Interview with Jim Benson, ModusCooperandi.com

Lean Increases Small Team Effectiveness Three-fold [VIDEO] - With Jim Benson

Watch the full video at:
http://www.isixsigma.com/lean-for-small-teams/

Michael Cyger: Hey everyone. My name is Michael Cyger, and I'm the founder and publisher of iSixSigma.com - the largest community of Lean and Six Sigma professionals in the world. At iSixSigma.com, we champion the idea that breakthrough process improvement can be accomplished by anyone within an organization.

When I think of implementing Lean at an organization, I think of a methodology to reduce waste and empower employees or people doing the job to do the right thing. A classic example that we've all heard of before is the Toyota Production System. Lean can involve tools like 5S and process mapping. But to me, the methodology gets a little amorphous at that point. So today, I've invited on a guest that's going to help us understand the Lean methodology better - especially as it relates to small teams. Today's questions include: How do small teams implement Lean? What does Lean look like? And what are the benefits - both, top line growth and bottom line savings?

Joining us to answer these questions is Jim Benson. Jim is the CEO of Modus Cooperandi - a collaborative management consulting company based in Seattle, just across Puget Sound from the iSixSigma Headquarters here on Bainbridge Island. Jim, welcome to the show.

Jim Benson: Thank you.

Michael: Jim, let me start with the bottom line up front. If someone were to invest the time to watch this full interview, what would they get out of it? How can Lean help small teams?

Jim: So, when we're talking about small teams, one of the things we're mostly talking about is small teams of knowledge workers; and the implementation

of Lean into the manufacturing sector is very well documented. But when you get into knowledge work, it gets a little bit more amorphous. And so what I think we're going to cover today is how we can plug Lean principles into knowledge working teams and give them more clarity into their work, more ability to estimate effectively, plan effectively, and then communicate more effectively with the rest of the organization. When we go back to Deming profound knowledge, one of the main things in the profound knowledge is understanding variation. And for small teams especially, especially for knowledge workers, there's a high degree of variation and I think that's what we're going to spend a lot of time talking about today.

Michael: Definitely. Okay. And so, that makes sense to the small team. But if somebody wants to talk to their managers and say, 'Hey, we want to do this because we're knowledge workers, we work on this team; we want better communication. We want shorter cycles times. We want improve processes', what can management expect by supporting this type of Lean implementation in small teams in terms of hard savings?

Jim: Right now a lot of the waste is inherent in knowledge work comes from a fundamental lack of understanding about what's involved in knowledge work. And that's on the part of the knowledge worker, the team that the knowledge workers in on, and the people who are managing those knowledge workers. One of the things about it; if we have an assembly line and there's widgets moving down that assembly line, we can visibly see when that line is at capacity. When you start running into the Lucy and Ethel at the Chocolate Factory problems where there's just chocolate everywhere, that's a visible thing. You can see that. But you can't see when someone's head is overloaded. And people do have actual capacities. And so what we want to do is provide some visual controls, and some tools, and some mechanisms to communicate to everybody involved what's going on in any given point in time and when people are reaching their capacity. Otherwise, if I approach you as a boss and I just have a ten minute task, and I give it to you, and I say, 'Here's your ten minute task'. And then I come back in thirty minutes and you're not done. I am annoyed because it was only a ten minute task.

Michael: Right.

Jim: I have no ability to actually see into what you are doing, so I can't respect what you're doing because I just simply don't know. And when that is understood by everybody involved, you also start to understand things like yesterday that task took me ten minutes; today when I did that identical task, it took me a half hour. Because there's a high degree of variation in my work. Well, that 3X difference is very, very disruptive to your ability to plan out your day, to your ability to estimate effectively, and so forth. So if I'm a boss and I'm saying, 'How long is that project going to take you?', I'm asking you to estimate that with no knowledge of what variation is going to come up. And then we're all upset when that goes offline. And we'll talk more about that as we go along.

Michael: No, definitely. And I had attended one of your training sessions at a Seattle Tech Startup Presentation that you gave in Seattle. And so I understand how it all fits together and that makes perfect sense that we're going to get into all those tactics. But before we move on, do you have data, have you read any research, do you have any situations that you've experienced where you've implemented Lean in small teams and you've actually seen productivity increase as more work output from knowledge workers as a result of doing the Lean, and seeing the process, and having better communication - not overloading people in their head just like you said?

Jim: So there's two standouts for me, but there's a lot of examples. The first is a team of support staff that I worked with in a company in West Virginia. And there were twenty-four people on, what they called, the Support Team. And this team had a backlog of two thousand tickets. And that wasn't so bad because they were doing about two hundred and fifty tickets a day. The problem was that they were taking in two hundred and fifty tickets a day. So any variation went more into the backlog - the backlog grew, and grew and grew. When we first started discussing their team, we said, 'What are you doing to lower the number of tickets that you have?' And the response back was, 'Well, we're all trying to go through more tickets'. So they had a daily award - daily tracking of who did the most tickets. Psychologically, that destroyed everybody. Because everybody was fighting with each other to do the easy tickets, or they were upset because they got a hard ticket and that was going to drive their numbers down. They would prematurely end calls

and things or, worse yet for moral, they would actually do a good job. So the metric was saying, 'You have to do as many tickets as possible'. As good professionals, they were staying on the phone for four or five hours with really sticky problems. At the end of the day, they did two tickets. And then they would get in trouble for doing a good job. So the problem with that is that team - it was, what I call, a bowling team. Everybody was out there trying to bowl as hard as they could, but they weren't a football team. They weren't a baseball team. There was no collaboration going on that allowed them to enjoy the benefits of putting multiple minds on a problem or by taking a group of work and figuring out ways to increase the throughput of it. So what we did was we ended up dividing them into teams of six, and then those teams had a visual control and they were highly focused on team dynamics. And that ended up making the organization overall, in the course of about six months, get down to about seven hundred tickets in the backlog. And the cool thing about that was while they were doing that, they were constantly going through kaizen exercises. Their teams met in with agile methodologies, called Stand Up Meeting. So for fifteen minutes, every morning, they would meet in their individual teams. But then representatives of those teams would go to, what they called, a Deming Team Meeting. And that team would say, 'Here's the continuos improvement exercises that we did yesterday. What are we going to do today? Here's the results that happened'. So they were doing plan due, check back, every single day. Because their problem was that big. And, again, this is a team that's in the middle of West Virginia. It's made up incredibly dedicated professionals who, before we started this, absolutely hated their jobs. At the end - and this is the bigger metric for me -, they all really liked their jobs because they were allowed to be effective.

