Lean Kanban Lessons from

a Software Developer

Guest was Chris Hefley is President
and Co-Founder Bandit Software

Related Podcast:

[Lean Kanban Lessons from a Software Developer](#)

Lean Kanban Lessons from a Software Developer

Business901

Podcast Transcription

Implementing Lean Marketing Systems

This is a transcription of the Business901 Podcast with Chris Hefley, President of Bandit Software. Chris offered a perspective from not only a producer of a product for Kanban but as a software developer himself. A fresh perspective of not only Kanban but Lean in Software Development, this is not a consultant or an author of a how to book, but someone that is in the trenches living it day and day out. Chris's primary area of expertise is in helping development teams implement processes that enable a lean, continuous flow of work - focusing on the technical practices that enable teams to deliver quickly, avoiding bottlenecks and blockers.

Bandit Software, LLC.

Bandit Software was founded in May 2009 by Chris Hefley, Stephen Franklin, and Daniel Norton. We all worked together once upon a time at the Nashville-based healthcare giant, HCA, Inc. We'd remained friend for years, long after leaving to pursue other opportunities, and one day we just decided it was time. Time to strike out on our own and "*own the company we want to work for*." So, one night back in May we gathered in Stephen's basement, and Bandit Software was born. Six months and many sleepless nights later, we are on the verge of launching our first product, LeanKit Kanban.

Chris Hefley can be found on Twitter at @indomitablehef

Lean Kanban Lessons from a Software Developer

Joe Dager: Participating in the program today is Chris Hefley, the president and cofounder of Bandit Software, a maker of electronic Kanban tools for Lean software and system development teams called Lean Kit Kanban. Chris has been a software developer, architect, and coach and has experience in manufacturing, consulting, and healthcare. Chris, could you just tell me where you came up then with the name Bandit and why you named the company that?

Chris Hefley: That's a good question. We were struggling to find a name, and Bandit is actually the name of the one of the cofounder's dogs. He's a bloodhound dog called Bandit, so that's what we came up with.

Joe: Now could you tell me how you first got exposed to Lean and a little background about yourself?

Chris: Absolutely. When I first got out of college, or actually even in college, I was a co-op student at a TRW plant and my first exposure to Lean was actually in manufacturing. They had set up a line that used work cells and were trying to implement some Lean methodologies there in the plant. My first several years out of college, I worked in various manufacturing plants, in DuPont and Frigidaire and places like that, where there were Lean and Six Sigma efforts going on. A lot of that was peripheral to me.

About three or four years ago when software developers started talking about Lean as something that would apply to the software development process, the Mary and Tom Poppendieck book about Lean software development came out. I really started getting interested in it then. I was very much into the Agile software development movement

before that and started really seeing a lot of value in applying Lean principles to software development around that time.

Joe: The interesting thing to me is Kanban has really come up very quickly here. There is more of an association with Lean than maybe what Agile is, is that correct or not? Or how has that worked?

Chris: Sure, absolutely. I think Kanban as we sort of define it in software development is a little different than what it has traditionally. It's taken on more meaning than it traditionally had in Lean manufacturing. It's a more complex system with lots of rules that we're sort of defining for ourselves using the basic concept of a Kanban as it came out of manufacturing. A signaling system, a visual signal, or when work is ready to be done or when work is available to be done.

So I think Kanban as a tool has really taken off, maybe even more than Lean in general in software, and I think there's some dispute about how good that is. But I think it's a good gateway into Lean principles for a lot of software development teams. Kanban gives you a chance to start where you're at and begin to make evolutionary changes as opposed to revolutionary changes to get to improvements in your software process.

Along the way, as you start seeing the result of limiting your working process and visualizing the working process, you start being introduced to the Lean principles simply just by going through the process.

Joe: How is Kanban distinctly different than other Agile methods?

Chris: I think Agile methods are almost entirely based on iterations. There are lots of

Lean Kanban Lessons from a Software Developer

technical practices that go with it, and there are several different flavors of Agile. But we generally basically see one month iterations or two week iterations or one week iterations. It's still a schedule based kind of thing, so if you batch up a batch of work and you try and complete that by the end of the week or the end of the month. Kanban is different in that we try to use it to get to a continuous flow of work so that you eventually can get rid of iterations. Now you can do it in combination with iterations as well. We'll talk about that later, but the key thing with Kanban is to learn to limit the working process, to level that out with the actual capacity of the system. So the less work you have going on at one time the faster all of it can flow through the system.

