



Podcast Episode #62 – How a hardware SaaS company wants to help you not get lost in your data during product development, with Michael Corr of Duro Labs, USA

RAW TRANSCRIPT OF INTERVIEW

Balint: Well, welcome, Michael, to the show.

Michael: Thank you.

Balint: We got connected about a year ago as I saw at that time that you were very much into agile, agile framework, agile development. And I also think that's the future. Agile taking over both software and hardware worlds. Then we had some discussions. I read some of your posts at that time and also recently the newest ones, and we stayed in touch ever since. Now we have even more to discuss these topics even more openly via this podcast episode. So I'm really, really excited about this episode.

Michael: Yeah, me too. Thank you very much for this opportunity. These are topics that I've been talking about and thinking about and working on for 10+ years now.

Balint: Yeah. So the topics that we are going to talk about I think it's highly interesting for hardware entrepreneurs and even engineers who are working on design aspects and actually the whole team because in the end we should not work in silos. And your work is connected to the whole development, how you can optimize, how we can make the development smoother. But I of course, I don't want to jump too much into that topic but I just want to highlight that it's a really interesting topic and I can't wait to discover more with you in this episode.

Michael: OK, great. Well, thanks again for inviting me.

Balint: So to start off, what can you tell us about you, what you do now and how did your past activities, 10+ years of background, hardware background led up to this point?

Michael: Well, sure. So today I'm co-founder and CEO of a company called Duro Labs and we offer SaaS tools to help hardware product companies manage the product workflow from prototype through production. So the idea for Duro came from my own experience designing and manufacturing hardware. I've been working for



almost 20 years in various industries. I moved to San Francisco after finishing grad school back in 2001 and I've had great opportunities of working on products like telecom equipment, cleantech, wearables, drones, IoT devices, quite a nice range. I worked for several different companies leading engineering teams. I've had my own consulting business. I even lived in China for a little bit where I helped startups bridge that gap from prototype to production.

And so throughout this career there was a consistent pain point across all of these endeavors and that was basically too much of my time was being spent administrating the data associated with the hardware products, the build materials, the supply chain. So there's a specific product category called PLM, or product lifecycle management, which in principle is an immensely valuable tool and its goal is to capture all this information in an organized fashion and be the communication portal with your suppliers. And these tools and concepts are great for reducing risks, great for communicating, great for converging and getting all the teams together on the same page. However, in practice I just found that all the available solutions on the market at the time were just...They are antiquated, they're horribly inefficient. They are not really conducive to today's engineering culture and I was just spending way, way too much time using these tools for the value that they're providing. And so what ended up being the result was that it's always easier to revert back to simpler tools like Excel and I knew Excel wasn't the solution either.

So a little over two years ago I moved from San Francisco to Los Angeles and I had a nice break in my career, take some time think about what I wanted to do next and honestly just like all the stars aligned. You know I've been complaining about this problem for a while. I had some flexibility to do something new. And I said, "You know what? No one else really seems to be doing anything to solve this problem. Certainly, a real consequence." So I decided to stop complaining and just start doing something about myself.

Balint: And can you tell us more about what you're doing at Duro Lab?

Michael: Sure. So you know at a high level Duro provides a cloud based SaaS solution help hardware companies manage the design and manufacturing of the custom hardware products. Our solution is designed to be as simple as to use as a spreadsheet but it integrates a very powerful automated backend engine that assists with the data entry and validation to enforce hardware product development standards and best practices. So we help both less experienced and more experienced teams guarantee that they'll be able to bring their products to market on time and on budget.

Balint: Can you tell us some of the early adopters, some of the users or at least the fields where it's used?



Michael: Yeah, so we've had some fantastic early customers who are really gravitating, resonating with the product. A lot of our customers today are what we call the small OEMs, the 5 - 15-person teams, the Kickstarters, the Indiegogos. We have companies developing VR products, IoT, drones, even smart mattresses. There is really a need for these companies. They're kind of at the cusp where they know Excel isn't the right solution but they're too intimidated by these larger enterprise solutions like SAP and Oracle and so they are looking for something that's just the better cultural fit to what they need.

