The Theory of a Promise Guest was Mark Burgess

Designing through Promises and *Designing through Promises, Part 2*

The Theory of a Promise

Note: This is a transcription of an interview. It has not gone through a professional editing process and may contain grammatical errors or incorrect formatting.

Transcription of Interview

Joe Dager: *Welcome everyone. This is Joe Dager, the host of the Business901 podcast. With me today is Mark Burgess. Mark is a theoretician and practitioner in the area of information systems, whose work is focused largely on distributing information infrastructure. He is known particularly for his work on Configuration Management and the Promise Theory. He was the principal founder of CFEngine and is the professor of Network and System Administration at Oslo University College. His latest work is captured in a new book from O'Reilly, Thinking in Promises. Mark, I'd like to welcome you, and I think the Promise Theory for many of my listeners needs a slight introduction.*

Mark Burgess: Thanks, good to be with you all. I think Promise Theory is something that I started working on after I reached the point in my main line of work as supposed you could say, which is Configuration Management, desired state computing. It's what I like to call it. I reached the point where it became important to take maybe a larger view of the system. Configuration starts out, you know, just single machines and the state of that single machine but as you get into systems you want to look at the state of the whole system as being many cooperating machines and people across in between as well. People are quite important. We often forget them when we're talking about computers, but they're pretty important piece of the puzzle. Like to think of it as a change from the way we want

to think about control and traditionally we sort of had this law giving idea, this sort of kept and be caught, make it so or thou shalt this and that. We try to instruct the computer but in Promise Theory we turn these things upside down and say, try to turn it into a model of agency where we say, 'Each component agent in the systems will say, "Hey, look what I can do for you." and we start with that.' We try to build a cooperative system by saying what each of the compartments can do.

We could get more formal and say it's about trying to unify intent and behavior. What you intend to come out is a system with what actually comes out of the system. We don't have to go into too much detail but, you know, science usually tries to do away with the intention part and complicated semantics because it's trying to make everything universal and impartial. It's describing behaviors without much in the way of interpretation. But, that kind of breaks down as you get into more complex systems, let's say more information rich situations where I think context is an important key.

From an I.T. perspective it looks, you know, a lot like services, service-oriented architecture and today, of course, micro-services are a thing, except that it's nothing at all to do with a particular technology or implementation. It's a kind of a science model approach to try and to describe the behavior of systems from an agent point of view. Look what I can do.

Joe: *It sounds just about like machines could be promising each other. Is there any truth to that?*

Mark: Absolutely and what I really wanted to do is to make a model where it didn't really matter whether one of these agents was a human or a computer or a machine or a robot or a carpet or a piece of furniture. All of these things, basically, play a role and they exhibit

some form of properties, or agents see the ability to decide things in some cases, not in all cases but in some cases it can make decisions and even though they're not the source of that thinking or the source of that agency necessarily they're proxies for the human intent that goes behind it and machines are basically proxies for our human intent in I.T. And so, the model I think tries to sew it together, the human and the machine aspect in one.

Joe: *It's somewhat mathematically grounded? I reviewed a previous book of yours, and it was like, I thought, "Oh gee, I might need t brush up on my math skills a little bit." Is that true and is your new book, Thinking in Promises, have a lot of Math in it or not?"*

Mark: It's a vicious rumor. They always say it. I bet that was Jeff Sussna that said that to you. But no, I tell you what, well, it's a reality I suppose, some people have thought that the first book I wrote on Promise Theory was quite mathematical. My partner in crime in that book, Jan Bergstra, who's been one of my close collaborators over the years, he laughs if anybody thinks that it's mathematical because he can't see any math in this at all. It's got a couple of symbols in it, you know, but no mathematics, and I already think there's a lot of mathematics in Promise Theory, a little bit of set theory to formalize things.

In the first book you could say there were a bunch of symbols in it because we were trying to find a way to formalize things without all of these clumsy words. But, for the last book I was asked by O'Reilly actually on pain of death or decompilation or what everyone says in I.T.to avoid using any kinds of symbols which is quite hard for me because I'm a symbol kind of guy. I tried to turn up everything into cartoons instead, so it has cartoons, I think, in the latest book to try to make up for the lack of mathematics.

Joe: *I thought you did a nice job with that. Where did Promise Theory originate? Is this old*

thinking that fits new theory thinking and expanded or where did it come from and why now, I guess?

