The Digital Service Economy
Guest was Jeff Sussna

Sponsored by

lean marketing lab

sales and marketing learning community dedicated to the user experience.

main   media   lean marketing system   lean engagement team   tools

We are CONNECTING!

Related Podcast: The Digital Service Economy

Business901   *Podcast Transcription*

Implementing Lean Marketing Systems

Transcription of Interview

Joe Dager: *Welcome everyone. This is Joe Dager, the host of the Business901 Podcast. With me today is Jeff Sussna. He is the Founder and Principal of Ingineering.IT, which is a Minneapolis consulting firm that helps companies adopt post-industrial IT practices. Jeff has nearly 30 years of IT experience and has led high-performance teams across the Development, QA and Operations spectrum. His interest focuses on the intersection of the areas of development, operations, design, and business. Jeff, I would like to welcome you to talk about your philosophy and particularly your new book, Designing Delivery.*

Jeff Sussna: Hi Joe. It's nice to be here and thanks for having me.

Joe: *I start off with this, and I know a lot of people know what post-industrial is, but it's a kind of a fuzzy term to some because is that 1920ᵗʰ post-industrial? Can you start off by defining that for me?*

Jeff: Sure. It all comes from a book that a sociologist named Daniel Bell wrote way back in the 70s called The Coming of Post-Industrial Society and he believed that we were in the process of a shift away from 250 years of Industrialism to this new societal and economic

structure and I find where we are now, given what he said 40 years ago that it was pretty amazing how much he got it right, and he saw 4 characteristics of post-industrialism.

The first was a shift away from products to services as the core of economic activity, and I think we can see that all around us right now. You read reports every day about, you know, 80% of new jobs are service jobs, so on and so forth. The second was moving from a focus on work that was about either manual labor or management and bureaucracy towards knowledge work.

And again, all you have to do is read the papers or read the blogs to see that happening. The third one which kind of threw me for a loop a little bit but I think it's really true is that previously corporations and enterprises were fairly conservative in the sense that they were focused on how to do the same thing in the same way day after day after day. We make ball bearings, or we make band-aids or we make cars. We figured out how to do it well. There was a certain resistance to change. Everyone talks about how hard it is to change things in large companies. But now, it's becoming their actual purpose to change.

There's all this talk about innovation and innovation labs and how do we respond to disruption and so on and so forth. The whole notion of why companies exist is shifting. And then, the final one was his belief that for all of that to work it would require pervasive computerization which is where we come in and where I come in. I add one more characteristic that I see which I call infusion, in the sense that our physical ordinary lives are becoming ever more infused with the digital realm.

It's becoming harder and harder to do anything that doesn't have some online component, even driving a car. The second part of that is that the different parts of our lives are infusing each other. It becomes harder and harder to separate them out. If you go to a coffee shop, are you're going there because you want a cup of coffee or are you going there to work? The answer is both. Again, if you're driving a car, is it really a car or is it a Spotify client? The answer is both and that creates interesting challenges both for IT and for people if you can't play music in your car. It used to be just a matter of, well, maybe the cassette tape was twisted. Now, is there a problem with your car? Is there a problem with Spotify? Is there a problem with internet? So, it makes the whole process of figuring out where we are and what we're doing and how to make it work much more complex.

Joe: You bring up an interesting point, we used to be able to separate things and organize our life somewhat in chunks but there's no longer chunks because we really need to embrace the actual, the overlapping of everything and it's just not really defined, but it's a culture that we have to embrace, don't we?

Jeff: You're absolutely right and I think when you look at the trends in corporate organization, whether it be small cross-functional teams or open source culture or Holacracy or what have you, that's what they're all about. No longer can we just say, "Well, this is the Manufacturing Division, and this is the Software Division or this is the Finance Department and this is the IT Department." They all interact with each other, and that's really at the heart of what my book is about is that the parts of organizations interact with each other and the organizations interact with their customers in new ways.

We need new models for thinking about it, and we need new models for managing it and doing it.

