

Business901

Podcast Transcription

Implementing Lean Marketing Systems

Mixing Scrum and Kanban in Agile

Guest was Yuval Yeret of AgileSparks

Business901

Related Podcast:

[Mixing Scrum and Kanban in Agile](#)

[Mixing Scrum and Kanban in Agile](#)
[Copyright Business901](#)

Business901

Podcast Transcription

Implementing Lean Marketing Systems

Yuval Yeret is a senior Agile Consultant on the AgileSparks team, helping individuals and organization ease their path to Agility and Engineering excellence, focusing on Kanban, Scrum, Lean, and effective R&D in general. Yuval comes from the R&D management world. Starting 1994, Yuval held various positions in IT and R&D, leading up to VP R&D of several IT technology startups where he introduced agile methodologies, as well as served as Product Owner on various occasions. Today, Yuval focuses on designing, and implementing solutions involving Kanban, Scrum and other approaches for enterprise-scale product development organizations, working with various levels of organizations ranging from Team Members up to the Executive Team.



Yuval is a practicing CSM, CSP and CSPO as well as a graduate of David Anderson's Kanban Coaches program. Yuval holds a BA in Math and Computer Sciences from the Tel Aviv Open University.

Yuval has been asked to deliver a pre-LSSC11 webinar on the topic of Lean Product Development flow. He is going to introduce an approach to mixing Lean and Agile in order to achieve the end to end the agility. This is also the topic he will talk about in my Agile Israel 2011 session "Techniques and experiences for managing end to end Releases, Projects and Programs using Kanban and Flow." Yuval is focusing on an aspect of this approach – how to use it to address challenges in quality assurance/testing approaches when dealing with agile projects.

Joe Dager: Welcome everyone. This is Joe Dager, the host of the Business901 Podcast. With me, today is Yuval Yeret. He is a senior Agile consultant on the AgileSparks team in Tel Aviv, Israel. Yuval has been helping individuals and organizations ease their path to agility and engineering excellence focusing on

[Mixing Scrum and Kanban in Agile](#)
[Copyright Business901](#)

Implementing Lean Marketing Systems

Kanban, Scrum, Lean and R&D. Yuval, you started out R&D, did you not?

Yuval Yeret: I started actually in IT around '93 and '94 in the Israeli army actually. I have been in IT directly about eight years and then made a move to R&D, to product development. I've been in the storage networking world a couple of years and then found the light around Agile practices. In the last couple of years have been trying to show the light to other teams based on my experience.

I've been Vice President of Engineering in two companies and found myself dealing with all of the dysfunctions of classic traditional development releases that don't finish on time, hard times towards the end and lack of flexibility. I turned towards Agile as the leader of those organizations as a way to deal with the problems. I have lead two Agile transitions on my own in my organizations, one of them with Donco from Agilesparks team came in to help at some point, and then the connection was made. I fell in love with Agile even before joining the team.

Joe: You've talked a little bit before about the different ways to look at planning in an Agile environment. Can you kind of talk about that a little bit?

Yuval: Actually, one of the complaints that we hear very often when we talk with people whether it's around Scrum or Kanban, people don't really get it that there's planning in Agile. What they understand is that with Agile you're agile, you don't have to plan, and you just do whatever you want to, when actually there are very elaborate ways to do planning around Agile. One of them is if you're managing releases, we do Agile release planning, which is to look at the capacity of your teams, of your groups to look at the backlog and amount of work that you need to do on each of the features in your backlog and try to match those two using both the capacity, both the velocity of history or past versions

Implementing Lean Marketing Systems

that you ran and try to see what you can do for a certain time. Or if you want to look at it differently, you would care more about the scope than about the time you can ask, "How long will it take me to reach that point?" That's for the release aspect.

Another way to look at it, which is the service delivery or the factory kind of planning is more similar to what you do with, I would say, Theory of Constraints, Drum-Buffer-Rope (TOCDBR) is the closest thing to it and that's where Kanban comes in and then what you mainly ask is, "OK. I've got this feature, when can I deliver this feature?" You look at it less from the perspective of a release and more from a perspective of specific items that you need to drive all the way.