So it shies away from the iterations, and it really focuses on limiting the working process and living within the rules that that kind of restraint puts on you. Like you say, it's an evolutionary thing, too, whereas when you come in with Agile there's some upheaval that has to happen in order to get everybody to agree to do things a certain way.

With Kanban, you can really start by and you should start really by visualizing the work the way you already do work, visualizing the way things are now and putting that up on the board. Once you do that you start to notice things about your process. You start to notice things stacking up and bottlenecks appearing and baulkers.

When you start noticing that, it's easy to take that as evidence and say, "Oh, we should change this." And you can start making those continuous improvement kinds of changes, little small incremental changes along the way to get to a better and better process and again starting right from where you are now without having to make a revolutionary upheaval.

Lean Kanban Lessons from a Software Developer

Joe: Now when you talk about that it's different than the iterations, but doesn't the iterations allow you to do better planning or be able to do better estimation?

Chris: Sure. I'm not necessarily here to say anything bad about iterations. I think one of the things that Kanban does, there's a guy named Henrik Kniberg who wrote a book called "Kanban and Scrum: Making the Most of Both." He talks about how basically that you can do a lot of the same activities that you do in iterations in a Kanban system. It's just that they're decoupled from the iteration itself. So with Scrum, for example, you end up doing your planning and estimation, your development, your testing, your deployment, and your retrospective all on the same cycle. So every month you do all of those things. With Kanban, you can still do all of those things. They just don't all have to be coupled to the iterations.

You can do your planning every week. You can do your estimation every other week. You can do your testing every quarter, and you can do your retrospectives on demand. You can still do a lot of those things. They just don't have to be as rigidly structured with the iteration.

Joe: I always think of software development as a fairly sophisticated field, and a lot of technical things go into it. Then all of a sudden I see Kanban taking hold in it where Kanban seems like the simplest of all practices. It seems counterintuitive to me a little bit there. Is that the secret to it? Is that why people are using it?

Chris: Well, yeah. I think there's a lot to that. I think that as sophisticated as our jobs maybe are, we often struggle I think as software developers to be good at process. I think there's about a 30 or 40 year struggle that we can look back on, and nobody that I know of

Lean Kanban Lessons from a Software Developer

is really quite satisfied with the way things are in terms of process in software development. There are the people that have their pet methodology that they push, but I think everybody realizes that we have a long way to go to be able to say that we do this effectively across the board

A lot of time's software development teams don't have quite the autonomy that they wish they had to define their own processes. I think that's one of the reasons that Lean and Kanban are taking hold in places where it was difficult for Agile to take hold.

Because Agile, in my estimation of it is basically the developer is sort of rising up and saying, "We really need a better process to do our work better and we're going to tell you the business what that process should be." Where when Lean comes in, there's a history there. There's a legacy of Lean. Business leaders recognize the validity of it from its history.

When we start saying, "Instead of doing Agile things, we're going to gradually start implementing Lean principles in our organization to help us deliver software better," that's more easily accepted at all levels of the business than some of the more revolutionary Agile practices have been, I think. And I think there's a lot of hope in that for us to apply that and get better at it.

Joe: I have this feeling the reason I always thought Lean has taken hold so much better than, let's say, like Six Sigma and maybe like Agile as you described there a little bit is that the stories are better. The people understand the stories when you're sitting here talking about standard work or you're talking about value stream mapping and you're doing things like that, people understand the stories. They relate to it better, and that's what's made it

Lean Kanban Lessons from a Software Developer

easier.

Chris: I think there's a lot to that, absolutely. And all the stories from history and all the idioms from Japanese that have built into it, I think there's a lot of knowledge bound up in all that.

Joe: Now why an electronic tool for Kanban? I mean Kanban; I just put it up on the board, right? I stick the Post-it notes and I might use different colored Post-it notes up there to signify different things. Why an electronic tool?

Chris: Well, there are actually several reasons for that. When I first started doing Kanban for my developer teams, we were doing it on the board with index cards and tape. But we also had developer in Nashville, Tennessee, in Italy, in the Bahamas, and in the Ukraine. So every morning on our call, I would get up and move their cards around, and they would tell me what it was. They never got to see the board except for some snapshot pictures that I sent them. So there's the working with distributed teams.