Joe: *Tell me a little bit about what your book is about, Designing Delivery, should I take that literally? I mean is it about deliverables? Is it about design for deliverables or is it just about software since you're an IT guy?*

Jeff: The answer is yes. It's a bit of a play on words. On one level it means that as software becomes service, you know we're pretty much at the point where we interact with just about all of our software in a service model, what we're delivering is digital service not just software. So, on one level that means that when we think about software and software quality, we have to think about things like technical and business operations and customer support and marketing and documentation. That's all part of what we're delivering. On the second level, the design of that is part of what we're delivering as well. We're delivering the ability to continually redesign ourselves. We hear talk all the time these days about disruption. Kodak went out of business and IBM is in trouble. Ford is reinventing itself into a software company as much as a car company. One of the characteristics of post-industrialism is the need for the ability to continually adapt. The ability to do that is actually part of what we have to deliver to our customers.

Joe: *The readers of my blog will recognize many of the terms you used throughout the book, Service Design, Service Dominant Logic, Jobs to be Done, Design thinking, all big terms but how have they changed or complemented your thinking in recent years? Is there an underlying thread to all this?*

Business901   *Podcast Transcription*

Implementing Lean Marketing Systems

Jeff: Yes, I think the underlying thread is Systems Thinking. My background is interesting in that I've worked for software product companies, and I've also worked for service companies, both in the IT space. I worked for a company that basically managed data centers and servers for people, but I've also worked in companies that were essentially design agencies. I have some understanding from my background of the service model. As I got involved in cloud computing and started seeing software becoming more about service, I started thinking about that and I got introduced to the world of Design Thinking and Service Design and it occurred to me that, "Oh, we're really talking about something different here, and we really need a different view.

We need to shift away from thinking about things to thinking about relationships. What is the point of this software I'm building? It's to help my customer accomplish something. I think the theme for me again is it's all about Systems Thinking. Basically, it doesn't just say, "Well, there's a big system with a bunch of parts. What it says is the key is to looking at the relationships between the parts, not just the parts themselves.

Even if you take something straight from my IT world which is this new methodology called DevOps, the reason it's called DevOps is because it's looking at the relationship between development and operations within companies and thinking about the idea that, well, we really need to think about them as one thing because just because I build software doesn't do anybody any good if you don't operate it for them. If I use an online invoicing service and I love all of its features but it's offline half the time or it's slow, or it loses my data or it gets hacked, that doesn't do me any good. I like to use the metaphor of a restaurant. If there's a new restaurant in town and somebody goes out to eat and you ask them

afterwards 'How was it' and they say, "Well, the food was great but the service was terrible." I'm maybe not going to go to that restaurant. DevOps supplies the same to IT. The theme for me is really about thinking about all of that in the largest possible context.

Joe: *It's really kind of a digital service type of atmosphere that you create. I think you talk about four dimensions of Digital Service. Can you tell me what those are?*

Jeff: Sure. This all drives from the idea that just building software that works is necessary but not sufficient. The first dimension of Digital Service is outcomes, and this is kind of a service dominant view of functionality. When we think about it in terms of quality, the first question is, "Does the software work properly? Does the functionality work?" On one level, if we build a widget that lets somebody enter their phone number, we want to make sure that that widget works and actually accepts the phone number and stores it in the database properly and so on and so forth. But, what we're really after is 'Does this widget actually help the customer do what it is they're trying to do?'

The second is Access. The fact that I have this nice widget and the widget is useful is great but if it's slow or I can't get to it or all of those things I talked about before then there is no service quality. The service doesn't work. The third is what I call Coherence, which really applies to the idea of the customer journey from service design. So again, I have this nice widget but how does the widget work? How do I learn how it works? How do I get my system configured so I can use the widget? How do I integrate the functionality of the widget with the rest of my business?

In service, there's always a journey where the customer interacts through a variety of steps that happen over time and those all have to hang together. Let's say that I'm having trouble with the widget and I called customer support and customer support doesn't know what I'm talking about or they don't understand either the software or me well enough to understand the context of my problem and they're like, "Well, it just works this way." "Well, I don't understand it. I'm trying to do x and I can't." If the customer support person doesn't understand me as a customer and what I'm trying to do with their service well enough then that's a service quality failure just as much as if there were software bug.