Then you start getting into questions like, "OK, if this goes into development right now, when is it going to be out?" For example, a month, but if it goes into the end of the queue and it's right now last in the queue, how long will it take? So it's kind of like Disneyland, I use that example quite often in workshops or discussions with teams and they grip it.

If you go into the top of the queue with the fast pass or V.I.P access or something, you don't need to wait much. You know how long the ride takes, and that's the amount of time it will take you to get through it. If you had a really large group that goes into the fast path, then you need to ask, "How many people does the ride take, per minute or per hour?" Then you can know when a group can be finished.

You can add the queue that comes up before. You can ask, "OK if you get into the queue at the end of a long line where you see the sign that says 60 minutes from here, you can know that in more or less 60 minutes you will be, you will reach the ride, or you will be up to the ride." It doesn't really; it depends on what you measure.

Implementing Lean Marketing Systems

Now the main issue that people don't understand in depth is when you do this planning, you need to understand the meaning of commitment? I see that as an area of a lot of confusion and that drives actually, drives the performance of teams lower and creates a lot of friction between management and teams and frustration around Agile. So that's an area I've been trying to help teams understand in more depth recently, and just today in the workshop I had a good discussion about it with some people.

Joe: That's interesting, because commitment does seem that when you first hear about Agile, and you look at it, it seems like there's a lack of commitment...but it's not really true, is it?

Yuval: It's not really true, but it's indeed one of the more common myths around Agile in general. And when you go into Kanban, a feeling that or people think that Kanban has even less commitment than Agile itself for Scrum as an instance of Agile, when in reality, if you look at, let's start with generic Agile for a minute and their release planning. The issue is that there is variability in the performance of your teams. The team has an average velocity or average performance, and there is a good chance that they will perform more than that average, and there's a good chance that they will perform less than that average. If you come from Lean and Six Sigma, a control chart is a good way to look at the velocity of a team in Agile.

So the question come up, what do you commit on? Do you commit on the average? Do you commit on the higher control limits, the best case? Do you commit on the lower control limit, the worst case scenario? So, typically, managers ask teams to commit to higher levels. What they say is, if the team will not commit to a high level, then they will not deliver.

If a team commits to a higher level and there is a good chance that it will be hard for them to deliver, then what you get is

Implementing Lean Marketing Systems

teams that have unrealistic commitments a good percent of the time. So what do teams do? Teams lower their commitment and lower their actual performance because they know that if they from time to time deliver something that is more than their commitment they will be expected to commit to a higher level and you have a cause and effect that drives performance lower.

What you want to create is a situation where teams commit to something that something is for sure something that they will deliver in the release, in the iteration, it doesn't really matter. But you have the expectation of a team that if they always deliver on their commitments, that's OK, but that's not really good. You want them to deliver on the average and higher most of the time. But just make the separation between the commitment that the teams make and their actual performance.

Joe: Well, I think that's a lot of just management understanding numbers. The control chart, for example, is a great way look at it and what you want to do over time is raise the average.

Yuval: You want to raise the average, and you want to reduce the variability, improve the predictability of the team. So be able to make more aggressive commitments and have fewer surprises in what you're doing. That's something that I'm trying to explain to teams how to do. When you go to the Kanban world in service delivery, you have a similar discussion just with cycle times where it's maybe a more natural discussion, I would say. You have a cycle time, when you will, the SLA that you can commit on. But if you always meet the SLA then it means that it's a dangerous SLA because there are some times that you will miss it. So you expect to beat the SLA most of the time.

If you go back to Disneyland, I would expect that most of the time if you see a sign of 30 minutes from here, most of the time it will take you much less than that. The park manager or

Implementing Lean Marketing Systems

someone like that will probably be worried if you're just making the SLA that you want to aim for on most of the days.

