

Jim Benson on Personal Kanban, [Modus Cooperandi](#)

Watch the full video at:

<http://www.isixsigma.com/tools-templates/kanban-tool/personal-kanban/>

Jim Benson: Hello everybody. Why are you guys hiding behind the pillar? Why don't you come back out here and hang out with us a bit. So as you can see I have no computer, I have no slide deck. You will see no slides tonight as I think Power Point should be eradicated from the face of the planet. This is so easy. So I'm Jim Benson. I am the CEO of a company called Modus Cooperandi. And we do collaborative management. One of the major tools that we use is a Kanban. Several years ago we did our first kanban project plan, my software development company that was 2006/2005 and noticed almost immediately that kanban made an extreme difference in the way that teams operated. Raised the level of clarity and the level of awareness, the level of conversation, gave rise to continuous improvement. Made that continuous improvement real time and completely changed all of the rules to the agile processes that we were doing before.

So I will rewind a little bit. I went to a university and studied psychology and then one day I looked around at the people in my department and said man if I become a psychologist these guys are going to be my peers. So I figured I had to do something else right away. And I got a degree instead in Urban Planning. And so for about a decade I ran around and built light rail systems and bridges and freeways and did things like re-plan 405. That was six maddenly week times. So I have managed projects of billions of dollars without thousands of people involved.

So then I open a software company and everybody is like wholesale development is hard. We have big teams and it's so hard. And I'm like seriously? And our budgets are really gigantic or they are really small. So it really kind of screwed with my mind. You have got eight people on your team, it's not a big...so when we started out software development company our clients were government agencies and we were selling Agile to government. And we were very successful. So the company ran for nine years until we got sick of it. And we went off to do other things. And for me it was very easy to sell Agile to government even though it was very, very

foreign to them because we could speak their language in essence. So we could run that translation between what our teams were doing and what they needed to do.

And again in about 2006 we did our first Kanban project and I was completely blown away by how different it was. The first thing I did was fire myself as a product owner. So we had no single point of contact between our development team and our client. And then after that it was just kind of a heyday for breaking all of Agile rules and finding out they weren't quite as sacred as they thought they had been over the past eight years. So we were talking about Scrum a second ago. If you really want to pull my string make me talk about Scrum.

So step back here...

Audience question: Let's talk about Scrum.

Jim Benson: Yes I'd love to. So I'm going to start here by talking a little bit about what Personal Kanban is. I could but I need both of my hands. For Personal Kanban we have two rules. It is visualize your work and limit your work in progress. And in the end we find that if you give teams those very two loose rules they go crazy. They become very productive but more importantly they become more effective. They stop doing stupid things. Whereas before they do a lot of stupid things because we are trying to either use the Agile world where we are trying to smash a bunch of stuff into two weeks or the no holds barred waterfall techniques to where you test six months and you try and do 50,000 things in those six months. So visualizing your work. Initially you have your value stream which is the steps that you take to create value. And the simplest one possible is ready, doing and done. Basically the stuff you haven't done yet, and the stuff that you are doing and the stuff you have completed.

Audience question: Could you use a thicker pen?

Jim Benson: I could get a thicker pen, if that makes you happy.

Audience question: (Inaudible 00:05:09)

Jim Benson: I would but these ones don't write very well. Really they are tough. So at any rate as my lovely assistant will write right behind me, what this has done here is Bill's...A quick story of how we work. And it is a very universal story. All languages, cultures, every song, movie, epic poem is based on this. It is a birth, life, resolution story. So once we started doing something like populating this with our work, whether that is tasks or whether it's user's stories or whether it's bugs or features or whatever, however we may be setting those up doesn't really matter. What does matter is we are talking about stuff that needs to be realized, that excitement of realization and then the completion.

For our Personal work right now every single person in this room has expectations based upon them. So we have stuff from work, stuff from family, our own personal expectation and stuff from our friends. All that stuff right now is jumbled up in our heads in a very conceptual or amorphous blob. And if we looked at it it might look something like this. And this is not very manageable. So what we want to do with the Kanban is grab these things and these are super sticky ones and sometimes they are hard to get off and we populate our backlog. And when we are doing this we are starting to make sense of all of these expectations that are upon us. And what we have done is we have turned that amorphous blob now into something that is cognitively much more pleasing for us. We've turned this into a rectangle. And while it seems kind of ridiculous now we can see our work and it's not quite so threatening.

One of the most kind of heartwarming emails that Tonianne and I have received...I should say I've written a book called Personal Kanban with my co-author Tonianne who is lurking about somewhere around here. One of the best emails that we received was from a family who is a very strong patriarchal family. They have totally loved their Dad and it looks kind of like this if it is held my Michael. And they suddenly found out that their Dad had Stage Four cancer and this guy was like the glue of the family. And the family is like bwah we don't know what we are going to do and they were completely ineffective in doing anything. So they set up a Kanban just to manage his treatment. And after they got the treatment up they are like oh this is just a bunch of stuff to do. It's stressful and it sucks. But it is just a

bunch of stuff to do and now we can focus on helping our Dad and making him feel better and healthier and get through all of this.

So this comes back to a concept that we call Existential Overhead. And Existential Overhead is all of the fears, all of the pain that you have that is following you around right now and that lowers your capacity for work. So just like in normal Kanban where we will set a width limit here for the team and say that the team can only be working on three user stories at a time, we also in Personal Kanban set a width limit and that width limit directly relates to how much work you can actually do right now. And if you have a large amount of Existential Overhead the amount of work that you can do decreases. And a lot of the Existential Overhead that we give ourselves is work in progress. So if we are doing six, seven, eight, nine things at once or we have six or seven or eight or nine things that we haven't completed, we never stop thinking about them. So it is something that is called the Zeigarnik Effect. And the Zeigarnik Effect says if you finish a task you tend to kind of forget about it and if you don't finish a task you never forget about it. So if you have 25 things that you have not finished or haven't reached some resolution on they are going to bug you at night. I'm not done with that, I'm not done with that. So it is hard to get to my kanban here. So we will just move on here to flow.

