Podcast Episode #66 - How to apply lean startup principles to hardware, with Ash Maurya of Leanstack, USA

RAW TRANSCRIPT OF INTERVIEW

Balint: Ash, welcome to the podcast.

Ash: Yeah, pleasure to be on.

Balint: I read your book *Running Lean* after I came across Eric Ries's book *Lean Startup*. I love your book. It's very actionable, one of the most actionable books actually in how to scientifically manage innovation.

Ash: Thank you.

Balint: Here in Zurich I go sometimes to the Lean Startup, a meetup where you've also been to as far as I know and looking at the meetup thinking back how it was. Unfortunately, I missed that opportunity to meet you in person and look at your presentation but I'm glad to have this conversation with you now. Your book and concepts give important principles, very important principles. One doesn't have to run and start the company fat by being fat in a way so taking in a lot of investment from the start, jumping into the creation process providing a solution without thinking about the problem. You know when I was reading your book Einstein, Albert Einstein's saying came into my mind and I have this connection kind of with Einstein because he was a physicist and I have a Ph.D. in physics. That's why I think it's relevant to mention this quote.

Ash: Right. He was in Zurich, too.

Balint: Yeah, exactly. He was in Zurich, too. "*If I had an hour to solve a problem, I spend 55 minutes thinking about the problem and five minutes thinking about the solution."*

Ash: Exactly.

Balint: I think in your case it could be modified somewhat. It's not just thinking about the problem but actually talking about the problem, interacting with others. So I think it's extremely important your message and especially considering that in hardware the stakes are high when developing a solution that there can be high consequences if things go wrong when developing a physical product because simply there's just more commitment in the process. When you develop a feature, it's hard to make

modifications later. So talking about the Lean startup principles is especially crucial and relevant here. Can you tell us about the Lean Startup innovation framework for the listeners who are not completely familiar with it. I think we hear about this a lot in the media and a lot of people are talking about it but of course it would be amazing to hear it firsthand from you, one of the pioneers of this field, how you could brief and summarize it.

Ash: Sure. So it's a new mindset, a new way of thinking about products. And the thing I often like to point out is that it's not a new way because a bunch of people got into a room and said, “Oh, let's do things differently.” It's a new way because the world changes. If we look at the way we used to build products 100 years ago in the factories, there were very different challenges and realities and the methodologies we used then were the right ones for those problems. And as we transferred over to Waterfall and then Agile and then now Lean, they were all done in reaction to market. So it's not you know pioneers or thought leaders that just say, “Oh, let's all do this new thing.” So that's the first point I want to raise.

But what's changed is that the way we build products is very different today. So what you're describing as the fat way or the more resource heavy way of building products we used to work at a time when it was needed because building products, all kinds of products, was very, very complex and difficult. Even software, which now we have a lot of tools like open-source and cloud computing, it's become easier. But going back several years it used to be quite expensive and very hard, a lot like you know hardware today and even hardware used to be significantly harder, it's a lot easier because again there's a lot more components and thinking and that is just advancement in the way we work.

So fast forward today it has become easier, it's not easy but it's become easier to build products and the bigger risk now is understanding what is it that we should actually build. And that kind of trigger and that way of thinking is what sparked the principles that got codified into the Lean Startup. In the beginning it was a bunch of entrepreneurs in Silicon Valley, I'm not in Silicon Valley, I'm in Austin and so in Austin and other places beginning to engage in this conversation and that eventually became what got codified into the Lean Startup and we can talk about some of those principles but one of the big ones you touched on already which is that we don't any longer need to start with a solution, we can actually in the interest of going faster start with problems and then do a bunch of validation around those problems to make sure that we have the right customers and markets for a solution before we invest our own time, money and effort building out a perfect solution.

Balint: What are some of the hardware companies that have been using these principles? Because you know I took one of your courses, *The Continuous Innovation*

Roadmap recently. There I came across some examples like Tesla. But are there some other ones that you could maybe bring up?