Joe: Now, part of this, what always interests me is the estimation part of it because that is so important. We can sit there and say we're delivering an average; we're delivering plus or minus, but it's all based on what you estimated in the first place on what the average, is?

Yuval: Well, actually it depends. Ideally you have items in your Kanban board or in your Scrum screens or in your releases that are more or less the same size. Once you get into the same ballpark, then once you go to big numbers the differences are not that great. So that's the ideal. But there are many situations where you do have different sizes of things. You have those features which are; I don't know, 10 days and the features which are 100 days. So obviously, you cannot compare the cycle times or in the release plan treat those features the same way.

What you can do is do just enough estimation to be able to have the amount of accuracy that you need in your plans. If you are committing not to, not based on 100 percent of your capacity in your release, for example, then whether a feature is really 90 days or 120 days shouldn't be that much of a difference.

What we basically say with Agile is that most of the time when you plan a release, you don't really know. So you know it's a 100 kind of thing or a 10 kind of thing, it can be an eight versus a twenty, something like that. The relation between them is what is important, not the actual estimate.

If you have the featured size or the gross, the rough order of magnitude of the feature, then you can use the cycle time numbers that you have or your velocity numbers and see what comes out if you use this feature size. For example, you can say, "I do 10 engineering days per month." Then you could have a

Implementing Lean Marketing Systems

feature of 100 engineering days that you can know it will take 10 months.

Having said that, again if you are able to break those features into smaller pieces and write them one by one, you get a lot of advantages. You reduce the variability. You get a faster time-to-market on the top priority parts of this feature. You get a lot of good things by doing this. I really like what Don Reinersten, the way Don puts it in his product development flow book. Recently finished it, and it's a very good view of all of the technical principles behind all of this Agile and Kanban thinking.

Joe: You do most of your projects is in the Agile methods, has there been an increase in time, increase in quality. What's the difference between, let's say the old typical waterfall methods that people used? I mean you've been through both haven't you?

Yuval: Yeah, well, if we start from my personal example, the organization that I took to Agile. We switched from a version, which was supposed to take six months and was finished after around a painful year. We switched to a version, which was supposed to take four months and was finished after three months and three weeks. So we believe we are one week early to an OEM version, which was in very high quality. So that's one example, a small example of what it provides. It provides quality, predictability. You sleep well at nights.

Being in all of the places where we see organizations go to Agile, we see more or less the same things. We see the quality go up. We see the versions end up on time. We see a lot of the problems that were hiding around being surfaced, and the organization starts to deal with them. The other main thing that we see is that the projects become smaller. So you get faster flow in the system.

Whether you are doing Kanban or Scrum, it doesn't really matter. Once an organization understands the concept of delivering in

Implementing Lean Marketing Systems

smaller releases and delivering smaller parts, the time-to-market, in the R&D area at least, enables a faster time-to-market.

We do see a lot of big organization when, where this capability is not leveraged by the organization. So the organization is saying, "Come on guys we don't have anything to do. We have the releases that you are giving us or the features that are ready. Field is not willing to pool versions from us in this space." So this is an ongoing challenge for the enterprise software market. We obviously don't see that in the software as a service domain.

Joe: Now when you go to develop user stories, I mean who works on the user stories? I'm always intrigued by that, because a good user story really makes things work well at the very beginning; I think. Do you get a customer involved or is it still an internal operation?

Yuval: Well, most of the work that we've been doing in the last couple of years, and my recent history has been with product development companies, so the ISVs of the world, whether small or large. Basically, it's a game between the product management organization and the R&D folks. There are the cases where the product manager is doing the user stories. There are the cases where the R&D people, they have managers, team leads, sometimes the teams are doing it. But there is a very good question about what's the right way to do it.

Because a lot of the time the product managers, well, the books and the theory, if you look at the traditional or the classic Agile, says, "Yeah, the product manager should be the product owner and be in charge of priorities and break things down and everything."

