and then you want to refine it. Then what I talk about in the book is how to go about hiring someone who’s going to just build out the prototype. That could be a consultancy, as in a software development agency that builds the prototype for you. It could be hiring a freelance developer but being very, very clear about what’s going to get built, the timeframe, and what your expectations are going forward.

I oftentimes think that that’s better than going out and looking for a technical co-founder because again, there’s a huge risk factor before you launch where the technical co-founder might be like, “I don’t want to give up my day job so quickly. I want to see that you’ve made a little more progress.”

In the book I provide a number of interviews and one of the interviews in it is with Melody McCloskey who is the founder of StyleSeat. Melody’s background is she’s non-technical but she has been a product manager before and she was bootstrapping her business. So she couldn’t hire a technical co-founder and instead she worked with a developer to build a very rudimentary prototype, launched it, started acquiring customers. That actually attracted the attention of her now technical co-founder. So it’s a way to think about it being a process, an iterative, rather than going out and trying to find the perfect person to build this product from the beginning.

Roger Dooley: That’s really an important point I think because I've seen these events that are like co-founder speed dating events, where folks talk for a few minutes and meet a bunch of people and hopefully try and find that co-founder for their business.

But often at that point, these are nothing more than very raw ideas. Where somebody’s got a rough idea where what you're saying is before you make that commitment to sort of a serious investment in a technical person, whether it’s a co-founder, even a full-time employee, consider trying to do something rudimentary with a contractor or agency or something to prove the concept so then first of all you perhaps have a better idea of what you’re looking for and also you’ll have the ability to recruit a better person than you might otherwise have.

P. Vijayashanker: Yeah, the way I liken it is I talk about building a house, right? Let’s say you’ve already got a house built and you’re remodeling it, you wouldn’t just go out there and look for the first contractor and hire them, right? You would sit down, you would explain, “Hey, this is the room I want to have remodeled. This is why we’re doing the remodeling. I'd like to get some quotes. I'd like to get a sense of what your past projects have been.” You look through references and you go through all of those steps because you're making an investment and at the end of the day, you're the one that has to live in that house.

So it’s the same process but people often feel like it’s a disconnect because it’s a product that’s intangible. They can’t touch it. They somehow feel like software is vastly different but it’s essentially the same thing. So you’ve got to apply the same process as you would when it comes to hiring a contractor or anybody for that matter. Thinking about referrals, looking at past projects, understanding if they're interested and how they could add value. Does it seem like it could form into a long-term relationship or are they just there to do the initial work and then move on from that. Because a lot of people do that or they're only capable of doing that where they're good at building the prototype but then if you need to scale, you need to bring in people who have a different set of skill sets.

Roger Dooley: Well, how would the, say, typical entrepreneur who doesn’t have a strong technology background identify those technical people who really have the skills versus those that just sound good. There's a lot of evidence showing that an interview process is a very imperfect way to hire people, particularly technical people. If you're hiring somebody for a sales position, an interview is almost like a work simulation so it may not be totally bad. But the hiring a coder on the basis of an interview isn't really all that effective if you don't know the questions to ask.

Years ago, I was in an IT outsourcing business and we hired a lot of network engineers. My technical skills were rudimentary compared to the people that I was hiring so I had a general idea of what I was talking about but I really couldn't ask the probing questions to see, is this person really savvy or are they not. Eventually I had people working for me that were able to sit in on the interviews or do their own interview and say, "Okay, yeah, this guy really knows his stuff." Or, "Wow, she's spooky, she knows more than I do, so hire her right now." But you know, up to that point, and again, for folks who are just getting started, they may not have that kind of person at their beck and call. How does the non-technical person choose a technical person who's the real thing?

P. Vijayashanker: Yeah, one of the pieces of advice I give people, and I know it's hard to take, is I often tell them I understand you're very interested in building this product and you have this idea but it would serve you well to go and work for somebody. I know that's a challenge for a lot of entrepreneurs because one of the benefits is not having to work for somebody, right? But even if you could do a small project, even a hackathon, or something where you're just getting comfortable with the process, then that can be a really eye opening experience.