Ash: Yes. So the point before going into even the hardware that I'd like to point out is that a lot of what kind of Lean Startup, the principles that I touched on are rather universal. They are universal because they can be applied in many different domains but how we apply them in different domains becomes very tactical. So I know we're going to talk a lot about hardware and so I would say that the principles are the same but sometimes the speed with which we move in hardware is going to be different than say software or services type of a company. At the same time, the point to raise here is that the new unfair advantage is speed of learning, speed of implementation. And as long as a hardware company is moving faster than its competition, that's still a huge thing, that's an unfair advantage. They can still win.

So I think the reason I bring that up is that oftentimes people say, “Oh, this works only in digital but we're doing hardware or bio sciences” or things where they are long cycle times, the thing that internalizes that companies that I will share here are different because they have taken the principles and then find tactics to go faster than their competitors. So Tesla you brought up is one of the examples I give. Now you know they don't go around saying, “We are doing Lean Startup” but you can see the principles and everything that they do, in the way they think and the way they tackle problems. So some of the principles and I think that may help here is when you're trying to solve for a solution rather than falling in love with the solution we instead try to think of the entire business model story. What is it that we are trying to do here? Who are our customers? What value are we creating for them and how do we deliver that value? And the reason we want to go quickly is because today there's also a lot more competition and then also going back to the speed of learning, if you can learn what to build faster and you stand to take more of the market.

So what Tesla did is they were building a hardware. They had a hardware problem to solve with the hardware solution in mind which was building a car. But the way they broke that down is by thinking in terms of what might be the riskiest assumption, what might be the hardest thing about building an all-affordable electric car and then they divided and conquered. So they started with the highest risk you know build their first car and we can talk about those tactics later. Your question was “What other examples?” So I would say other than Tesla we can see many other examples in the medical field, you can see companies like Philips, you can see companies like GE, they are taking these principles and applying them to building everything from medical says to jet engines. So those are all hardware examples. You can see companies you know at smaller scale, many startups hardware startups as well Pebble and others who have taken these principles and have found ways to again implement them in those contexts.

Balint: And do you see some specifics like tools what hardware startups use so that they can go with a higher speed than their competitors?

Ash: Yeah. So I would say the first thing and this is something we teach is that irrespective of what kind of solution you are building whether it services, software you know, digital, low-tech, high-tech, hardware, it doesn't really matter. The first step for everyone is the same. So we tend to get people to deconstruct their idea on a one-page business model and the tool of preference for us or choice for us is the Lean Canvas. That's a tool where you can begin to describe your idea, you can describe who is the customer, what you're going to build, why you're going to build it, so what problem is it solving so you can sort of begin to plan that out. The alternative used to be a business plan. But the problem with a business plan is that it takes too long to write, no one enjoys writing them, especially if you're a maker. So more people write business plans for funding but this document, the Lean Canvas, is more for designing, it's more for designing the product before you build it. So that usually is step one.

Step number two is doing some customer validation. So again, the big leap of faith here for most people is “I should just build something and people will love it.” But we don't know. And actually, most of the times people build things and nobody wants it. So what we want to do here to avoid that trap especially with hardware, to use your words, it's a very expensive proposition, it takes a lot of time, money and effort. So especially in that context what we would recommend that you do is actually going to test whether the thing you're building can actually solve a problem and you don't need to build a solution for that. You need to do some problem discovery and then you need to do some demoing and solution testing but not solution building. And once you do that the confidence level goes way up because now you know what customers really want, what customers really need and then you can begin to even scope down the hardware that you're building. So oftentimes people want to build a big product but after going through this process, they realize that they can define a much smaller initial launch that could be their minimum viable product. They could launch with something and then incrementally add onto it. Now there are strategies that are in place so when you are trying to do hardware specifically one of the things that's challenging and you touched on this is the difficulty of making changes and cycle time is important because as you get feedback once a product is out there you need to be able to update it.

So some of the strategies or tactics that people have used, one of them is just batching. So when people initially build an initial product they start with a small batch of products. So rather than building a million watches or a million units, you start with the small batch, use that in an early adopter setting to learn and then you come up with improvements from there. Others will build in infrastructures. If you look at Tesla, Tesla has actually built a hardware-software combo. And I actually heard even Apple

describe this once as they see the borders between hardware and software blurring. It's actually if you think of it as one thing you can do a lot of amazing things. And so Tesla for instance will sell you a car which is a bunch of parts but everything in the car can be programmed over the wire so they can give you new features, they can reconfigure things, they can make a car go faster, they can make it more efficient. And that's powerful because that means that they can change things without having to give you a new car. So that's another tactic that we sometimes see is building that into the infrastructure of hardware can be powerful.