Mark: Yeah, it's a good question. I suppose you could say, not so much new, it's a synthesis of all kinds of different ideas and I don't think there's any one part of it that was new when I did it but perhaps that part that I'm associated with emphasizing from my own work is this notion that autonomous agents or these local agents can decide things for themselves and I suppose the breakthrough that I experienced was this idea that by making a description in which these autonomous pieces decide things for themselves, what can I do for you. It's sort of what it does for I.T., what Autonomic Theory did for chemistry. It says these are the building blocks from which you can compose a cooperative system, and you can start to combine them, you know, you study the chemistry of this cooperation between these elements, kind of again like this idea of micro services that you start with very simple building blocks which promise relatively simple things. And then, you try to build up a chemistry of cooperation between them.

Historically, I mean it came out of where I was in my thinking in Configuration Management because the system that I originally built, CFEngine, the configuration engine was built as a distributed system in this way. Each machine was an individual autonomous thing without a centrally, without a central controller instructing them. I had sort of made this software intuitively and suddenly it was running all over the world and almost sort of beyond my control. I wanted to understand it better and it also gotten a bit messy with features and language and trying to appease users' wishes and all these kind of stuff, so I wanted to clean it up, and I needed to understand it better. I felt as a physicist by training, I needed some guiding principles and having put this thing together intuitively I wanted to set myself the goal of trying to, is there a theory behind it, can I come up with an explanation of how it works? I sat down for actually 5 years doing this in enormous detail.

But, one day, I came up with this name, Promise, for this idea describing your intent, what you intend to do.

In CFEngine, we call these things configuration statements or commands or whatever but these things, it really promises that you're trying to keep because you don't just, make them, throw them over the wall and hope for the best. It's not like setting something up, deploying it and hoping for the best. The point of CFEngine was that it would continuously try to verify that this promise was kept and repair it if it wasn't. It was self-healing, self-repairing and I could see that this idea of a promise sort of fitted that idea and I kind of liked it because it also fitted this notion that it promises of something that kept by a person, an individual, entity or a proxy. I could also see it fitted into a much wider story too which appealed to my physics brain, I suppose. It fits in with a lot of ideas in physics and economics and human computer systems which have become my sort of research topic at the university because I've been trying to sort of give operations a scientific basis as part of my work at the university, and I'd understood early on that you must see it as a human computer system. The interaction between agents that are both human and machine because you need both aspects. You need the intent, what was intended and you need the execution of it as well.

I met my friend Jan Bergstra in, I think, it was April of 2004, and I told him about this notion of promises and as a logician he was sort of intrigued because it's sort of upside down in the way of logic. Logic is this kind of obligation; this makes it so way of thinking. He encouraged me to kind of develop the philosophy of this before getting too technical. And then, I presented it in Barcelona in 2005 and I guess it went on for the next 10 years, actually with Jan developing a lot of the ideas in all kinds of different areas before eventually coming back in 2007 and 8, re-making CFEngine in the image of all of that, that theory. People were getting interested in the ideas. I was trying to popularize some of it

and explaining some of it at conferences but in I.T. people are often scared at the abstraction. When we did the first book and people said, "That's great, but we'll read it when it comes out in Ruby."

Joe: *I think it's a brilliant statement because I think Promise defines it so well. I know exactly what you're saying there. I have to ask you; you wrote another book, In Search of Certainty. Is that book a beginning of your synthesisation, if that's the right word on the Promise Theory? Is that a tie into or it's something completely different.*

Mark: Yes, what I wanted to do is In Search of Certainty was trying to explain the scientific viewpoint of my thinking. Being a science guy, my background being academia and science and physics in particular, I drew on many ideas which were just there, natural and available to me from my cultural background. Science is part of our culture, perhaps not as big a part of our culture as I would like to be, but its part of the cultural narrative that we have in society and there are a lot of ideas that I could draw on by discreteness, information theory, non-determinism, a lot of these things which are just natural to a physicist really are not part of the cultural narrative in IT. The background of IT is, it comes from 2 places actually, one is the world of Turing in mathematics and the other is the world of engineering from Norman's side, McCullough's side, the engineering, electrical engineering side which is close to physics batch but still not quite bringing in all of that narrative.

