



AGILE / DEVOPS

At this 01.12.23 Roundtable 55, NOREX Select Members from Fortune / Forbes 1000 organizations discussed future organizational growth of Agile / DevOps; organizational changes necessary to ensure DevOps success; DevOps success with development lifecycle outside of SDLC and projects with a hybrid approach; implementation of DevOps in organizational silos; setting up DevOps teams and assigning responsibilities; security requirements in the DevOps process – integrated or considered +1; economies of scale; lessons learned adopting a DevOps culture; and transitioning from Waterfall projects to Agile.

EXECUTIVE SUMMARY

On the topic of organizational changes necessary to make DevOps successful, a Member believed many changes are needed as you are moving away from a traditional project management mindset into a completely different type of work world. Everything is broken up to the point where there is more efficiency. You now identify the obstacles quicker so you can move forward and get the right resources. You start thinking and talking about Scrum Masters and Agile boards and how to map all the information to be able to change the reports that you create so that the senior leadership still understands the status and progress. A Director of Infrastructure Systems stated your Organizational Change Management plan needs to go beyond OCM and into individual transition management. How do you help individuals go through the grieving process of whatever they have to give up moving to something new? Secondly, you have to have a definition of what DevOps or Agile mean to the organization and recognize that there are many different flavors of Agile. Sometimes that recognition is what you do not want to move to Agile.

Discussing how companies are setting up DevOps teams and assigning responsibilities, a Principal IT Solutions Architect stated their DevOps team is responsible for creating the pipeline specifically for QA and staging training and production environments. When ready for release, DevOps has the responsibility to go through the pipeline, run it, and approve it. They also take on troubleshooting and look at the server in the event of a pipeline failure or system issue. A Manager of Solution Engineering shared they have a Platform Services Team residing in the infrastructure department that is responsible for the enterprise. This platform team is broad across the enterprise while the Solution Engineering Team is narrow and focused on the applications that serve their development teams and business customers. A Software Development Supervisor has his team reside on an AppDev organization. Their job is to ensure that the Dev teams can push their code at any time without any impediment. As far as team setup and what you are looking for between Dev or Ops, they have a combination of both. Everybody started off primarily on the Dev side to have some knowledge space within the infrastructure aspect of things. They are learning that they have to have a true mix of people that have that Dev experience versus those that have that infrastructure networking experience. When you have the right combination, it builds a strong team with the right mix of both the Dev and Ops.

Additional headline topics:

- Successful DevOps with any kind of development lifecycle outside of SDLC.
- Economies of scale when using DevOps approaches.

TABLE OF CONTENTS

Definition and the future growth of Agile and DevOps at each company	3
Organizational changes necessary to make DevOps successful	4
Can DevOps be successful with any development lifecycle outside of SDLC?	7
Setting up DevOps teams and defining the team responsibility	8
Security and the DevOps process.....	9
Economies of scale when using DevOps approaches	10
Managing longer, larger projects.....	11
Tools in use.....	12
Scrum Master turnover	13
Products/Vendors/Technologies shared in this Roundtable 55:.....	14

This transcript is from a videoconference. It may contain misspellings and grammatical errors. To preserve privacy, names have been abbreviated and organization names have been deleted. NOREX retains the unedited version in order to facilitate future networking. For networking assistance, please contact your NOREX Member Success Manager.

© Copyright by NOREX, Inc., 5505 Cottonwood Lane, Prior Lake, MN 55372. The opinions expressed in this document / recording are those of NOREX Members, not necessarily those of NOREX, Inc. This document / recording is for NOREX promotional purposes and for use by NOREX Members only. Unauthorized use or distribution to non-NOREX Members is strictly prohibited.

NOREX Select Roundtable 55 Transcript
Agile / DevOps
January 12, 2023

TOPIC: Definition and the future growth of Agile and DevOps at each company

Agile is the methodology to develop and deliver while DevOps describes how to continuously deploy code through the use of modern tools and automated processes.

Moderator: What is the future growth of Agile and DevOps at your company? Where are you at right now? Are you almost fully engaged in a more Agile DevOps type environment, or more hybrid in how you do things? We invite you to share what the future growth in Agile / DevOps is at your organization or if you're completely all the way there and basically that's your dedicated environment. Anyone want to start us off?