And what we would like to achieve in this kanban is an understanding of how our work flows or how it can flow gracefully to the ground. So right now there are three things in our doing column and we have this width limit of three. We finish one and then we can pull another one. So this is a pull basis system. So Scrum is kind of a hybrid pull/push base system. So what we have done here is we have pulled this because we have capacity for it. And Scrum or in Agile what we do is we have defined this two week period and we believe we have this much capacity so we are going to pull this much work. But in reality what we have done is we have pushed work onto ourselves. So what we want is to understand what the cadence is of our work and that starts to tell us some things almost immediately. So things flow through here and flow through at a regular rate. And this one gets stuck here for a couple of weeks where things usually take a couple of days then we understand that there must be something the matter with this. So that is like the quick version of flow there.

We talked about pulse, we talked about clarity. So what we found when we started working with kanban was that the teams suddenly had a greater clarity not only on what they were doing but what their peers were doing. And this is in real time. So before when we asked the team what is going on they would say well we are in this sprint and we have these user stories and we know that we are being worked on or that they are going to be worked on. What happened with this transition was people said this is exactly what is happening right now, this is exactly the state of completion and this is exactly what is done. The upshot of that is that you can make more decisions in real time. So you have at the beginning...you have these thing prioritized in this do, do, do, do like that. So this is the one of the highest priority but you get to the end of the day and you know that you can do this in an hour and you pull that and you get it done in an hour and you go home and you try and leave that open.

So I wanted to go through that pretty quick tonight because I wanted to talk more about the Lean start up ascetic and how this relates to what is talked about in (come on just stick to something) to the Lean start up books. And one of the things that we have discussed quite a bit in the last few years is what the role of failure is in an organization. We want to fail fast and we want to have pivot points. We want to make course corrections along the way so that we can avoid a lack of success. And that does not mean if we do the Lean start up methodology we will be successful because business fail statistically so that's something to keep in mind. We will talk about that in a little bit. I'll look more in a little bit.

When we are working with the kanban and when we are working with a visual control one of the things that we are blessed with is this understanding of this is what we have completed this is what is coming up and we can start to set experiments up that are real time and extremely provable which is pretty vague I know. But what we do here is as work is flowing through we have what we call cycle time and cycle time is the amount of time it takes to pull something out of here and complete it. And when we start to understand that then we start to understand what our rate of completion is. And one of the things that I have found in working with now a lot of different software projects is the distribution of tasks or the distribution of stories is fairly

consistent. So we spent a lot of time in planning meetings saying is this a three point story or five point story? And I know that we are not supposed to but we end up spending a lot of time doing that but in the end it doesn't really matter because it all balances out. So what we do instead or what the teams I work with do instead is we will do regular releases, two or three or four or whatever or real time releases depending on what the industry is. And whatever makes it into that is what gets released.

So here, oh I could go five different ways with that. So one other thing I will say is if you guys have questions, ask questions as soon as you have a question. Yes Ken I believe you have a question.

Audience question: I still have no concept of what the difference is between the way I execute Scrum and what you are doing here.

Jim Benson: Do you have width limits with Scrum?

Audience question: I don't know what width limit is in particular, we have the ability to work on one thing at a time each and so if there is six of us and we are working on six different things then we work on six different things at a time.

Jim Benson: Okay.

Audience question: There is a very distinct limit on some things. As far as we talked about the sprint I think that the way you talk about it seems to be with people who have tried to impose waterfall expectations into what has been dropped to Scrum. But I think that is more of a limitation problem. The way we have always been doing this, and since 2000 we have been doing this, if we get it we are going to work on it for two weeks and we expect that we are going to get this done but we don't know. So we get it done when we get it done. Other than this being the breaking it down into the kanban board, breaking it down in a slightly different way between the back log and what we are actually working on at the moment and what we have completed I don't understand any real functional differences between the two. I was hoping maybe as you talk you will be more clear about why this is different.

Jim Benson: Did everybody hear that? Good, okay...

Audience question: Could you repeat the question?

Jim Benson: (Laughter) The question was what? So there are a couple of things that I will try and work out quickly. Number one is yes the way you are describing it is the way that it is supposed to work. And works that way very, very, very rarely. In an interview Ken Schraver said that 75% of all Scrum implementations failed because people aren't doing it right. When I hear that I don't think there is a problem with the people.

Audience question: So again one of the questions is how is this going to make people smarter because again I don't think that it is that hard I think that people have a mindset, they have that mindset and they will have that mindset.

Jim Benson: Who are the people?

Audience question: The people in management in general. They have the concept of saying I'm going to get this feature on this date. That makes them warm and fuzzy inside. So whatever system you have they are going to try to force it into a way that it tells them I'm going to have this feature on this date. Agile was never designed to do that. It was designed to tell them you will probably have this feature around this date. So what is the difference between five years from now or eight years from now when people are forced, the same people are trying to force the same thing onto companies?

Jim Benson: So there are like five conversations there. So the first thing is the devolution of any management idea. Okay? The reason that we go through different phases of management or different management fads is not because the previous ones became or were flawed is was because we as human beings will fuck things up over time. So Tonianne and I had a client, they had five different teams and the teams were mandated that they would all do 100 points per iteration. It was completely ludicrous. And of course they did. They were always very successful at getting that done because it was complete garbage numbers. Having said that management is necessary. And so what this has told me is that story points and iterations have not been

entirely successful at being a communication mechanism between coders or coding teams and the rest of the organization or their clients or whomever they are working with.

So with kanban we wanted to do was put up a board that says dude, this is what we are working on, this is what we have done, this is what is coming up. Do you ever have a problem with that because we are going to show you this 24 hours a day, seven days a week? Not only that but if you come down and tell us that we need to do more we already know that this is our optimal processing capacity as a group. This is our speed limit and if you go above you are going to have either breakdown in quality or a breakdown in cost - And so the width limit is a fairly explicit elucidation of what the team can do in a high quality way.