Michael: Right.

Jim: And when we moved into the software world, a lot of teams that we work with are software teams. Especially in organizations where your company doesn't really do software, but it has a software group, which is almost every company now. But the problem is that since your processes are geared towards manufacturing, or finance, or healthcare, or whatever, relating to IT is traditionally very difficult. So what we've noticed is teams that implement visual controls and begin to understand what their variation is,

begin to understand how the teams learns; understands Psychology. So basically, going through all of the steps of Deming's profound knowledge. They're better able to communicate to the rest of the organization. I can actually give another example that's actually an anti-example.

Michael: Okay.

Jim: So we had one client that was a large company that was traditionally extremely topdown and wasn't geared for Kaizen events in any way, shape, or form. So when we started the project, we went to upper management and said, 'Look. You've hired us to come in and teach your people to be autonomous. And we're worried that this isn't the culture of this company - isn't autonomous'. And they said, 'No, no, no. It's exactly what we need and that's what we want to do', so we worked with the teams. The teams went through a couple of Kaizen exercises and they basically reorganized themselves so that they could be more effective. The numbers that I was getting back was their effectiveness went up somewhere between two hundred to three hundred percent.

Michael: Wow. Okay. So we're seeing effectiveness of teams going up as a result of multiple benefits going on. The employee morale is increasing because they are actually able to control their destiny and make those changes that allow them to be more productive. And then, as a result, their output is increasing. They're either reducing their caseload, or they're getting more work products out. And at the end of the day, the customers are then more satisfied because they're being handled more quickly; they're getting a better product.

Jim: And the funny thing about that is that when you make professionals visibly aware of the impediments to them doing a good job, and that sits up there on a visual control - it just stares at them all day long -, it becomes intolerable for them to allow that to go on. So, waste reduction is a byproduct of Lean. It should not be the focus of Lean. The focus of Lean is to know what's going on. And then, when you learn that, you will start to do continuous improvement exercises, which, if successful, will, by definition, reduce appropriate amounts of waste. Now, conversely, reducing waste can lead to really bad process. So if you're lean focused - and this a problem with

a lot of upper management or people who come to Lean from a cost cutting standpoint as apposed to what you describe earlier, which is more of an understanding of work; understanding why we're here and why we're building what we are. If we have faith in people, as professionals, which I think is pretty much entirely what Deming wrote about, that's where the real savings come in. That's where waste disappears.

Michael: Yeah. Great. Well, I'm excited to get into the specifics. Because I think a lot of people quickly summarize Lean or Six Sigma, and differentiate the two from themselves, by saying Lean reduces waste and Six Sigma reduces variation. And it's more than that. And so, that's why I'm excited to get into this interview in detail. But first let me take a step back and ask you a little bit about your background. When did you first discover the benefit of using Lean in small teams?

Jim: Well, I'll go back further than that. So I started off in Psychology. And as I was going through University, I kind of looked around and was like, 'Wow! If I get a degree in this, these guys are going to be my peers'. And I wasn't sure that I wanted that. So, I went into Urban Planning and I built giant things. So I built freeways and light rail systems; subways, neighborhoods, and so forth, and got an appreciation for what it was like to manage gigantic things that were made up of bunches of different components. So to have massive, massive teams and then smaller teams, and teams within those teams; and understanding that from the individual all the way up, everybody had different information needs. At one point, I became frustrated with Urban Planning because my government clients were sometimes actually policy bound to destroy the work that we'd done. So if I went out and gathered a whole bunch of geographic information for a region on a project, a lot of my projects had line items that said, 'You cannot save this information. You have to give it back to the government'. And so, there were a few times in there where I had back-to-back projects with an agency. I would go out and gather a whole bunch of information. Give it to the government. And later on they would say go out and collect it again. And I'd say, 'But I just gave it to you'. They'd say, 'Yeah, we killed it. We don't know where it is'. And so, I made it into a software company that was designed to make collaborative software for government that made them better storage of their data. And we actually did a pretty good job. And so, we were in for

about ten years and during that time we were, what was called, an Agile Shop. And so, to software development, Agile is kind of the Lean of software development. Until now; we really introduced Lean for real in the software development. And Agile taught a couple of things that were antithetical to government contracts. So you needed to do rapid release, so about every two weeks we would show up with a new release. It would be fully realized. You could have discussions based on it. The mess of contracting was that when you got to that, you could migrate your software development based on what the feedback was. Well, all of that flew in the face of a rigid scope, which is how government usually operates; so we had to get very good at selling this. And there were a few key components that weren't working well for us. And one of them was a central tenant of most Agile methodologies, which is something called Iteration or a Sprint. And thats, just like I just said, every two weeks you release something. When that realization hit us, there was a huge Agile community, which had its processes very well set. I will not say Dogma, but by saying that I'm not saying Dogma, I'm saying Dogma. And we kept bouncing up against that and we went off in search of other mechanisms that we could augment our Agile practices with that would help us out. And then we basically discover Lean through the concept of Flow basically. We wanted to set up a system where people could visualize what was happening across the team(s). We could see the flow of work. And when we started doing that, we suddenly realized two week interruptions in work is waste. Because you have to have a beginning and an end to every two week cycle, and there are arbitrary constructs. And that was basically our introduction. A guy named Don (Unclear 18:36.3) in Southern California introduced the concept of Kanban to my friend David Anderson. He and I started discussing this. He went off to work with it at Microsoft and I was working with it at my company, and we both started noticing very immediate results. And then it wasn't long after that before we were digging into Womack and Jones, and Deming, and everybody saying, 'What else is in here?'

Michael: Right. Great. And so, your current business - I love the business name - Modus Cooperandi. I think people can understand Modus Operandi means a particular way of doing something. That's my MO. Why did you choose Modus Cooperandi as your business name?