Computer science just tends to focus on things like logic and sets and discrete, formalizations, it doesn't have access to a lot of the language of statistics and non-determinism and flow and all of the things that are very natural to physicists. But, all these ideas are super important to describe what's going on in a computer and I kind of realized this when I a Post Doc in Physics, after leaving the university and I was kind of looking at

the computer on my desk and thinking, "There's this idea in computer science that it computed simply does what we taught, and we can all laugh at this story now because, you know, everybody knows today that we're pretty lucky. It's a good day when the computer does anything like what we actually told it to do. Clearly it's related to what we're telling but there're all these other influences on. I thought we kind of got to look at this thing as a phenomenon, like a botanist watching the animals in the jungle through binoculars and we got to observe it and learn its behaviors first because it's really doing something that we're not OK with, and we need to learn those behaviors.

For a few years I just actually started measuring some of these behaviors to see what went on and discovered all kinds of interesting things like circadian rhythms and so on which are pretty well-known now. But, back then we didn't really have that kind of clear understanding. I wanted to tell that story and bring what I considered to be, you know, second nature almost to an audience that really hasn't grown up with that background and popularize it. You know, I love these books I read when I was a kid like the Selfish Jean and the First Three Minutes, Steven Weinberg and Dawkin's book, the great popularizations of science. These are things that inspired me when I was a kid and I wanted to sort of get back and try to see if I could write a book that would inspire people at a more cultural level I suppose and by telling the story of how all these pieces came together.

Joe: *I think you did a great job because I think the Promise Theory is really interesting to me because it starts putting much more perspective on things. I'm a sales and marketing guy. I'm not into all this Configuration Management and different things but the Promise Theory really resonates with me because with big data I think we need to start humanizing the big data, and I think promise theory does a little of that for me.*

Mark: Yeah, I think we focus a lot on data today, and I scratch my head a little bit at that.

I do recall being at the university and there was a group, the Space Physics group; they were big on data. They had big data. They collected stuff from satellites, and they measured the weather and the upper atmosphere and all these particle streaming and they got tapes, it was tapes back then of course, they had tape after tape after tape of data coming in and these tapes used to pile up in the side of the lab, and they literally didn't have time to process these things with our old sun workstation. These things were building up fast, and they could process them. I was thinking isn't the goal actually to try to reduce the amount of data by looking, by trying to construct an experiment that minimizes the amount of noise that you're getting if you're sucking in absolutely every signal that you can possibly do like scanning the sky for the SETI project or something.

There might be a whole bunch of messages from, like text messages from Proxima Centauri in there which you probably didn't mean to capture or maybe somebody's sifting through that soup in the future and might find it but is that really the best use of your time to just get a whole bunch of data. But, if you could redesign this around some intent, so call it a promise, focusing and defining your intentions more clearly then you could align that behavior with that intention in a more explicit way. Then, you might be onto something that could filter out a lot of that hard work so that you would be using and perhaps making better use of your time.

Joe: *We only promise something that we're pretty certain of and that's the thing that jumps out at me and is that a layman's way to understand the promise theory or is there more to it?*

Mark: Well, if only that were true, I think people promise stuff all the time way too freely and this notion people say 'I promise' so easily, and they have no intention of keeping this promise. That's a lie by the way.

Joe: *I'm in sales and marketing, and I mean sales guys never overpromise, okay.*

Mark: Oh yeah, don't get me started. That's a lie by the way, right, when you promise something that you have no intention of keeping but I think the key is that trust play an important role in promises, and I think that's one reason why this notion of a promise is important because it's a little bit related to knowledge, how much knowledge you have and if you don't, you know there's this great Russian proverb, the "Trust but verify" proverb, which just says that you can save a lot of time by trusting somebody or something if you, because verifying is expensive, right. If you have to verify, meaning you don't trust anything it's expensive to do that.

If you can trust a promise, you can save a lot of time and effort. It helps you to form expectations, and that expectation is knowledge that you build up by having a relationship with something or some process or some personal device. I think this is one of the important things about 20th century science was this idea that all of the theories in 20th century science were about incomplete information, having incomplete knowledge but still trying to make the best of it. Statistics came out of this. Trying to analyze the statistical variations in measuring the orbits of the planets but later got turned into thermodynamics by Boltzmann and then quantum mechanics, and then statistics apply to almost every area of modern life and this idea of certainty being sure about something super important.

We have varying degrees of certainty about things, but we still make promises nevertheless if we have some fair assessment that we're going to be able to keep the promise. What I like about the promise thing though is that it doesn't get too hooked up on this idea of a guarantee which perhaps is the sales pitch. You just guarantee me that this will be the case. It's what people want to hear, but a promise kind of admits that you may

not be able to keep your promise and there are all kinds of reasons, right, you may be unable to keep a promise, you may be unwilling to keep the promise. It may be beyond your control.