One of our other cofounders works in a consulting firm here in town and moves around to client sites a lot. They ended up taking the science project boards, the folding cardboard things, and that was where they put their Kanbans. They would tuck that under their arm, load it in the back of the car, and take it with them to the client site and back and forth to the office.

So the distributed team aspect of it is great, and the communication you get from being able to interact with people on different time zones and continents, even if it's just in different rooms is good. The other thing is, as developers, I like low fidelity tools. Most

developers do. Scrum walls and card walls and just stories and those kinds of things.

Developers like that, but I have yet to meet a project manager that does. A project manager or functional manager usually sees that as lost information that I can't write reports against. So with an electronic Kanban tool, the reports about how the work is going on is basically captured automatically as you move the cards.

So you can get a lot of good information that's actually more truthful than the information you get from a project management system where developers are coming along and giving you estimates and saying, "I'm 75 percent done, or I'm 95 percent done, or I'm 80 percent done on this particular task." Because the statistics about what's being done and how fast it's being done are actually captured realistically.

I think we talk about project management systems being repositories for lies because everybody [laughs] knows that last 10 percent of about 90 percent done is like the last minute of an NBA basketball game. It can last a lot longer than you expect. So doing away with some of the guessing involved and just capturing the work as it actually moves, there's a lot of valuable statistics you can gain from that.

That's one of the great things, too, about Kanban for software development. I think we've pulled in a lot of interesting new statistics and analytics tools into that methodology. Some of it has probably come from Six Sigma. I'm not that familiar with Six Sigma. But I think some of them do, like the statistical process control charts and things like that.

But some really interesting metrics that are easily understandable by everybody and really point out some really interesting things about how work is being done--cumulative flow,

looking at cycle time and lead time, overall work in process, and how that changes over time. So I think the statistics captured and the distributed team aspects are probably the two biggest things.

Joe: Well, I think you bring up an interesting thing because there is still a project manager that has to fulfill needs of other people outside the team.

Chris: Absolutely.

Joe: And that seems to be a great reason for it. But what do you lose by putting it on a screen? What do you lose by putting it into software versus on the board?

Chris: I think the constant challenge is to not let the software get in the way. With a board, anytime you want to add another lane or draw a diagonal line across somewhere or do any kind of change to your Kanban board that will help you get things done better, it's as simple as walking up to the board and drawing that line. With a Kanban tool, a web based tool, the constant challenge for us is to keep it flexible enough that you could still do everything or almost everything you would want to do on a board. We think we're doing a pretty good job of that.

The only one as far as I know among our competitors that can do things like horizontal lanes and splitting lanes vertically and horizontally several levels deep so you can break things down and do little sub-compartments in lanes, rolling the work in process when it's up that chain, those kinds of things.

So that's something that you don't want to let the software get in the way of making your continuous improvement efforts on your Kanban board.

Lean Kanban Lessons from a Software Developer

Joe: Corey Ladas said in his ScrumBan book, he's never drawn the same Kanban system for anyone. He doesn't repeat it. They're all somewhat a little different. And that customizable aspect of it is you hate to hear the words. We've all heard these words from software people when we call in with a problem or something like that, "Well, the software doesn't allow me to do that."

Chris: Absolutely.

Joe: And at any point all software has limitations, but it sounds like what you're doing is that you can split it. You can change a lot of things with your tool.

Chris: Our product roadmap is basically all about adding more and more flexibility. There are tools out there that are more opinionated than ours, that give -- that put more constraints on you. And their simplicity, part of their goal is simplicity there, and I think that that's a great goal for some segment of the market. We're going after the people that are really wanting to do Kanban and have the freedom to do all the things they can imagine doing. We're working now on customizing the different card types, and adding classes of service, and allowing you to switch.

Right now, the card color is what indicates which type of task it is. Is it a "feature of" or "task" or "improvement?" We're working on letting you switch that to a different field so you could say card color could then communicate class or service, and a little icon could indicate the card type.

Ongoing, that's what our product road map looks like is just continuing to add more and more layers of flexibility.

Lean Kanban Lessons from a Software Developer

Joe: Now how would you recommend someone getting started with Kanban doing it with software or do you recommend putting it up with post it notes and doing your own out there?