Jim: Well, ultimately there are significant waste reduction cost savings, quality improvements, when we collaborate. Two people working on something produce better product than one person working on something. They may or may not produce it faster than people working parallel, but usually they do. And the quality improvements of having multiple eyes on something, multiple hands on something, people actually interested in what you're doing - that increases quality, reduces waste, and also, because we are social animals, makes us happier when we're at work. So the more than I go and toss Dilbert into a cube and say, 'Don't talk to anybody. Just get your work done', the lower the quality of product I'm going to get and it's going to keep decreasing over time. Because that person is going to feel more and more distant from the work that's actually being done. The other thing is that there's cooperation between levels in an organization. So one of the problems that we had in Agile is that the rhetoric was - Agile is kind of a liberation theology for coders. And very important, software would be decades behind where it is now if Agile hadn't come along. But before Agile, coders were incredibly mistreated because, again, no one can see what was going on in their heads, so they just kept tossing more and more work on them. Agile provided a mechanism for them to put some limits on that to start to limit their work in progress. And that's what we've continued on with Kanban, but we're continuing in the collaborative model that not only says inside the team this is what's happening, but says up and down stream, this is what we're working on. You guys can see it because it's on the wall, or it's on the screen, or wherever the Kanban is. You can see it. Come stop us is we're insane. And the conversations then become less people showing up in your cube and saying, 'What are you doing?'. But I see you're doing this. Let's have a conversation about it. So you start from activity. You don't start from an open ended query.

Michael: Yeah. Okay. So we're going to talk about Lean for small teams, which is applicable to a subset of Lean for software development. But I don't want to make this software development focused. We're going to talk about Lean for small teams. We're going to talk about personal Kanban visual management. And that's all great. But let me take a step back and get you to define a term for me.

Jim: Sure.

Michael: What is Lean?

Jim: What is Lean is like what is love, or what is quality, or some things that Supreme Court Justices couldn't quite get down. So, for me, when I kind of deconstruct Lean, and I would like to think Jim Womack would agree with me on this, is that I liked your definition at the beginning. That Lean is about respect for the people that you're working with and for the teams, and for a single minded drive toward better conditions. And that may be quality improvement. It may be quality of life improvement. So, I can do all sort of things to improve quality of the product. But if it makes the working conditions more dangerous, then it's going to be self defeating. So, there's different levels of quality. So I keep coming back to Deming System of Profound Knowledge and those things are usually appreciation of a system, knowledge of variation, the theory of knowledge itself, and then of course the Psychology aspect of it. And, for me, that last Psychological component when we're dealing with small teams and we're dealing with knowledge workers is imperative. Because just like you're not going to operate a non-lubricated lave, you also shouldn't be going and destroying the morale of people who's entire output for you comes out of their heads.

Michael: Right.

Jim: Because if people feel lousy, they will produce bad thinking. They will stop thinking. They will shut down. So, if there are lousy increments, every lousy increment you add on is actual destruction of value for the organization.

Michael: Sure. And there have been organizational studies that have proven if workers bring their outside personal life to work that they're less productive. If they have health issues, they're less productive. And that's why a lot of forward thinking companies are offering these human resource plans that help the employees feel more comfortable, take care of their personal issues on their personal time, help them manage their issues, so that they could have more happy and healthy employees at work so that they can get more productivity out of them. So it's the same sort of relationship.

Jim: And also to understand that there is a very close linkage between effectiveness and productivity for knowledge workers. And so, as an employer, I need to understand that when the bell rings, it's not Fred Flintstone jumping off of his Brontosaurus and going home and saying yabadabadoo. That people actually go home and keep thinking about their work. Conversely, when people come the office, they don't stop thinking about their lives. So, providing ways to elegantly allow people to be human while they're at the office is huge. And sometimes, like let's say that you have your Kanban set up for your personal work and in there there's a wip limit of three things that you could be working on at a time, but there's something in your personal life that's really bothering you. You should be able to walk up to the whiteboard and scribble that thing out, and put up a two and just let your coworkers know, 'I'm dealing with some stuff and I might not be as pleasant today as I normally am'.

Michael: Yeah.

Jim: And some places that works better than others. The one thing that we have noticed though is when we have people and they finish a task, annotate that task with some indication of how they felt when they were finished with it. So we used smiley faces or angry faces. And people can get very creative with what they put on there. But if you get to the end of a day and you look at the completed column, and you see that everybody is angry - and you can even see this after two or three things go by -, then you know something is wrong. So, for knowledge workers, disposition is a leading indicator of future quality problems, or cost overruns, or things like that. So you say, 'Why are you guys upset?' Very seldomly, you have a whole group of people upset because of something external to the group.

Michael: Yeah, that makes sense. And I've actually got my personal Kanban set up on my office wall on the other side of this wall over there. And it's great. I got a lot of things in my queue. I've got four things that I'm working on because I'm a major multitasker even though I'm probably not as effective. I've got my four things and I've got my done. And it's great for me to visualize what's going on every single day and what's important in my life versus what comes up everyday. And, actually, I was just having a conversation with one of my entrepreneur friends that runs a customer

relationship management software system - phenomenal product - over in Seattle yesterday. And he picked up the phone; he was talking to his development team. And they were talking about the same sort of thing. What are the personal emotions that you're feeling when you're working on this item for this release, or for just completed it? And I got to listen to him interact with the CTO and development team about that, and for knowledge workers, it's very important. Because something may have been completed, but somebody may not be happy with it. And just checking a box on a database that has a interface for the headquarters office over in a different state, let's say, is not indicative of what's going on in the organization.

Jim: Right.

Michael: And so, we'll talk about that. But let me ask you one more question regarding what is Lean. Jim, can people look at it as a framework? Can they look at it as a series of steps that need to be completed in order to make sure that the right things are being done in order to have a more effective Lean organization?

Jim: Yes and no. For me, when we start to get into finite steps, we start to come up with rules. And the more rules that we come up with, the more rigid our system becomes. And the more rigid our system becomes, the more it doesn't tolerate variation. So, for personal Kanban, we only have two rules. And that's visualize your work and limit your work in progress. And, again, because if people see what the problems are, that will drive the quality improvement. There are implied rules beneath that, which is: you should be continuously improving. But the understanding is that poll system isn't just a poll system of work. There's also a poll system of quality.

Michael: Right. But that's an individual tool. Kanban is one tool that fits under the Lean umbrella, right?

Jim: Right. Exactly.

Michael: So, can you refer to Lean as a methodology or a framework that includes a bunch of tools that you may or may not use depending on the exact work conditions?

