Innovative Development with Lean

Guests were authors Tim Schipper and Mark Swets

Related Podcast:

Innovative Development with Lean

Tim Schipper and Mark Swets are the co-authors of [Innovative Lean Development: How to Create, Implement and Maintain a Learning Culture Using Fast Learning Cycles](). The authors have helped development teams cut development time in half and increase speed to market, while delivering award-winning, quality solutions. In this book, they share those examples while providing a road map that all companies can follow to reach a Lean development culture, one where creative thinking and practice converge in ways that lead to innovation, improvement, and success.
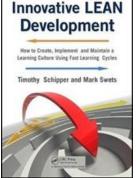
Tim Schipper is a graduate of Calvin College (Bachelor of Science in Mechanical Engineering) and the University of Michigan (Masters of Science). He has been a tool designer, engineering educator, CAD specialist and trainer, senior product engineer, IT manager, and most recently a Lean expert and internal consultant for Steelcase, Inc., in Grand Rapids, Michigan.

Mark Swets has facilitated more than 30 Lean workshops at Steelcase, Inc., in Grand Rapids, Michigan. He has led teams who have applied Lean development techniques in product development, information technology application development, sales strategy development, and business model development.

Related Information: **Single Point Lesson**

Amazon: [Innovative Lean Development: How to Create, Implement and Maintain a Learning Culture Using Fast Learning Cycles]()

Joe Dager: Thanks everyone for joining me. This is Joe Dager, the host of Business 901 podcast. Participating in the program today is Timothy Schipper and Mark Swets. They are the author of "Innovative Lean Development" which is a book on how to create, implement and maintain a learning culture using fast learning cycles. Mark and Tim, could you just shortly described what prompted you to write the book.

Timothy Schipper: What we did was we were looking for a method to speed up development at our own company, and we wanted to apply the Lean principles and the principles of fast learning cycles to the process of development. And so, we worked with a number of people to do that in IT development and in practical development and what we have discovered by applying these principles were valuable enough that we made it into a book. So, we initiated the writing of the book.

Joe: I read your book, and I liked it. I struggle a little bit with a couple of concepts in it. It kind of went onto the sideline for me but then when I saw your Single Point Lesson in Target Magazine; it kind of re-kindled my interest. It was really like an A3 page breaking out your Lean Development cycle. I went back to the book, and it kind of filled in a few gaps for me. I have to compliment you on to getting it on A3. Was that the purpose of it?

Mark Swets: It really was the purpose. AME Target magazine was a great vehicle to do that and the effort was to try to fill in some of the holes, exposure for the book and exposing the concept to people who are interested in doing some Lean development.

Tim: It wasn't easy necessarily... we took some time with this because we wanted to try to capture a number of the key concepts that we talked about in the book. But to do that in a way that would be quite practical for people and give them something to relate to. So,

one, the format was not new to us. The format is the format for other single point lessons in any Target Magazine. We fit our material into that format which really starts off with a situation. If this is your situation, this is the problem that you're trying to solve, here is some things that we as starting steps to solve that.

Joe: You did a great job explaining how to use Lean development. I just have to compliment you on it. I thought you did a fantastic job.

Tim: Thanks Joe. And the thing you talked about, the waste and how it relate to practical development is a difficult concept to talk about because in the early stages of development, you actually want to put in a fair amount of searching and looking for various solutions that maybe even throwing solutions away. It is a little counter intuitive of creating rework and just creating to solve a problem in development. Actually, it is required to make the process go faster in the long run.

Mark: That's probably the number one question that people get. "How is it that these Lean guys are going to be telling us, 'I want you to take things forward' that we know we are going to throw out. We are going to take in concept and ideas forward and throw them out at the very last minute, delaying our design freeze to the very last second. I know I'm going to be working on things that I am going to throw out. That sounds like waste to me." And, in fact, it is. The difference is that when you got this concept that you are moving forward and set this design, you can be using some of those solutions. You can decide which parts of the different one you like, and you can be putting them together in different ways that you maybe would not have thought of before. What you have done in the beginning is hitch your wagon to one concept, and you carried that all the way forward. If

it doesn't work, then you get to do it all over again. The major rework of going back and doing it all over again is avoided. There are some wastes in there built in, but you end up going faster as you take multiple concepts forward.