Audience question: And wip is...?

Jim Benson: Wip is work in progress. Sorry about that. So we want to limit our work in progress as individuals and as groups. We want to limit it to a point where we are comfortable completing that work. And there is a couple of interesting charts that I can (I can't flip this) so there you go. It is sticky and I can run back and forth with this and re-stick it repeatedly. This is a deadline metrics. So here we have 'errors' and here we have 'here is your deadline' so this is 'time to deadline'. I don't care what the deadline is. This is how that curve looks. So what happens is as we get closer our arbitrary deadline if that arbitrary deadline is anything other than what we can actually complete in that amount of time as we get closer we get more and more stressed. And this is technically debt generation unit. Now in Agile the original goal was we were going to say we believe that over the next two weeks, or the next iteration we are going to be able to complete this much stuff. And then if you didn't you would say crap. I hope I can do it next time. But as Ken pointed out the people that we work with don't speak that language. So with the kanban we wanted to remove that language altogether and to say you know we understand that we have this cycle time down here and this cycle time we've generated by (I really didn't think I was going to go all technical tonight) by plotting out the completion times of the stuff that we have done over the last couple of weeks, or the last year, or whatever that might be.

Audience question: That is yellow.

Jim Benson: That is yellow. It's gorgeous, (you guys are tough), so over time what we could do is we can get a composite cycle time number that we can actually start to offer, an SLA or an SLA like thing to the people who we are working for. Because we know that 85% of the stuff is going to be done, if not at this point significantly before this point. Most of the stuff that we do that doesn't make that point is just outside this line and you can have a quick conversation and say you know what we thought it was going to take two weeks, it's going to take two weeks and two days, sorry. Then the farther you get out here you are like do you still want to complete this or this is insane.

And what we have also found is that if you give a group of coders a bunch of stuff and you say give me story points on a Fivenati (spelling 00:25:05) curve from whatever to whatever they will do it. But if you also give them that same bunch of user stories and you say tell me which ones of these are stupid, they will do that too. And the second one is actually more valid than the first one because the ones that aren't stupid will fit into this area. So rather than starting a two weeks sprint and using the horrible word of commitment...

Audience question: Ugh.

Jim Benson: We can actually now say over the next two weeks we are reasonable sure that we are going to get this stuff done and since we are not committing to a group of features we can pull anything out of this back log that we want over the two weeks and it will hopefully make sense. One of the other things that happens is (let's take this one off) and let's say you are sitting here and you have got three things in WIP and all of a sudden your load balancer explodes and then you can say you know what I've got three things WIP you have to wait for my load balancer, and I will fix the load balancer next week. I hope that the 65,000,000 hits that we were going to get aren't too upset. But no you can put it in here and fix it and no one dies. Right? There is no WIP police that comes in and says dude I'm going to totally kick you around for going past your WIP.

So what we want out of this is to be able to exercise our professional judgement at any given point in time about what is the important stuff to be doing. So we want our teams to be able to truly self-organize to use the Agile term. Not only in their composition of who is on the team but also in their composition of what work is being done at any given point in time.

Audience question: This technique applies per client who totally understands how a massive delivery is broken down in the past and who is doing what. And so typically working on itinerary projects that are quite big you need some sort of a... What should I say I see my tasks in what they call more fine grained but I'm saying oh okay this is not looking like a one week business. I've got to spend more than two weeks on this. So what I'm saying is that the more I look at that board what I visualize is a lot of refiguring as I get more clarity, as the team gets more clarity. I mean does it allow for that, I mean is it malleable enough to support that? Because all said and done if every project is a nice right down to the past or kosher, we can go with that or leave that. Big when a big wig client comes in the process of tasks breakdowns itself makes sense only when you are in the WIP stick. Is it can you juggle things around, is it malleable enough?

Jim Benson: I'm really enjoying the fact that people are asking questions that are loud enough to hear. Did you guys in back hear that? Awesome! This is cool. Okay so the whole point of this is to make a system that an A fault tolerant and be eminently flexible. So number one I don't care what anybody puts on their tickets and so I see boards all the time where people will set up...they say that we are going to work on one epic at a time. And they put that epic in and it sits up here in a column and it explodes into a bunch of stories and then they will actually have those stories explode into a bunch of tasks that may be pulled by any member of the team. And in the middle of that should something explode or should something become more important you can find ways to move that through your stream if it is really that important. So, one of the things that this is not doing is expecting that people will have to decompose those tasks up front.

So I have here three stories of three start... Yes?

Audience question: Just one more question before you are done.

Jim Benson: Okay.

Audience question: So one of the things that you were talking about was it takes time to get done toward the end of the iteration. So I can imagine positive and negative effects that happen in an organization based on illuminating that sort productivity curve from the measuring perspective. Can you talk about what the effects are?

Jim Benson: Oh absolutely. So did everybody get that or should I repeat that one? So the question there is can I talk a little bit about the positive and negative impacts of when we get to the end of a deadline and then the removal of that impediment. So one of the big shocks for me from one who had come from government and then from software development was that when you removed the threat of deadlines they actually did more. It completely freaked me out to the point where I didn't believe it for several years. I just kept gathering data. So it just keeps working. And so what I figured out happened was that when people see their work individual control it has a really interesting psychological impact on the team. The team, like I said before, they see their work, they see the flow of their work and they start gaming that system. Here I'll, because this is actually kind of fun and I hope that it is illustrative.