A lot of the product managers, their head is somewhere else. They want to be in the strategic level, in the higher level. The lower level, the lowest level that they want to discuss is features. They don't care about user stories that are in the size of days,

Implementing Lean Marketing Systems

and those are the stories that teams need in order to drive effective flow in their experience or in their Kanban boards.

What you need is some way to bridge the gap between R&D and product. Some organizations have functional architects that do that, some of them have business analysts that do this kind of thing.

Some of them don't yet have those people that do those stories, but are starting to understand that there is a difference between owning the product at a higher level and deciding what features to do, and owning the product at the practical team level and breaking into print-level stories.

It's something that a lot of big organizations are struggling with those days.

Joe: I like to do the user stories and get the customer involved and have them describe it. The real work takes place when you take that from the strategic to that tactical side. It's that interpretation of those stories to give the team something to work with is really what makes it flow later on to me.

Yuval: I agree, and I see a lot of teams that are struggling with breaking into effective stories. We see a lot of technical stories, which are stories that are actually tasks or stories that don't really make sense from a functional point of view. On the other hand, we see cases where there is a story at the functional level but it's too big. So what we're working on these days is to collect best practices around how to slice and dice those stories, how to detect what kind of practice or tool will work best to slice a story based on how that story appears, and kind of a cheat sheet for slicing stories. We see a lot of need for that when we do product owner workshops, or just help teams break stories.

Joe: Jump over to Kanban a little bit here, which is somewhat of the rage the last year or so. What, from your vantage point, what

Implementing Lean Marketing Systems

grabbed you about Kanban? What made it interesting to you, when you saw it?

Yuval: The first time I went into Kanban has been while trying to do the Agile transition in my organization. Trying to recall what grabbed me there was its openness, the way it pragmatically looks at the situation and allows you some more flexibility to deal with complex situations like high levels of specialty, like dealing with a team that was doing field work development and had dependencies on team that was doing the UI, and you could not really go into a full Scrum feature team that does everything or you didn't really want to do all this revolution. So there were a lot of things about Kanban, which brought the principles to bear, but without some of the baggage that Scrum has.

I think I first got into it, I actually read David Anderson's first book, "Agile Management Using TOC," while trying to find what Agile method to adopt in my organization. Actually went through feature redevelopment and the way and some other alternatives, but since that time I have been tracking David and obviously when he started to talk more seriously about Kanban that's something that grabbed my attention.

And actually it's, once I joined the AgileSparks team, a couple of years ago, it was already a point where we were trying to understand what is required beyond Scrum in the clients that we're seeing and we saw a lot of blue ocean out there, of organization which didn't want to do the Scrum revolution.

Where Scrum per se couldn't really help whether it was, or it was too painful for Scrum to help them, whether it was areas with high specialization or areas where there was lot of changes or simply areas which didn't want to jump into the water head first and wanted to try it first. We saw that there was a lot of interest or openness around Kanban.

Implementing Lean Marketing Systems

A year ago, we brought David to Israel to deliver his Kanban coaching workshop for the entire AgileSparks team. So we stopped everything, got into a room and let David kick some Kanban sense into us, and we had some great discussions around it.

It wasn't trivial for him to go into a Scrum native crowd, no Kanban there, so there were some situations where we had to, we had some serious discussions there. You know the Israelis we don't hold anything back. But it was a great experience and since then we have been raising our game around Kanban.

We have been running Kanban workshops; a lot of customers are starting to do, whether it's Kanban end to end or more typically we have a lot of teams which are already doing Scrum, but are starting to adopt Kanban for an end to end view for managing their programs, the release or some of the teams which cannot really or don't really want to use Scrum whether it's their maintenance teams or IT teams or the work of the product owners so there's been lot of work around Kanban that we have been doing there.

Joe: You probably read David's book before he came over, I would assume it was out, I'm not sure, it was right around that time. What did you really learn from David's class that surprised you? I mean what did you take from it that maybe you would have never gotten on your own without attending the class?