Joe: Are you software guys?

Mark: We are product development guys, primarily from our educational background. I did spend some time in the IT area. I was a manager in IT venturing systems where I was responsible for software development and, in fact, one of the first things we applied this process to was a software development process. We were installing SAP major business system into the company and what we were discovering have taken us many, many months to put that system into each plant. We were using a well-documented process which we struggled with because it assumed we could get all the requirements up front and then go into development and show the users. Hey, we solved your problems. But in reality, when we got to the users they said, well, you didn't listen to what we told you at the beginning. You didn't give us solutions that would actually work for us. The first applications for this were in IT development, and we took a 20 month project that was thought to be a success, but actually contained 80 percent rework at the end.

In the next cycle through, we discovered that we could do a similar plan for in just under six months by including the users in the requirements gathering and in the actual building of the solution. We decreased much of the testing that had to happen and a side benefit is that we didn't have any critical issues after the roll out.

Joe: I find that interesting because it reminds me a lot of agile or scrum development cycles.

Business901

Podcast Transcription

Implementing Lean Marketing Systems

Mark: Similar to IT development. We had projects that were taking far too long. We were going two to four years for some of this projects and our business, it was...That significantly too much time. We had too many projects that were being done concurrently. There was a lot of rework due to the requirement changes. Just as what Tim talked about IT. We had many changes in there. We didn't involve our suppliers or our manufacturing base until far too late. Engineers or others left the team and so we lost that knowledge. We didn't prototype or validate from the field early enough or often enough or sometimes not at all until they were at the door. And testing happened late to change any other designs. With the Lean principles that we have applied, the announcement initially seen or similar sized projects in eight to twelve months; significantly decreasing our time, time to market or hitting our cost targets much more regularly. Most importantly we're canceling projects earlier when we know there aren't going to be the right thing for us to take forward.

Joe: I like the way you broke that down, and you called it a "Learning Cycle." Can you explain what a Learning Cycle is?

Mark: A learning cycle starts out with scoping or planning and in the scoping and planning section we are interested in understanding the problem or a subsystem that needs to be solved for. We're taking a larger problem, whatever it is; maybe a large system design or system problem and we're dividing it into a smaller subsystem which we place in the learning cycle. So, we list the questions to answer, and when we list the questions to answer, we are listing the things that we don't know, the discoveries that we need to make about the problem. Those questions go into the learning cycle.

In each learning cycle, we really want to do the design so we can solve those questions. What do we design, whether it's an experiment, whether it's a test, whether it's some part of the system we want to prototype, and we actually build that and do the test and then look at how it informs the questions that we asked at the beginning of the learning cycle.

Tim: In a learning cycle, the way we sum it up is it has these things. It has a plan, a design, the build phase, a test phase and then a review of the results.

Joe: How do you keep track of all that, of what stage you're at and what you're doing during a product development cycle? Are you mapping it somehow or how do you keep track of all this stuff?

Mark: The learning cycles do lend themselves to fit over our traditional product development, or IT development process. We put the learning cycles right on top of our existing process. They fit very well on top of it because as Tim talked about, the idea behind development is; its things we don't know the answer to. It's a fun part of the business, but it's kind of a messy part of the business because we don't know the answer. So, these are really fitting inside of our concept phase or inside of our development phase.

Tim: Our product development phases stay the same. What we're doing here is trying to accelerate how we come up with the answers or how we come up with the solutions.

Joe: How do you obtain, let's say, voice of customer or voice of market? How do you get that? I know that feedback is probably going to be different in many different instances, but is there a certain way you've found that works well?

Mark: There are a couple ways we do that. One is in the planning of the learning cycle we will think about how we can bring in market validation or user validation into the process, even right into the learning cycle. That's one key way that we do that. Another way we do that is as we evaluate multiple concepts that are being considered, we use a concept selections matrix that uses those voice of the customer type criteria in order to rate the concepts that are coming out of the team.

One of the big parts of learning cycles, and maybe the most important part, is to actually test the prototypes that you're building. We look at learning cycles as promoting a prototyping culture. If we want to be a learning culture here at our company, we would need it to be a prototyping company.