Balint: You mentioned Apple that they see hardware-software border, the line between these getting blurred. And also, that you know that the minimum feature set that you should use, that you should build, we are releasing the product. And in your course, you mentioned the iPhone. I think that's also a good hardware example because when it came out in 2007, they emphasized three features like wide screen iPod, revolutionary phone and like breakthrough internet connection device and that was the big thing, the big USP, the unique selling proposition of value proposition that they had.

Ash: Yes. Yeah. And I guess in Apple's case it was even interesting because they have taken a hardware business like phones and made it a subscription business. So in the U.S. here you have an Apple upgrade program, you pretty much pay them a fixed monthly fee. I think it's $35-$40 a month and you can pretty much go there every year and just trade in your phone and get a new phone and walk out. And they'd handle the rest. They handle any recycling or reselling of the phone, you know refurbishing, doing all that stuff. But from a consumer perspective it's now just like you buy a software subscription or a magazine subscription, you're now buying phones. So you know yes, it's a physical product but they have changed…You know you brought in this business model innovation and kind of change the way that we even interact with hardware.

Balint: In the Lean Startup framework there are three stages that you define that have some major milestones connected to them - the problem/solution fit, product/market fit and growth stages and then during the development you make sure that you don't go with a really high speed but you go with a speed that is sustainable and you learn with the right speed so that you'd make sure that the stages have its milestones fulfill the requirements. So my question is can you give us some examples about some KPIs or some indications people, startups use when they want to make sure that they reach these stages?

Ash: Yes. So it's important to go as fast as possible but also don't just fall in love with the scale stage which I find a lot of innovators and entrepreneurs do is they worry about you know their 1000 customer when they don't even have the first customer.

So it is important and I find that and that is a trap, it's a premature optimization trap where we start you know polishing the product or we start building for scalability. If you were Tesla, for instance, you know launching a car is massive, for a new company there's so many things to do. You have to worry about the brand, you have to worry about, if it's electric, the charging stations, you have to worry about you know your retail center, you know all these things, your servicing, how you do all that. It's just overwhelming. You have to worry about the factories and the design, the automotive engineers. And so how do you tackle all of that?

So when you break it into the stages it actually gives you permission to scale in a controlled fashion. So problem/solution fit is all about testing. Does anybody care about my idea? Does anybody care about this concept? And so the way Tesla and I'll use Tesla because I think it's just easy to always bring an example in. So the way that Tesla did this is in the early days Elon Musk and others in the company would share the story of their vision. So they would share the concept. Over time they began to show some pictures of what the car might look like. It was essentially this Roadster that they were going to build. They then began to talk about you know the price it would have and then they got a number of people interested and saying you know if you can deliver this car, they would buy it. So problem/solution fit is all about the demo phase so you are sharing your USP or unique value proposition, you are sharing with your customers the concept, so a demo. It doesn't have to be a working demo but just a demo and then you're talking about price and if you get enough people saying, “I want this.” or “I will buy this and here's a down payment.” that’s problem/ solution fit.

I did the same thing with the physical product, it's not hardware but it's a physical product with the book. When I was writing my first book before I wrote a single page, I just put the table of contents out, I talked to a number of people. I said, “If I wrote a book that answered these questions and here's an outline with a table of contents, would you buy it?” And I got a number of people. In my case I got, you know, hundreds of people saying yes, they would buy it, and not just saying “yes” but giving me their e-mail address and ready to give me payment for the book. So that would be an example of problem/solution fit.

We haven't built anything but we have tested this concept in the sense that if I build this, will people buy and I am not just asking verbally “Will you buy?” We wanted to put some clear call to action. So the call to action might be you know give me money. If you have a Kickstarter campaign, another great example, many products hardware and non-hardware, many, many of them are physical. It's a great example. Crowdfunding in general is an example of a problem/solution fit is if enough people liked the concept, they will back it with their money. It doesn't have to be money. It can be

email addresses. If you're in a B2B context, it could be a letter of intent. But all those are examples of problem/solution fit.