Yuval: Well that's a good question. Actually we read the book it was in draft level just before he came by. I think that in general a book is not a replacement for a workshop. So the discussion, opening things, trying to understand in depth the Kanban change meant and what is the real difference between going into an organization and doing Kanban and not just a tool, but a way of thinking, the mindset of not changing everything up front, but

Implementing Lean Marketing Systems

driving slowly, slowly evolution style of change is not something you really just get from the book.

Personally I have been, I would admit that I've been quite familiar with how David thinks. When he came into class, and there were some others which the class was more shocking to them. But there were a lot of things around classes of service and how to deal with this whole area of commitment and the importance of slack and things like that we digested or internalized better after the workshop.

So if we look at it, the importance of what we did there for our team was not just what we did with it around Kanban. I think it changed the way we look at all of the Agile work that we're doing, so it started some changes in the concepts that we used in our work. So it was a great experience from my perspective.

Joe: Was there any resistance from the team on a certain subject of the Kanban workshop? Was there a certain thing that they just kind of shook their heads about and just, and walked out of there shaking their heads like, "This isn't going to work," or something?

Yuval: Yes, I think one of those areas; we had a great discussion about the need for not committing all of the capacity to your highest priority. So this is related actually to the commitment discussion from earlier and how to meet your SLA's. But the example that we use there was, "Look at your calendar. OK, if I look at my calendar not all of the days are the highest priority work or committed work. If I fill my calendar with full-time of meetings with clients that are really important, and I cannot move them then I don't have any space to exploit emerging opportunities or to deal with surprises, things that run longer." What I need to do is I need to squeeze into my planned schedule some slack, some time to work on blogs and new

Implementing Lean Marketing Systems

workshops and articles and things like that. This will allow me to be better, to be more flexible when things really come up.

One of the understandings that we have is we that cannot plan on 100 percent availability. We try to be at around 80 percent availability to projects. Most of the time what we find is that, yeah, even with 80 percent availability we actually have utilized more than that so it's not that we sit at home waiting for the phone to ring.

That's a different, difficult understanding for an organization to make. It's something that is not trivial for people that are working with a plain vanilla approach where you just look at the backlog; you commit 100 percent of your capacity except for what you set aside for the support, you commit 100 percent of it just based on the product owner priorities and you all of the time work on the highest priorities.

It was a hard time understanding this different mode of working. In the workshops and sessions that I have been running, yeah there is a challenge there. So you need to find examples like the fast path in Disneyland or some other examples or metaphors that people relate to that help them understand this point. The calendar one is a good one actually.

Joe: If you give them Reinertsen's book that should convince them of that theory if they would read that...he discusses that a fair amount in it.

What has AgileSparks taken from it and have they added something to the Kanban training? I mean are they doing it a little different than maybe what David presented to you?

Yuval: Well, as the one that's currently running the Kanban workshops, my main emphasis these days is on release level Kanban. We see a lot of organizations currently doing some kind Agile, most of the time it's actually Scrum. We find ourselves

Implementing Lean Marketing Systems

spending more time comparing Scrum and Kanban and understanding where they, where is the synergy between them, that's the way I put it. Most of the times I look at the cases where I've put Kanban on top of Scrum in order to manage the end to end release and how to deal with end to end bottlenecks using Kanban. So that's something that's, it's a different emphasis or it's something that David covers at some level but I think there's more emphasis in our training on those aspects.

I have been adding recently more and more coverage of demand management, the classes of service and this kind of discussion. I know that David is spending more and more time on this area as well. We both use the Kanban game from Russell Healy. It's getting rave reviews every time you run it, and you can use it for further discussions as you go through the materials afterwards.

I think that's, if we look at the emphasis of the last couple of sessions that I've been running, that's where the emphasis has been. So I would say it's from the same school of thought. There are more discussions in our case studies. A lot of the work that we've been doing of AgileSparks is with big clients doing enterprise level Kanban, or are starting to play with enterprise level Kanban.