Buzz W.: We're fully engaged. We have been for at least 5 or 6 years. We have not only gotten out of our lines up from a DevOps point of view, but a couple of years ago we actually started moving it into other areas of the business. First one was within our infrastructure environment, trying to follow Agile processes. We use Jira boards pretty much exclusively across the company now for all types of work that's done. We cross pollinate the boards between different organizations. It's worked out very well for us, even to the point where we think about DevOps and Agile initiatives. We're using that in our audit. Our internal audit is now using Agile. It's tremendously helped the effectiveness and efficiency of the audits. Actually, one of our Directors, Clarissa, has spoken several times at many of the conferences. She's got a book coming out on it, actually about Agile and the audit practices from it, and she learned it from the DevOps world. It definitely is working really well for us.

Moderator: That's interesting, Buzz. Thank you so much. When you say audit, is that literally outside of IT?

Buzz W.: Yeah outside of IT. When they built this process a lot of work with my team, my organization to come up with it. But they do audits and sprints now, and they track it differently, so you don't get caught up with okay, we're going to do this audit. We're going to need all of this information, and then we're going to need it by this date. It's broken down much more effective, much easier on the participants of the audit. It gives you a good comfort level about how things are moving and progressing, instead of we're being audited and then 4 months later you go oh look, we had all these issues. Now you're having an opportunity to find out sooner, remediate quicker. The whole DevOps Agile environment has been a big improvement for us.

Moderator: Excellent! Do you know what her book will be? You said it's going to be published soon?

Buzz W.: I can find out it. It's supposed to come out in May. I'll see if I can find out. I'll post it in the chat.

The name of the book is *Beyond Agile Auditing* by Clarissa Lucas. Here is a link to pre-release information: <https://itrevolution.com/articles/beyond-agile-auditing-an-introduction/>
GREAT point! Yes.

Moderator: Thank you, Buzz. Were you going to speak up, Troy?

Troy T.: I was just going to say we are in many ways kicking and screaming and struggling. We've been trying to do it for probably 5 years, maybe even more than that. We struggle against the typical project Waterfall versus Agile formatting stuff. I think yeah, it's a struggle. We're a local municipal government body, and so we kind of move slower than most organizations. We're making good progress. But it never seems fast enough for all of us.

Jocelyn G.: I kind of agree with everything that Troy said. Everything we do here is kind of our own version of it. We struggled a bit as part of the Agile part. The DevOps though, I will say we started out of that journey probably two years ago, and on automating our code deployments and stuff like that. That is really where we we're still in our infancy. But that's really something that we see that's going to be very helpful for us. We really enjoy that.

Moderator: Thank you for taking in the discussion today. We hope you get some good ideas. That's what we're all about. Anyone else want to jump in on future growth of Agile DevOps and / or the status at their company?

Matt D.: Hi! Good morning. We've been using Agile and DevOps in certain pockets, particularly around our consumer marketing and our digital marketing websites for a number of years. We tend to kind of move forward and then take a step back and move forward and take a step back. I think the biggest challenge that I'd be looking for inside this group today is how you manage long term project planning and resource allocations and break that down into short sprints. We traditionally follow Waterfall, but what we found some challenges is how do you map an annual planning cycle with an Agile DevOps mindset?

Moderator: Let's add that to the list, Matt. That's great. Thank you. Any other comments before we get going on more of our specific topics? Appreciate these warm-up comments.

TOPIC: Organizational changes necessary to make DevOps successful

Moderator: What, if any, are the organizational changes necessary to make DevOps successful? I put a few in here that you may or may not agree with in this next portion. Would folks agree these bullets listed here potentially are with preparing for that culture shift, etc.?

- Prepare for a culture shift
- Create a continuous integration and continuous deliver platform
- Create a continuous testing environment
- Establish a continuous deployment system
- Make use of blue / green deployment
- Continuously monitor performance

Barry R.: Most of the items that are listed are the technical side, not the organizational change side. I think the first one, prepare for cultural shift, I think the biggest thing is that it is a mind shift. There's a couple of attributes of associated with the mind shift is your executive team. Does your executive team believe in failure? Do they support essentially quick fails? Quick learning? Does your organization support that? Are you from an organization that believes in perfection before deployment, for instance? There's a couple of attributes, and those are just two. I mean there's a long list of things that are different in the Agile mindset, but I think the cultural shift is really kind of what the question has to do with. You could argue that the question also talks about organizational structure change, and I'm not a huge advocate of change in organizational structure specifically for DevOps, unless you have no experience with Agile. If you're not an Agile shop at all, then you may

have to do some team realignment or team structuring that you don't necessarily have that you wouldn't have to do if you're already there specifically for DevOps.

Moderator: And where are you at right now? Could you give us a little highlight of where you're at with your DevOps?