Now once you get enough of those, enough is going to be determined by how big of an idea you have and this is where I will point to some content that we have on traction modeling and traction road mapping. This is where you can size the idea. If you were Tesla, you need to know how many cars you need to be able to ship now to be able to make this a big enough idea 3-5 years from now and to be able to answer that question you have to do a little bit of estimation and calculations and we have some tools for doing that. But that's how you would determine how many preorders or what interest level you need to generate for problem/solution fit. But once you have that, that gives you permission to start building the minimum viable product and then you spend several months, sometimes you know one or two years depending on…In hardware sometimes those time scales make sense. You know building and finishing out and then testing the MVP and then refining the MVP and that would be an example where Tesla would take their concept, turn that into a battery that actually works, put that in a car, ship that to their first handful of customers. And with their initial launch of the Roadster they shipped about 500 cars and that gave them confidence that the battery works and the car works. And then in their case they moved into their next scale. Their next stage was getting them to product/market fit and that was moving to the Model S which was their stage two, building their own cars and then starting to scale up from there which eventually became Model 3.

So in each of those stages there are specific metrics that are driven by a roadmap that you can create right from the beginning on how many units or how much interest level you need to generate or you need to see, you need to validate to convince yourself that you move onto the next stage. At each stage you are only tackling certain risks. So maybe just to summarize that, the problem's solution that is all about “Is there a customer and market for this idea?” and we test that with some of the demos and things we do. Stage two is about the minimum viable product – “If I build this, will I be able to deliver value to customers? Will they pay me? Will they get value from it? Will it actually work?” Those are the questions we're asking. And if you can do that, then you worry about scale, then it's all about growth and channels, partners, all those types of things.

Balint: I think that's a great example because you know Musk based on his background and his success in entrepreneurship, we might think that he can just start a company from scratch without validating things, just listening to his gut feelings and he is going to be right because he developed such a strong gut feeling and such good sense for innovation and what people want. But even he was doing things in stages, as you said it, starting with the Roadster where the body part was borrowed from the Lotus Elise. And then they had to do the minimal amount, minimum work to

basically electrify that car, that they already proved the demand. I think it was good that you mentioned that the product/market fit was the stage when they started building the Tesla S. And now this is the growth stage with the Model 3.

Ash: Correct. Correct. Yes. No, it's exactly true. So I would say that you know what's different is that as you get more experience you can take on bigger challenges. Of course, building a car company is not a trivial task but it's complex for anyone and you can see a systematic breakdown of high risk to low risk in a systematic rollout in the Tesla's story. You also see it in Facebook. You know unlike many social networks, Mark Zuckerberg didn't launch publicly. Now he didn't do it in the beginning because that was the design but because there was no funding so they had to do it with that way. Startups kind of fall into that small scale to large scale stage rollout but where we see people going astray are some of the bigger companies with resources. When you look at big corporates you know they want to do everything at scale and the Tesla story kind of teaches us that there's a pattern. If you can embrace staged rollouts, you can actually move faster. It's counter-intuitive but it seems like we should just go fast to scale in the beginning but by going slower you actually go faster in the long run.

Balint: In your course you're talking about some other concepts like new concepts and you mentioned now the traction modeling which is not in your earlier book, the first book that came out, the *Running Lean*, but it's in the next one, the *Scaling Lean*.

Ash: Yes.

Balint: And in your latest work you are talking about Customer Forces Canvas, Innovator's Gift. Could you tell us a little bit about the status of these latest thoughts and work of yours? And what can be expected there and what's going to help us innovate even better?

Ash: Sure, sure. So I would say that the trigger for the *Running Lean* book was this realization that all of us, all makers in general whether you're in hardware or software, it doesn't matter whether you an entrepreneur or an innovator, we tend to fall in love with our solution too quickly. We tend to lock ourselves inside the building, in the lab, in front of the computer and we are doing this work and then we release it and then realize, "Oh, we built something that maybe was not the right thing." So the first book was really all about this process of talking to customers. It was a conversation between the innovator and the customer to uncover problems who we are solving to talk about you know that's when I introduced the Lean Canvas and that's this modeling business, modeling tool to then go through the process of defining the MVP and that's where the book stopped.