For example, the case study that we presented in last year's LLSC conference is a good example. Scrum teams with Kanban looking at the upstream and the downstream or other organizations which adopted Kanban end to end, the uniqueness of our work.

Joe: You can use Scrum and Kanban within the same organization. Are they compatible without changing the entire thought process?

Yuval: The answer is yes. If I want to look in more depth, there are a couple of ways to look at it. The mindset change that you need to make is dependent on the level of integration that you are doing. One way to do it is take existing Scrum teams, look at

Implementing Lean Marketing Systems

them like a black box and then put a Kanban view on top of that. That's something we see emerging as a best practice, and we're using it in, I would say, most of the projects that we're currently doing.

The team doesn't need to change anything really. You use Kanban to manage the up-screen works, or the work of the product team, the work of discovering, what are the details of the features, the design, all of the things that are the flow towards the iterations and the flow after the iterations.

You could take cumulative flow diagram that adds to the release and you get a much better-big picture view of your project, of your product line, of the other one, portfolio.

It's kind of like; you can look at it like, doing the aspects of multi-project critical chain. So you stagger the features. You stagger the projects. You get this level of visibility and control. That is one way to do it.

Another way to do it is starting to use Kanban principles inside of Scrum team. So, what you do, there is you look at the Scrum team. You identify that the team has some dysfunctions or things that might benefit from doing Kanban.

For example, the classic one is that you look at the way the teams deliver during the sprint. You see that most of the stories are delivered towards the end of the sprint. That's something you can see from the burn up chart of the team, for example or just ask the teams, "When do you deliver all of your stories?" Or, even better, ask the testers on the team, "How does their stress level look throughout the sprint?" And you will see in a lot of teams that are not really Agile that it's still kind of a waterfall within the sprint.

The team starts all of the stories together, and all of the stories reach testing more or less at the same time. They just took their

Implementing Lean Marketing Systems

waterfall and inserted it into a sprint. What you can do there is you can start talking to teams about "Stop Starting, Start Finishing."

Focus on a few stories at a time, get the capabilities that allow you to swarm through a story, to have more people working on the same story and then deliver a pipeline of stories to testing, for example. Then you have a kind of a Kanban within the Scrum sprint.

This is the first stage of something called Scrum-ban, a combination of Scrum and Kanban. Then, if you want to take it further, you can stop committing at the sprint level, or stop cleaning the desk chart and finish of a sprint.

There is a lot of overhead that Scrum has for planning the sprint. A lot of time, a lot of energies, is spent just on deciding what you need to commit to. And a lot of energy inefficiencies there around what happens at the start of the sprint when you have to start clean, and at the end of the sprint when you have to start clean.

If you look at it from a manufacturing point of view, yes, it's good to clean the floor once in a while. But you will probably not be very effective if you need to clean all of the lines, for all, don't have any inventory inside the line every hour or so, or even every day, because if you have that, then you're bottleneck doesn't have anything to work on until work reaches it.

In the sprint level, the testers sometimes don't really have things to work on at the start of the sprint, and the developers are looking for things to do at the end of the sprint. So, that's some kind of inefficiency that you can live with when you start doing Agile. It's probably a lot better than what you have when you're doing Waterfall.

Implementing Lean Marketing Systems

But if your teams grow more and more mature, probably you start to be frustrated by this level and you start to look for ways that you can reduce this waste.

Kanban, and having a flow throughout the sprint, and not stopping the flow at the end of the sprint, is a way to deal with that. But that's something that's for mature Scrum teams, so it's not for the faint of heart.

Joe: In your opinion, Scrum isn't going to die as a result of Kanban.

Yuval: No. No, I think Scrum is a good way to go to Agile for teams that want a real revolution, need a structured way to throw out what they're currently doing and start fresh. Scrum has the advantage of having a prescribed way to do that. A lot of things that Kanban doesn't prescribe, Scrum does. We see a lot of situations where organizations need this change. On the other hand, there will be cases where you want to go in stealth. You don't want to say even that you're doing Kanban. You just want to provide visibility, start to use the limited working process to avoid bad multitasking, to avoid big gaps between the stages, and start to improve slowly, slowly from there.