We are however certainly getting some traction even enterprise companies so even large companies are recognizing that they're trying to move fast during the NPI process, new product introduction, and their development tools are allowing them to do so but their administrative tools aren't. The SAP, the Oracle, their arenas are just too slow and can't keep up with the rapid iterations that they're going through during the NPI and so they too are looking for something that's more nimble and faster and allows them to do what they want to do.

Balint: Yeah but you mentioned that your product is like a spreadsheet but it's not an Excel. It doesn't have that simplicity. I think it just reminded me of Airtable, the startup that's up and coming. I'm using it myself for more and more of my work.

Michael: I love it.

Balint: I also love it. And we had some discussion before this interview and you mentioned the word like a single point of truth that it little bit reminded me of Airtable. It's good even for managing your inventory. I think even Tesla to my knowledge used it for managing their inventory and also WeWork.

Michael: The coworking space.

Balint: The coworking space. They use it for setting up their coworking spaces in an efficient way by documenting what kind of equipment for each area they have so that they can create the right environment, this positive environment that facilitates the people connect with one another. So it reminded me of that. But we had some discussion that you're solving even more problems with it. And can you tell us more?

Michael: Of course. First off, I also love Airtable. I think it's a fantastic product and I think they're really giving companies like Microsoft Excel and google sheet to run their money. We use it for a lot of our administrative tasks as well. So that's actually a great analogy because for the less informed we compared Duro to the combination of Git and Airtable. So it provides a simplicity in the power of a spreadsheet but it has a revision control the management behind it, the source control version management tool like Git. So there's a couple of different ways that Duro is differentiating itself from the market.



So first off just to kind of paint the picture. Our big picture vision is to be the global hub for all hardware manufacturing and commerce. So we're connecting hardware companies to manufacturing partners and suppliers throughout the world. And we're going after it in slightly different fashion than some of our predecessors. So as you alluded to you know there's currently one approach to solving this problem is using things like Excel and Airtable. And those are great because they're simple and everybody knows how to use them. However, they are not context specific. They have no automation. They don't have any overlay of hardware industry standards. And so the data that people put into it are only as good as the person managing it. So it still relies heavily on manual effort to manage it as well as expertise to know how to set up properly.

On the other extreme, you have as we talked about earlier these industry specific products from Oracle and SAP and NetSuite and even Arena. But as I mentioned they are just culturally not a good match. A lot of them were built in design decades ago and haven't really updated since then to match what today's expectations are. They are very complicated, they are bureaucratic, they are antiquated, there's a long onboarding times. Some of those products even advertise that it takes three to six months on board. And three to six months for a small team just getting off the ground is an eternity. They can't wait six months just to get their administrative tools set up.

So Duro is basically taking the best of both worlds. And as I mentioned the design is very spreadsheet-like so it's intuitive and doesn't require long onboarding times or a lot of industry expertise. But the backend is a very powerful database with automated algorithms which overlays these industry standards. So we heavily focus on automating the data entry and validation process to take the burden off the engineer. And our goal is to guarantee that your build materials and supply chain data is correct and complete. And so that when you're ready to head it off the suppliers you know you're good to go. There won't be any delays.

So another difference that Duro is bringing to market is kind of a philosophical difference between Duro and other BOM management products. So in the current ecosystem of tools you have your PLM, product lifecycle management, PDM, product data management, and ERP. These tools are very powerful and provide a lot of value but they are in essence what's called a garbage in, garbage out problem. If the data that goes into these tools is erroneous or incomplete, then any exporting for reporting functionality is also going to be flawed and useless. And furthermore, if you can't root cause a mistake in this database, then I would argue that the entire database is tainted and you have to scrub the whole thing.

So from my experience what the status quo is in all management tools is they really focus on the exporting capabilities and the recording tools and I think that's fantastic and there is a lot of value in it. But if the data that's used to generate these reports



isn't correct or isn't complete, then the reports have no value. So Duro really focuses on the data entry and validation process and we actually have built IP and patents around this. So our goal is to catch any mistakes or issues that could potentially become a risk at the time of data entry and we highlight those issues for users so that more efficiently they can address them right then and there versus weeks or months down the road when you actually go to build the product. Because remember in our industry it's very common for an engineer to pick a part or update a design today it will be three to eight weeks before they actually go to a producer. And over that course of that time a lot of time has passed they may have forgotten the context or maybe that engineer is gone. And so if you don't catch the problems until you export the data, it's very difficult to recover from it or it's very difficult to efficiently address them.