One of the things I learned after that from my own experiences but also looking at other entrepreneurs who were beginning to practice *Running Lean* is that when they would do this they would get some early validation which was all positive. But when they came back, when they talked to their stakeholders whether these were investors or budget gatekeepers in a big company, they were not able to get the funding they wanted on the basis of this small validation. So it's almost like you know, “I've got 100 people interested in this idea you know give me money to build it.” It was hard for them to communicate that. So the stakeholders they wanted to see more metrics, they wanted to see the financial model you know the spreadsheet, the forecast and one of the dangers with the forecast is that there are just too many numbers. People get lost in the numbers and we start inventing and making up stuff like we do in the business plan just to get the funding and then the problem is that when we don't hit those numbers, then bad things happen. If you are a startup, your company can be taken away, you can lose your seat, you can be fired. So all those things can happen. So the second book was really a big dive into metrics. The question you asked which is “How do you tell when you are the milestones and the stages between them?”, that’s what the second book gets into. So that's where all the traction modeling, traction road mapping comes in and it takes the qualitative story of the business, of the idea that we capture on the Lean Canvas and turn it into a more quantitative model where you can begin to measure things and you can begin to communicate to your stakeholders and investors, “Look I am making progress and this is the progress I have made.” So everyone's kind of on the same page. So that's what the second book was.

The newer work, The Customer Forces is really a bit of going back to problems before solutions and that's become a new mantra for me personally but even for the company that I run stack is that I've found that it's so easy to you know to keep falling into the solution trap and so I wanted to find better ways for uncovering problems. And the Customer Forces Canvas is rooted in jobs-to-be-done thinking, maybe some of your audience has heard of it but it's basically this idea that we are using products to meet certain unmet needs and we are triggered to do them and if you understand the psychology behind it, so when you are hungry what restaurant do you pick. You know it depends on where you are and the context but that can be very, very interesting to understand. So this third kind of work is in that space which is if I describe it, it's trying to understand the psychology of why people by products and how they consume products and how they make choices. If you don’t understand that, it's a big qualitative but if you understand that piece, it's a bit fuzzy, but if you understand it you can then begin to build better marketing, you can begin to build better products that kind of fit like a glove, the problem/solution fit becomes a lot easier as a result of those insights. So it is newer work. So of the three, that’s the newest stuff that we're working on. There are some product that I kicked off called The Innovator's Gift

where I'm kind of sharing some of these concepts. So it's rather new but maybe in a year or two years it will kind of get codified into a book and maybe show up as a book.

Balint: Yeah. I had the luck of interviewing Alan Klement who's you know into jobs-to-be-done. And interestingly when I read his book, I read his book twice *When Coffee and Kale Compete* and he actually brought up your example, the Leanstack and how you uncover some further problems to solve. I remember this story. I think it was some fisherman or somebody like that was using your product, the Lean Canvas, like in South America and then he had problems... Maybe you can tell the story better.

Ash: Well, I would say that what I mean in many ways that's what opened my eyes to this that the importance for this kind of thinking is that even… We have taught everything so far about you know start small scale and then work your way toward scale. Along that way your customers are going to change, the customer segmentation is going to change, their needs are going to change. And if you don't keep uncovering that, if you stop talking to your customers, you can fall into this trap where you think you know your customers and that's a big dangerous place. So in our world like so we have this tool Lean Canvas, our early adopters were typically Lean Startup practitioners because as you said I started a lot of my early work there and so I knew that audience very, very well. I was actually one of them. So I would build a product for that audience and mine. And then this fisherman was what happened. So many years later we were kind of building along. We noticed some of our metrics were going down and we were worried and we were trying to improve them but we didn't know why this was happening. And so rather accidentally somebody sent me an e-mail but it ended up being a fisherman in South America trying to use our tools. And for me that was a big surprise because most of our audience were startups and there were entrepreneurs you know they were not fisherman running businesses in South America.

