



The Needed Mindset for DevOps

Guest was Shirly Ronen Harel

[The Needed Mindset for DevOps](#)

[Copyright Business901](#)

The Needed Mindset for DevOps

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.

Joe: *Welcome everyone. This is Joe Dager, the host of the Business901 podcast. With me today is Shirly Ronen. Shirly is engaged with both small and large companies optimizing working processes, team resource allocation, and providing insights using Agile and Lean methods. She is also the author of the Agile Kids book and currently writing my new book about personal agile coaching, or it may already be out. So Shirly, could you finish that introduction for me, clean it up a little bit, and give the elevator speech?*

Shirly: I don't know about the elevator speech but first of all, the book is done, and it's already sold - The Personal Agile Coaching. But let's go back; I will introduce myself, and I will add something because our topic is going to be a bit about DevOps and testing, so I want to add some relevant things about my background that relates to it, that sets my beliefs for this issue. I'm a Lean and Agile coach and in my past, I was Director of Social Quality in a few big organizations which were very traditional organizations. So when you are the quality director in a traditional organization, you're usually located at the end of the process and you get all the bugs into your hands, and all the blames into your hands, so I had no choice but to stick into other people's processes and start learning these processes and start understanding them. This is where I got the process view in understanding the entire process of an organization.

20 years before, I was a social worker, so I was engaged in family therapy, in health issues, so I learned how to move people and change perceptions of people and change behavior in the difficult situation. I used to take care of very sick people and sick families, so if you're taking both processes and people, I think they should work together. Especially say DevOps or Agile, you cannot ignore the people inside the process. This is where I came from, and this is what I believe in.

Joe: *Working with maybe some very sick people and working with some very sick organizations, would that be a fair way to say it?*

Shirly: Yes, but I would drop the word 'sick.' I would say, every organization has its own personality. Some are more comfortable personalities, and some are more troubled personalities.

Joe: *And they have personalities, right?*

Shirly: I think so. Some people like to call it DNA. I like to call it a personality. You need to fit your processes, and your tools, and your statements, and your mindset to a specific and special personality.

Joe: *Yes. Well, we're here to talk to DevOps, so what is that DevOps mindset? I mean how do you define a DevOps mindset from a big picture there?*

Shirly: I think that before we start defining the mindset, I will set straight what I think DevOps is. Because if you ask people in the room, maybe if you take 20 people in the room, and you ask them what is DevOps, you'd get a lot of answers. If you take 20

Israelis, you'll get 25 answers, and they will keep on arguing because this is what they like to do. So, I will set straight my beliefs on DevOps and then we'll get to mindset. Let's say we have to go from A to B, where A is your dream and B is the things that you can get out of your dream and make it happen, and see them, and touch them and whatever. So, at the A side, you have those Dev people who want to change everything. And on the B-side, you have those Ops people who want everything to stay stable. So you need to go through this role which is not really A to B, like A to Z. It's a very long and complex world with a lot of people, and a lot of processes, and a lot of things and decisions, and a lot of professions in the way, and you want to do it really, really fast, and you want to keep the quality, and you want to change along this way with everything.

First, to make your world easy, you can use tools. Because setting and placing tools around this road like continuous integration, some monitoring tools, some log in tools, this is usually the point where you meet the Ops people who call themselves DevOps. They connect the DevOps to those tools they put on this road, from A to B. But tools are not enough, because you need to know your processes, and you need to know your road. You need to know where is the best place to put your tools, what kind of tools, and when. So in order to do it, you need to develop some kind of end to end thinking about process, and you need a lot of information coming from this process. You need the people along this road. You need the information from different professionals. You need the perspectives so you can have all this information, knowing what is the best way to grow fast because you cannot place all the tools all the time, changing all the processes, because it will take years. So, in this way, you need to develop some kind of culture where people know they are part of the big process, they need to collaborate, they need to give their feedback, they need to create fast decisions, and you can go fast and still keep your quality. So, this is what I think is DevOps. Those 3 items – tools, process, and people.

But let's go to the mindset now. If I go to the mindset, then I need to put it very, very clear in my organizations what I want those people to achieve. I want them to remove the barriers between those who want to change and those who want to keep everything stable between Dev and Ops. Actually, I want to move the barrier between Dev test and Ops. I want more quality or better quality or 100% quality, and I want fast progress and reaction. But this is also not enough. It's not enough because you need to start translating it to things people can relate to. Usually organizations, when you come to an organization, you see these posters on the walls where they said collaboration, innovation, quality...