I always use the story of King Canute. He sat on the beach with his throne to ridicule this idea that kings actually had power and he tried to hold back the ocean by saying, "You shall not come in. The tide shall not come in. I decree it." The tide has washed him away without too much concern. Being able to promise, you need some agency. You need to be able to be in control. You also don't make promises about things that you're not in control of and that's the key, I think, of the Promise Theory. It's kind of the rule, you know, "An agent can only promise things about its own behavior, the things that it controls." You cannot make a promise on behalf of somebody else because you don't control that stuff, and this is why, you know, look what I can do. This is want I can do for you is the way to start trying to put that together.

Joe: *You bring up a point there, and as we go out further in time, there's more uncertainty. So, how does time play a role in promises?*

Mark: That's, yeah, that's the jackpot question, actually, perhaps maybe, more ways than you realize. Was a promise kept or not? Was it kept on time? Before the stakeholders got fed up and retaliated, or you went to war because you didn't keep your promise before you sued the delivery company for not providing you with, the medicine arrived too late, and you died before it was useful. I think this is an important point. A service that's provided too late or a promise that's kept too late may as well not have been kept at all or promised at all and this goes to reliability in systems and as well as trustworthiness of course over time. But, reliability is traditionally that's been thought of as some kind of objective measure of time to failure. A component will eventually blow up or melt or something in a

certain amount of time, and then it will be repaired or replaced, and this is how systems are measured in terms of reliability.

Actually, Agent Carkhuff likes to emphasize that a service performed too late may as well not have been performed and I think this is absolutely right because if you fail to keep your promise on time then you simply haven't kept your promise, and that's an important aspect to this. It's a little bit like the notion of rounds in game theory in economics where you're playing different rounds, and you're repeating, time is like a new iteration of trying to keep your promise time and time again. The notion of time repeating itself in cycles is also an important aspect at promising because often you don't just keep a promise once. It's not again this idea of throwing something over the wall and be done with it. It's not really like that right. People are testing our resolve to keep this promise maybe again and again. Does the system always respond within a certain time frame? Does it have a certain latency? Is the system always responsive within a certain number of microseconds?

There's this notion of, a perception of time as well where the autonomous nature of the agents also comes back into it because one agent may perceive time differently from the other. This expression where 'the watched pot never boils'. The wisdom there is that when you're waiting for something to happen, eagerly waiting for something to happen, you instead of twiddling your thumbs and you're focusing on smaller and smaller intervals of time. You keep sampling again and again and again, and you get faster doing it. Time seems to pass more slowly because you sampled more in the same length and the same interval, so you seem to have more ticks of the clock going on. Whereas when you're busy, you don't have time to sample very often, and so it seems like there were fewer ticks of the clock in between. Time flies when you're having fun but the watched pot never boils. Every agent has its own perception of time depending on its circumstances. The perception of a promise being kept is also extremely personal and subjective. I think this is also

something that Promise Theory captures fairly well.

Joe: *I think that goes back to my comment as a sales guy, the customer is always right and that Promise Theory really needs to be made together. Would that be a correct way of saying it, that it has to be… can you co-create promises?*

Mark: That's it. You nailed it. Yes, okay, so the sales things obviously is, sales is about trust, yes? And, the salesman knows it. Sales is about building relationships and relationships lead to trust, and you would buy something from someone you trust. Someone comes up making ridiculous promises or claims with whom you have no relationship, you wouldn't make the sale.

What I like about promises is that it kind of explains the algebra of how that comes about, how that trust gets built in the first place, and you can see it almost through the diagrams or through the algebra from what I would call, I call it the principle of Promise neutrality. In so far as the, I labeled promises with plus or minus signs, you know, signs plus, what was like an electrical charge. A plus is like to give something, a promise to offer a service to give, you know, to deliver the medicine, to deliver the e-mail and a minus promise is a promise to accept something that you receive something. If I'm minus-ing something I can accept your package, I promise to receive the message if it arrives at my desk. I promise to process your, respond to you within a certain time, and this principle of neutrality says that you need both of those components to make a binding.

Bindings are neutral. You need the plus sign. If I don't promise you something, you're not going to get it, but you're also not going to get it unless you promise to accept it and no amount of me promising to give it to you is going to force you to receive it because again this principle that you cannot force another agent to promise its behavior. You can only

make promises on behalf of yourself. That emphasizes that every promise actually to transfer something from one to another, to make a trade, to make a sale if you like, to pass something from one to another. There has to be this affinity between the 2 which is, "Does my promise to offer you something overlap with your willingness to receive it and to what extent, you know, how much will you receive? How much will you take and so on?