Balint: You do see this problem right in practice?

Michael: Of course. Yes. And again, a lot of this comes from my experience and my co-founder Kellan O'Connor who was a former SpaceX engineer. I mean he had the same problems when he was working at SpaceX. You know this problem isn't unique to small companies. Even large companies have these same issues. And so that's what Duro is trying to address and that's why I think Duro really has some potential for success because we transcend both small and large teams.

Balint: Yeah. You mentioned at some point, actually twice during this interview, the differentiation from Oracle and SAP. And that kind of reminded me of the topic that in CRM tools you have Salesforce but that one is like the behemoth, the monster to handle. And I see this that some startups are using it. I used it also myself at the startup where I've been doing sales and marketing. And I think it's not the right to start with. It's too expensive. If you want to customize it, you need to bring in an outside help, the onboarding is in a similar way long. And you know for a startup you have by definition an unproven business model so even the sales pipeline is unproven so you need to do customization and you're learning constantly. So it's not a good thing to have. You need the flexibility. For example, Airtable can give you such flexibility. I think it's wonderful for sales purposes and business development purposes, partnerships, documenting such things together with of course other tools because it gives you the flexibility. So in a way yours is also is more flexible and it's not something complicated to handle.

Michael: Correct. And one of the key factors is as we're trying to give power back to the engineers and so those tools as you mentioned like Salesforce and Oracle and SAP, you're right, those are very, very successful companies and they get all the credit they deserve. But their business models I feel are just not a fit for what today's engineers are looking for. Their business models require you to not only buy this expensive license but also require to buy and purchase or find your own consultant to



customize it and so it's additional cost. And the lead times associated with that are just, as I said, they are orders of magnitude greater than what teams want today. Engineers today want to be able to onboard themselves, get up and running and focus on what they really were hired to do which is to engineer good products. They don't want to spend a large percentage of their days just setting up or configuring their administrative tools.

Balint: Continuing this point, the differentiation, how do you see your product being different from other offers on the market, especially relatively new offers on the market?

Michael: Well, you know as I was saying we're really focusing on the data entry portion versus the reporting. We're automating a lot of the process. A lot of these other tools still require industry expertise and we're just taking a lot of the administrative burden off the engineers' hands so they can focus on what they want to do. I mean the analogies in the software industry are things like Git. You know every software engineer uses that tool but it's such a simple tool set up. You know within moments they can have it up and running and it's just there, it's a safety net. It allows them to focus on actually writing good source code and releasing products, not on administering the data. So that's the philosophical difference that we're trying to get to. We're trying to get to the point where today software engineers the very first line of code they write I you know I joke is Git in it. We want to get to the point where hardware engineers the very first thing they do is Duro in it.

Balint: And a little bit changing gears. How do you see in a wider sense the state of hardware development, especially hardware using electronics? Because you have a lot of experience in electronics. What topics make you excited?

Michael: Well, to be honest I'm really excited about the activity that's happening in the hardware industry today. It was definitely dormant for a long time but it's going through its own renaissance moment now. It took a lot of catalysts in advance to get to this point. What I think is one of the major catalysts was the convergence of hardware and the Internet, there are essentially IoT products to get people excited about the value that hardware itself can bring to our lives. With that along with the reduction in various entry in hardware development have really stimulated this industry. So things like 3D printing or even Arduino or other rapid prototyping technologies has now lowered the barriers entry and so people who have less traditional educational background in hardware can now be equal participants and consumers in this space and because of that the market's just exploding, there's lots of great products that are coming about, new companies I feel are popping up every day. You know I've been attending CES for the past few years which is a major hardware conference in Las Vegas. The startup Eureka Park Startup area has just grown tremendously year over year the number of companies that are presenting.