Jim: May or may not is exactly the phrase I was looking for.

Michael: Okay.

Jim: So, if there were steps, I would say that the steps would be something like: become aware of your work, stay aware of your work, and then try and become better at your work continuously. And here is a wide range of really amazing tools. Here's different questions and answer mechanisms. Here's appreciative inquiry. Whatever it is that you want to employ. One of the problems is - and this is true for Lean; this is true for Six Sigma; it's true for Agile; it's true of A3; it's true for everything that is a process - people try to apply that process rigidly where it doesn't belong. Then it doesn't work there and then they claim the process to be a failure.

Michael: Right.

Jim: But what's actually a failure is a lack of understanding about how you as a team or you as an organization create value and what your customers want. And if you understand those two things via any of these mechanisms, then you can apply any of these other mechanisms to solve problems when they come up.

Michael: And you just described the Paramount issue - facing corporations on any large scale implementation. Whether it's lean, whether it's Six Sigma, whether it's ERP system; if you do it wrong, if you're not implementing it in the way that the business needs it, it's going to fail and then people are always going to say, 'Well, we tried that before. We're never going to use that again'.

Jim: If you're not doing something thoughtfully, then you're not doing something thoughtfully. That's the end result. So, ERP is an amazing example for that because so many people thought two things. One, it's out of the box, which ERP is never out of the box.

Michael: Never.

Jim: And then number two, that the professionals - the SAP professionals or whoever - can come in and solve all of our problems for us. And that's not the way it works. Because they'll only know your problems in how you relate that story to them. You have to be the introspective business owner. Maybe you hire a consultant to come in and help you become more introspective, but you don't expect other people to come in and overnight figure out exactly the way everything works culturally and systematically in your organization.

Michael: Right. Exactly. Okay. So we're talking about Lean for small teams. And it's a methodology framework including a bunch of tools and, overall, you want to measure what you're doing, analyze it for improvement, and then improve it. That's the goal of Lean. But small teams is a little amorphous to me. How do you define a small team? Is it two people? Is it ten people?

Jim: Small team is also a catchall. And one of the things for identifying what a small team is, is how simultaneously independent and dependent are the actors on that team. So, initially, that group of support staff that we were working with - they called themselves a team, but they were actually just employees. But the moment that we developed the subteams, not only did they become team members in the sense that they were actually on a six person team, but they also became team members in that those teams related to each other as one large support team now. The green team. The red team. The yellow team. The blue team. They all knew what their role was in relation to the other teams. So it gave them definition and it gave them clarity. The people on those teams had a lot of discretion in how they handled their own personal work and how they drove the work for their teams. If you have a group of people that are only receiving orders from above, that's a group of people receiving orders from above. It is not a team. Now, the word small can go anywhere from an individual up to fifty, sixty people depending on how all of those people are interacting. But I think that there's more of a systemic or a functional definition than there is a numeric definition for what a small team is.

Michael: Okay. Well, we're talking about a team; so it may be anywhere from one to fifty people that is going to use the steps that we're going to talk about next and they're going to organize their work around that.

Jim: And I can actually tell a little story around that.

Michael: Yeah.

Jim: There's a rule in Agile methodology that say the ideal team size is eight people because that's kind of a human relationship - human dynamic - limit. When we introduced, at a large company here in Seattle, a Kanban, they had fifty developers. And we were able to use the visual control as the focal point for the group, which allowed the cognitive load of understanding what everybody else was doing a lot less; which meant that those fifty people could get together, they could have a daily stand up meeting in fifteen minutes that actually covered everything that all fifty people were doing. Small team is going to be defined as the size of the team where everybody knows what's going on. So the more tools that you can give to prove that clarity, the larger your small team can be.

Michael: Alright. That's a great way to define it, Jim. So, before we get into implementation, tell me how does Lean for small teams differ from Lean manufacturing like the Toyota way, from Lean services like Lean Healthcare that we hear so much about nowadays, and from Lean Startup popularized by Eric Ries?

Jim: Okay. You do know that this is a long answer?

Michael: Yes, I do. But I want to compare all three just so people get a better clarity in their own minds.

Jim: Absolutely. So, with Lean manufacturing, we have a long history of proven results based on the optimization of, and some people might criticize me for this, but based on rather predictable events. So there's a model called the Cynefin Model, which talks about the complexity of an environment. And there are four major sections along this continuum. It's drawn kind of in quadrants, but it's more of a continuum. And the simple part of that is simple stuff. And simple stuff is: I put in a certain amount of energy and I get out a widget; and that happens all the time. And my optimization of that is: How can I put in less energy to provide the same widget?

Michael: Right.

Jim: And/or can I change my processes to do that? And I believe it's somewhere that Deming's definition of quality was something like the amount of work efforts that you have divided by total cost. So a fairly mechanistic system. And in that simple domain, we can apply best practices. Because the best practices is something that I can do over here and I can give it to you in your company and you can do exactly the same thing; because this is a simple system. Above that is the complicated system. And the complicated system has a higher degree of variation and you can apply, what are called, good practices, which are like best practices with more leeway. That's the ordered side. That's the Lean manufacturing side. Manufacturing should never get into the unordered side. So, for teams, we're dealing a lot with the unordered side. And I'll get to that in a little bit.

For Lean services. Lean services is awesome and kind of what we're doing. But the enactment of Lean services was largely flavored by Lean manufacturing without brining in the Psychology of work, or fully understanding, I believe, the unique role of variation for knowledge workers. So it got a bad rap because people were coming in and they were trying to implement rigid systems that didn't belong. So the checklist manifesto style management for specific operation procedures in an ER is awesome. The checklist manifesto across the board in a healthcare organization is awful. Because there's a lot of listening and a lot of emotional feedback that doctors get that you can't put down; like the checkmark, 'Did you get your emotional feedback? Did you work appropriately with that emotional feedback?' So what I've been seeing in Lean Healthcare and the other services is that's become apparent and basically in continuous improvement of Lean services themselves; that that's becoming a lot better.

The Lean startup world is interesting. Because that's applying Lean techniques to entrepreneurs, which are almost chaotic. They have crazy ideas and they want to go out and do their crazy ideas. And so, before Eric Reis wrote Lean Startup, what people tended to do was they would have an idea and then they would go out and build it - like, the whole thing.

Michael: Right.