In that relationship, I think, is the basis of most of what goes on in economics. It allows you to formulate pretty much all of traditional game theory based economics. It also allows to see the value of cooperation because you need 2 parties for every single thing that gets passed from one agent to another. I think that really nails it, you know, it's all about cooperation and in the past, I think the mistake we've made in computing is to think in terms of commands throwing stuff over the wall, obliging, firing a missile to your counterpart and hoping that it's received rather than emphasizing the fact that both those parties need to be complicit in the cooperative aspect.

Joe: *I'm going back quite a few years here but I'm thinking that in some type of programmes, we used to call something the command line, and that goes back to your thing that it's, that's how we thought about machines and the way things work were commands.*

Mark: I may still be stuck in that past. I'm going to admit that. I'm still partial to my command line. But actually, even the command line is somewhat illusory because, you know, what the command line prompt is there's a promise accept your commands and unless that thing made that open promise to take your commands, now if you are not allowed to log into your machine you don't get that promise to accept your commands.

There is an implicit promise there which you don't tend to think about but it is actually

there, a promise to accept those commands which is the service provided by the command line service and you as the dancing monkey or whatever that types onto the command line your commands is promising to in a sense provide input to that process that's sort of flashing there waiting for you commands to be typed in but without again, the binding between the 2, nothing would actually get done.

Joe: *You make a statement in your book that 'autonomy leads to greater clarity', and I think it's interesting. Can you explain that a little bit to me?*

Mark: Yeah, so the autonomy principle is again, I think it's the central point of promises where the Promise Theory really started from. It's this atomic idea that brings this chemistry of intent or cooperation into focus. I think what it does is it hones your focus into what's right in front of you. In physics, we'd call this a local theory and what it means in promise theory is that agents can only promise their own outcomes, you know, their own behavior. You can't make a promise on behalf of somebody else that's not you.

In Physics, we call that nonlocal because it's not where you are. It's somewhere else, and theories of obligation are non-local theories. Remote control in front of the TV, you're trying to oblige your remote, device, object, and person, whatever to behave in a way, in a certain way. We tend to imagine that these things simply must do as they're told.

We all know that neither people nor machines nor devices even mechanisms that we design to behave and do as they're told, they don't always do as they're told. We can't keep their promise on their behalf because we're not in control of all of their circumstances. They are surrounded by environments which are intertwined complex we call them these days, many influences, information-rich scenarios where we don't know all that information about what's going to go on. We can't make that prediction. But, Promise

Theory tries to focus on the things that we can know with greater certainty.

Autonomy leads to greater clarity because it's focusing you on the things that you know can be delivered. Each agent makes its own promises and is therefore in control of its own faculties and it has knowledge of its own state, its own information without having to go to some other party and rely on that. It's about minimizing the dependencies on things around you that you're not controlling. The control thinking, the obligation thinking, the make it so thinking is you can stomp your foot and shout and scream, and you still can't make these things do what you want. But, by turning things around and thinking in terms of cooperative promises you're maximizing in a sense the likelihood of success because you're basing your estimations on information which is local, current, quite well-known, and it's straight from the horse's mouth in a sense.

Joe: *How do you think the book has changed your work? Has it taken on a different persona per se that's opened up other doors for you?*

Mark: Of course, I'm hoping for riches in a small island where I can be an autonomous agent. I might let some people visit, but mostly I'll drink coconut milk and watch Star Trek reruns. I got interested in, where Promise Theory led me I suppose is towards this idea of knowledge about systems, certainty, it starts with the certainty. What can you say with some certainty? Those things that you can say with certainty because you've revisited them, you've got this trusting relationship going on, that's what we call knowledge.

I made some statement back in, I think it was 2007ish when I was into this that the management of knowledge was going to be the big challenge of the next decade, and I do think that this is the case, and you mentioned big data earlier on. I think, today we're still stuck a little bit on data, but data isn't knowledge. Data is just signal which maybe, you

know, interesting it might be nice; it's just stuff. It doesn't have any meaning until you actually put it in the context of a model and the promises help you to do that. But, you know, when you do that, you can turn data into information and information when you trust it becomes knowledge that you can use.

When you can say that this information is when you know it like a friend when you have that relationship with it then it's knowledge. Know it like a friend. Knowledge I think is the challenge because we get so much information thrown at us these days, through Google, through Lookups, through big data, through census and pretty much everything in our lives, houses, fridges, carpets, you know, it's going to wired up to the networks squirting data at us. It's not going to mean anything unless we try to figure out this knowledge problem.