The final one, which is the most subtle and sort of the deepest level of the book, is what I call Continuity, which refers to this idea of services needing to change over time. To go back to the online invoicing example, I started using an invoicing service a number of years ago. I'm an independent consultant, so the ability to invoice is very important to me. When I first started using it, it solved probably about 70% of my problem, which was find because it was great because the simple idea that I could log onto a website and enter my time and generating invoice was so wonderful that I didn't care about the other 30%.

A year later that's not good enough anymore. I need the service to continually adapt and improve itself to follow along with my needs and the changing market and I need it furthermore to be able to deliver that change. These days I'm sure we've all had the experience of logging into a site that we use every day and all of a sudden it looks different. They have this nice, sexy, new UI that's better than the old one. But, from my perspective it's just confusing because wait a minute, I expected everything to be blue and have round buttons and now all of a sudden everything is green and it has square buttons.

What am I supposed to do? This process of delivering change and helping people understand what it means and how to use it, again, is another aspect of service quality.

Joe: *You're looking at something that is continuous that I'm hardly even to some extent noticing the changes that are taking place. Is it really how the world should be now?*

Jeff: Well, it maybe that you don't notice it, or it may be that you get educated as to why it's good, and you get an opportunity to engage with it. For example, Google started a practice where when they were preparing to release a major change, they would start to put up, pages when you log in would say, "Hey, we're going to do this new thing and here's what it is and here's it's cool." And then, they would transition to a point where they would say, "Well, if you want to try the new thing, click here and if you don't like it you can switch back." So, they actually made the introduction of change a very explicit design practice.

Joe: *It's educating the user as he goes about it and giving him some options, and you're actually probably learning from who chose the options and what they did, right?*

Jeff: Yes.

Joe: *That SD-Logic thinking where you co-create value together and, value is at the point of use that you determine that. Also, you use Customer Journeys to explain the process of validation in your book. I sometimes struggle, if I have a customer co-creating value, how do I perceive validation anymore? I mean how do I know or do I just accept that it's never finished. Can you relate that to me?*

Jeff: Sure and that's a wonderful question. So, the first thing is when we design something, we are basically creating a hypothesis, not a solution. In the sense that we say, "Okay, if we built service that does a certain thing, people will use it in a certain way, and it will have a certain value to them." The first thing we have to do is, well, let's assume that hypothesis is correct and let's make sure that what we built actually works according to the hypothesis. That's traditional QA and co-creation doesn't mean that you don't have to do QA anymore, right, if your software is riddled with bugs, whether it's the right software or not, it's still lousy software. So, as a company, we start with an idea, and we have to make sure that what we've built actually fits that idea.

The next thing is when we release it into the market we have to understand that it's not necessarily going to get used the way we expect and that happens on a number of levels. One of them is sort of the relatively straight forward c-creation level which is I'm using your invoicing service to do my own thing which is run a business, right? I don't buy your invoicing service because I want to buy an invoicing service. I do it because I want to get paid on time. So, we need to think from that perspective, and we need to get customer input in order to validate that our hypothesis made sense. But, there's another level in which the whole idea that we control what things are as designers are somewhat of an illusion.

It's interesting if you look at things like Facebook and Apple and Google and you look at some of the challenges they're facing, right. Facebook had a really interesting one recently which is they were insisting on people using their real names to sign up for accounts, and they were rejecting certain names which turned out to be native American names. They

were assuming that if someone signed up as Ben Runninghorse that that would be a joke. Well, it wasn't.

The point is that I can pretty much guarantee that Facebook could not have foreseen that when they started. It's not something we look at that, and we say, "Well, they should've designed for that. They weren't empathic enough. They weren't user-centered enough." But, the point is that it's not until your software gets out into the market that you really can understand all of the problems. So, this process of validation and design has to be continuous. You have to be prepared for the idea that you're going to get stuff wrong, and you have to have the humility and the process to be able to repair that as you go, and that's in my mind where things become continuous.