Barry R.: We are a legacy shop in so many ways. We're actively undergoing some major monetization issues where we're modernization, our manufacturing shops play it technology. We're modernizing our distribution technology, and we're modernizing our back end technologies. As such we are just beginning the DevOps journey. We are on a dog and pony show about what DevOps looks like. On the Agile side, we have a couple of teams that are operating in an Agile scrum methodology. We have nobody operating in an extreme programming XP methodology, or anything more advanced. It is just such a large, cultural shift for us that it is a very slow cultural change here.

Moderator: Thank you, Barry. Would anyone else like to chime in on ways to successfully prepare from a cultural perspective, what worked for you and your company to embrace it? Noel, you've been in this environment for quite a while.

Noel S.: Yeah, good memory. I've been with this organization for nine years, and we've been an Agile shop since I started. It certainly has matured over the years. We got into the DevOps world probably about five years ago. As it relates to this I think it really had to do a true assessment of what DevOps means within your organization in and of itself. You know we jumped on to the whole DevOps bandwagon. But DevOps by its true definition, if you look at Google and those large organizations that you have people embedded in development teams that we're doing what they need to do to help with getting those environments spun up for you automatically and help with those code pushes. We weren't like that. We have a separate team here where we are creating those pipelines and that that Cloud infrastructure for the teams to build and push their changes through. Really, by definition, it wasn't really DevOps itself. We've popped over more on the SRE side. Site Reliability Engineering is what we're doing. But the big thing and my manager hit me up with a few months ago and got my thoughts on it's called platform management, and that's really what our team does. We're providing essentially a service to our internal customers, giving them the pipelines and the infrastructure that they need to be able to do their jobs. It takes that overhead off of them, because development teams inherently don't necessarily want to take on that overhead to manage the infrastructure, whether it be on-prem or in the Cloud, or worry about the pipelines. They want to get their code moved from dev stage and show production as quickly as possible, let alone having to worry about the build and releases in and of themselves and the performance of those. That's what we're at in that in that realm. Yeah, it started off as the DevOps buzzword but we've kind of projected from that and more into hey, we're providing a platform for the team to use.

Moderator: Noel, thank you. And what is your role or your title?

Noel S.: I'm a Manager of that SRE team.

Moderator: Great, thank you. And if you were to answer a question that Matt had posted a moment ago, I'll get to that in a moment.

Buzz W.: The answer is lots right there. There are lots of changes. I mean you're really moving away from a traditional project management mindset into a completely different type of work world. Everything is broken up in the little chunks, and even in your work breakdown structure when you used to think about maybe you had something that would go two weeks. Well, that two weeks might be broken down now between 3 or 4 different people, and each one of them have little tasks that they

have to complete. It breaks it down to a point where it's more efficient, because you're finding out where some of the roadblocks and obstacles are quicker, so you can move forward and get the right resources. You start thinking and talking about Scrum Masters and Agile boards and how you are going to map all of this information to be able to change the reports that you create so that the senior leadership still understands status and progress of what's going on. I know the question earlier from Matt was, how do you think about budgeting and that type of deal? It's pretty much the same way. You look at it as what's the project going to cost you to do? How much effort, how much time? Just the way you break down that effort and time is different and how you manage those efforts are different. It gives you the opportunity to do a lot of parallel work, or it is especially good when you're doing offshore work, because you could say, the onshore group's going to do this much today. Here's the tickets so that your cards that you had to do offshore. It allows that type of collaboration and move forward. I'll tell you one of the biggest org challenges is probably the naming conventions. It's a whole new lexicon when you start moving into DevOps and different words that you use, different phrases, they mean different things right? I always used to kid when they first start off we're going to have this sprint. It's like, okay, where's the starting line? Where are we going to line up? What are we doing? But it's this type of things that are different. The biggest change from an org process to is to make sure that you've got senior leadership engaged. Because there's going to be a lot of pushback. This is a change. People don't want to have to do things differently. Why do we have to do this? This has always worked before because they can't see it. And the first thing is, people are trying to micromanage me, and they're trying to figure out what I'm doing every minute of every day. And a lot of times lean practices get tied in with Agile because they work very well together. Lean is always all about reducing and eliminating waste. And some tie that to well, we're going to get better at how we're doing things. And now I may not have a job. That's actually the furthest from the truth. It's all about how you communicate it. What's your OCM plan around getting this new process out and in front of everybody? Those are just some of the challenges we had when we first started.

Moderator: Thank you. Barry?