I think if you are coming from another industry or another background and then immediately trying to jump in and build something from scratch, it’s just not the best use of your time. You’re not going to be successful. It’s going to be demotivating very quickly. So the more sort of baby steps you can take or the more practice you can get, the better off you are. Even if that means you serve as a sales person or a PM role or something else but in an environment where they're building tech products, you can learn a lot. That’s typically the types of people I see that served in business functions in a tech company and then transition to becoming an entrepreneur. Not just waking up one day and saying, “I’m going to be an entrepreneur in technology.”

I would highly advise people to take that initial step, even if it was for three to six months getting yourself comfortable. Or, like I said before, even just doing a hackathon or a side project. Once you’ve gone through that, I think you’ll learn a lot of the jargon. You’ll learn how people think and interact. The phase that comes after that and I think you’ve already hinted at is not necessarily to sit down and do a formal interview because, yes, you're not going to know the right technical questions to ask.

What I've found works better for both founders who are technical and non-technical is to work on a small project together. I mean, almost think of it as an internship. We do this with new college grads.

We don’t really trust them, we’re not really sure if they can do what we need them to do, so we’ll have them start off, work for two to three months, give them small tasks that lead up to a project and test their abilities. If they do a good job, we’ll invite them to come and work for us full time or whatever the longer term arrangement is.

I think you have to take a very, very similar approach when you're thinking about your startup and building a product. To work on a small project, see how that goes, see if you like working together. Because truthfully, a lot of the breakdown between technical and non-technical folks is not because of the technical side, it’s actually because of soft skills. It’s because of miscommunication or misunderstandings.

Sometimes that’s because the technical person can’t convey what’s going on in layperson’s terms or because they aren’t showing up on time or they're not delivering things in a way that is expected from the business founder. So all of these things you can iron out if you do a small project together rather than sort of jumping all in from the very, very beginning. So that’s one, I would say, interview technique that I suggest. You could even do it within a week or a weekend.

I remember when I was getting started with BizeeBee, I brought on an engineer. Being an engineer you would think, “Oh, it’s got to be so easy for me.” But I’ve gone through a lot of rotten apples. One of the things I did was I had him work on a very small project, I gave him about a week to complete it. I paid him for it. We set a small budget. At the end of the week, he just did a phenomenal job. He took it above and beyond what I had asked for and I thought, “Wow, this is the guy I want to hire.”

Surprisingly enough, within a year, he became my technical co-founder. So it was great to have him be promoted rather than start the conversation off with, “Yes, I’m looking for a technical co-founder. What can you do for me?” Right? That small project was a great way for us to test the waters working together.

Roger Dooley: Right. That makes a lot of sense. When you're entering into any kind of relationship, you wouldn’t propose to marry somebody on a first date. At least most people wouldn’t. But that’s kind of a behavior that does occur when you're recruiting people, whether it’s technical people or some other kind of person. You have a brief encounter and then you say, “Okay, let’s get hitched.” And maybe for an extended period of time. So, makes sense.

P. Vijayashanker: Yeah.

Roger Dooley: Poornima, some people suggest that schools should teach everybody coding. In other words, every student should learn how to program in the same way they might learn math or English or whatever. What do you think about that idea?

P. Vijayashanker: I think according to that, they should also learn some sales, some marketing. I don’t think there’s any one skill that’s indispensable. I think they're all necessary. So I think, sure, it would be great to have a rudimentary understanding of technology so that you know how it works. But again, I think at some point in time, we're not going to know all of the inner workings so it's up to you to decide what's important.

Are you a founder who's a strong leader who can delegate and hire amazing talent and keep them motivated? In which case, that's what you should focus on and getting some rudimentary knowledge is sufficient. Or is your desire to be that technical co-founder or be that CTO long-term? In which case, you should invest more time in learning the technology, in practicing, and working with engineers, and perhaps learning how to manage them. But it's important to get that sense.

Again, I had another interview with Alyssa Ravasio who was the founder of Hipcamp. She went through one of my courses. When she first started she had just finished Dev Bootcamp and she was kind of concerned because she felt like everything needed to be perfect before she could launch and I told her, "Having a prototype is sufficient." She built the version one, she built the prototype of her product and took it to a point where she could launch it and start making money.