Chris: Well, if you've got everybody in the same room, put it up in post it notes first. When you get to the point where you feel like you been at it for a week or two and you want to start collecting better statistics, then a Kanban tool is probably a pretty good way to go. If you've got people in different places, then you're going to want to move to a Kanban tool, probably faster. But when you're actually sketching it out, probably, at least draw it out on the board and see how that feels for a few days, or a week or two, or a cycle or two before you get too deep into choosing your tool, probably.

The other thing, too, though is ours, with a lot of the tools, and ours included, are actually really simple to start, as well. I tell people to start with three columns; To do, Doing, and Done. And start splitting those out from there. Start figuring out how work enters the system. At what point is customer value actually realized, and so you basically start building your value stream map and reflecting at least a section of that in your Kanban.

I'll also tell people it's good to try to do the mapping stream exercise in great detail on a board somewhere, and then decide what part of that actually goes on your Kanban board. I think the tools are getting better in terms of the flexibility, so you could very well start with the software. I just think that that initial brain storming session where you're just trying to figure out "what is it we actually do now" is still a good thing to do on a board.

Joe: Do you feel that Kanban is in its infancy, do you think there's a long way for it to go?

Lean Kanban Lessons from a Software Developer

Chris: I think there's a lot of adoption that's yet to happen. I think that we're about to see a pretty significant surge in adoption of Lean and Kanban in software development. I think it addresses a need that has been nagging for a while, and it addresses some of the need that Agile tried really hard to address, but was not as successful in some areas as it could have been. I think that a lot of the promises of Agile about being able to react quickly and to turn things around fast and to be able to actually make sure that we deliver customer value because we're doing it in short cycles and getting the valuations. I think a lot of the promises of Agile can be better realized in a system that moves more towards Lean. I don't think we have to throw the baby out with the bathwater, but I think like I said with Kanban, it can be an evolutionary thing.

If you're already doing Agile, you can start adding elements of Kanban and learning more about Lean and beginning to apply some of those principles, basically from where you are. And I think that you mentioned this before a lot of books that basically takes you from that progression where you start with Scrum and you gradually introduces element of Kanban until what you have is definitely a hybrid, and it's not much like Scrum anymore, it's more of a Kanban system.

I think Dave Anderson's book on Kanban, Dave Anderson, as far as I know, introduced the software world to Kanban as a tool. He's definitely been a champion of the community for a long time and his book is about to be published and I think a lot has gone into finding what the steps are, what the principles are.

I think it's beginning to come into some level of maturity and definition. I think there's definitely going to be some changes along the way, but I think we're getting to the point

Lean Kanban Lessons from a Software Developer

where it's a viable, robust body of knowledge that goes along with it. The community has been really active in creating for the last three or four years.

Joe: I see the different Post-it notes going across the board, can you describe that note to me. Is there technical data within that note? Does that note change as it goes through the system?

Chris: Well, on a physical board, it can be as much or as little as you want to put on there. So a lot of times it's a user story card in Agile teams, so it will have a user story like; As a "whatever role I am, " I want to be able to "whatever function I want to perform, " so that I can "achieve some business value, " form. Maybe it'll have "acceptance test" written on it or those kinds of things. An electronic tool, I'll just speak to ours specifically, it has fields for describing it, and so you can put that user story stuff. It has some things like "status" and "priority" and "class of service" and "card type."

We also have a discussion form on each card. So each card, as it moves around the board, as developers need to talk about it, it actually starts to discuss the card right on the card and have a threaded conversation about that. The history of the card is kept there. As the card moves, every time something happens, we capture an event and that can be seen on the card and later in the aggregated statistics.

There's some effort to do some leveling I forget the Japanese; it's one of the Japanese words that they use for that. But basically things work really well if you can make some effort to make all your cards of similar size, similar effort, and it can get easier to decipher statistics if you don't have things that are vastly different in size.

<div align="center">Lean Kanban Lessons from a Software Developer</div>

One thing that takes 700 hours is a card that moves really slowly across the board and one thing is a bug fix that takes two hours is a lot faster. Some effort to make the cards a similar size is done in a Kanban system, but you can only do so much of that, too. I think you have to mitigate that with some kind of size information on the card.

Joe: So you actually very close to taking a story and developing certain story points into it in your backlog there, let's say.