Barry R.: Just to build on what Buzz articulated very well, your OCM plan needs to be go beyond OCM. Not just Organizational Change Management, but individual transition management. Because you will have an organization plan, says, how do I move my group and the transition management side of a move individuals? How do I help them go through the grieving process of whatever is that have to give up to move to this something new? That's the first part. The second part I just wanted to build on something that was said by a couple of people. Sometimes it's not, I think, Noel referenced it. You have to have a definition of what DevOps means to you, or what Agile means to you and recognize that there's so many different flavors of Agile. You have to have that recognition, but sometimes the recognition is what you do not want to move to Agile. For instance, we do it here, but I've seen other places where they move service requests into an Agile process. And that means that what should take 30 minutes, 5 minutes, an hour to deliver now moves into an Agile process as a story, even though it's a standard service request. As a result, you end up waiting weeks as the customer to get something that should be delivered in hours or days. You have to have a good scope and what you're trying to move in the actual process, and what you should be excluding and provide me an alternate mechanism.

Moderator: Thank you, Barry. I'm going to keep us moving along here, and I think some of these points will be shared again as we continue into the questions here. Elsa is not here but had this question.

TOPIC: Can DevOps be successful with any development lifecycle outside of SDLC?

Moderator: Can DevOps be successful with any kind of development lifecycle outside of SDLC? What about projects with the hybrid approach, the Agile, Waterfall. Barry, you talked a little bit on the fact that you guys have both Waterfall and Agile. It's also interesting that any questions submitted today have all been more DevOps, so it made me feel that of course a lot of our larger companies have embraced a quite a bit of Agile in general. Can it be successful with any kind of development lifecycle outside of SDLC?

Noel S.: I think it comes down to how people are defining DevOps, and are they separating DevOps from Agile? How you go about working within the teams and the culture and the management of that work and the DevOps. For me, as it relates to NFM is how are we getting that completed work to production. It's going from either you do a monolithic push, or you're doing these pushes on a more frequent basis with smaller code releases that are going out to production. My answer to that question would be I don't see why you couldn't be practicing DevOps and doing those small code releases with any type of approach, whether it be Agile or Waterfall. We're Agile so we'll do things in sprints. Teams or dev teams are releasing code throughout their iteration. But if someone was a Waterfall methodology, I would assume that they would just have in their project plans, specific checkpoints. Would they be releasing to the various environments? You could do that. It's just probably not making the most of the powerful capabilities that you have available to you. If you're truly doing DevOps. If you're doing Waterfall, essentially you're going to take everything and wait for one designated date. All right, we're going to push everything we completed up this point in time to dev. Or pick an environment, and then you've got to go back and try to test everything all at once. Whereas if you've got DevOps you can do these builds and releases on a regular cadence. Why not push those smaller releases of code out there so it's easier to test and isolate if there is an issue all those isolated to this particular release, whereas if you're doing one release every month or so it's like all right, what piece of the code is it that really isn't working?

Barry R.: Noel raised a good point. But sometimes it's a hybrid. He's specifically asked the question about hybrid approach. Maybe you have an Agile methodology, but you do monthly or quarterly releases because you have an impact on a community, stores, plants or whatever. You can still do DevOps, because DevOps is about the delivery mechanism and the integration on how you handle that stuff. To the point Noel made, I don't see that it's a barrier. What your build mechanisms are or build processes are, where there's Agile, Scrum, Agile XP, whatever it is. I don't see that DevOps becomes a barrier there. But, in fact, it's actually very advantageous if you are building in a "Wagile" environment and releasing in maybe a Waterfall environment.

Moderator: Wagile, I like that! Thanks. I'll keep us moving. Organizational silos can impede the success of implementing DevOps. How do organizations tackle this kind of issue? The silos between development and being a developer and being more operational, infrastructure, that sort of example. Let's go to Matt.

Matt D.: One of the things that we've had some opportunity to do is to normal change management like what's in it for them? How can we help minimize support issues because we automated the resiliency of some of our applications? And another thing that it's kind of a journey we're embarking on now, but it's building out templated solutions to make the right thing the easy thing and help decrease that barrier to entry. Because we've had to do a lot of rescaling around infrastructure as code and deployment pipelines and using modern code repositories like get for the deployments versus the traditional way of doing things. We're not quite seeing the organizational adoption that we're hoping for but it's early on. Most of our work has been grassroots versus executive down. And so that's another challenge that we're navigating through.

Moderator: Thanks, Matt. I think, Buzz, you had your hand raised at one point. I don't know if you have more to interject here.

Buzz W.: Yeah, I think the point that has been brought up has been excellent from a standpoint of I refer to Agile as DevOps, and really Agile is a framework of how you accomplish it. And so the answer to hybrid is absolutely. What we found and why we've expanded the Agile framework outside of just software development is because it makes it easier to work with all the different entities that we might have to interface with. It gets them on the same cadence. It aligns the two parties that says, okay, we're going to be doing this. You have to have this done by this time. Let me track your two-week sprint and how are you doing. Both can work. It's just that with anything, consistency is important. If you can get a consistent framework that you're using across every place it just makes it easier. It doesn't mean you can't, just if you only have to manage it one way that's a lot more beneficial. But yeah, I was remiss in not being clearer. Agile is just an end to the means. It's not truly the DevOps mindset.