Jim: And that ended up not serving a lot of startup's very well. They ended up burning through a lot of money. And so, the Lean startup concept is borrowing something from Agile. Agile had the minimal buyable feature, or minimal marketable feature. And Eric borrowed that and created the minimal buyable product. And basically what that is, is I've got an idea, and I really believe in it, and I'm smart. So I'm going to go out and I'm going to create this minimal buyable product as quickly as I possibly can. I'm going to release it to the market as quickly as I possible can. Maybe I make money from it and maybe I don't, but I get back feedback right away. So you set up feedback loops right away. And Eric kind of set up a plan due check act kind of thing, which is Build Measure Learn. And what the benefit of that is, is it's starting to bring the scientific management model to what was previously an entirely chaotic, unmanaged system. Lean startup doesn't yet - it will - address the actual production systems. So this is more of a testing your hypothesis for a product. So this is applying to the product development process; not necessarily the product creation process.

Michael: Right.

Jim: And so, what's nice about that is that we ended up working a lot with Lean startup groups now. Because people are recognizing, okay, I really understand my customer, but I still don't know how I'm building the software.

Michael: Right. So there are tools from the original Lean Manufacturing that have flowed forward into Lean for Services that have flowed forward into Lean Startup. A lot more database decision making across all three. A lot more structured form to implementing it although you pick and choose what may or may not be an appropriate tool to use at whatever phase in the process you are. And a general methodology, like Six Sigma - Define, Measure, Analyze, Improve, and Control -, Lean you have a three step. Maybe it's PDCA. Same as in Lean Startup, or similar as in Lean Startup. So that's sort of how they all compare; even though they are variations to them.

Jim: Right. And one of the things is there's a concept called Schoharie, which is basically this. You're at kind of a learning stage, or you're at an incorporation stage, or you're at a synthesis stage, if you will. And one of the

issues is it's very hard to describe to people the synthesis stage before they've gone through the first two. And when you're going through the first two, you're learning comes from martial arts. So you're learning how to kick. You're learning how to block. You're learning how to do all of these things. And while you're learning those, there's a right way. When you get to the synthesis stage, there's no longer a right way. You can do it any way you want as long as you're successful. But as thousand of people have come through, like the black belt process, or in Agile, the Scrum Master process, they get to stage two and they freeze. And so, one of the hardest things as a Lean practitioner is to understand that the notion of Lean says that our ideas are always wrong. That things can always be improved. And so, improving Lean becomes as hard as it was for Agile community to expect that maybe there was some improvement we could make to Agile.

Michael: Yeah.

Jim: That is one of the hardest things about selling this into a company. Because people who are buying want to buy a product. Just like the ER people want to people come in and solve all their problems. And what the product actually is, is teaching the organization to not have to buy anymore products like this.

Michael: Right. To always be self-improving themselves.

Jim: Yes.

Michael: Exactly. Alright. So let's talk about nuts and bolts, Jim. How is Lean for small teams implemented? What are the steps?

Jim: Okay. So what we do, generally - I'm trying to think of a way to not talk about this in the first person.

Michael: So let me preference that by asking. Can Lean be implemented step one, two, three, four, five differently at five different consulting companies that you go and ask?

Jim: Probably, yeah.

Michael: Okay.

Jim: Because everybody has their own methodology. So, Jim Lomack's methodology is to go in and then say, 'I would like to walk Gemba'. And then he goes out and he walks to Gemba. And Gemba walks are fun. It is a riot. It is so much fun. But one of the issues is, when you move into knowledge work, that the Gemba, like I said, is no longer a series of steps on an assembly line. It's just a bunch of people sitting in a room.

Michael: Yeah.

Jim: And so, walking to Gemba would be walking a circle around a bunch of cubes.

Michael: Or trying to get into somebody's head.

Jim: That's right.

Michael: Right. Okay. So it's different for any type of organization? Just like implementing Six Sigma at any type of organization may look differently. It may be a top down deployment where the CEO says, 'We will do this like Jack Welsch'. Or it may be bottoms up where just a software development group may say, 'We want to do this'; or just an HR group may say, 'We want to do this'.

Jim: Exactly.

Michael: So, having said that, I'm asking you what your process steps are. So what are the steps for implementing Lean within small teams from your perspective?

Jim: From my perspective, the initial process - it almost sounds like a 12-step program - is to admit that you have variation. 'I am Jim and I have variation.' And that's not a bad thing. That is endemic to the work. And that's hard for a lot of professionals to accept.

Michael: Sure. And what might that variation look like? I don't want to talk about software development. Let's say that I am processing loans. What might that variation look like if I'm the one who's actually reviewing the details of the loan, running my analysis on credit worthiness - things like that -, and determining, at the end of the process, whether they are approved or not?

Jim: Right. So, let's divide that up into a couple of sections. So the first section would be variation based on failure demand. Failure demand is John Seddon's concept that we often create our own waste by building systems that have failure inherent in the system. So, when I was in college, I got a temp job in the middle of the night processing loans. And I was the only person in the bank in the middle of the night sitting there in this lonely room processing loans. And, probably, three loans out of five didn't have all of the information on them. So I had to kick those back. So that was failure demand number one. Failure demand number two was, I think, pretty much once every three days the fax machine would stop communicating with the fax machine on the other side. And then you would have to do a bunch of stuff, and then wait. I never did find out what was going on, but then there was like that mechanical failure demand.

Michael: Okay. So all of those failure demands are variation that occurs within the day-to-day process of doing that job. It could be related to the specific tasks; like hey, they left this field blank so now it's an exception. I need to kick it back and it's going to reenter the process again. Or I drank to much coffee and I got to go to the bathroom, or the fax machine doesn't work, or the boss keeps e-mailing me. Just all these sources of variation that are going to affect how well I do my job.

Jim: And from the other side of that, it was frustrating for both, the loan recipient and the loan officer.

Michael: Sure.

Jim: Because they were taking stuff to me where literally my work was fifteen minutes alone. And because of the failures, things can be delayed two, three, four days for fifteen minutes worth of work.

Michael: Right.