But there are still problems, there are still issues. So people are definitely entering this market and new products and new companies are starting up every day. But there are still issues. You know we're not there yet. You know there's still a large gap between the prototyping stage and the mass production stage. So while prototyping tools and expertise has gotten better and simpler, the mass production technology and processes haven't improved at the same pace. So there's still a lot of inefficiencies, it's still complicated, it's still intimidating and this to be honest causes a lot of good companies to fail. What we need is the same magnitude of improvement in the mass production capabilities as there has been in software technology and prototyping technology before the hardware industry can really take off. And it's certainly coming and we're on the cusp of it but we're not there yet.

Balint: Yeah. Recently we've had the Apple special event and there they announced a couple of things. And I think even Apple sometimes is struggling with mass producing things. For example, just like you know last year they were talking about the wireless charger. I think it's called "air" something, AirPower for charging your devices like the Apple Watch, your iPhone. But in this announcement, they somehow didn't even talk about that. There must be a reason behind it and they had the prototype, low volume produced I think to show unit. But if big players are sometimes struggling, startups can also struggle with mass production.

Michael: I completely agree. You know Apple they've had some issues along the way. You remember Antennagate. But even more recently companies like Samsung if you remember the Galaxy Note 7 battery fiasco. That was reportedly a two-billion-dollar loss to Samsung because of this battery issue. And so Samsung is very well respected company with some top, top talent and they have great products but it just shows to show you that these problems are not isolated to just start up through inexperience. Even the largest companies with the most experienced teams still have manufacturing problems. And again, it's because in my opinion the production capabilities and technologies just aren't where they need to be. There's too much room for errors, too many inefficiencies, there's still too much manual maintenance of the data of a company's recipe.

So a unique thing about hardware compared to software is you know a software product you know a team building a web app or a mobile app you can basically have the same three to six software engineers developing the same product and the initial prototype all the way to a product launch. And so while documentation is important it's not as high a priority because it's the same team. But that's not the true for hardware. In hardware no product can be built from prototype all the way through mass production by the same team. Inherently there's different disciplines, there's electrical engineering, there's mechanical engineering, there is packaging, there's maybe firmware. But more importantly the team that designs the product is not the same team



that manufactures it. And so for that reason the reliance on the documentation that defines the recipe for how you build your product needs to be accurate and it needs to be complete and it needs to be in sync between all these teams and if it's not, stuff happens.

And so today our industry works on email and Excel. Unfortunately, it really does because again it's the simplest tools to use. Everybody has e-mail, everybody knows how to use Excel. And so because of that it's still a manual management effort. And so people make mistakes, they get lazy, they cut corners. You know when they're in a rush to get a product out, they skirt the details. And until that ecosystem is being converged, until it's all digital and it's all centralized we're always going to have these inefficiencies and these inefficiencies promote risk and they ultimately lead to mistakes. And the other unique thing about hardware is there's no CTRL Z. There's no undo. So if the wrong recipe for your products gets to the manufacturer and it doesn't get caught and you have 10000, 100000 units coming off the assembly line with the wrong build materials inside, it's very hard to undo that. You know and in some cases yes you can rework them at a cost but in many cases they just have to go straight to the garbage can. It's not like you know a web application where you can just you know switch branches or just quickly revert a change. And so the cost and impact of these mistakes is huge. And we really need to get to a point where we remove the human in the loop because that's what's slowing us down.

Balint: Yeah. So digitalization is even becoming more important. And yeah, taking out the human mistakes, risk for mistakes.

Michael: The point of vulnerability right now is the human.

Balint: Connected to this, to the hardware development, how do you see the state of agile development? I had on the podcast earlier at the beginning when I started out Joe Justice, the president of hardware at Scrum Inc., you know the company that started the scrum and we talked about this agile development. And how do you see the status there and what are some of the bottlenecks for agile development?