TOPIC: Setting up DevOps teams and defining the team responsibility

Moderator: Our next question is great for this discussion. How are companies setting up those DevOps teams and what are those teams responsible for? Maybe we start with things like where do you find those skill sets on your teams? Or is there one area that seems to become a little more successful in finding or developing the skill sets for more DevOps? If you're pulling from Ops, or pulling from the development side or other. What are the DevOps teams responsible for?

Anas H.: I can explain a bit about my previous experiences in my previous job because they are more organized and they already started integrating DevOps and they are working. We have the DevOps team that is responsible for creating the pipeline specifically for QA or staging training and production environments. The developers they can do whatever on their environment. But what is like a higher than their environment DevOps team? What will be their responsibility? Their responsibility also to do like the deployment on a weekly basis. Once we are ready for the release, so DevOps will take the responsibility to go through the pipeline, run it, and approve it. Also, the DevOps team is more responsible also on troubleshooting. They are like in middle here. They also take a look at the server and see what's going on in case the pipeline failed, or the system is having any issue. And to your question, Dev or Ops, I think it's more like Ops. They are not like from development background. Those guys more like operations.

Matt D.: I can share what we're trying to see if it resonates with any other organizations. We have a Platform Services Team, which sounds similar to what Noel mentioned earlier, that resides in the infrastructure department is responsible for the enterprise. Then we have a smaller unit that's focused on a specific business area, and if the platform team is broad across the enterprise, the team that I lead is a Solution Engineering Team, which is narrow and deep, focused on the applications that serve our development teams and our business customers. We try and focus in on developer experience. How do we make it as easy and as fast as we possibly can for developers deliver high quality code. We know our applications, but we go to that platform services team for the standards, the governance, the controls, the high-level architecture recommendations. It's kind of just a different focus area for us.

Moderator: Great point, Matt.

Noel S.: I'll just piggyback off of what Matt was saying there. Honestly, my team resides on an AppDev organization. To your point, our job is to ensure that the dev teams can push their code. Our goal is to make sure they can push their code at any time they want to without any impediment. So that's why, as far as setting up DevOps teams it comes down to defining that. Are our folks going after what's out there in the books that are published by those large tech giants and trying to mirror what they do? Or you trying to do DevOps with what it works best within your organization? What worked best for us was not having those DevOps engineers on Dev teams per se. Essentially kind of goes back to the previous point. We were a bit silent. We were a unique team that that really was just there to help support those dev teams. So again, it comes down to organizational needs and how they define it, and what those needs are. And as far as the team setting up, and what you're looking for between Dev or Ops, we have a combination of both. Everybody started off primarily on the Dev side to just have the propensity to have some knowledge space within the infrastructure aspect of things. But we're also learning that we have to have a true mix of people that have that Dev experience versus those that have that infrastructure networking experience. When you have that right combination, it really adds to a really strong team, provided that's the way you go. Now, if you go to have it embedded within the Dev it's a little bit different. But for what works here that's kind of the route we're going to, trying to find that right mix of a combination of both the Dev and Ops.

TOPIC: Security and the DevOps process

Moderator: Thank you, Noel. Let's throw security into the mix here as well. How are security requirements considered in the DevOps process? Is it always integrated throughout the DevOps process, or is it considered a kind of a +1 with the Dev setup?

Noel S.: We're the +1 model. We definitely have a network security team and that's their responsibility. They work with our team hand in hand as far as hey, can you add these bits and pieces into your pipelines for monitoring a learning perspective from security aspect? That's how we work. Certainly, you have to have a security mindset within the team in and of itself to ensure that you're not injecting vulnerabilities into what you're doing. But ultimately, there's enough to manage without having to worry about the security aspects of it. It's very comforting knowing that we can refer back to a team that's truly dedicated to that.

Moderator: And are they considered DevSecOps or they more security team and they are engaged with what you're doing?

Noel S.: They're more a security team engaged with what we're doing. We have corporate infrastructure as well, so they work hand-in-hand with various parts of the organization. But they're separate for sure.

Moderator: Great. Thank you, Noel.