So there are cases for doing each. I think that for the typical Agile team, starting with Scrum and having the structure of the sprint is a very good way to start if the team is willing to do that, and the organization is willing to make the changes that you need for a feature team.

I think that some of those teams will evolve to Kanban throughout the years. I think it will take some time until we see a lot of teams that do that. I think that there are ways to get to hyper-productive teams without making the move to Kanban.

If you really can break the stories into very small slices, and if you do have very high-level engineering practices and collective

Implementing Lean Marketing Systems

ownership, maybe you don't need Kanban to deal with your wastes. I think it will be a question of the organization prioritizing what it is willing to spend effort on. And sometimes they will say, "Kanban provides a shortcut to higher productivity without doing the things that Scrum requires." But I think the jury is still out on how the real hyper-productive teams will get there.

I don't think that's an interesting area at the moment, by the way. I think much more interesting area is the organizations that are currently doing traditional development and want to go to be more Agile, but Scrum currently is too much for them. I think that's the main green field for Kanban at the moment. And it's an opportunity; we're currently not leveraging as a community or industry, but it's certainly something that we need to focus on.

Joe: For you, Yuval, and for the AgileSparks team, what's in the future? What's coming up for you?

Yuval: Well, we have conference season coming up actually. I am going to be at the Lean Conference in the West Coast on May. Actually, we'll be talking about some Kanban aspects for Agile testing actually. It's another area where we've been working on, taking the Agile testing practice forward in Israel and in our region. There are nominations where I've been chosen for some reason by the Committee as a nominee for that. It's quite humbling to get this nomination. So looking forward to seeing what develops there.

Then there is, go to conference in Copenhagen, the week after. And also we have for the Israeli crowd, Agile Israel 2011, which is in April, which is turning up to be kind of an international event with, I think we have five or six international guests that are arriving.

So one of the next steps that in general we're trying to make as a team and I'm trying to make as part of the team is provide the services, leverage the services and the learning that we've been

Implementing Lean Marketing Systems

doing here in Israel to expand more and more into Europe, Eastern Europe and the area. So whether it's Agile testing, whether it's Kanban, Agile Project Management or Scrum, it's something that we're looking forward to helping things in the area some more. I think one of our main agendas for the year.

I think that things will; we're trying to help teams understand that engineering practices, the things that the teams need to do on the technical level are things that you shouldn't disregard. So at some level you do need to spend time on continuous integration and the testing and development and the acceptance test during development. It sometimes not as popular, but I think it's very important and we're trying to understand how to help teams reach that point more effectively.

Joe: Well, I'd like to thank you very much. Business901 podcast, of course, is available on the Business901 website and the Business901 iTunes store. So thank you again, very much.

Yuval: Thank you.

Business901

Podcast Transcription

Implementing Lean Marketing Systems



Joseph T. Dager

Lean Six Sigma Black Belt

Ph: 260-438-0411 Fax: 260-818-2022

Email: jtdager@business901.com

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

Twitter: [@business901](https://twitter.com/business901)

I

***What others say:** In the past 20 years, Joe and I have collaborated on many difficult issues. Joe's ability to combine his expertise with "out of the box" thinking is unsurpassed. He has always delivered quickly, cost effectively and with ingenuity. A brilliant mind that is always a pleasure to work with." James R.*

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

An example of how we may work: Business901 could start with a consulting style utilizing an individual from your organization or virtual assistance that is well-versed in our principles. We have **capabilities to plug virtually any marketing function** into your process immediately. As proficiencies develop, Business901 moves into a coach's role supporting the process as needed. The goal of implementing a system is that the processes will become a habit and not an event.

[Business901](#)

[Podcast Opportunity](#)

[Expert Status](#)