So let's say that we have this backlog and hopefully it is set up a little bit better than this. And we have a software development team. And that software development team is made up of people and those people are human beings and they all fall somewhere between these two dots. And this dot is the ADHD dot and this dot is the aspergers dot. And that dot is the hapless manager's dot. So we have our team. It's made up of these people. And actually like this and this. While I go around software developers that they are really not all that special we do really kind of be special. And so what ends up happenings is we have our kanban here and our ADHD coder who can take any weird problem and solve it in like five minutes. He has the three things in WIP and I come by a couple days later and after awhile it is like this and after awhile you are like dude three. So he is like okay and so what he has done is he has done this much of each of these tasks. And what are those dots? Those dots are the cool stuff. And the rest of this is not done. And so

you have to finish that and he is like oh it pisses me off so he quickly starts finishing things. And it goes over here.

And a couple of days later Adam and the other testers show up at his desk and they are like dude I don't think so. And what does this become? This has become real work. The only thing worse than doing stuff outside this circle, to an ADHD person, is to do rework. Rework is like oh I hate it. Then they really gave him the system and say I'm going to do the least of this stuff that I need to to finish these tickets and make sure that they don't come back because I don't want to do any rework. So what has happened is these guys have optimized for how they view the world and how they want to do their work. They have turned this into a game and they have just written the best game possible because our job as coders isn't to write code, not to write code. So we have taken this and allowed the system to break and then allow the system to self-heal. So that is awesome.

But then so what happens next? So we have our asperger buddy and she has pulled a ticket and it is sitting on her kanban and come back the next day and it's still on her kanban and I come back the next Monday and it is still on the kanban. And then we go on maternity leave and come back and it is sitting on the kanban. And so after awhile are you going to finish this and she says it is not perfect yet. When is it going to be perfect? When it is perfect. Human beings can't know perfection and Buddha says so. So now we say so what? You are pulling the rest of the team down. Our goal is to finish this stuff. So, now all of a sudden perfect has a definition. Perfect is how quickly can you write this code so it goes over here and it doesn't come back. So we have taken the completely different world views of our team and we have suddenly given focus and we have done something that we never really able to do in Agile which is we have now defined done. And we didn't have to do it with statistics or measurements or anything. We are using our own professional judgement as software developers and as entrepreneurs, this is done when it goes over here and it doesn't come back. This is fun.

And I've watched team after team who have been after each other's throats and after just a couple of days of fiddling around with this thing and they start having fun. And then they start doing things like team member A pulls that over and team member B says oh hey I see you pulled that task for that story,

I did something like that last week if you Google this stuff it is going to save you like four hours worth of work. That is just common human courtesy. That is what we do naturally. But if we don't have the visualization, if we don't know the other person is doing that stuff we can't help them. So being able to allow the team to see what the team is doing creates natural opportunities for collaboration which creates huge opportunities for costs time and time saving and increases in quality. So did that answer your question?

Audience question: Absolutely! When we pull from kanban we use a graph hopper tool to do our task breakdown and monitor but I think try (Inaudible 00:38:11) and it works like a charm. Everybody can see, to me the specific problem was that it doesn't enable much in terms of past breakdowns per se. It happens organically over a period of time. But the clarity of (Inaudible 00:38:36) in terms of others helping you out and sharing that while actually doing and looking up work is invaluable.

Audience question: I would sort of like a re-factor of his question a little bit. Something specific.

Jim Benson: Okay.

Audience question: He is talking about these various problems not very well defined and gets into works in progress, what actually happens to the board when I go oh shit this is a five week problem and I need to make it into smaller pieces?

Jim Benson: Right, so there is a couple of things with that. So which one to go for first. So we will go ahead and rip some more paper off here and do this again. And by the way your value stream can be whatever you want it to be. Don't think that this is the be all end all of value streams. The other day one of the teams that I was working with said to me we miss having planning meetings. And I said why did you stop doing planning meetings? And they said because we are doing kanban. And that is the funny thing about it. That is what people used to say when we were doing Agile. We don't do planning any more because we are Agile. No, that doesn't work either. So we did our first planning meeting using a kanban. And we do this also every Wednesday

morning at Kakao Coffee which is just around the corner here on Westlake, 415 Westlake. We do a thing called Lean coffee and what happens is we get everybody into a room. We set them down to a table. Everybody gets post it notes. They write down whatever it is that they want to talk about. We introduce the topics on the cards, everybody gets two votes, you dot vote on the tickets of what you would like to talk about and all of a sudden you have a prioritized list of things to talk about. This works incredibly well for planning meeting in retrospectives.

What I found in the retrospectives in a lot of teams where the teams were always looking for somebody to lead the retrospective and the moment you do something like this there is no leader other than the team because the team is creating the backlog of conversation. And when you step away from an agenda based and into an open collaborative meeting you start to see the same benefits that I was talking about from actually using kanban with the team. So when you get into a deposition issue or you get into an issue like where this explodes you need to do a couple of things.

One: make a determination about how, what the damage is. And then figure out who needs to be alerted to this situation. So if it is a start up and there is only six of you in the room you all have decision making authority and you can just go with it. If you part of NBC Universal you might actually have to give somebody a bit of an alarm. So what we do in this team, since this is a conversation we are going to talk about one thing at a time, so the team...let's say this was something that had a lot of dot votes will discuss these things in order basically until we run out of time and if we run out of time and there are things that people feel absolutely had to talk about we will continue the meeting. But we really don't go any deeper than this because the team at this point can say okay what are the issues that we are having? Well there is a technical issue we are not able to x, y, z, the a, b, c's. Okay well then what are the ramifications of that? But opening that up to a conversation and not to an apology or to a mia copa...And the other things is when you bring in management, because management should be involved in things like this, then they (or we as the case may be) then everyone gets to see the professionalism of the team and what needs to be discussed.

What I've seen is that when things like that happen and a team has already gone through and said these are acceptable and these are not acceptable stories. When that thing explodes it is usually such an exception that you are able to discuss that exception.