Joe: *We're all Agile now, right?*

Shirly: But people doesn't know what to do with those. So, if I will draw my poster, it would be written on it, no walls, no gates, quality, trust and collaboration, early feedback, continuous improvement, Lean thinking – but I still need to translate it into behaviors, actually behaviors that the people will know what I mean. I will give you an example. Let's take collaboration for an example. In the traditional world, when a tester comes to a developer and asks for help, he usually works, or she usually works on something that was developed in the best case, two weeks ago, or maybe two months ago. And the developers have other priorities; then she needs to approach the manager and then. Usually, she doesn't get help. When an Agile tester comes to a traditional organization, she will ask help directly from the developer, and there will be a clash. This clash and this Agile developer or tester that used to direct communication used to get answers and are very fast, and the same priority with the developer is her priority, is not the same mindset as the rationale.

So when we are talking about collaboration and we want to make the culture of

collaboration, we need the organization to set those stages of collaboration. They need to approve them sitting together. They need to approve them having the same backlog and the same goals. They need to respect the ceremonies. They cannot cancel dailies. They cannot cancel retrospect. They need to have those rooms to collaborate. They need to have maybe screens if they're co-located teams. They need to have all those stages of collaboration. I think that the development manager cannot shout on their support people if they have bugs. The mindset of collaboration will make him think differently, like how can we prevent it, and why did I get this support person into trouble in the first place. So this is the differences of collaboration, and there are a lot of things, small things that you find along the way, that you need to start and fix into behavioral actions, into actions. It makes the people act according to what they want, but what you want to achieve – quality and fast moving. I have a few more fun examples, but only if you like.

Joe: *A lot of it has got to do with creating that frequency of interactions so that the upside doesn't feel like they're just the testers and the developments, giving them things to test. Explain to me, how do you test let's say in an Agile environment?*

Shirly: This is a big question because nothing stays the same in an Agile environment in terms of testing. Just think of the constellation; in a traditional environment, the testing is usually located at the end. They have their own group. They sit together. They meet together. They have these Gemba professionals. I used to manage those groups. They're really excellent groups. They're really motivated and effective and stuff, but they are blamed for everything. And now you want them to start testing everywhere, from the first line of code, which is probably impossible for a tester to test everywhere all the time for all the codes. So, their entire way of working is changing. The way you planned the testing should change because you're not at the end anymore. You need to plan with small

chunks. The way you execute the testing changes the way you monitor the testing. The way you manage the defects has changed because you have no zero defects approach. What the hell is this and they don't believe they cannot get zero defects approach, just give the testers software and you'll find 100 defects. So, everything changes. If you start testing from the beginning, you need to change even the smallest methods. You cannot automate anymore the software, top down at the end using GUI. You need to find a way to automate from the first line of code. The unit test, acceptance test, service level test. The GUI tests now become maybe 10% of your tests. Your skills are changed. Many old testers are becoming less and less common in Agile development and in Agile testing because most of the tests are automated. So what are we going to do with manual testers? I have the answer, but not for this podcast. But everything changes, so I think the main change is the mindset change. We need to separate between the tester and the test.

Joe: *So is there a difference between Agile and DevOps testing, or is that what you're explaining?*

Shirly: I think DevOps is just a phrase, where publishing and automated testing and testing is the glue for going fast and for DevOps. So, DevOps is just a phrase. We are just starting to create turbo testing and a lot of automated testing. You can call it DevOps, and you can call it whatever you like, but a lot more automated testing everywhere and in everyone and this is a big change. Because sometimes they come to a developer, to development groups, and they say, but we cannot test it if we don't have a tester, and I said, the tester is not a human being. And they said, what? I said, yes, testing is just the task. If you can add testing everywhere, you don't need to have a specific person to do it. You can perform unit tests by yourself as developers. You can perform functional tests on each and every user story. Write and perform service level tests and even run them. You

don't have to have a specific group. It's even cheaper that the developer will write his own automated test because he's closer to the code. Someone should raise the flag and should raise the issue of quality and planning, and take a look at the big picture of quality. This demands really quality people, but it's a hat. It doesn't have to be a person. I think that in the world of DevOps and in the world of Agile testing and in the future, we'll find more and more developers doing automated tests and thinking about how to make their code more testable. We will be able to decrease those handoffs between developers and testers. This is part of growing fast; thinking about how to break those barriers are in our mind, that they need a tester.

Joe: *That may or may not be true but you know, for me, the Ops side, I don't have to change my mindset and everything, but I'm responsible for getting things done. I mean, that's the majority of my job, right? Do I want to open something up and experience something new? I mean, you talked about that mindset. What's that fine line of productivity and continuous improvement? Well, that's tough, isn't it?*

Shirly: I think the world is going in a fast way, a lane of progress. The amount of changes that we will experience will be greater than we experience now, so the line is changing. The fine line between innovation and stability was different in the 90's and different today. Today, you can find an organization Ops people really understanding continuous integration, but they don't really know the difference between continuous deployment, continuous integration, continuous delivery, but they're really open-minded people. But when you teach them new tools and new stuff, they learn, and when you teach them the way of going fast. But I think that each and every role of the organization needs to think about the entire organization in one hand and understand their places or roles in other hands.