Buzz W.: We're kind of a +1. We have within our information risk management organization teams that work directly with our DevOps groups to make sure we're getting different tools implemented to ensure secure coding practices. Like we recently wrote a contrast, and some others that catch things prior to them going into production so that you don't have to go back and try and fix. There's a number of other tools that that we're implementing right now, especially as you get into the Cloud arena there's different parameters. It is kind of a +1, but it's getting to where it's blurring. We're actually having discussions now of perhaps implementing a BISO role for all of our lines of businesses, a Business Information Security Officer that would have a better understanding of all the best practices. If you look at a BISO, a BISO understands for that organization how all of their apps work, what platforms, etc. Kind of a combination of an architect and a business leader and a security

officer that can balance all of those pieces and keep help keep the environment safe. I don't know how far we'll go with that. We kind of do that now. We call it our front office within information risk management. We have individuals that are aligned to certain size of applications to where they help with those business delivery leaders in the DevOps arena. But we're seeing that due to the velocity of it, the number of them, we may need to have people over those people to make sure we're keeping those organizations and syncing in the line. If we decide to go with the BISO role I'll talk about that more maybe in another time. But right now, our front office kind of handles that piece.

TOPIC: Economies of scale when using DevOps approaches

Moderator: What economies of scale have companies seen using DevOps approaches and in what ways? Are we thinking of those as sort of what are the benefits you're getting from it somewhat? Things like cost reduction, speed, quality, performance, customer satisfaction? Go ahead, Matt.

Matt D.: Thanks. One of the things that we've noticed is because once you get the infrastructure as code and you have your deployment pipelines built out in a repository that's reusable, it does tend to make it easy to share that with other groups. You have similar, like ETL based solutions or CRUD (Create, Read, Update, Delete) application solutions. The copy / paste idea is real. The challenges that we face, though, are that when we approach to that nonlinear innovation on a project, like we have a new use case for a new business area where we have to build out something new, we're teaching people infrastructure as code, we're teaching how to use Get application, then that has caused some project delays that we've had to be intentional about getting ahead of to make sure expectations are appropriate or on delivery timelines when those challenges occur. I guess that would be my watch out is doing it the first time is going to take longer, but then doing it the next time is much faster. Then the more you use it, that economy of scale really resonates over time.

Moderator: Thank you, Matt. Barry?

Barry R.: The other thing I think, is the value of the ISC not economies of scale, but cost management. If I allow my development community to do whatever they want, then I tend to be oversubscribed on Cloud services. So infrastructure in Cloud allows me to essentially publish t-shirt sizes. It allows me to publish enablers that they will use because it's easy. It makes their developer experience very simple. But it also allows me to direct them to resources that are cost-effective except on an exception basis. It prevents some of that sprawl and out-of-control costs. It's not a scale answer. It's more of a cost management answer.

Buzz W.: I really like what Matt had to say. It definitely improves interoperability between all of the applications. It helps with that build once, use many mentality, because a lot of times you'll see you got three lines trying to build an interface to it to the same database. It's like now you get this, and you post it. It also helps from the standpoint of API development, which again helps with interoperability between not only your internal applications but external applications when you need to. There are some great economies that you can pick up from it. But the ramp up, I mean those are some pieces If I go back to the organization, it's like, okay. Where's your class on scrum? And where is your classes on Get? And where is your classes on API development, and there's a whole list of new types of things you want to think about. You're also then getting everybody developing with the same tools. That's the other piece that this shows and helps out it's like, Well, I like to use Java. Well, I use Python, or I like to use this, or I use this. You don't eliminate it, but you greatly reduce that type of sprawl that comes from well, I want to use the thing I'm used to when someone new comes in as a developer. It's like here's our toolset we use. It's that type of thing. Over time that momentum really gets going forward. In tasks or projects that used to take maybe 3 or 4 months can be reduced to maybe 3 or 4 weeks, because you can start picking pieces and putting them all together and just

change a few things. And now it all works, and the out product of it also works with everything else, because you've been consistent in what you're doing.

Moderator: Definitely saw Matt nodding in agreement, too. Architecture and process patterns. Thanks, Barry in the chat. And Buzz, thank you for that link to the pre-release of that book you mentioned from a member of your team.

TOPIC: Managing longer, larger projects

Moderator: Let's go to the question Matt mentioned, just to make sure we covered it enough for you. How to manage those longer, larger projects, those longer timeframes, and how you separate those into maybe shorter or sprint portions. You explain it, Matt.