So now I'll actually talk about some cognitive biases. And the things that were in the rest of my event here. So I worked basically with a ton of different start ups but my first start up was called Info Move. It was over on the east side. Was anybody here in the late '90s and heard of Info Move by any chance? Awesome I'm sorry. I hope you weren't working there with me. So there was a couple of things about Info Move. Number one Info Move got a ton of money up front. So when I went to work there we had about 40 employees. And we were building an in-vehicle computing of platform in 1999. And you would have your Palm Pilot and put it in a cradle and you could give it these voice commands and it would do all these great things for you. The goal of this was we could get in, hit the button and say 'wonder machine I would like you to find the nearest Barnes and Noble with a Starbucks and I would like to get there with the minimal amount of traffic. Thank you very much'. So this was 1999. And we were working on Palm Pilots. None of us were ever allowed to see the system. And so we were going to all these architecture meetings and all these other meetings and we were doing all of this work and stuff. And it is apparently being folded up into some kind of system. And we are going to launch at Christmas. It is the second week of December and we are going to have an internal unveiling. And all of us have like polished the part of our car where we were going to mount the system in and stuff, it is going to be awesome.

And we go in and I walk up to the system and I say 'Show me traffic on I-5'. And it doesn't do anything. And then I said why isn't it doing anything? Well it is keyed to his voice. So his walks up and he says 'up, up, right, select, select, select' and he hits the reset button. I about lose my mind. Not only does it not part sentences it can't even navigate a menu. And the funny thing about it is anybody with half a brain in 1999 would know that speech recognition technology did not exist at all to handle that type of stuff let alone work out of a Palm Pilot. So we were suffering from something called the Availability Cascade. And this is huge in start ups. Is you get in and you start

to discuss what your dreams are and you discuss them so much that you think that they have already happened.

And so my colleagues were being laid off after a point, like the next week because we didn't get our next funding. At that point we were 105 people. When I went there there were 40 and now we were 105. When I left there were about 60 people. I was there five weeks. We swelled up that much to please credit squeeze, and then credit squeeze said you actually don't have a product. So they weren't suffering from the Availability Cascade and laid everybody off. And it was a horrible experience.

So we are in the process of writing another book call Scrum Bon II - A follow up from our Scrum Bon book from a couple years ago. And these are companies that have kind of... The book about Scrum Bon was all about continuous improvement, how to migrate your processes from something that isn't working for you to something that is working for you. And one of our teams I can't name yet, because they told me I can't yet, so I call them x77d, they also got tons of money. And they staffed way up and they got all these offshore teams in India and they blasted forward and two years later they had produced absolutely nothing. And during the last year of those two years they spent the whole time fighting about what their processes should be. We are not doing Scrum right, we are not doing this right, we are not doing that right and they are fighting back and forth. So then they went back for their next round of funding and much like Info Move the V.C. said 'dude you didn't do anything'. So they laid off everybody but 12 people and six months later they had a working system.

So there were two issues there. One is what psychologists call The Sunk Cost Fallacy. And the other one is Status Quo Bias. So both of these biases interacted to these teams in horrible ways. They basically said we have all of this work to do and we are not getting it done the last thing we should do is let people go. The last thing that we should do is radically alter our processes. But when their system was radically altered for them all of a sudden they were able to create a system that actually worked for them.

The second things was the Sunk Cost Fallacy. They had sent everybody out for specific training and they wanted to use the processes the people had been

trained on, and they weren't saying what are our individual needs, what are our contexts, how do these things work for us. The third start up was my software development company. We got no money from anybody. We were entirely self-funded and before that we were civil engineers. So it is not like we came in with a huge war chest because believe me civil engineers don't get paid very much money. So we needed to set up systems that would allow us to do a whole lot of work with very few people. That is when we really got into the planning over plans. To use your word regigging our back log so that we knew that we were always providing the best stuff at the best time. We fell into all of these traps all the time. But we never stopped talking to each other. And we also just happened to be the nice word for it would be introspective but the other things was that we were all really bitchy. So we wouldn't allow anybody to ever kind of rest as far as process improvement went.

So my first company had no clarity. The second one had some clarity but it wasn't well distributed and then our company had fortunately for us a great deal of clarity. And one of the things that I figured out is that I think that money and clarity are like butter. So if you are making butter sauce or you are making something baked you add butter into your mix fairly slowly while you are cooking. You don't just dump it all in. And if you dump a whole bunch of that stuff in to begin with it ends up clumping up and destroying what you are doing. And that is also true for our backlogs.

So we talked about Minimal Viable Product or Minimal Marketable Features or Minimal everything else's because I think that we pretty much loaded the word minimum thingamabobs. When we are talking about that that is what we are trying to do. We are trying to introduce work into this backlog in a way that doesn't overload or over whelm our doing column. We have all had project where we have literally thousands of functional requirements or user stories or whatever. And they are all loaded up in Jira or Rally or whatever. When we take a look at that long list of stuff it is not exciting. It is demoralizing because you just see all this stuff that you have got to do. There is no context to it; it is not functionally driven in any way. What we want to do with this keep this list to a level where we have a really good idea what these are and why they are there. And if we don't we either have to get rid of them or we have to have a conversation about that.

I'm going to talk about one more cognitive bias real quickly here and that one is called The Planning Fallacy. And it is one of the most important ones especially for start ups. Every business plan is a Planning Fallacy playhouse. And the Planning Fallacy basically says that we can't estimate at all. We are really, really bad at it.

Audience question: I'm not.

Jim Benson: Except for you because you are super human and that is why you own North America. The Planning Fallacy goes on to say that we actually are worse when we are planning for other people. So if we do an estimate for other people we will always underestimate the amount of time that it takes to do something, but there is good news because we are even worse at estimating ourselves. And the worst of all possible scenarios is if you estimate for yourself with witnesses. So if you ever want your plumber to give you a really good deal don't get the estimate over the phone. Have them stand in front of you and then guarantee that estimate. And this seems to be, among all the cognitive biases, if you go onto Wikipedia there is 117 cognitive biases which is scary because it basically means we are really kind of fallible bags of walking meat.