Jim: So there's that type of failure. That's a type of variation. There's also the variation where, let's get out of a processing mode like that, but let's say I'm writing an environmental impact statement for a construction project. And I'm writing away and all of a sudden I find out that there's a situation where the data that I've been given can give me two conclusions. They're two completely logical conclusions and the only way I can find that out is to gather more data and/or talk to other people. That is variation. And so, ordinarily, writing a section might take me an hour. Now it takes me question mark. And that inventive data driven process is highly susceptible to that not only for me, but for all of the inputs to me. So I might say, 'Okay, I'm going to write this on June 3rd. It's going to take me two days'. And then on June 3rd I get there and I only have seventy-five percent of my data because of data upstream variations.

Michael: Right.

Jim: And then I'm, of course, late. And then that causes variation downstream.

Michael: Sure. Alright, Jim, I want to be respectful of your time. Because you're so generous to come here and walk us through. And so, I don't want to necessarily have to go through too many examples to exhibit what the steps are. Because we just want to make people aware of the overall steps. I think a lot of people that read iSixSigma are educated; or they have additional resources that they can go to to be understand individual steps - if there are twelve steps. I don't know if there are or not. So, step one is understanding or admitting that there is variation in the process.

Jim: Right. The second step is recognizing that the people on the team are all professionals and that their goal in life is to do a good job. And, yes, there will sometimes be people on your team who's goal is not to do a good job. And there's a quick remedy for that.

Michael: Fire them.

Jim: Yeah. But there, ordinarily, when we don't recognize that, we blame systemic problem on people and then we come to the conclusion that people routinely do a bad job. But in almost every instance that I found, it's be a systemic problem. It hasn't been an individual problem.

Michael: Yeah. And in Six Sigma, we teach: it's the process, not the people.

Jim: Yes. And that's kind of a mobius strip statement. Because it's the process; not the people, but it's not the process. So the people need to; I like to say, in knowledge work, we moved from walking to Gemba to Gemba activation. And so, when we understand it, we kind of end up in a Marshall McLuhan situation, where not the people are the process. Because the team, itself, is responsible all the time for its own process, for its own process improvement, for its own Kaizen events, for its own remedial actions, measuring, acting; the whole bit. Now, it's the people and the process; and they are one.

Michael: Okay.

Jim: And marrying that group directly empowers everybody over the entire value stream as apposed to the silo of their station.

Michael: Right. That makes sense. Alright. Let me ask you for a couple of definitions, Jim. Because I'm not sure if they're in the iSixSigma dictionary, so I'm going to give you credit for them. Can you define Gemba for us?

Jim: Okay. So, in TPS, Gemba is the crime scene. It is the place where everything happens. And so, when you do a Gemba walk, you go walk the line and you talk to the people on the line. You ask them what they're doing, how things are going, if they have any suggestions, if there any problems that have come up, any solutions and so forth. If there's a specific issue that happened, you will go address the Gemba and say, 'Hey, we're seeing these numbers coming back. How can we, as a group, fix them?' So, the Gemba is a concept that was developed to respect the fact that line workers were actually professionals with brains that could give value to the company more than just turning things and shipping things along on the line.

Michael: Got it. And it's a Japanese word if I understand.

Jim: Yes.

Michael: Okay. So that's the definition of Gemba - where work is done. And walking the Gemba is walking the line, or walking the ER department, or walking through human resources and talking to the people who are doing the work, and respecting the fact that they do the job on a daily basis, they know what the problems are, and they're probably the best to solve them.

Jim: Yes. And so, when you walk into the nursing department, for example, you will see a bunch of nurses. Maybe you can look at how they do paperwork. Maybe you can look at how the screens are located. And those are all good things. But those are all the built environment. Those are all manifestations of existing process. And if you're looking for ways to improve, looking at the existing process may not be the best way to go because you're only going to improve the existing process, you're not going to have any radical new ideas. With visual controls, one of the things that we're trying to do is create a Gemba in a place where previously there was none. So, previously everything's happening in people's heads. Now, we actually see on the board these are the people, this is what's happening, this is what's being built, this is the stage they're in, this is what's blocked, this is what's not blocked, this is a dependent on this task, and blah, blah, blah. That means that you can walk the Gemba. Whereas it's a literal Gemba for manufacturing, this is more of a figurative Gemba - symbolic Gemba -, but a very, very precise one. And value stream mapping is - I'm hesitant to say - something that I would demand. But it's very, very close to something that I would demand. And what I found with knowledge working teams is that when you map a value stream, the first reaction is everybody is annoyed that they have to come in and map the value stream because it's obvious. Now, we sat down and I said, 'Okay, if this is obvious, this is great. It'll only take fifteen minutes and then you can all go back'. And so, we write up the initial value stream and I say, 'Okay, this is what happens'. And everybody is like 'yes' every time.

Michael: Well, there was that one time and suddenly your fifteen minutes turns into three hours. And you've got people moving things around and you've got exceptions to the rules.

Jim: And so what happens there is that the team that you're meeting with recognizes that fundamentally they had different ideas of how value was created in the first place.

Michael: Right.

Jim: And that's wildly powerful.

Michael: And even what the definition of value is - to whom? To the customer. To the company.

Jim: Yeah. And that's one of the other things. In a lot of Lean and Agile rhetoric, and Lean Startup rhetoric, there's this focus on value to the customer. And so, you start to have disfunction because you are pathologically focused on value to the customer, but you don't recognize things like the SEC actually should be paid attention to from time to time. You should probably do your bookkeeping even though it is not a final value to the customer. Because if your company closes...

Michael: That's not a value to the customer. Right. Exactly.

Jim: So that's another kind of anti-pattern that I've noticed. Is that people are like, 'I don't want to do that anymore because it's not value to the customer'. It's like, well, it's a value to a customer. And that customer is the SEC.

Michael: Right. Okay. So, step one was admitting variation in the process. Step two was recognizing that people on the team are professionals and they do want to do a good job. Step three?

Jim: Let's go ahead and say that value stream mapping would actually be step three.

Michael: Value stream mapping. So that's process mapping and then actually determining what steps are value added and what steps are non-value added.

Jim: Right. And then, for me, step four would be to actually start tracking work through that value stream. And to have that be a participatory process. Bosses do not move work items. The person doing the work. So, just like any Kanban, as the object moves through the system, the system is moving the tickets.

Michael: Yeah.