Joe: *I think that's a theme that resonates through the book. I go back, can anything be really continuous? To me that means there's got to be a certain amount of co-ownership also doesn't there or not?*

Jeff: Yes, I think there is co-ownership, and I have to say I think your questions this morning are wonderful. I love them. They're really, really good questions. I think this goes back to the what we talked about earlier about this idea of complexity and the parts of our lives coming together. So again, if we take it in a very straightforward way, let's look at something like Amazon web services which is a cloud service. They basically let you rent virtual servers by the hour.

On the one hand, if I use AWS I have outsourced my IT infrastructure. But, on the other hand, I've made them part of my own IT group because if my systems go down because Amazon is having an outage, my CEO isn't going to go to Amazon they're going to go to me. I have to, on the one hand, that means that the service that Amazon provides to me has to be a little different. It also means that how I think about them, I can't completely blame them. If they have an outage, well, why didn't I plan for the possibility that they had an outage?

So again, it all gets much messier and we all have to take more responsibility for what we're doing and we have to take more responsibility for understanding what others are doing so that we can kind of make this messy thing appear to be nice and neat and actually work. The interesting is that, you know, 99% of the time it actually works pretty well and it actually looks pretty clean and neat.

Joe: *How you said that was brilliant. What popped up in mind is I can understand having that relationship with a vendor, I can understand working with Amazon in the way you explained it and everything but then to turn the tables and have my product and service be that way with a customer is a kind of, "Hmmm, does it really work that way?" Can it really work that way? I think it was wonderfully stated, the way that you did because you have to include them, don't you?*

Jeff: There's a very interesting anecdote that illustrates that. A couple of years ago, Amazon had a major outage. It was like 20 hours long or something like that and the entire East coast went down and for about the first 10 hours or so they didn't communicate

much and they got a lot of grief for it and they actually posted a little blog or something where they said, "Hey, we thought you wanted us to focus on fixing the problem not talking about it." And, people said, "No, we need to understand what's going on. We need to tell our management what's going on, so we need you to talk to us."

Amazon changed their tune, and they started publishing a lot of information, a lot of details about problems they were having and how they were trying this fix and it didn't work and the interesting thing was that nobody got mad at them. People started rooting for them because every, these were a bunch of IT guys, and they all understood the pain that Amazon was going through, and they now had the information they needed so they could relax a little bit and basically what suddenly started to arise was empathy. Instead of taking this hands-off approach, people were actually talking to each other almost as if they were part of the same company.

There's a lot of talk in my industry these days about operational transparency, about revealing yourself to your customer instead of trying to keep secrets and we can actually see this in the ordinary world when we look at security breaches. We'll look at one company that'll have a major security breach, and they will talk about it for 5 days and then they'll be kind of cagey and people get really pissed off at them and they'll lose brand goodwill. You'll look at another company like Zappos a few years ago, they had a security breach and they basically just came right out at the beginning and they said, "This is what happened. It's our fault. This is what we did wrong. We're really sorry. This is what we're doing to fix it. This is what we're going to do for you." They actually reinforced their brand instead of degrading it.

Joe: *I think software guys get it, okay, I think they understand it and I think part of it was because they were kind of the lone wolves within their company so 20 years ago and they grew up with that mindset that we're open and we talked about problems with other people and outside our company. A lot of companies really have a difficult time with sharing with customers problems. Is my analogy right there? Is that something that you see also?*

Jeff: Well, I think you may be giving the IT guys too much credit. But, aside from that, one of the things that social media does is it rips control of the message away from the corporation. If you're an airline, and you're having problems and you have 10,000 or 100,000 people talking to each other on Twitter, trying to be secretive about your problems doesn't work anymore. And again, you can see this on Twitter. More and more judge companies by how they respond when I complain to them, not about them but to them on twitter and some companies will talk to me and even though they had a problem, I will then turn around and say, "Hey, this is a great company because they helped me out." So, I think, to go back to the beginning, one more thing that's happening in the post-industrial world is the whole model of who controls the message is being flipped on its head.