One of the things that we would like to do with the kanban by measuring cycle time and lead time is truly understand the rate at which we are doing work and then try and remove us as much as possible from the estimation game. And I could show you all kinds of charts and graphs about that if you want but I don't think you really want to see that. Although there is one that is really funny so I'm going to go ahead and do it. So I will do it over here.

So I had this team and I was suggesting to them that they might want to look more at their cycle time than at their story points. They said you know we are almost perfect with our story points. And I thought that was amusing so I said do you have numbers around that. And they said yes we have been charting them. And they brought these charts out about how their story points related to their actual completion rates. So they have their numbers here and the look like 1 hour, 1 hour, 3 hours, 7 hours, 3 hours, 3 hours, 3 hours. And yes everybody that you ask we all know that we are not supposed to relate these

two as completion times. I love it. Everybody, even everybody that comes into contact with this will always lower it to completion times even though we all know we are not supposed to. So it would look like this, and then like this, and like this and they said if only we could get rid of these tall ones we would be perfect. And that made me laugh. These were people who worked for a financial services firm and I said if you could get rid of these tall ones you are on the wrong floor. You don't want to be in IT you want to be trading because you have now been able to predict the future. But what I see here is beauty. I see the exact same curve each time. This is the variation in software development. And if we get rid of these peaks we make \$10 an hour because software development is not predictable. So understanding the role of variation which Eric does talk about in his work, understanding that role is very important. I do disagree with the statement that you control it but understanding that it is there and then statistically understanding how that relates to our ability to complete work. This is when we can really do some serious, serious estimates. So how am I doing for time?

Audience question: You have as much time as you need.

Jim Benson: Oh that is the most dangerous thing you could every say to me.

Audience question: Could you give us an estimate?

Jim Benson: I could give you something else.

Audience question: You mentioned that you had 1,000 appointments or something like that. Do not put them all in the ready because that will confuse things. So where do you put them?

Jim Benson: With Personal Kanban in the personal productivity space, like the current I'd guess you'd say industry leading practices, GTD. GTD is really awesome for holding that huge backlog of things that you want to do someday or maybe. And we use Personal Kanban to the more of the immediate work adjudication and calming things down. So in the same way I would still use Version 1 or Jira to hold those because part of the thing is the kanban or the Personal Kanban is a map of your work. And so already I've showed you a bunch of different views that you are getting of that. You are

seeing how quickly things are done, who is doing what, what has been completed, where things are blocked and so on and so forth. So if I show you a map of Puget Sound: You see the water, you see the Olympics, you see the cascades, you see Seattle, you see all the political boundaries, you see where the highways are and you instantly know that. It seriously takes you less than five seconds to know almost everything about the region. If I gave you a long list of all of those features in Jira you would have no idea how anything related to anything else.

So what I like to do is break things into those ethics – into those minimal marketable features and then have Jira hold those. And then when you move them in you say like in the next couple of months what are the MMFs that we want to get to. And then you start decomposing as they get closer to reality, to implementation. But one of the big things about Lean is to do things at the last responsible moment and that is absolutely something to save for the last responsible moment. Hopefully what is in Jira are these big huge blocks and then when you move them out here all of a sudden they will break into a bunch of smaller blocks. Yes?

Audience question: Where do you keep that, the whole team (Inaudible 01:00:52) or is it like on the do or die spreadsheet?

Jim Benson: I prefer if the team is all in one spot I definitely prefer that you use a physical board because as messy as this is that is how messy yours is going to get. But it is going to get that in a good way because you are going to continue to personalize it and I'll talk a little bit about how that works. There are over two dozen online kanban tools now. The favorites that one of my clients uses is just Google Draw because they can also customize that because one of the things when you use a rigid tool (and there are some awesome rigid tools out there) when you use them you can't doodle on them. You can't make them silly. And fortunately sometimes we want our work to be kind of silly. So if the team is distributed and you need some sort of electronic board there are a lot of different options for that. And they can go from something as simple as Tello, which just allows you to put cards up. It doesn't allow you to do WIPliments. It can be something as complicated, Rush (spelling 01:02:15) is complicated but as full featured as Lean Kit Kanban which is pretty phenomenal piece of work.

I'm going to cover one thing really quickly and that is that we like to talk about metrics. (Inaudible 01:01:36) loves to talk about metrics so really quickly we will make this a three... I will show you my favorite metrics because it is a pretty deep psychological concept but if you try it you are going to find out it works. Let's say that you pulled this ticket and you are working away on this story or this task or this feature, whatever it happens to be. And you finish it and you want to remember for your retrospective how it went you can annotate it with something that looks something like this. Very, very, very complicated stuff here but what ends up happening is you say I was really happy when I did this task. This task completely pissed me off. This task yea whatever it was just a task. But if your board starts to fill up with these very, very quickly you know that the sentiment is going in that direction. This is a leading indicator of every problem that we will face. Team dysfunctions, technical debt, delays, whatever you might have. Since we are professionals and we became software developers for a reason generally the only reason that these things are going to show up is if there is some issue. But when we are just sitting there plugging through work we tend not to pay attention to when we are upset because we want to get on to the next task. So, yes?

Audience question: So you talked about keep things in Jira, that they become a bunch of things in the ready state but you are not ready to work on them. Does that happen in some sort of planning game or is that a big thing for work in progress for a little bit and bounce back?

Jim Benson: Absolutely it depends on the group. So there are groups that have fairly stable and successful product donors. And product donors can do initial decomposition of that if that is a good thing. Sometimes it is and sometimes it isn't. With one of my teams every Monday morning they have a 25 minute planning meeting and they just decompose all of the work for that week. And sometimes that meeting ends in ten minutes. So they have scheduled a half hour and they realized they needed a five minute hug. They and just try and do 25 minutes. But one of the things that I want to make clear is that there is no single process for developing software and that what works for your team is what works for your team because we are all made up of people along that scale. I've seen teams where people just want to go to work

and want to go home and watch TV. I've seen other teams that want to go to work and they can't wait to go to the slopes to go snowboarding and they get work done while they are snowboarding and they are still talking about what they are doing as a team. Why would you stop that?

