Podcast Episode #064 – To transform or not to transform your hardware company to an all agile one next year, with Balint Horvath, Switzerland

Balint: Welcome to the show, to the podcast. As you know by now, after many episodes I published, I talk about stories, tools, trends, tactics, strategies that my guests or I think are important for building a hardware startup. This time again I'm on the show after the last episode and this episode is kind of the continuation of the topic. What should you do once you know that you have a problem worth solving and you verified what solution you should build, together with all the features? Should you just go ahead and develop the solution based on the specs that emerged, the whole development potentially taking years?

Perhaps you're new to development or you have loads of experience. Typically plan-driven or waterfall has been the norm for a long time, when you develop something based on an initial specs and you break down the tasks into project phases. This has been around for a long time both for sw and hw dev. Whereas for sw the dev has become a lot less plan-driven for the last 20 years, in hw it's still mostly the standard way of making things. But should you really continue using that method both for your development and other parts of your business when there are more superior frameworks, techniques are available?

What is some framework that's being adopted by more and more organisations these days, such organisations as Tesla, Bosch, Mercedes Benz, BMW, Boeing, Saab Defense or also some of the smaller companies, startups? These organisations realized that you cannot keep developing and operating in a rigid plan-driven way since the whole world has turned upside down, speed of working has increased and if you don't come out onto the market with a product faster than your competitors, you lose. The question nowadays should not be, should I speed up but what happens if I don't? It is true that changing a running system is dangerous, tweaking the way you work is risky, but if you don't even try it, you risk being out-innovated, becoming obsolete. I kept you waiting I guess long enough,...I'm talking about agility, agile frameworks, and its most widespread implementation, scrum. We already talked about this topic in Episode 10 with Joe Justice, President of Scrum@Hardware at Scruminc, the company led by co-

creator of scrum, Jeff Sutherland. Actually that's the most downloaded episode of my podcast of all time. So there is actually interest towards this topic by you, the listener. If you haven't listened to that episode, I encourage you to do so, because it's a very motivating one and it's a good intro to this topic. Another episode where we touched on this was with Michael Corr of Duro Labs just a couple of episodes ago, episode 62.

Now in this episode I wanted to talk briefly about some aspects of scrum. I've recently been lucky enough to visit a high-tech hardware-software company here in Switzerland which is going through an agile transformation. They're transforming not only the way they do development but also the rest of the business is affected, sales, marketing, how management works. The team has grown to a size of a few tens of people, so it's no longer a small company and they realized that projects were delayed, there was no real synchronization among teams and team members. One of the reasons was that teams were organized according to functionality, so mechatronics people were in one group, sw developers in another group, and so on. They were developing solutions to their customers who cared about the end product and from a lean point of view things that don't count very much towards giving value to the customers, those things are waste. It turned out that the big disadvantage of their previous structure was that accountability was on a functional team level, also the distribution of tasks, and actually building the product required skills that crossed through different teams. In team meetings individual efforts were discussed and such efforts were shown off instead of focusing what the team achieved together for the customer. Scrum solves this issue by creating cross-functional teams, so teams that have all the skills necessary to get done with the work. So they reorganized the company structure to have cross-functional teams instead of functional teams. Another issue was the plan-driven nature of the organizing work. If you set the specification of your product and start to develop, but later the environment changes, to the market, competitors come up with a competing solution or customers change their mind, plan-driven way of managing things would require you to continue developing the solution. This means, let's go into the wall. Let's build another Nokia phone and drive into the wall with that, which is what happened to Nokia when the iPhone came out in 2007 - they didn't take into account that little change on the market, caused by Apple. Scrum as an agile tool overcomes this issue by welcoming change, welcoming interaction with the customer.

The company that I visited had several teams developing different products, one team developing a product for a given market segment, for a given application field. I liked the maturity they reached. Teams were collocated, they used a kanban board where the work for the current sprint was displayed, so the focused work for 1 or 2 weeks or a month, depending on the length of your sprint. Each team had a somewhat different

implementation of scrum, some had in addition an elaborate electronic way of documenting things, tracking tasks and items on the to do list, others were more focused on the physical board. But that's OK, as each team should have the freedom to work the way they want. And anyway scrum is based on a continuous improvement process so the way you work will evolve over time.

As for the development to do it using scrum, of course there are challenges. How can you iterate and release a new product version after every sprint? This is not straightforward. If you want to learn more about how to do it, I encourage you to check out some of the presentations by Joe Justice and also the book by Paolo Sammicheli, Scrum for HW which is the first book on this topic. I've recently read his book and it's a good one, especially the 2nd part. The first part is more about some of the players in scrum, thought leaders and Paolo's journey discovering scrum, and scrum for hardware. There are a number of tactics scrum teams adopt. One major idea is that the product gets split up into modules with well-defined interfaces between modules, with built-in redundancy in the interfaces. These interfaces ensure that each module can be worked on separately, so you can optimize each module in increments instead of having one monolithic, complicated design where each part's changes affect other parts of the system. Another tactic scrum teams adopt is that every team member develops skills over the whole spectrum, but keeping his or her specialization in a field, T-shaped distribution of the skill set. You can do this effectively by working in pairs, one person acting more like a coach, the other one as a coached person. This allows one to not only have a fresh pair of eyes on problems, but one can develop a common language. It can certainly slow down development at the start, but it should speed up things in the medium-, long-run.

I want to close off this episode with a great example for achieving extreme speeds in hardware development. Probably you've heard about Tesla's solar tiles, right? Remember when they announced the sexy tiles that you can have on your roof so, using Elon Musk's words, you don't have to bolt a fish tank to your roof? Tesla came to them and asked: what would happen if we used the protective films on tiles that is used on laptop screens and mobile phone screens? Since our screens are transparent when looked at from the top and dark from the side, so it works there, then they might work also on solar roofs to let the incoming sunshine through, and from the side they'd look like normal roof. 3M seemed positive that it might work. Tesla said, well, what if you invent the technology, test it on 3 houses, make it available in big quantities, all in 5 weeks? Well, this is what happened. The whole product was developed, tested in 5 weeks by 3M. This includes filing 7 patents with the patent attorney being on the team.

Incredible. I've included a link to this episode where you can see the 11 second demo of the product that shows off the tile's performance.

What is your solar tile, what is your product that you want to develop and bring to the product super fast? As some of you know, originally I come from Hungary, where I grew up in socialism. Then capitalism came in 1989-1990. Capitalism is not a perfect system but I believe it's better than socialism for a few reasons, one reason being that even if the idea about socialism might be right, the implementation doesn't really seem to work on humans. Scrum is not perfect, but at the moment this is the best framework I know. Plan-driven development works in theory but not practise, not in the real world where things change in the market, people are creative during development, so they might come up with better ideas than before, to delight the customers.

Well, thanks for listening and let me know privately or on the website as a comment what you think, if you do your developments already using scrum, what your challenges are, what problems you've overcome or any other observations.