So I got him on the phone and I talked to him. And the thing I realized is that the job he was trying to do was very different than what our core audience was doing and our early adopter audience was doing. So early adopter audience were trying to practice Lean Startup and they were using our tools for that. He was trying to raise money for his business. He was trying to write a business plan and some of his friends told him, “Don't write a business plan. Go to Leanstack, go to Lean Canvas instead and do that.” But when he got there, he was completely lost. And so that prompted a trigger in my mind. I began to ask myself, “Is he just a fraction of our audience or are there many people like him because our numbers are going down?” And so we began to run many interviews over a two-week period and what we were surprised to learn is that the majority of the people that used to be that 80 percent of our audience knew Lean Startup coming into our platform. And at that time when we began to pull

and do that research, we learned that only 20 or 30 percent actually knew Lean startup. Many of them were like him, they were beginners.

So that's kind of insight for us that caused us to realize that there is a bigger job that we are trying to do here. There's a whole different mindset that people are coming with and that prompted us to kind of redefine the product and kind of make it simpler, make the language simpler. So all these things happened as a result of that one conversation. But as the story that Alan puts in his book because it's this realization that the job changes, the job can change over time. If you look at Apple, why the early adopters bought the first iPhone. Very, very different than why somebody goes to buy the iPhone 10. It's really mostly about the camera and the quality of the camera and you know many other things that they are now adding onto it. But the reason that somebody buys it they weren't very, very different than why they would buy it two-three years from now.

So uncovering you know those insights is very, very powerful because one of the things that we are in here for if you want to build a business that lasts, you have to continuously be uncovering problems. In the beginning to get to your first, let's say 100 customers, your solution is going to have problems too. And if you don't fix them, somebody else will. And guess what? They're going to take your customers away. So that's how disruption happens. So the way that you stay on top of your game is you have to uncover your own problems before your competitors do and build something better and remove some of your old stuff and replace it with something better before your customers get upset and leave. And if you do that fast enough, then you keep your customers. If you don't, then one day they're just going to get up and leave and you're going to be there with a disrupted business model.

Balint: This was for you I guess the trigger event that made you move in this direction, the jobs-to-be-done concept.

Ash: Correct. Yes. I knew about it in theory and it was interesting but this was the trigger event absolutely that may made me think about it a lot more seriously.

Balint: Yeah. Using what I really like this terminology that you actually also use in your course trigger event that we have to look at when people switch from using one product to the other.

Ash: Yeah. Correct. Correct.

Balint: Yes. So I think it's really good talking about all these Lean startup frameworks. But I would love now to move on to the ultrafast round which is a little bit more personal. Is it fine?

Ash: Yeah. I'm good.

Balint: All right. So this means that I would ask four questions and it'd be great to get relatively short answers. So if you could go back in time to the time when you were in your younger ages, what notes would you give yourself?

Ash: Well, I guess it would be the mantra I just shared with the audience here is that it's this notion of love the problem, not your solution. I spent many, many years trying to build and brute-force the solution. If I had known that it's a lot more effective to find a problem worth solving and then build a solution. It's like understanding the door you want to open and then building a key instead of building a key and then trying to open, find the door. So that would be the advice I'd give myself.

Balint: Yeah. I love that metaphor. Yeah. The second question is if you had to name a book, which one had the biggest impact on your entrepreneurial thinking?

Ash: Yeah. I guess for me that's a very hard question because I tend to read a lot and I tend to take ideas from many books. I don't have one favorite and one of the things that I do is I read in many different topics. So definitely when I was starting out as an engineer I could point to very technical books. But I'll say for a lot of people I find… I'm guessing a lot of the audience here is going to be technical. And so I tend to push people more in the marketing direction. So one of the books that I particularly like is *The Battle of Your Mind*. I guess for your mind. It's called *Positioning: The Battle for Your Mind* by Al Ries and Trout. They also wrote a second book *The Immutable Laws of Marketing*. They are very simple books to read but when I read them it kind of changed my perspective completely on how important marketing is. Before that I used to say marketing is just for sales people and marketing people is not what technical people do but everyone, at least in this new world, everyone needs to understand customer needs and that's a very good intro to understanding what marketing is really about. So I put that one out there but I can give lots of recommendations on books for different topics.