Tim: That's maybe the biggest cultural shift that we've come across especially in engineering circles. Prototyping turns out to be not very easily done unless it's very professionally done. I'm from an engineering background, so I know how this goes. Prototyping is a reflection of your work and a lot of people (I included), are not interested in showing a reflection of your work out there that may not be completely safe. That's a big cultural shift to be able to come out with something that isn't perfect, but it can help you learn.

That's where we need to really get our culture to shift to that we are taking things that aren't perfect, but being able to test and learn from them and go forward.

Joe: That's a big change in most development that's taking place now. It's more about learning from your customers or learning from your market because we've realized that we

don't have all the answers inside. We can't design things inside a shell anymore and then just go out and say "Here it is people, let's buy it."

Tim: Exactly right. That's exactly right. What we don't want to find ourselves in the spot of, we don't want to find ourselves in the spot of having solutions that we think the customer wants and then either they aren't going to buy it or they're so expensive because we hadn't taken their input into account that we actually hope we don't sell very many of them because we're not making any money on them.

Joe: How about the definitions that I think people might struggle with. Most of the listeners will understand what a stage gate is but is it different in Lean development? How do you know where to... stop, what action and approval step? How do you determine that in Lean development?

Mark: That's a great question. So, a stage gate process as Mark was saying can lie over the top of learning cycles. We've seen this done a couple different ways both here at our company and other companies that use similar methods. One way to do it is you walk through a series of learning cycles within a phase. So, if you have a phase that's 12 weeks long, then you might break that into four, five or six learning cycles. We always say, "Keep the learning cycles short." Our goal is no longer than about 30 days. We like to keep the learning cycles in about a three week to four week duration. You break that phase into a series of learning cycles that you just repeat.

Sometimes you have multiple teams doing this. You have multiple teams or sub teams doing this in parallel, so they're all breaking their tasks and questions into learning cycles.

You have to bring them together at the end. That's a key concept is breaking the phases into these learning cycles.

Tim: Traditional project management has a lot to do with what you are going to solve and learning cycles we try to get the language to be about how to solve it. And again, in traditional project management we have just a few toll gates, phase gates, so we don't get the input as often as we'd like any corrections or things like that to the path we're going down. Learning cycles, as Tim talked about, have multiple integration events all along the way. So, we have regular leadership interactions so they see where the project is and can make adjustments based on what they've seen and where it's going.

Joe: Do you measure any of this or how do you measure it?

Mark: Well, there's a couple ways to measure it. One thing that learning cycles allow you to do pretty easily is use visual control to track the learning cycles. We'll actually post the learning cycles on a visual control board where we can keep track of the tasks on an accountability board of what has to happen within the learning cycle. So, that's one point of control is that visual control. And then that also allows management to check in on the process more often. So instead of waiting until the end of a traditional phase gate or stage gate where there's a management review of the project, we encourage management to look at the progress of each learning cycle.

At the learning cycle review, you're at your, OK, what were you trying to learn, what did you learn, and what do you need to learn next which would then be the material for the next learning cycle.

Tim: Another way to ask similar questions are three questions; "Are we there yet, where are we, and what do we need to do to get there if we're not there yet?"

Joe: I'm going to shift gears just a little bit, but you talk about the use of multiple prototyping techniques in the book. Can you tell me about what you mean by prototyping because that seems to be a central theme of what you're talking about, and I think people have difficulty sometimes envisioning even early prototypes when they're in product development, in what stage is it good enough to send to a customer or to get feedback on even if it's internal feedback. Can you explain that to me?

Mark: Sure. Prototyping is really one of the keystones of this whole process as you're bringing out here. The earlier you can have a prototype and have some testing of that prototype, the better informed you're going to be and the further down the road you'll go. So early in the process, very early, you have very few detailed requirements, and you really don't have very high confidence in any of your concepts that you're taking forward. You might have some that have better confidence than others, but your confidence is not high.

A lot of times you can't prototype those to a level where it's very pretty, or that's perfect or anything. You may just have sketches. Then, you would have subsystem prototypes and testing going along as you become more confident. As those subsystems become able to be put together, we have a total system prototype where you might even start putting some tooling together.