Michael: Sure. So we're going in the right direction. I'll say that for you know true agile for hardware. So how I see it you know the principles and the concepts behind agile, agile sprint are really to create these type feedback loops – design, develop, test. Design, develop, test, rinse and repeat. The goal being you incrementally make progress forward evaluative of what you're providing is of interest to your clients and then iterate or change as necessary. And so that you're constantly getting feedback on these short bite sized timescales. Hardware we have that loop but the timescale isn't the same and the testing capability isn't the same. And so what happens is so in hardware our lead times are inherently days, weeks and months whereas software it's seconds, minutes, hours. And so because the software team can do a full agile



sprint within a matter of a couple of minutes it gives them a lot of freedom to experiment because the cost of failure is low. If it's not right, they lost five minutes, ten minutes, maybe an hour. It's not a big deal, it's not a big consequence. So it gives them that room to be creative and try new things. Hardware because our timescales are still much longer you know days, weeks, months, we're not as free to experiment because the impact on making mistake is much greater. You can't recover a month in time or the cost associated with that. If you designed a mechanical part, went to machine shop and it took two to three weeks to come back and it was wrong you lost two to three weeks. Hopefully, you were able to learn something from that. But if the part itself wasn't built correctly and you can't even use it for a test, it will provide no value. And so because the lead times today in hardware prototyping manufacturing are on those orders of magnitude I feel like it's holding us back from being creative. We're a little too constrained and we don't take as many risks and so opportunities aren't opening like they need to be. So once those lead times get reduced down to hours and days then I think that's when we'll be able to truly unleash the creativity that the hardware industry's on the verge of reaching.

Balint: And do you see some developments, some good signs that this is actually happening?

Michael: I do, actually I do. There's some fantastic work being done in this space. So obviously 3D printing was probably the first chip to fall towards reducing these lead times. You know today for not much cost someone could purchase a 3D printer, have it on their desks and within minutes to hours they can design, build and test the mechanical part. And that's fantastic. And so that is certainly contributing to mechanical engineering being much more agile. Electrical isn't quite there yet, certainly to having something on your desk next to your laptop. It's coming and I've seen some good inroads into that but it's not quite there.

But where some progress is happening is on the supplier side. So not that long ago you know it's pretty common if you have a circuit board or a PCBA to be prototyped or a machined part, CNC milled you know it would take one to three weeks, you'd send an email to a supplier, you'd send them the design files, they'd send you an email back, maybe a couple of calls you know back and forth, they'd ask you to make a change. Basically, one to two weeks will go by from your first point of contact when you finally get to an agreeable quote and design and you are ready place the order. Today there are some companies who have completely digitized that process where you can upload your CAD files to their website and within minutes without engaging with any human you can get a quote with some DFM feedback you know different options and you can place the order yourself. So there's been some great progress in this space on the mechanical side for companies like Protolabs and Fictiv. On the PCBA, on electronic side I know Tempo Automation is also working in this space



shortening these lead times. And I know half a dozen other companies were also working in a space and doing great work and are not that far behind. And so these suppliers are definitely contributing to shortening these lead times which again I feel is what's holding back industry to reaching its full potential.

Balint: I had Fictiv before on the podcast and they work only so far in the Bay Area and they deliver in a very short time 3D printed parts. But also they're into CNC milling. And I am also really excited to see such developments. Protolabs they're doing injection molding. This is how they started out with plastic. Now they're doing a lot more things, buying even companies. In 2007 they bought a company. It's going to be quite interesting these developments. And electrical engineering I've had some listener who I had a conversation with a startup and they're working on the electronics side helping designers so in a way getting rid of the design houses and they are with AI, using AI. So I think there is going to be quite a few developments coming. But of course, it's somewhat slow because the industry is relatively slow and just too many things that have to be shaken up. It's not like software that is already more agile. So also the response to such changes is just faster. But I think it's exciting. You know when there's a problem, such a big problem, it's a huge opportunity.

Michael: Yes, I agree. And I think also there's some cultural shifts need to happen. You know there's some of these tried and true suppliers or you know the lack of a better way of saying this is just old dogs and they are just stuck in their way. And that model I just feel doesn't work anymore. You know having to call the salesperson and negotiate over a week or two to get to a price you want. As I said earlier in the show you know engineers want to be empowered to manage it themselves. And so these companies that we just talked about Protolabs, Tempo and any other supplier who's digitizing their coding in order placement they're giving the power back to engineers. And that's what I just feel that today's culture wants.

Balint: And where do you think agile way of working could be especially applicable? In which part of the hardware development?