Jim: The limiting work in progress is obviously going to be the next step. Is to understand that the more out of control your work in progress is, the higher your variation will be. The lower your quality will be. The more often you're going to miss deadlines. I keep going back to Deming. At this point, and reason this hasn't been mentioned before because it's Deming's point one and for me it's like six, is that now everyone can see that there is a system.

Michael: Right.

Jim: So now we can appreciate that there is a system. If you start off by saying you guys are part of a system, they'll say things like, 'Yeah, I'm part of a system. And everything that breaks is his fault. Because he's the broken part of the system'. So getting people to understand systems thinking by actually seeing the machine of work move along. And then, I have two tools that I tend to use rigorously. And the first I mentioned already, which is the Standup Meeting. And that comes from Agile methodologies. And it's basically a fifteen minute meeting that the team has every morning and you basically talk about what's happening. Before having a Kanban, everybody would have to verbally say what they had done and what they're doing. Since that's already handled on the Kanban, you can now talk about the flow of work, the structure of work, what's blocked, and things like that. So you can have much more substantive conversations. You want that to be a short conversation and if longer things come out of it then people can peel off and go to those. Ideally, if you can handle it, that's an awesome time to swarm on a problem. For instance, I started this yesterday; I'm hung up on this; can you guys help out? And you get five mind on it in a very short period of time.

The second part of that is, what Agile would call a Retrospective, and what Lean would call a Kaizen event. To have regularly scheduled team meetings where you talk about the process, talk about continuous improvement, and make sure that continuous improvement is actually happening. And while it is possible to go through some points in time where continuous improvement is not happening right now, if that goes on for more than a couple of weeks then you know that people aren't paying attention.

Michael: Yeah.

Jim: That would be my major steps. The training in the toolkit, which Ken includes personal Kanban, or A3, or 5S, or any number of other things. And to understand the differences in the different loops that are out there.

Michael: Right. Sure. That makes perfect sense. Because you've got the Lean Deployment Planning and Training. And then you've got the actual team training, which they then implement on a daily basis. And there's a difference between the organizational implementation and the team implementation. And so, I appreciate you bringing that up. That we're talking, mainly, about the team right here. And that if somebody wants to go implement it in a team, they can follow your seven steps or so that we just went over. But there's probably more if they wanted to make their entire company more Lean focused.

Jim: Right. And I would say another, not necessarily a step, but a warning is never feel like your team is autonomous.

Michael: What do you mean by that?

Jim: When we draw up our value stream, then we create our own silo and we don't recognize that there's business value up and down stream from us and concurrent. And that ends up creating single focus wasteful teams that don't collaborate across different parts of the organization.

Michael: Right.

Jim: And large consulting companies - you'll see this happen all the time. They will go out and get sub-consultants for things that are done in-house. Because the company is so big and people so don't know what's going on that they don't even know that there are resources internally to turn to for those.

Michael: Yeah. That makes sense. Alright. So, many times, when I hear about Lean for small teams it's almost synonymous with Lean for software development. How does it shake out in all of your implementations of all of the companies that you've worked with? How many are software teams doing Lean? And how many are non-software teams?

Jim: I'd say fifty percent are software and the other fifty percent are a wide range of other things. So, we've worked the UN; with the World Bank. We used these to write the Human Development Report for Vietnam a couple of years ago. And that was zero software developers. All social scientists. We've adapted it to accountancies; to education. Starting to work with healthcare on that. Pretty excited to get into that. And where I'm really excited in healthcare is not necessarily in like big Lean healthcare, but in working with research groups or with sections of hospitals who are trying to track certain things about certain patients. And also just to make everybody in the organization, or that portion of the organization, more aware of what's happening in their pod. And I've worked very hard to keep a balance because IT is definitely the low hanging fruit. Because everybody is very frustrated in and out of IT. IT is very frustrated with everybody else and everybody is frustrated with IT. But any improvement that we make to an IT group has to have a commensurate improvement to the rest of the organization. One of the clients that hired us - a huge, huge company in England - called us up and they said, 'We have a group of software developers and they're crazy. And we can't make them do what we want'. So we thought that the problem was that they weren't Agile, so we sent them off and go them Agile training. Now they're crazy ten times faster. They're much more efficient at being crazy. Can you come in and either fix them or fire them? And so we said, 'Okay'. So we went in and by and the large the team was fine. They were just getting conflicting requests from the rest of the organization, literally, on an hourly basis. And so, that ended up not being a software problem at all. It ended up being an interface problem between the rest of the organization and one part of the organization. So while our credibility in software came into play there because we were

able to stand on that and say stop torturing these people, the actual remedies of that were all systemic outside of the software development part.

Michael: Yeah. Alright. Let me look through my list of questions here. If anybody has questions about Kaizen events, we have a lot of information on iSixSigma about that. We've defined Gemba. I'm going to ask you about Kanban. We talked about personal Kanban, which I think is great. I'm going to give a plug for you, Jim. I taped your presentation, like I mentioned earlier, on Personal Kanban in Seattle. That's live on iSixSigma. If people want to watch that, I'm going to link to it below this show and also you can go to, on the ride hand side in the main menu, tools and templates and Kanban is listed there. And you can see your video there with the entire transcript and everything. But what if a team is distributed? What if you have people that are knowledge workers? They could be a software company. I'm here in Bainbridge. I've got my developer across the US. I've got another developer in India, let's say. Or let's say that I run a financial organization where I don't care where people work. I know you're a knowledge worker. You're intelligent about processing loans. I'm going to hire you and you're over in Seattle and I'm down in Southern California. How can you use these type of tools? How do you make your process more Lean when you have a distributed workforce?

Jim: So that goes back to the Cynefin model. The more distributed your team, the less fidelity of communication that you have across members of that team. The more that fidelity is decreased, the more you're going to have to move your tasks back towards the simple domain of the Cynefin model. So, simple tasks you can outsource any time, any day; no problem because they have a very set scope of work. You say, 'Here's what you got to do'. The person says okay. They do it and they hand it back. So typing the word 'Hello'. I can outsource that to anybody at any time that can type. The more that you move up into the complicated domain and into the complex domain, the more you're going to have communications issues. And that's not to say that you can't outsource complicated things. It's to say that once you do that, you need to have mechanisms in place to facilitate that level of communication. The first, and most important one, is to know those people individually. So, one company that I work with in Tennessee. They are constantly flying people to India and flying their Indian outsource people for

extended stays in Knoxville. And there are always people from India in office in Knoxville. So that when they're having conversations overseas, those conversations aren't limited by culture and they aren't limited by unfamiliarity.