If the DevOps people will understand, I don't know if there are people, DevOps is a thinking, if an organization will understand that because they want to move fast, keep quality and change, they need to have tools, they need to reduce waste, and they need to collaborate, you will find a lot of innovation inside, because it will be part of the role. So they will do their work, and they will also find innovation inside. I see this happening and if it's happening today in development teams where the old Agile fashion is development and QA, why can't it happen with Ops people. They are a little more rigid, I agree, but we can change them.

Joe: *I'd go back and I'd use this example is that I walk out of the airport, I hit my Uber button to grab my Uber and see where I'm going to go and the I get this update, and I'm sitting here waiting for an update to happen because I haven't used it in two weeks, and all I want is a car. I want to get where I'm going. It seems the clunkiness of what was supposed to be convenient challenges that DevOps or that Agile mindset a little bit. Or am I just getting too old?*

Shirly: There is a lot of waste in every process, and sometimes you solve waste with waste. You hire more people to save one waste, and then you get a lot of waste of processes on the other hand. But I think I like this technology. I like the fast moving technology. I like the fact that by clicking one button, I have a taxi, and I don't need to wait, and I can call my taxi driver, and he talks to me, and he waits for me after 10 minutes, and it's very, very convenient, and it saves a lot of time. Now, go back to an organization where the organization environment is very, very competitive, especially for this kind of application that you just mentioned. If you have a long update, then it's a problem. And a customer, nowadays, you won't wait. You will download a new version

and then you will throw a few words on Facebook or on Twitter and along with you, 100 people will download this version, and you will close the company that gets you a long update. So this is a problem, but this is also a very competitive market and I think that people inside the organization need to know that the need to go faster doesn't mean they need to lose their head, but there are techniques that they need to learn in order to go faster.

Joe: *Well, I want to jump on into a different area here a little bit because when you explained the different practices that you participate in, it seems that you rely on Scrum as your methodology more than others. Is there a reason for that, or am I just looking at the wrong material that you're putting out?*

Shirly: It's just because the material I put out is based on the customer requests, my customers. When I have a conversation with a customer, how can I involve DevOps in my processes and most of my customers are working Scrum right now, so I will write about Scrum, which is more common, and people can relate to it. But, since DevOps is a mindset and the mindset of culture and collaboration, you can do DevOps with Waterfall, and you can succeed, as long as you follow early feedbacks, long chunks, continuous improvement, going fast close collaboration. So, someday we'll find something. And by the way, they're more than just Scrum to Agile practices. You can have Kanban or ScrumBan, which I like the best. It's just for the sake of I cannot write a post about Scrum and Kanban and everything else.

Joe: *Probably one of the more common methodologies in Lean people adapt to what you're saying from that.*

Shirly: Yes and I want my customers to read it; those who asked it.

Joe: *Well in Scrum, they have this thing called a Sprint, and you talk about buzz words lately. I mean this year's buzz word seems to be 'Sprints.'* What makes a Sprint a sprint? I mean, in Scrum, maybe in the purest sense and maybe expand on that, outside the pure sense.

Shirly: I want to distinguish the Sprint from the Scrum because I really like Sprints. I think it's a wonderful thing, and I want to thank the people that invented this. It's just a time frame that has a beginning and an end, and it's short. It's short enough that you can look at what happens inside, and you can retrospect over it, and there are things you need to do. It's like a heartbeat. You have a heartbeat of constant sprints, and even intuitively, you can measure your progress. You say, I have this amount of work to do or this type of work to do and how did I do it? And then, you try again. It's kind of practice. In a psychological aspect, it's like you're playing a computer game, and you're doing your round, and you failed in your round, and you do it again. Then you score more points, and you do it again, and again, and again. You can create small, and you can check your fails, and you can change your fails. It's small enough that you can actually look at what you did, and you actually can pick one thing and change it. I see organizations and teams grow using sprints, just because it's kind of a routine where you do the things over and over again. It's like when I came to practice for this interview, I looked at your question one time, and then I looked at them again after a while, and then again after a while. And from time to time, I find myself giving a bit different answer, or maybe a better answer. So, sprints give you this opportunity to improve, and it's wonderful thing.

Joe: *When you describe a Sprint, to me, it sounds like you like to establish boundaries or*

a capsule that you put what you're going to do in there so it's definable and you understand it, and I can just get it done, kind of.

Shirly: Yes. In my book that you mentioned at the beginning, about the personal coaching with Agile and Lean, Sprint is very, very key because Sprint is a heartbeat about holding those things that you want to do, and it grows you. It grows you just by using those boundaries. It's like in adolescents and children, it's a routine, and people are moving very, very well inside routines. If you add to this routine's innovation and empowerment, then you have boundaries from one hand and rules, ceremonies, and symbols, and empowerment, innovation, and growth on the other hand.