At Modus when we had our offices on Lake Union we had this beautiful view. Every day we would break at 3:00 and we would do 3:00 Scotch. And we would break out a bunch of bottles of Scotch, we had whiteboards all over the place and we would just think our butts off for the rest of the day. Highly recommended! Now what did this mean?

Audience question: The time thing where we should probably get to the closure whenever you want.

Jim Benson: Directly goes against your 'you have as much time as you want'.

Audience question: And you said that was dangerous so...It's an improvement!

Jim Benson: Thank you. That is a good segway (Spelling 01:06:46). I can tell you it's an improvement. We will close up with that right there in the doing column. Tonianne and I wrote out book Using Google Docs and Skype. She was on one coast and I was on the other coast and we pair wrote pretty much the whole thing, watching each other type in real time in Google Docs. Over the course of that we noticed that we were formulating new ways of working together without having discussions. There was emergence systems coming out of how we were working together as a team. When you are in a band and you are playing with your band mates you will be playing and you will go through chord changes without any communication with anybody else. You just know after playing with those people for a period of time. That is how things work.

The more we subscribe myself to rigid systems the less we allow that emergence system that to coalest (spelling 01:07:55) around us and our teammates. And that is my biggest message for everybody and it is the biggest message of the book. Is there is no right way to do kanban and there

is no wrong way to use kanban. And I think that that is important because I have spent the rest of my career trying to figure out what the right way was to do things. I did want to talk a lot more about the whole pivoting and failure and tracking failure and all of that kind of stuff. But we didn't quite get there. But it is still in my head. I guess I want to talk about it afterwards. Can I do it quickly? I don't know. Can I do it quickly? Tonianne says I can do it quickly.

Audience question: Well ask how many people want to hear you do it quickly?

Jim Benson: Okay how many people want to hear me do that quickly? How many people want me to back up and not talk about any of this shit I talked about before I wanted to talk about that? What we want to do, what I love to do is to develop visual systems that allow us to see those pivots, those pivot points, those areas where we can do minor course corrections as quickly as possible and with the least amount of information necessary. So what I don't want is for people to get all caught up in metrics and go out and start measuring everything on earth and then become completely obsessed with the way that those numbers are working. Because that is not really the way software development works. I'm giving a talk in a couple of weeks and it's called We Are Not Engineers. So there is no software engineering. It is an inventive creative emotive process. And software people we have to work with communication, we have to work with mathematics, we have to work with beauty and design, we have to bring together all of these areas. And none of them are manufacturing. None of them are assembly. So when we are going down a path we have some plans, some hopes, some desire and we want to start seeing where those pivot points come up. We have to be very, very honest with ourselves that sometimes those pivots will undo a lot of work that we have done. I guess I have an example with you Toni after that as well.

Tonianne and I have written a 320 page book called Instant Karma – The Ten Principles of Social Media for Business. It's not been published. We are a publisher, we could publish it at any time, but we got almost done with that book and then all of a sudden the Personal Kanban stuff came up and we pivoted almost immediately and very radically into getting that stuff done. And we had some pretty painful conversations about what it was that we

were getting rid of. For the completely lay audiences the example that I use is to say if I put you in your car on I-90 and I said okay sit back, sit on your hands, the car is aimed at Idaho, call me when you get to Idaho and you just push the gas pedal and you shoot out to I-90 and you can't touch the steering wheel. You are probably not going to do that. Nor are you going to take me up on my offer if I say okay well if that doesn't work for you how about if you just touch the steering wheel every 20 minutes.

So we need to be constantly pivoting. I mean it's not...I've run into team where they save all of these things for the retrospective. Or they have a specific pivot meeting and they will do this once a month. And part of the reason for that is that they think that there needs to be a lot of conversation and thought behind it. But you get to the end of two or three days and everybody's tickets are coming up like this you better not be waiting until your retrospective to have a conversation because this is money. This makes you money, this loses money. Mad programmers will cost you more money than any other thing in the world. No, those are crazy programmers.

So there are two other cards over here. One is see the problems and the other one is provide slack. Getting things done quickly is not a business model. What we want to do is do thoughtful things, do the correct things and when things are complete then we are happy because they were the correct things to do. You want to talk about anti-patterns of kanban I've seen plenty of teams like not have any meeting, not talk to each other anymore because the game becomes moving the tickets as fast as they possibly can. I've seen that several times. People will break any system that you set up for them. But when you start to set up a system where they are actively seeing the problems so they would do things like the feature ticket would go over and then they would throw it away. And then they would treat the bug ticket as part of the feature so the bug was like new work. So it wasn't like anything ever came back. Incredibly twisted view of the world.

And I have talked to teams and said what is your defect rate? Tell me. And they will say we don't have a defect rate. Like awesome. So we want our teams to be introspective and in order to be introspective again we need to see our stuff. We need to be doing the amount of stuff that we can handle so we are not overloaded. When we plan on to work with teams and we will say

okay today we are going to this class thing but starting tomorrow 7:30 in the morning we will have coffee and donuts and we are going to do a Lean coffee. Everybody laughs. Like we are not coming in at 7:30 in the morning, but by the second day we will have 20-30 people every single time regardless of the company because most people are so busy they don't have time to talk about work. So how can you pivot if you don't even have time to talk about work?

So we need to actually build slack back into our systems and that is going to be independent to the teams. Like I said some teams it might be we are going to you pull three cards of work and then you pull a card of research or reading. Or you pull three cards of work and you pick a card of drinking Scotch. You pull three cards of work and you can go snowboarding. But give people; give their brains time to actually process what they have done. One of the techniques that we use is called the Pomodoro Technique. Does anybody know what that is? Just a couple.