Chris: Sure, and we actually take out and break a story, too. We'll have a horizontal swim lane that represents a feature, so we'll have the first block in that will be a green feature card and then within that it's almost like a board within a board. On our tool, you have a "To Do, Doing, and Done," for that feature. We break that card into... so maybe a feature that we break out into user stories, or tasks. When you scroll the "To Do" column that's next to that feature, and gradually move those things across the lanes in the middle of the board until we're all over to "Done." Then we move the main feature card over that and to the next stage of testing and then to deployment.

So the cards can be all different kinds of, you can break out sub-cards basically for a feature that needs a task breakout. You can track those on the board as well.

Joe: We always talk how Kanban sounds so simple. There are three columns To do, Doing, and Done, but there are really sophisticated processes out there. How large of a Kanban board have you ever seen? As far as, let's say maybe different swim lanes and different columns across?

Chris: Obviously, the physical ones, I've seen ones that fill an entire wall, floor to ceiling,

Lean Kanban Lessons from a Software Developer

and have lots of different things going on. I think there's some value in having a board for each team. Not necessarily per project, or per whatever, because what you're leveling things against, what you're limiting capacity or what you're limiting your working process against, you're limiting capacity of the team.

If a team is working on multiple projects, they need to be represented somehow on the board, and horizontal lanes are a good way to do that.

Our board, we thought, the board that we use internally for development at Bandit Software, we thought it was pretty complicated and sophisticated and we let some folks at the development team for Canonical, the Linux distribution web tracking system, started putting some boards up, and they really got ambitious with their boards.

It can get very complex and lots of layers and levels, and we have a zoom in and out feature on our tool, so you can actually zoom out and see quite a lot of work on one screen. It gets really tiny, but you can hover over things and see some descriptions there.

I think there just needs to be some effort to chop things up into manageable sized chunks, but you can definitely model a very sophisticated and complicated process on a Kanban board, and people are definitely, definitely doing that.

Joe: Do you think Kanban is being used by larger organizations, more so than say smaller ones?

Chris: I think there's some of that's happened already in larger organizations, like the BBC is a big Kanban shop, it's permeated throughout lots of areas of that organization and there are other examples of that. Some of our customers are large. We're not filling the

Lean Kanban Lessons from a Software Developer

entire organization yet. Obviously, one at several large companies but I think that... like I said with the Agile thing, the Lean adoption in places that would have been resistant to Agile, like large companies, will actually help Kanban adoption in that area.

I think personal Kanban is definitely a growing thing, too. It's very useful for limiting your own working process and for somebody who lives and breathes Kanban night and day like I do; I'm surprisingly not good at personal Kanban.

Chris: I'm not very good at limiting my own working process and keeping things leveled out there, but I'm trying. I do have a personal Kanban board that I use and try to use it every day. But yeah, I think large companies are going to see a lot of value from inputting Kanban and being able to see. Especially because it will focus the Lean principle about "focus on customer value" and only define value in terms of how the customer defines value.

That kind of insight in having people look closely at how our process works and what steps is value adding and what steps we should get rid of, and how in the end we create value for the customer. I think large companies will definitely gravitate to that.

Joe: One of the things that you mention about Kanban is that it helps with the continuous flow of work or avoiding bottlenecks and blockers. Why does it do that?

Chris: Well, I don't know that in itself it does that, it points out those things, right? The first principle in Kanban is to make the work visible. When you throw the work up on the wall and you say "Here's our development and testing and deployment. Here's a stack of 35 cards that's waiting for testing." It's obvious that there's a bottleneck at testing. Where

the testing team doesn't have the capacity or they're getting work in too big of a batch to be able to clear it quickly, whatever that situation is. The Kanban board and the visual nature of it help you see those problems quickly. It's really easy to get organizational support to do something about it when somebody can come and look at the problem and see it visually on the board, as opposed to hearing a developer complain about it, or a project manager complain about it.

I think the principles then also involved in -- from there, the principles that are built into the Kanban system and the Lean principles will help you to address those bottlenecks as they appear.

Just the same way they do in the manufacturing world where the Eli Goldratt book talks about how every system has bottlenecks and this is not good or bad, it just is. As you clear one bottleneck, others will pop up, and you just have to deal with those as they come up, but the Kanban system makes it really obvious where those things are.

The continuous flow of work, and limiting the working process is, again, is the second cornerstone to the Kanban system, at least the way that we're defining it in the software world is limiting the working process and not letting things backup. Placing limits on how many items can go into this column or limits of how many items of a certain type can go into this column can help you stop those bottlenecks and backlogs from happening mid process and keep the overall cycle time lower.