Joe: *Maybe you explained it, and I didn't catch it as I was listening there, so the difference between the Scrum and the Sprint.*

Shirly: A Sprint is a tool inside Scrum It's something Scrum is using. I just like Sprint as a sprint, in any case. If I use Kanban., I usually recommend Scrumban. And taking from the Scrum, the Sprint, giving the teams kind of boundaries where they can stop for a minute this race.

Joe: *Operational people, do they understand the Sprint process easily or even Scrum? I mean, is that foreign to them when they move into that process? Does development lead the way or is it a learning curve for both of them? How does that normally work in an organization that you work with?*

Shirly: There is no normally. Each organization has its own personality, but as far as I can tell, I see that they don't understand and at least in this point of time, they think of

themselves as tools people. What are the tools they need to set? So, at first when I come into an organization and I used to manage DevOps myself, I do some kind of evaluation and start understanding what is this mindset, the organization mindset, what is the level of the agility, because people say they work Scrum, but there are times, they are not. They just have a Sprint in the best case daily, and things are not going so well. So the first place, you're not taking the Dev people inside. They need to do other things. But let's say you have a good Scrum inside or type of an Agile, you need to start to involve the Ops people inside this Agile process in a way.

I usually take this maturity model that I find on the internet. There are a lot of maturity models for DevOps on the internet. It's just the stage for the conversation. It's not really a good model, but it's a stage for conversation, starting to let them know that DevOps is more than just tools. It has other dimensions. It has monitoring dimensions, maturity dimension, Agile dimension, process, continuous integration, and what they call CM or change management. When we start talking about it, we start picking up or picking out those things we want to take care of. As a coach, working with them and growing them into the next stages of things that I think they need to know. So, it's a process, and it's doable.

Joe: *Can you scale with DevOps and Scrum? Is it made for smaller projects? Big projects? I mean, how does that work?*

Shirly: It's like the question about Scrum 10 years ago. Can you scale Scrum? People said you cannot scale Scrum. Of course, you can, but there is a price for scaling. The price for scaling is a lot more processes and sometimes command and control. So, scaling is the same as going first time to anything else. You need to keep the mindset intact, and you

need to have a strategy. In one organization, you scale everything at the same time, you get the same tools, and throw your process, and pick the tools, and then get your process intact, and then train the pole, and then coach them for the mindset, and in another organization, you do the exact opposite. But the things that I like the best when you scale, and I did it only one time, is going into the organization, building a separate DevOps, and we start picking all those amazing, homemade automated tools, and homemade fast-moving glues that the development developed during the years, and we build them as one line, and we start it from there. And then we started the Agile coaching and developing the Agile with the DevOps inside, so scaling as easy. So everything you did with Agile, you just did with DevOps.

Joe: *If I want to start out if I want to become and move towards DevOps, if I want to start developing that mindset, and I may even partially have a mindset there of it, but what do I do on Monday morning? What should I do to get started?*

Shirly: Listen to Pink, and read one article, and get coached because mindset... Mindset is a way of thinking. I don't want to give political examples, but there are a lot of political examples. Let's say; there are countries that people are raped in, women. And when a minister says, boys are just boys, when you are listening to it, you say, "What!?" How can you say that? This 'what' means that you are living in a different mindset. Changing a mindset is not only something that you read one time. It's a process of understanding your reality and giving a different perspective on your reality, so you can change the rules and the way people think. So, it's a blend of reading and a lot of practices. This is the reason I believe an organization needs an Agile coach, because when I come to an organization and I see a specific behavior, I change the way people think about this behavior and then I change the way they feel, and then I change the mindset. This is kind of a reaction. When

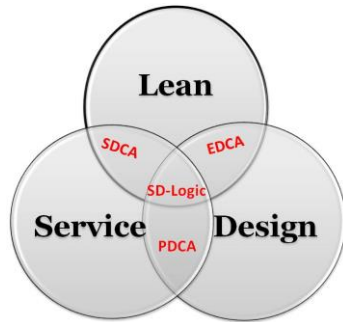
you change the way people see the reality, you change the way they feel, and you change the way they behave.

Joe: *What is upcoming for Shirly and the best way to contact you?*

Shirly: The best way to contact me is Twitter or LinkedIn. This is the places I like to hangout, especially Twitter. I love Twitter or read a lot on Twitter. As for what is coming, I have two new boys. They were just born a few months ago, so I will not write any new books this year. I will be busy taking care of my boys. And you'll hear a few inventions that my business partner Tony and I has added we'll probably publish them during the years.

Joe: *Well, I would like to thank you very much, Shirly. I appreciate your time and everything. This podcast would be available on the Business901 iTunes store and the Business901 Website. Thanks everyone for listening.*

Shirly: Thank you very much.



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](https://twitter.com/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.