Pomodoro is also ridiculously simple thing. You work for 25 minutes and then you don't work for five minutes. And when you are not working for five minutes, no we really mean it don't work for those five minutes. It's not like you can work for 25 minutes and then you can answer your email. So you take one thing, you focus on it for those 25 minutes and then for the five minutes you go take a walk. And part of that is we didn't evolve as a bunch, we didn't as a species evolve sitting in cubes. We weren't sitting there saying oh it's time to go hunt and gather so let's all get out of our cubes and go. It was we were basically meant for learning things while running along. So we would learn things like I am walking over here and get berries or I'm walking here and there is a saber tooth tiger and holy shit I'm running and I'm really going to remember that now because the movement of my body is imprinting on my brain in the urgency of running into the saber tooth tiger.

So how many of you guys have ever have to fill out a timesheet? And how many of you guys have gotten to the point of filling out your timesheet and said I don't remember what the hell I did this week.

Audience question: Did you ask if they kept numbers?

Jim Benson: The thing is that was 40 hours of our lives right? I mean that we are living during that time, you know, we just flushed that time away. And the reason is exactly what the Zingara Effects said we were done with that thing so we forgot because we were moving on to the next thing. We can't find pivot points if we don't remember what we just did. But I think that logic is unsalable. So when you have a system that will remind you what you did, remind you how you felt about it, remind you who you worked on it with. That is going to give you the real information that you need to see the trajectory of your project. Providing the slack time, you know Google has its 20%; lots of companies have lots of different ways of doing this. But giving people the ability to A: process what they are going and then B: be expansive and explorative I don't think there is anybody in this room that would argue with that.

But I do think that as people once we sit down and get in the zone, whatever that zone might be we don't come out for a long time. Right? I'm sure that we have all...Tonianne and I we are humans right? So when we were writing the book there were days when I would wake up at 5:00 in the morning which is 8:00 in the morning for her. We would start writing and then at about 8:00 or 9:00 p.m. I would say something like Tonianne isn't it like midnight for you? And she was like yes but I can't stop. So getting in the zone feels great. And it's what we want. But when we do that we end up burning ourselves out and what ended up happening was a lot of that zone writing that we did (oh that is another thing to talk about) we ended up if not tossing out, radically altering when it was complete.

And I can go back to this Sunk Cost Fallacy. And this was almost like every day that we got into the editing phase. We had a block of text and Tonianne was this block of text isn't working for me. And we would pound on it for hours. And then finally I would highlight it and it would delete it and everything would read better. But we couldn't get rid of it. We sunk our time into it. And because we were like in love with two sentences and a paragraph, but the paragraph didn't add to the product. So if we would have left all those paragraphs in the book would be an infinite jest. It would just be utterly gigantic. So I'll try and come up with two pithy sentences to close this.

There is an old pogo quote which is we have met the enemy and he is us. And that is absolutely true for us as people who are starting ventures. We are our own worst enemies. And we are fraught with our own biases. And our biases do crazy things for us. We can sit there and say I'm doing this because it is pragmatic but all of the pragmatic things that we are setting up are suffering from what is called experimenters' bias or confirmation bias which is we will set up experiments and they prove the things that we want them to prove. So if I want to prove that people fall over I can run out there and start pushing you guys over and say people fall over all the time. My experiment said so. So we have to recognize when we are setting up like ab tests and we think that we were getting back independent data because we are not actually clicking. That may not be valid. When we are setting up av tests and we are not actually saying is the product something that people want we might not be gathering the right information.

And Daniel Kahneman is kind of like the king of Cognitive Bias. In his book Thinking Fast and Slow which I recommend to absolutely everybody on the planet, one of the things he says in it is there is no way to inoculate yourself from these cognitive biases. So he has been studying them for 50 years and he absolutely is as subjected to them as anybody else. So what I try to do to show those pivot points to free ourselves from those biases is keep things visual and constantly be changing the methods of my visualization in case the visualization is suffering from the same biases. That is pretty long and convoluted.

Audience question: Could you tell people how you - that can follow up with the Personal Kanban and cognitive biases and kind of what is next as it is moving?

Jim Benson: Well the easiest way is that you could walk over here and buy the book which is \$5 least for me than it is from Amazon. And \$5 will pay you back for coming here tonight. How's that? It's a rebate. We wrote a mini book which is a Kindle only thing which is \$2.99 about Cognitive Bias in Business. And it's on Amazon and we also have book cards over here if you want to go check that out.

The other thing that I would recommend is again Wednesdays at 8:30 is Lean coffee. One thing I will say about Lean coffee is my friend Jeremy Lightsmith and I set it up two years ago. And the system is so self-perpetuating that we don't have to show up any more. So we have built a very sizeable community of practice around Lean software engineering in Seattle without having a board of directors, without having to have a location, you know basically just people coming to have coffee and talking about what is important to them. And those would probably be the best ways.

Oh two more, in the summer we are going to have a Lean camp which might be called something slightly different. But we had it last year at the Center for Urban Horticulture and last year it went really, really well. And what I really appreciated about last year were two things. One: gender parity. And two: Only 50% of the people were from software.

Audience question: That sucks.

Jim Benson: Oh yes and we had awesome food trucks too. And we do our best to keep it incredible reasonably priced. And then the last thing is the Lean SSC - The Lean Software and System Consortium. We are having our annual conference in Boston a very pretty place and amazing speakers. So if you just Google LLSSC it would be great if you guys could come to that. It is certainly a bigger operation than just me because I'm doing hardly anything for it except maybe talking about it. Although you can come take a cooking class from me because I'm doing Lean Cooking.

Audience question: Action kitchen?

Jim Benson: Yes, in the action kitchen, which I did not name. So that is it. We will be around wondering around here for awhile if you guys have other questions or what have you.

Audience question: Thank you.

Watch the full video at:

<http://www.isixsigma.com/tools-templates/kanban-tool/personal-kanban/>