Joe: *You end the book about talking about promises and that terminology I have only seen in the Lean Construction world when we talk about the Promised Conversation and the Last Planner. Can you explain how and why that terminology is important now?*

Jeff: Sure and the terminology comes from something called Promise Theory which was developed by Mark Burgess who is really one of the luminaries in the IT theory world. He's

an entrepreneur and a professor and a researcher and the basic idea is that when we build systems, complex systems and they might be IT systems or they might be human and organizational systems, we make promises to each other. I promise that I will get the budget done on time or I promise that this website will serve all of the traffic that you can throw at it or I promise that if you fly on my airplane I will get you from New York to Amsterdam without falling into the ocean.

The reason that the word promise is used is because we don't always keep our promises, right, websites go down, budgets don't get done on time, fortunately, planes very rarely fall in the ocean but they don't always arrive on time. Right, one of the promises I make to you is if you fly my plane I will get you to Amsterdam by 10:30 in the morning. That doesn't always happen. What we traditionally do is we try and ignore that fact and just say, "Well, damn it, we have to get everything done."

The reality in a complex world is failure is inevitable so what's more interesting is to think about what do we do when we fail. How do we fix our broken promise? You can see that with outages in digital businesses. You can see that with even a warranty on a car says, "We're selling you a car that we promise we'll never break but guess what if it breaks in the first four years we'll fix it for free." The irony is that when we think in terms of promises as commitments that might not be kept, we begin to deal with uncertainty and failure and by dealing with it, ironically, we can make systems that are more robust. In the world of service in general and digital service in particular, you could say that a service is a set of promises and then the opinion of your brand on the customer's part is based on how well you keep promises and how well you repair the ones that you can't keep.

Joe: *I think you could say that for just about any product, let alone just services. If you think of SD-Logic, the value is in the use of it, right?*

Jeff: Exactly. Absolutely right.

Joe: *You're promising someone that they're going to be able to use that product/service, and you can fulfill that promise in what goes with that. That promise may be taken differently by your customer also. So, there's some give and take when you're having a conversation?*

Jeff: Absolutely. I couldn't have said it better myself.

Joe: *Is there anything else you would like to add that maybe I didn't ask?*

Jeff: Well, I think just a couple of quick things. One is my hope for the book is that it will really appeal to a market across functions. Now, it talks about development and operations and QA and it also talks about design and product management, and part of my thesis is that in the world of digital service they all have to come together. I'm hoping that people across various domains will find it interesting. The second is I'm beginning to do workshops based on the book. I'm doing 3 in Europe this Fall, and I'm available to do others. The first one I'm doing is in Minneapolis next month, and I'm actually very excited because the audience is very cross-functional. The folks that have signed up are product management people, designers, developers, QA folks and even a DBA, database administrator.

Joe: *I have to say after reading the book, what it really did for me more than anything else is it made me want to learn more.*

Jeff: Oh good.

Joe: *Where can we find out more about the book and contact you?*

Jeff: You can find the book on Amazon or the OReilly.com website. O'Reilly is my publisher. The best way to find me is on Twitter, @JeffSussna, or on my company website, Ingineering.IT

Joe: *I would like to thank you for your time today. This podcast will be available in the Business901 iTunes store and the Business901 blogs. Thanks everyone.*

Jeff: Thanks for having me.

Business901   *Podcast Transcription*

Implementing Lean Marketing Systems
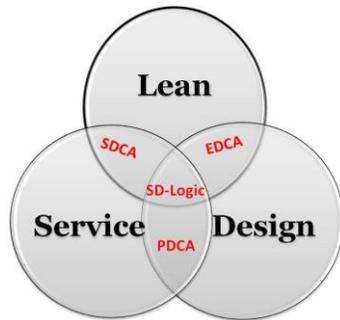
Joseph T. Dager

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.