Michael: Well, certainly earlier on. So in the traditional lifecycle stages early on you have the EVT, the engineering validation testing, and DVT, the design validation testing that's collectively the kind of the prototyping early stages of life cycle and that's where you want to be creative. And that's where you want to be able to move fast and try new things and take risks. And so agile for hardware I think will have the most impact on the earlier stage of the product lifecycle. As your product matures you certainly still need the flexibility to move fast, certainly as problems come up and every company has an issue like when they go and they think have all their problems solved, they're leaving the DVT and they're getting ready to go to PVT, the production validation stages. But something comes up invariably and so you want to have the flexibility to respond to it quickly, make a change quickly and move forward. But



you don't necessarily want a lot of flexibility or you don't want to be making changes as you get to the later stages because you really need to be converging on your design at that point because that's when the manufacturers get to effectively take over and start doing their job well and they can perform much better if your design is stable and it's not changing. So you want to allow them to do that.

Balint: Yeah. The contract manufacturers, the CM, they're not happy species when it comes to changes.

Michael: No, exactly. And that's what I think I often have to explain to less experienced hardware teams is suppliers hate change. And especially engineers who come from a more traditional software background where they literally can be making a change last minute because they have all these fantastic automation tools like continuous integration and continuous deployment in all these automated test suites that they know if they make a change, the test suites will catch it and so that doesn't exist yet for hardware and so engineering teams can't be making changes last minute because it just kind of sets up the processes or it kicks up the processes that the suppliers have to take. The suppliers ultimately take the risk of making a production mistake and so they don't always appreciate or know the impact of your change. You may know, "Hey, it's just a small thing. It doesn't really impact something." But your suppliers don't know that. And so to cover themselves they still have to go through a litany of checks and processes that just has a lot of overhead associated with it. And so it's always best to minimize your change in your designs as you get towards production.

Balint: Yeah. And what about the testing part? You did mention it that's the other bottleneck. We discussed now you know the lead time part. What about the testing?

Michael: Yes. The other tenet of agile is being able to test and obviously the more you can automate, the richer your tests are and the more effective they are and more efficient they are. And so what's another unique thing from hardware to software is software has standard programming languages. And each of these languages, these modern ones all have a corresponding test framework like an xUnit framework or you know there's more and more testing tools coming out every month now I feel. But either way they are still standard. And so the interface between the test framework and your actual code is standardized it's understood and so within minutes you can be writing code and have a test framework up and running and immediately you can be designing and developing and testing and start creating that feedback loop.

Hardware doesn't really have those same standards. By definition every hardware product is unique. And so the physical interface of those products has to be customized. Now there's certainly standards and what you should be testing and how you should be testing. And certainly, there's some fantastic testing tools and capabilities



but the interface between them between your test equipment and your product needs to be customized. And for mechanical testing I don't want to oversimplify it but it is pretty straightforward. You know you have a physical impact test, you have environmental tests, or certain types of stress tests and those interfaces are pretty simple. You know I can easily take my products and put it in an environment test chamber and start running a series of tests or put on a vibration table and start running a series of tests.

But the electronic sprint circuit boards those are a more complicated interface and those obviously are unique. Traditionally what's done is what's called a bed of nails. It's kind of this clamshell fixture that we usually put on the end of the assembly line and the goal of this fixture is to verify that each individual circuit board is functionally working properly after being assembled. There's just enough variance in the assembly process that there's some probability that the boards have a failure rate. But the problem here is while again these processes are pretty standard and anyone can look it up on what's a good strategy to build these things but these tests picture themselves as a bed of nails. They talk about those themselves are custom, they are products in and of itself and they take a long time to design, develop and test. You need to test the test. And a lot of people underestimate the amount of time it takes to build those products. Sometimes they are more complicated than the product testing itself. And so again because of that it contributes to the lead times, people will just skirt details like, "Well, these test fixtures cost too much or take too much time. We just want a small test." Or they'll do you know minimal testing.

And so once there's kind of out of-the-box solution I feel for circuit board assembly testing that could be set up and run in minutes then that too will be a major barrier that will be broken down to allow more agile processes.

Balint: Hopefully some of the listeners will get excited about this opportunity and will jump on this.

Michael: I hope so too.