There's some information there about queuing theory and the Little's Law, where they talk about the average cycle time of items going through their system is directly proportional to the overall number of items in that system. So the lower you can keep that number over

Lean Kanban Lessons from a Software Developer

time, the faster things can move through it.

Kanban and implementing the principles that go along with managing this system and continuously improving it can help you avoid bottlenecks and block through and managing a continuous flow of work.

And a pull system, we talk about pull a lot and how instead of batching up a set of work and pushing it into an iteration schedule, we actually try and pull the next highest valued customer valued item from the queue as the system has the capacity rather than pushing things in. Doing that pull, again, using working process limits, to manage when capacity is available, can make things go faster there, too. We talk about working process limit as the lever that we apply to the Kanban system to get it to behave the way that we want it to.

So there are almost all the things that we end up wanting to do to improve the flow and reduce bottlenecks and that kind of thing, involve, at some point the working process limiting on the board.

Joe: We talk about Kanban and dealing with working process, which I'm a big believer of that, even in sales and marketing. I believe that people that many people keep working to fill up their funnel but never do a good job of managing their prospects that are in their funnel. To do that, and to manage a working process, from the old troubleshooter side of me where I would take service calls, I always would ask someone "What did you do right before this problem took place? What did you do right before this happened?" The problem typically was always the step before.

If Kanban really concentrates on working process, does it help to manage the queue?

Lean Kanban Lessons from a Software Developer

Because isn't that where it's coming from?

Chris: That's absolutely a big issue, I think that one of the principles that we talked about is to balance demand against throughput. Once you've clearly stated and understand what the actual capacity of your system is and you've placed intelligent limits along the way, to keep you from pushing more into that system that has to pass through whole. You can actually as business leaders and as the people that are building that queue and deciding what comes next you can make much more intelligent informed decisions about the priority of things. So even if you say you're going to push more work onto the development team, they're not going to be able to get it all done. Things are going to fall through the cracks, and the overall cycle time is going to suffer.

But if you've limited it intelligently, you can actually look and go, "I know the next time we have capacity, that this is the highest priority item." And you don't have to guess and reprioritize over and over again as things go through the system.

The limits, like you said, are the step before. The limits on the board can limit how much work you push into the system to begin with. And when you've limited that and it's clear what that limit is, you can make really clear priority decisions about what comes next.

Joe: I there something that you would like to add that maybe I didn't ask that's about Kanban that you feel that's important?

Chris: Just one thing. One of the things that we've started talking a lot about on our team and to some of our customers is the importance of some technical practices. Lean and Kanban in the software world don't come with technical practices like some of the Agile

methods do. But we feel like there are some technical practices that development teams can follow that will aid them in being able to produce that continuous flow of work and avoid baulkers and avoid bottlenecks. There's not a lot of discussion going on about that.

At least as the software developers talk amongst themselves about how to improve the system, there's some things we can do like branch per feature, source control strategies, release per feature, and automated testing at the regression testing level and acceptance testing level.

And getting to a place of where you can do things like continuous deployment where you actually automate deployments to production on a regular basis based on once you're comfortable with your automated testing suite. So doing those kinds of things as a development team can really help your ability to manage this kind of system.

There's a big conference coming up here in a couple weeks. I'm not sure when this will actually air, but the Lean Software and Systems Conference in Atlanta on April the 21st that we're going to be at, have a booth, and be showing off some new features of the software to you.
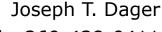
Joe: I would like to thank you very much. I appreciate the overview, Chris. It was very helpful to me in understanding that. How can someone get a hold of you?

Chris: Chris.Hefley@BanditSoftware.com. Or the URL for our software is LeanKitKanban.com, and you can send us an email from there and take a demo. There's a free version of the software. You can check it out there and a free trial for a larger team.

Lean Kanban Lessons from a Software Developer

Business901　　　*Podcast Transcription*

Implementing Lean Marketing Systems

Joseph T. Dager

Ph: 260-438-0411

Email: jtdager@business901.com

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

Twitter: @business901

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. Part of your marketing strategy is to learn and implement these tools.

Lean Kanban Lessons from a Software Developer

Business901　　　　　　Podcast Opportunity　　　　　　Expert Status