But then she hit a sticky point where she needed to scale the product. She needed more designers, product managers, etc. She then took on the role of the CEO. The person who sets the vision and recruits talent. So it was good for her to have a little bit of that background and go through the program but longer term, she needed someone who was going to be the expert that was going to take the product to the next level.

Roger Dooley: I guess I'm a little bit skeptical of the belief that sort of anybody can be a great coder. I think there's not necessarily any intellectual reason why they can't but I think, well I know from my experience in the IT business, there was a while where for a period of time a lot of people were getting certified as Microsoft engineers or Cisco engineers and so on because that was a path they thought to good income, good jobs, and so on. But when I talk to these people they sort of fell into two camps.

There were the folks where you ask them what they studied, what they were doing with that knowledge. Some their eyes would light up and they'd talk about how they took some broken computers and cobbled together this home network. They were using the beta version of something and clearly they loved what they were doing.

There was another group of folks who were career changers who basically had not been in a field that was kind to them. Perhaps they were in manufacturing or something that was in decline so they said, "Okay, well, I'll do this network certification class." They actually did learn the skills, they passed the test, and so on. But these folks typically, the one or two times when we tried one of them out because we had no choice and were hoping for the best, inevitably, they weren't able to really perform in the real world.

You know, there was this divide between those who loved what they were doing and would just about do it without being paid and then the other folks who just thought, “Okay, well, here’s my 9:00-5:00 thing and that’s what I’m going to do from now on.” Do you see that same sort of divide in the coding area?

P. Vijayashanker: Yeah. I think it’s going to exist in any field. Certainly we’re now in a time where I think a lot of people are excited about technology and they’ve seen the rise of Apple and Facebook and Google but what they might not necessarily fully grasp is the level of technical prowess and depth of knowledge that took to build those companies.

So they might see somebody like Steve Jobs or Mark Zuckerberg as college dropouts but what it took to build those products were the folks who had gone and either completed four year degrees, or if they hadn’t, they have many, many years of experience in building and refining these products.

So it depends on what your goals are and you can’t just assume you’re going to take one class and become an expert. It’s the equivalent of me going and sitting in on a Spanish class for about a year and then assuming that I’m just going to become proficient. It’s not going to happen. I've got to continue studying the language. I might have to visit a country to learn a new dialect, to understand the culture. There’s a lot that takes place so people have to see it as, this is a skill that you hone over time. It’s not something that you pick up and a certification is nice but it is not everything, right?

Even if you take a similar analogy to medicine. Doctors go and they attend many, many years of school but it’s not like we hand them the scalpel and say, “Great, now start doing surgery.” They go through internships and rotation programs and fellowships and then they get to a level where they can maybe under some supervision be allowed to perform their first surgery. Even then, it’s been many, many years of practice.

The same thing is true for coding. You go out, you learn initially you might build a prototype or you might build an application or work on one feature in a larger product but that’s just the start. Then it’s, are you going to stay dedicated to continuing to learn and develop and hone your skill?

Again, that might not be your goal. Your goal might be, I want to be an entrepreneur. I want to build many companies. I want to build a business and therefore, I’m not going to get sucked into the details of building software and therefore I’m going to go hire somebody who does have that depth of knowledge. But at any point, you will need that person. You will need that expert. You can start but you're going to need to find someone who’s got more knowledge to take it over and to scale that product.

Roger Dooley: Well, let’s change gears for a minute. I originally set up to talk to you because of your software book but then I found your *Present!* book which tells technical people how to engage an audience or it will tell them. I was fascinated by the way you are launching this book. You used a crowdfunding platform, right? Publishizer?

P. Vijayashanker: Publishizer.

Roger Dooley: Publishizer, right, okay. Explain how that works and how that experience was for you. Did they drive traffic to you or did you end up, do you think, driving most of the traffic to your project there?

P. Vijayashanker: Yeah. So two things I'll say first. One, it's actually not just for technical folks, it's for techies. So anybody who's in tech who has to explain hard concepts to customers, to employees, to just anyone out there. The book is geared towards entrepreneurs, product managers, as well as technical folks like engineers. So I just want to clarify that.