Matt D.: Oh, sure, thanks. We're used to a year long, 2-year long projects where we might have a planning phase that's 3-6 months and then a build phase and the kind of go through those separate phases. And I think we've been a little bit challenged on some of those more ambiguous or changing projects that change over time. But we still get into that habit of everybody's used to saying, okay in April we're going to say what we're going to do from May until April the next year. We kind of go through those annual planning cycles. As I've had those discussions, it's generally been about you kind of want to say this is our resource allocation to this project and we're going to commit to that. And you get as much work as you can done in that timeframe. But that doesn't always satisfy leadership. They're like no, I don't want as much as you can get done. I want X, and I want it done in in production and working well. I wasn't sure if anybody had any experience kind of maybe riding that paradox between needing to give teams room to innovate and do something new to accomplish goals as well as still meet the overall project goals. But know that you might not get exactly what you thought you were going to get at the beginning at the end. But that's really what you needed anyway. You just didn't know it then. Maybe it's trust or something along those lines.

Barry R.: I think there's a couple of different parts in that question. One is the nature of do they understand what Agile is? And it sounds like the answer is no, because you're being asked to follow a Waterfall methodology for planning and probably budgeting would be my guess, that doesn't match. What I've done in the past is two things. Number one is we still have initiative definitions and project definitions, but they're looser. You know it's not the same three-month planning cycle where you figured out it's going to be \$3,404,000.30, right? You just don't go to that level of detail. Instead, it's a budget allocation. What is your velocity of spend? And then the understanding of progressive elaboration, which is the whole point of Agile. I progressively elaborate on what that initiative is, as I learn more and walk further into the environment and I focus my resource on things that are more valuable. I have the right and desire to abandon or redirect that investment, because as I progressively elaborated it, I've discovered it's not what I want to pursue. The alternative to that is, I spend three months in the planning cycle, three months in the preparation cycle, or six months in the preparation cycle, I haven't delivered anything, and because of that lack of delivery I haven't actually learned the reality is of the value of what I'm trying to deliver. I think it's about the education, and then I come back to budget the velocity. Don't budget the project. That's how I've worked it in my past lives.

Moderator: Thank you Barry. And Buzz mentions that it enables failing fast and pivoting to the right solution.

Troy T.: I would say you need to find a Waterfall Project Manager who's willing to work with you in Agile. And you also need to adapt your Agile to somehow understand and work with a Waterfall project. I have not figured out how to do an 18-to-24-month project that can't go live in increments

outside a Waterfall methodology. Now we can integrate our daily work processes or our Waterfall work into an Agile format, sprints and hitting bugs those kinds of things. But we're still trying to figure that out and try to see that. I think it's more difficult when you're talking an enterprise-wide system that's used my 200-400 people to say I need this specific process to be able to work for me to be able to do my job and support my customer. Well, I can't go to this system and do this, and this system and do this, this system and do this iteratively. But I see Buzz has raised his hand. Maybe he can tell me how to do that. It's a struggle.

Buzz W.: My suggestion, one of the lessons learned, I think biggest was start off with a good tracking and reporting application or methodology. We bounced all around. We looked at this one. We looked at that one. We tried a bunch of different ones. We've settled on Jira because Jira was pretty much built for this type of work, the reporting in it, the graphs that come out, the ability to track tickets across different boards. I mean, it just has all the functionality. And it's relatively inexpensive. It works really well. That would be one of the biggest things. Once you get that nailed down, and then start with one. Just pick somebody and do it, and get them trained to be a Scrum Master. There's lots and lots of places out there that offer that training now. Usually if you look at somebody, if you have any type of Lean organization with within your enterprise, tapping with someone in the in that area to say, how'd you like to become a Scrum Master? And we're thinking about developing this project because they get that whole idea about eliminating waste and that side of the house. But once they see it work, it's like if you build it, they will come. Usually does, because they're seeing people save the money. At the end of the year your budget you might say well, how come all of a sudden, we have \$50 million left over when you told me you were going to spend it? You usually don't run into that problem when you're doing DevOps because you're spending the money, and you're pivoting. And okay, we're going to stop this idea. We're not going to buy that. I mean it's a lot more definitive. And you'll see you'll be surprised how much scope creep goes away because that's another iteration now. It's not like we're going to add it to the project, so well why didn't the project get done well? Because we kept adding all these things. Well, that's another iteration. We do this, and then we move on. I mean, those are some of the other benefits from it.

Barry R.: so. I just want to add a lot of the big failures I've seen is where they try and replicate the Waterfall bureaucracy. They try and take gates, that's one example. Or to Buzz's point, we send them to a great Agile Scrum, Agile XB, or Agile whatever class, and they come back, and they think you have to do it exactly this way. How many times have we seen PMPs come in, implement project management and they're rigorous in their discipline. But they don't recognize that the framework should be a cafeteria that they select based on procedures that they select based on the needs of the business. My advice is to examine the process. Don't bring in the additional bureaucracy from either the Waterfall side or the Agile side. Bring in the pieces that you need to be successful. Learn what those are. Do those really well before you introduce additional bureaucratic pieces.