Balint: I will put it into the show notes these recommendations. I read actually these two books. I think these are really, really good books. Classical, old examples that they work with but still I think it's evergreen content and principles, for sure. Yeah. So the third question. I'm amazed by habits. I think because they can set us up for a great day or some productive day in our work or in our personal life. Do you have some routine that you have?

Ash: Sure. There are a number. So I would say that one of them that I try to do I tend to be very accomplishment driven so if I go to bed and I don't feel like I've created something or moved something forward, I can't sleep. It's very, very unsettling and oftentimes when we get busy with work you know that happens. You can get very distracted and don't get the thing you want to do. So my hack for that is I, and it's not just my, I know a lot people do this but I block my morning hours. I tend to wake up

early. I will block two hours off where there's no interruptions. Sometimes I even wake up before anyone if I need to get some writing done and so I'll do something that's maker driven because, again, I'm maker by heart so I have to create something every day or write something or push some software or push a feature out. So those are things that I tend to do very early in the morning. Once I get that done, then I can relax and then I can do more manager type stuff which is tend to customer requests or talk to my team, you know do planning sessions and that way I can feel okay when I go to bed. So that's probably the thing that I do the most that's a routine every day.

Balint: Yeah. I think that that's really good. I try to do something similar. In the morning doing some maker things so that you can have this kind of sense of accomplishment at the end of the day. And the fourth question. In your work because you do give presentations worldwide and you interact with a worldwide audience, what kind of a critical cultural differences would you pick that you wish you knew before and how did you resolve those issues?

Ash: I tend to find that there's more similarities than dissimilarities. So I completely appreciate that there are nuances in culture so I will talk about what we are doing on that second part. But I would say the reason I started Leanstack is I was more surprised by how even though we speak different languages, look different. We all fundamentally are human beings, we all want the same things and fear the same things and make the same mistakes. So we all get hit with ideas. You know we all go through this… So the mistake that I'm describing here I found them to be universal which is why I felt there was massive potential for launching a company like this one. So that's where the similarities are now. There are definitely nuances.

One of the things that we have learned is that there is always the “I can see how it works over there but it may not work for us because we are different.” and this would be sometimes... So like I used Tesla. I use Tesla examples. People say, “Oh, yeah, Tesla can do that but we couldn't do that in Switzerland or we couldn't do that in Germany because of the laws.” So you know people come up with those reasons. So one of the things that I think is important is to generate more case studies. So that's something we are searching for and I'll share it with everyone, anyone who out there is listening that wants to participate you know drop me a note at ash@leanstack.com. We are looking for stuff for more stories of these types of thinking that are coming from really not just Silicon Valley or not just the U.S. but really all over the world so that we can bring in some of those cultural differences if need be. But I find that it tends to be more psychological than actual. That's my thesis right now. It's just more of a thing of if we can share enough stories, then we can get through kind of this mental block that I sometimes see people having.

Balint: I think that's also important to bring up case studies. Some people get motivation out of it. Others might even get some useful tools or takeaways from these cases.

So I think we came to the end of the interview. I'm pretty sure we could keep on discussing for a longer time. Perhaps in the future we should have another interview especially when you see that you go ahead even more regarding these latest topics you are working on, the jobs-to-be-done and design thinking concepts. So any final thoughts you might have and what would be the best way for the listeners to reach you? At least you know you can mention again at the end your email address that you mentioned.

Ash: Yeah. Well, my personal e-mail address is just ash@leanstack.com and leanstack.com is now where I do everything so I used to have my own private personal blog. That's how I got started. It was just my name. But now I do all my writing at leaststack.com. We've put everything in there. There's all kinds of free content and then there's some tools you can use. We also have some paid and premium courses and workshops so if anyone's interested in digging deeper, that could be the place to go and that's where you can find a lot about me and also a lot about the work there.

Balint: Awesome. Thank you, Ash. I appreciate it.

Ash: Yeah, pleasure. Thanks for having me on.

Balint: Thanks a lot.