Balint: Let's see. Regarding your product that you are coming out, you already discussed quite a bit but another aspect that I'm still interested in learning about is the revision management. You touched on this briefly. Can you elaborate on this? Because I think it is important. In software like Airtable, even there there's a revision management and in hardware I haven't seen something like this, for example, for CAD design.

Michael: Yes. Thank you. This is actually another very important topic to me and I think it's also holding the industry back. So version management is pretty much standard operating procedure for software development. What I can't figure out is



why hardware engineers don't do this as predominantly as software engineers. So there's no modern software developer who doesn't set up their Git repository as a very first step before running any lines of code. But the same paradigm doesn't exist for engineers. If you think about it, your CAD files, your mechanical CAD or your electronic CAD, that is your source code. It should be managed. It should be revision controlled. It's no different than a software engineer's JavaScript files. It defines your products. And yet really, I'm at a loss why this concept the revision management isn't as prevailing and to be honest, if any of your listeners know the answer, please contact me.

I do understand that there are some inherent differences. You know I think, this is just hypothesis, I think part of the problem is that hardware design files, CAD files are traditionally binary files and often a proprietary format. So the existing common version management tools like Git and Subversion don't handle binary files as well. They can't see inside the files. So a major value add that these version management tools provide is what's called a diff - being able to identify the differences between two versions of the same files. Or now, branching is a very popular and common best practice and so merging branches, you know Git does a very good job in merging two text files when you're merging a branch but it doesn't even try to merge and you shouldn't try to merge a binary file.

And so it's a hypothesis but I think that may be a reason why the version management tools aren't as good or aren't as used as predominantly in hardware development. There is just to note though you know not to dismiss this. There is a product category called PDM, product data management, which in essence is used for version control but I don't think there are any in the market. So these PDM tools are first off, they're often offered by the same CAD providers and so they only work with the same company's proprietary CAD files. And so there's no single version management tool. You have to have a different PDM for each of your different packages. And second, they're notoriously complicated to set up and maintain. I hear tons of complaints that people don't purchase these packages because they're just complicated. Going back to that talk that we talked about before that a lot of the tools and the hardware industry are antiquated and they just haven't kept up with the times. Today's engineer wants something simple. They want to be able to set it up and run you know quickly and they don't want to have to deal with it. You know as I said earlier with Git or even Subversion I can download and install Git on my laptop and be up or running in minutes and it will work with any ubiquitous type of file. And these PDM tools just aren't there yet.

Balint: Yeah. Let's hope it's going to change because it's not a nice thing that you cannot do a version management for design.

Michael: I agree.



Balint: So moving on to the ultrafast round. I'm going to ask you four questions. It'd be good to get relatively short answers. Ready?

Michael: I'm ready.

Balint: Good. First one is about time travel. If you could go back to the future into the past, when you were in your early 20s, what notes would you give yourself?

Michael: I'd definitely tell myself to get out of the office more. I had a fantastic job. You know I left grad school is working at SRI International and working on really hard problems, more on the R&D side. And I was a kid in a candy shop and I think I definitely spent too much time working because I just loved solving problems. It wasn't till later in my career that I recognized the value of networking and talking to peers and learning from them. You know there's just as much to learn from your colleagues as there is to solving an engineering problem. So it took me a while to recognize the value of networking and that's an advice that I would give to my younger self.

Balint: Yeah. I think a lot of engineers including myself because I have an engineering and the physics background, I also spent more time in the lab, in a dark lab rather than being outside and talking to people. And this is what I'm doing now.

Michael: A quick anecdote. Later on, another company work that was a wireless lighting controls company I used to work such late hours and I've got a reputation for that that I remember one night the alarm in the building went off at 2 a.m. and the VPs or the executives were called from the security company to see what was going on and they just assumed it was me still working in the office. It's not quite the right reputation to have.

Balint: Yeah, it can happen, for engineers especially. So the other question is if you had to name a book, which one had the biggest impact on your entrepreneurial career and thinking?