TOPIC: Tools in use

Moderator: Great lessons learned, great tips here. We'll take the next couple of minutes for what you want to talk about. We had a couple of tools brought up. Troy mentions Trello TFS, and now I'll move to ADO environment. I think Get has been brought up, and I'm not quite sure if those things do the same. And Jira was brought up. But are there any tools or anything else someone would like to bring up before we end our discussion?

Anas H.: For the people who are already familiar with DevOps and they spend time on DevOps, if there is any add-ons you guys you added specifically for security reasons, or for scanning for any vulnerability or anything you think is something we should have. We started like using SonarQube. But if you guys have anything, any suggestions, you can just add a comment or something.

Jocelyn G.: Yes, that's correct. We use a SonarQube for our code analysis, more from the quality assurance side.

Anas H.: But if you guys have anything to share with us, any other something that you think is important or it gives like a value, or it gives maybe more security on the DevOps side, please share.

Moderator: Sure. Buzz mentions Contrast is being used at Buzz's organization. Anything you might shy away from, too? It's always good to hear the good and the experiences that you decided to move from something else. I know that Troy had mentioned a few different moves there, and what? Why they were done.

TOPIC: Scrum Master turnover

Troy T.: I can't remember what we use before Trello. But we've tried all the tools. We've chewed up and spit out multiple Scrum Masters. It seems that they're a very fluid bunch. They like to bounce around from one contract to another one every 9 months, so it's hard to get established and settle on one. But yeah, the tool doesn't matter. It's the process that you're using. We actually had a room where we painted whiteboard material on the wall, and when we first started doing Agile we did Post-It Notes across the board. The tool doesn't matter.

Noel S.: Certainly, haven't gone virtual. I mean Post-It Notes have gone away. But we had them all over the place and the building. But you find different tools. It's whatever. People threw out Jira, we use Azure DevOps for our visualization of the work. Troy brought up something there that kind of just made me shudder a little bit. We said that the Scrum Masters seem to be coming and going. It's certainly how organizations are set up. But we've benefited from that being a contractor type of a role and we've been able to establish longevity. When it comes to the technical work, having someone there that knows, that can help facilitate and manage that team, we have managed to work in a mountain that has that history with the team and remaining with that team. I couldn't imagine what it would be like, Troy as you said there, when you do have kind of that revolving Scrum Master in the organization. Everybody has a different flavor of Agile and how they practice it, and that's hard on the Dev teams to adjust to that, and it's hard on the Scrum Master to adjust to Dev teams accordingly. Just the stability in that, I think, would benefit folks for sure.

Troy T.: After our last Scrum Master, I said to our Project Management Office I don't want any more external people. I went out and I got mine. Recently I have what I said, one of my employees, and so now he got his Scrum Master certification, functions as a Scrum Master. No more external contractors.

Noel S.: And the thing with that, too, is having that as a dedicated role. It used to be an organization where it was just different hats that were that and helped facilitate that. But we then transitioned over to having it as a truly dedicated role. We also have dedicated product owners as well, rather than having it be a dual role, and that certainly has helped with things within our organization.

Moderator: Great points on the Scrum Master's role and having it be a dedicated position within the company. Thank you so much for participating in today's discussion. We'll talk to you again soon.

Anas H.: Thank you.

Buzz W.: Thanks. Everyone.

CHAT:

Buzz W.: The name of the book is *Beyond Agile Auditing* by Clarissa Lucas. Here is a link to prerelease information: <https://itrevolution.com/articles/beyond-agile-auditing-an-introduction/> GREAT point! Yes.

Barry R.: Architecture and process patterns.

Buzz W.: It enables failing FAST and PIVOTING to the right solution. Start with one project, make it work and then grow from there.

Barry R.: Don't replicate the waterfall bureaucracy

Troy T.: We've bounced. Trello, TFS, and now moving to ADO environment.

Buzz W.: You have to do what works best for your enterprise.

Barry R.: Pivotal Tracker, Jira.

Buzz W.: We use that also and Contrast.

Noel S.: Thanks for the good discussion.

End of discussion

Products/Vendors/Technologies shared in this Roundtable 55:

ADO	ADO
Agile XP	API development
Audit	BISO
Business Information Security Officer	Contrast
Create Read Update Delete	ETL. CRUD
Get	Hybrid
ISC cost	Jira
OCM	Organizational Change Management
Pivotal Tracker	Scrum
Scrum Master	SDLC
Site Reliability Engineering	SonarQube
Sprint	SRE
TFS	Trello