For me, I think the challenge is still there. It's how to extract, define knowledge and systematize knowledge so that it becomes a useful service as well. You know, so it becomes a utility, something that we can make use of and that requires, I think, changes in us as a culture and it requires retooling, not just new technologies, and it's not about grabbing as much data as you can. I don't believe that. Now, I've been interested in the internet of things lately. I've been working on that a little bit with some, with various partners and companies. I think that challenge of bringing data to us, monitoring, sampling, interpreting and putting into a context it's meaningful to us, being able to ask the right question, we'll have good questions answered. These are the challenges that we still face, and I think that's really where this is going for me.

If you look at the history of monitoring, I think there was a movement some years back called Monitoring Sucks which turned into the Monitor Armor series of conferences. Now, I've been wanting to go there for years, I still haven't made it there but I think, for me,

that was a great sort of observation because for years we've been monitoring systems by simply collecting random stuff, you know, not data related to whether certain promises were kept or whether certain questions were answered but simply gathering a bunch of data typically what we could collect and then trying to make some kind of sense of it, just throwing it at a person and saying, "You make sense of that." Unless you're trained scientist or an expert that understands the context in which the data were generated, you haven't really got any hope of being able to make sense of it and yet millions, billions of dollars are spent on this kind of software every single year. It's obviously a thirst for knowledge but we don't really have access to quality knowledge today and I think that is, that's really where this is going in that. The latter part of my book, In Search of Certainty, I tried to explain how that connection's made there also.

Joe: *Is there something you'd like to add that maybe I didn't ask?*

Mark: I think maybe, just a word of caution that's we live in this awfully sound bite culture, maybe some would even say anti-intellectual culture where we treat ideas as slogans. Recently I got concerned a bit about the misuse of the idea of complexity which has certain technical meanings and comes from a culture of physics and so on. This idea of complexity gets hijacked as a slogan and we tend to get hooked on these ideas like Promise Theory, Complexity Theory, Theory of Constraints and DevOps and these slogans which we treat as slogans rather than trying to understand what they're really saying.

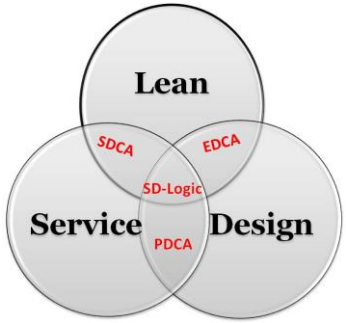
I'm that guy who likes to understand things in as much detail as I can so I often think we are perilously close to providing misinformation and misinformation is not a contest that you'd want to win for your resume. Just a word of caution that Promise Theory is a pretty well-defined thing. It has certain aims and goals and I hope that it won't be taken out of context, used to argue as an agenda or let's keep it science so ask the right questions and

use it to explore each scenario rather than as a manifesto for pitting somebody's agenda.

Joe: *I think that's very well said. What's the best way for someone to contact you and where can they find the book, Thinking in Promises?*

Mark: Okay, so knock 3 times on the ceiling. My website is probably the best way to get in touch with me, markburgess.org. I try to put useful stuff on there. You can apply for a visa to my island if I ever get it or, e-mail me, find me anywhere on Twitter and on Twitter, markburgess_osl. Love to hear from people and, of course, the book you'll also find me through my website or the O'Reilly website or Amazon and so on. I realized, I've actually written something like 17 books now over the years, and you can find all of these books on my website and hopefully there is some stuff there which people will find interesting. I do it because I hope people will find a use for it, and I love to hear from people, what their opinions are so please feel free to get in touch.

Joe: *Well, I'd like to thank you very much, Mark. I appreciate it. This podcast would be available in the Business901 iTunes store and the Business901 blogsite. So thanks everyone.*



Joseph T. Dager
Business901
Phone: 260-918-0438
Skype: Biz901
Fax: 260-818-2022
Email: jtdager@business901.com
Website: http://www.business901.com
Twitter: @business901

Joe Dager is President of Business901, a firm specializing in bringing the continuous improvement process to the sales and marketing arena. He takes his process thinking of over thirty years in marketing within a wide variety of industries and applies it through Lean Marketing and Lean Service Design.

Visit the Lean Marketing Lab*:* Being part of this community will allow you to interact with like-minded individuals and organizations, purchase related tools, use some free ones and receive feedback from your peers.