Michael: Right.

Jim: The quality that comes out of that company is phenomenal. It's mind blowing how they've been able to do that. But that is a significant commitment that flies directly in the face of why most people think that they should offshore. So, again, having an appreciation for the level of variation in our work. And as that variation goes up higher, and higher, and higher, and you get beyond complex or complicated into complex, outsourcing becomes really, really difficult. The bizarre thing about most knowledge work is that it's all over the dial all the time. So, me needing to fill my printer with paper is a simple task. That does not mean that my job is simple. So, identifying the portions that are outsourceable based on the fidelity of communication is the way to have successful Lean when you're outsourcing. Beyond that, you have to have make a choice about whether your outsourced people, or your distributed people, are part of the team or they're an external resource that you're drawing from. And if they're part of the team, then they need to be part of the Kaizen events, they need to be part of Standup Meetings; they need to have the identity that they are part of the team. And the moment that that identity starts to break down, your distributed team or your offshore team is going to break down as well.

Michael: Yeah. And I can see it my head right now. Because I do have a distributed team. And I'm just walking through the seven or so steps that we discussed and I see them myself. And we do track our work through the value stream map. We do limit the work in progress. But we don't do Standup Meetings as an entire team. And so, I can see how that would be useful for implementing today and also, doing Kaizen events on top of that so that we can improve how we work together. So fantastic, Jim. Let me ask you this as a final question. We talked about pitfalls that you've seen in your prior experience in implementing Lean in small teams. We talked about one of the biggest pitfalls is that thinking that teams are autonomous and they're not autonomous. We don't want silos and we don't want to just throw things over

the wall and say we're done with it. What other pitfall or pitfalls have you seen, and how can they be avoided if somebody wants to think about implementing Lean for small teams?

Jim: A whiteboard, but it's all the way over there.

Michael: I know how you like your whiteboards, Jim, so it's probably better that I can't get you over there.

Jim: The anti-patterns are rich and detailed on their own that we've managed to identify. But they can be kind of boiled down to a couple of things. One is fundamentally not understanding the nature of Kaizen. And the fundamental nature of Kaizen is that it's very (Unclear 1:14:32.6). The only thing that's constant is change. And so, understanding that when we're going through a Kaizen event, there is nothing sacred. There is nothing that we can't take off the table. That's a hard one for people and that gives birth to a lot of different anti-patterns. So, the fundamental understanding that our goal in business is to stay in business - that outstrips just about everything else. And we do that by producing quality products that people would like to actually purchase. And sometimes the product that we love doesn't give anybody value; or has been corrupted by the market. It used to be a lot more profitable to a pop star. The market has changed that. So now you have to have a line of clothing in addition to being a pop star.

One of the other, I guess root causes of these anti-patterns, is not being honest about the amount of work the teams have in progress. And it is hard to be honest about your work in progress when you're a knowledge worker. You don't know when a meeting is going to show up. It's hard to put tickets on your Kanban for every meeting that comes up. Your phone can ring at random. People show up on chat. They show up on Skype. We are so eminently interruptible now that not understanding how to limit our work in progress is pretty huge.

And then the third is by not being honest about what are true throughput is. And I try and get around to this by using the visual controls to actually create real cycle time. And then recognizing that until you understand the ugliness of your cycle time, you're always going to be demoting certain tasks. You're

always going to be late with things because you're never going to value the amount of time it actually takes you to complete stuff. And I've seen it happen a lot of times. People will note the start time for a task and the end time for a task. And when they get to the end, they'll rot it up and throw it away because they were interrupted. Well, that interruption is a constant interruption. So there's kind a helper app that we use called The Pomodoro Technique. And The Pomodoro Technique says for twenty-five minutes you take your task and you work on it for twenty-five minutes. And then when it's done, you're done. You put it back and go for a walk or something. You try and do something physical. And what happens when you do that is, while you're working, you're filling up your short term memory. At the end of twenty-five minutes, you go for a quick walk. It doesn't matter even if it's in circles around your cube, but just get up and move your arms. When you do that, it moves that short term memory into mid-term memory so that you can file it into long term memory when you go to sleep that night. If we tend to blast through the day, we tend to fill up our short term memory and then we start losing stuff. So that type of focus also means that when you're in that twenty-five minute bubble, you've shut off your phone; you've shut off Skype; you've shut off everything that can interrupt you. You put a note somewhere, like maybe on the back of your head if you're in a cube, saying, 'Leave me alone for twenty-five minutes' so you get actual focus. And what's interesting is that we found that when we introduced these simple concepts upfront, people feel so much better after doing that that they come and start looking for all of these secondary things that can make it even better. So one of the reasons why I was hesitant to list a whole bunch of steps is because our technique has always been to come in and give you as few steps upfront as possible and then let you kind of get in it and swim around in that. And then, expand that as you go along.

Michael: Yeah. That makes sense. Alright. Jim, this is the point in the show where I urge the audience. Right now, if you received value out of this interview, and I know I did, take a moment to say thank you. It's as easy as posting a comment below the show, sending a tweet to @OurFounder. Is that right, Jim?

Jim: That is right.

Michael: @OurFounder. I'll include a link below. Take a minute to do so. I'm going to say thank you again right now by mentioning the Modus Cooperandi website at www.ModusCooperandi.com. I'm going to include a link below. On that site you'll find some terrific information. You'll also find links to Jim's Books that are listed on Amazon about Scrumban, Personal Kanban, which I use in my day-to-day operations, and his other book, Why Plans Fail. And don't forget about Jim's other presentation that we've recorded on Personal Kanban. I like to download these, Jim, in MP3 format. I put them on my iPod. I listen to them while I'm working out or while I'm driving - in a lot of different ways. And so, I encourage people. There's more than just watching it on video. You can listen to it on MP3 or you can just download and read the transcript while you're hanging out at night watching some TV.

Jim Benson, CEO of Modus Cooperandi. Thank you for coming on the iSixSigma Show, sharing your knowledge, and helping others become more successful change agents and business leaders.

Jim: Wonderful. Thank you.

Michael: Thank you all for watching. We'll see you next time.

Watch the full video at:
http://www.isixsigma.com/lean-for-small-teams/