These are the kinds of prototypes that we're talking about. But, the important thing is to prototype in every learning cycle and to do it early and often and to get the validation from

either the users, from the customers that you'd be talking about from the place that would be accepting this, might that be in manufacturing or in the engineering or from an installer or other people like that. These are the kinds of testing that we do on our prototypes.

Tim: We all know the story of how Orville and Wilbur Wright designed the Kitty Hawk airplane in their bicycle shop. They actually achieved a pretty high level of prototyping just in their bicycle shop. We have to sometimes think about these prototypes as they're not going to look the prettiest, right? They're going to be very functional. When we talk about IT development, we're looking at the same types of prototypes. How can we show the users core functionality without developing a highly programmed solution? Or, how can we show them on screenshots or even on a simulation of what a screen might look like so that they can get a feel for how the solutions going to work for them without going off and automating the whole solution.

We're talking about this bicycle shop mentality where the prototypes are not perhaps what you're used to seeing out of a formal prototyping shop or professional shop, as Mark was mentioning.

Joe: The other thing I have to mention about the book, it stayed away from so many traditional Lean books where they spend a third of the book talking about how you've got to get leadership involved and you've got to develop culture. You guys talked about applying Lean and kind of stayed away from all that cultural stuff. "This is what you need to do" and I kind of like that. You also talk about how to map the process but to map the product flow?

Mark: Right. We really needed to take this to a practical level. With the help of some other outside consultants, Jim Luckman being a very important person in the development of this for us, we value streamed mapped an actual project, not what we thought might be the case, but very concrete actual project, maybe the last one we did or the last one that was approximately the size that we were looking at.

Tim: The other part of the Lean development or applying Lean to development is the idea of doing continuous improvement. What the value stream mapping does, if you're value stream mapping the actual projects, as Mark was describing, it allows you to point to real problems, problems that were truly experienced. You could pull those things into things that you need to correct, and you can use your traditional Kaizen approach with your A3 project plans and other things to get at those systematic or systemic problems that are plaguing the project development process.

Those are going to be true in any development process, but they're going to be unique to each organization, as well. So, that's why it's important to value stream map the projects going through the process to drive out those problems in the process.

Joe: Is there something else you would like to add maybe or expand on a little bit that I haven't asked you yet?

Mark: One thing that we didn't talk about is the purpose of learning cycles or a purpose of learning cycles to put pace into the process. We're trying to make the process flow by launching the learning cycle and then focusing on that learning cycle and letting that learning cycle complete. Before, we bring in any flow interrupters, we try to show in our graphics in the single point lesson that flow interrupters happen for projects all through the

project from the very beginning to the very end. They can come from the marketplace; they can come from management, and they can come from other places. The key here is to try to get some flow developed so you can go through the learning cycle, complete your learning and then take a look at your flow interrupters.

One of the things that we practice is putting those flow interrupters in the parking lot. We don't lose them; we just keep them on the visual board and say here are the things that we know are interrupters, but let's not let them disrupt the process that we're trying to do right now.

Tim: It's also true that if you can take those interrupts and put them in a visible location, we can take leadership to see what these interrupts are, and they can hopefully get an assignment from this to either eliminate or try to solve what those interrupts are off line from the development team.

Joe: How can someone get a hold of you?

Tim: We have a website which is the same name as the book, InnovativeLeanDevelopment.com. We also are available by email, and that's also on that website. I think the easiest way is to go the website, innovativeLeandeveloment.com.

Joe: So, I'd like to thank Tim and Mark very much.

Mark: Joe, thank you very much. It's been a pleasure.

Tim: Yeah, thanks, Joe, it's been a lot of fun today.

Business901　*Podcast Transcription*

Implementing Lean Marketing Systems

Joseph T. Dager

Ph: 260-438-0411　　Fax: 260-818-2022

Email: jtdager@business901.com

Web/Blog: http://www.business901.com

Twitter: @business901

.

Joe Dager is President of Business901, a progressive company providing direction in areas **such as Lean Marketing, Product Marketing, Product Launches and Re-Launches. As a Lean** Six Sigma Black Belt, Business901 provides and implements marketing, project and performance planning methodologies in small businesses. The simplicity of a single flexible model will create clarity for your staff and, as a result better execution. My goal is to allow you spend your time on the **need versus the plan**.