Michael: Definitely a book called *Inspired: How to Create Products Customers Love* by Marty Cagan. You know this was my first introduction to agile and how companies develop products and how the different teams that a company engage with each other and what their responsibilities are and what their interfaces are. That was definitely a big eye opener for me to kind of advance my professional career and know to expect from the different teams even myself. You know as an engineer with the engineering team I was responsible for but also how to know what a product manager is responsible for and even the sales teams. So it allowed me to engage with those other people in the office much better.

Balint: I will have to look at this book. I will put it in the show notes for the listeners. Another question is I'm amazed by habits and how these can have an effect on us, a



good effect. Do you have some kind of routines or habits, a morning routine or daily routine during the day in your private or professional life?

Michael: I try and get a good regiment. I'm a big believer in process. I think by having a repeatable process that's how we improve. Personally, I use tools like Evernote to take a lot of notes and I try and go through them on a regular basis to remind myself of action items. I try and read as much as I can. It's not always easy and often time I find to read as until I'm laying in bed and going to sleep. And I often fall asleep with the book on my lap having read the same page over three times.

Balint: I also use Evernote and though I also believe in the processes. I think especially when you work for companies that you get into this, into this process minded way of working. But I think it's especially useful even for startups which are less process oriented because you have to have some kind of structure.

Michael: Yes. I agree. And that's a key thing too. I want to emphasize as you know I've worked for a handful of startups and I love doing so and part of the fun of working at startups especially early on is it's so organic. Like everybody has an impact to what the culture will be or what the processes will be and you should embrace that. So you definitely shouldn't bring process into a startup too early because you can squelch some of that potential. But there is definitely points when you should bring in process or at least capture the organic process that has developed into a more wrote and written down procedure because that way you can build upon it and you can repeat it, you can evaluate it and you can improve upon it and that is the way I feel that startups really can scale and that's how where some startups fail as they don't repeat or build these procedures into their culture.

Balint: There is a saying by Peter Drucker. Something about measurement. If you don't measure it, you can't optimize it. And I think if you continue this thinking if you don't have a process, you cannot measure so you cannot optimize.

Michael: Yeah, I agree completely.

Balint: Yeah. And the last question would be in this ultrafast round in your work if you had to pick one or two cultural differences, which one you wish you knew about before and how did you solve those issues?

Michael: That's an easy one. So my current company Duro Labs was a pure software company and so this is the first time I've worked and managed a pure software team. In the past I've managed hardware team which includes electrical engineers, mechanical engineers, firmware developers. So I've definitely over my career learned to improve my skills on what each of those engineer visionary teams needed and how they interfaced and what motivated them. But when I started managing this web team was very different. I actually made the mistake of thinking the analogy would be



the same as the firmware teams have managed but there is definitely cultural differences for a web team, the front end and the back end developers, the UI designers, the UX designers. It took me a while to understand their own goals and interfaces and what motivated them. And so we did stumble a little bit in the beginning you know as I learned to work with them but I think we're at a fantastic place now and I'm really happy with the team at Duro.

Balint: I think it's unique that you've been in both worlds and you're digging deeper and deeper into this software world and trying to help hardware entrepreneurs. So it's a really good way to expand your mind and your skill set.

Michael: Yeah. Well, honestly it's a way that I can give back. You know I learned so much in my career from others that I've some fantastic experience and I think some good ideas on what it takes to be successful in hardware. And so Duro in essence is my way of giving back to the community and contributing to the culture as a whole. You know the saying though the rising tide raises all ships.

Balint: And being in both worlds is really helpful because a hardware cannot exist without software and also software is in a way for modern high-tech projects like the Tesla software can't exist without hardware. They support each other. We came to the end of the interview. So with the end of the ultrafast round, one last question is about reachability, your reachability. How could the listeners reach you - by email or social media?

Michael: Any of those will work. You know I'm on Twitter, LinkedIn, Facebook – Duro Labs, or you can reach me by email michael@durolabs.co. Our website www.getduro.com.

Balint: All right. Again, I will put it into the show notes. So that the listeners can check it out and contact you.

Michael: Thank you.

Balint: I appreciate it. It is really informative. And again, I appreciate it. It was informed and a very inspiring interview.

Michael: Well, thank you for giving me this opportunity. These are topics that are very important to me and I'm very passionate about. So thank you for letting me talk about them.