The second is in full disclosure, I'm an advisor to Publishizer. They went through 500 Startups and I was an entrepreneur-in-residence there advising them. So that's why I decided to test out their product. The reason I decided to use Publishizer is a year ago I did a preorder campaign with my own list when I did the first book, *How to Transform Your Ideas into Software Products*. I did this campaign for about a week and I think I made about $900. I was very disheartened because I thought, "Okay, $900. I can't even buy a couple hours of an editor. Maybe I could do a really tiny print run." But it's just not enough.

A few people advised me and said, "You know, preorders, they just don't work." I thought, I'm not sure I believe that because I preorder books from Amazon. I know other people do. There's got to be something else behind this.

I saw a few months ago that Eric Ries did a preorder campaign and he did it on Kickstarter and had something like a half a million dollars and I thought, "Okay, I'm not Eric Ries but I'm sure there's a few nuggets of wisdom that I can learn from that experience." What I saw was that having the book on a public facing platform, a crowdfunding platform, did a few things. One, it provided social proof. There were a lot of people who were buying into initially because they knew who he was but then other people were like, "Oh, all these people I know are also buying into it. So I should also buy this book." So that social proof kind of has a snowball effect.

The second is you can see the progress of a campaign. So there's a goal. People want to help you get to that goal but they also want you to explain what the benefits of the book are. So it's a great way to sit down and bullet-point out why someone would want to purchase this book.

The final is having a campaign where there is a deadline pushes people to act very quickly. For us, with this book, we had a number of people that we were talking to who were thinking about sponsoring us and it was a challenge to get them to write us a check because they kept delaying things.

Finally, we said, we're running this preorder campaign. We need the funds to do a print run and to pay our designers and all this stuff and the campaign ends on this particular date and everybody respected it. Everybody followed up with us. So it was great to have that product, that platform, Publishizer, to get us to that milestone. It was very effective.

Now you asked the question, did they drive leads to us or did we drive leads. I think I certainly drove the most leads because I have a pretty big list. I have a list of a little over 7,500 people on my email newsletter. Then I've got about 14,000 Twitter followers. So I have a lot of different social media channels that I can drive people to but I think if I had just told them about it, it wouldn’t have been sufficient.

I needed that platform because that platform exposed a lot of things. It exposed the number of people who had already purchased. It exposed how close we were to finishing the campaign. It had multiple rewards. It had us also explain what the book was about and do a video teaser. So I think it was important to have a product and I wouldn’t have just been able to do an email campaign and get the same results.

Now we did generate some level of interest from Publishizer. If we want, they will introduce us to a publisher or literary agents. We opted out of that because our focus is get the book out, get it in the hands of people in the next month. Since we’re operating on a tight timeframe, we’re not leveraging them for those sorts of things.

Roger Dooley: I think one thing that's good about that, it sort of ties into your software idea too where here you are testing the concept before you actually create the full-blown product too. I mean, if nobody was interested if after four weeks you had five people say they were interested in the book that would be the feedback you needed to know that something was wrong with your idea.

P. Vijayashanker: Oh, definitely, yes. I'm a firm believer in testing everything and doing a preorder campaign is a great way to generate interest and see if people actually want to pay for the product that you're building. That was one of the major reasons that we did the preorder campaign.

Roger Dooley: Well, great. Let me remind our listeners, we are speaking with Poornima Vijayashanker, author of *How to Transform your Ideas into Software Products: A Step-by-Step Guide for Validating Your Ideas and Bringing Them to Life*. Where can people find you and your content online, Poornima?

P. Vijayashanker: The best place would be to go to Femgineer.com. I've got everything listed there.

Roger Dooley: Great. Okay, well we'll have links to that site as well as any other resources we discussed including Poornima's book on the show notes page at RogerDooley.com/Podcast. We'll also have a text version of our conversation there. Poornima, thanks for being on the show.

P. Vijayashanker: Yeah, it was great to be here, thank you.

Thank you for joining me for this episode of *The Brainfluence Podcast*. To continue the discussion and to find your own path to brainy success, please visit us at RogerDooley.com.