Agile Operations and the Three Ways

Insights from DevOps Experts including Gene Kim

ca technologies

DevOps and Agile Operations:
Insights from Industry Experts

Moderator

Kieran Taylor has 20 years of high-tech product marketing experience with a focus on application performance management, cloud computing, content delivery networking and wide area network technologies.

He is presently Senior Director of Product and Solutions Marketing at CA Technologies and responsible for thought leadership and sales enablement for APM and CA solutions that help enterprises implement DevOps methodologies.

Prior he led product marketing teams at Adobe, Akamai, DataPower/IBM and Nortel Networks. His career began as an editor of high-tech publications at Mc-Graw Hill.

Nearly any DevOps discussion will highlight the transformation development teams undergo as they evolve from the sequential design of waterfall methods to adopt the iterative, agile approach of scrum methods. It's no wonder as it is dramatic, exciting and delivers impressive results. However, what is mentioned less often, but is equally dramatic, is the shift IT Operations teams must negotiate. Traditionally known for predictable stability, it could seem a misnomer to suggest that there is a growing set of practices known as 'Agile Operations'. That is until one considers how cloud computing, microservices, containerized applications and the like are changing the very definition of infrastructure. This new, agile infrastructure and associated concepts, such as planned failure, instill—if not require—collaboration between dev and ops.

Sounds great on paper, but for an IT Operations team traditionally measured, tooled and even compensated on maintaining resilience at the expense of change, how can the essential Operations discipline adapt—without exposing the business to increased risk from technical fragility?

Whatever our technical heritage, today's absolute reality is that digital business success necessitates more agile development and faster release cycles—meaning operational requirements, knowledge, skills, processes and tools must now be established earlier in the software development lifecycle. This demands a new approach. It demands Agile Operations.

Agile Operations no longer means keeping the technology lights-on, it means making the business shine brighter; working closely with Development by applying systems-style thinking to visualize and speed a constant flow of value to customers. It also means adopting automation, tools and methods to amplify the cross-functional feedback so necessary to improve quality and the customer experience. Finally, it's about future-proofing the business—managing technology at scale, yes, but never burdening the organization with additional cost, waste and complexity.

To better understand why everyone—including IT Operations—needs to care about DevOps, we sat down with four DevOps luminaries and thought leaders to get their take. Their deep insights make a compelling case for Agile Operations and provides immediate practical guidance on how to get started. We hope you enjoy the discussion.

The Experts

Gene Kim

is a multiple-award-winning CTO, researcher and author. He is the founder and former CTO of Tripwire. He has written three books, including coauthoring The Phoenix Project.

Benjamin Wootton

is the co-founder and Principal Consultant at Contino, a UK based consultancy helping organizations adopt DevOps and Continuous Delivery tools, practices and approaches. He has worked with over 30 organizations on DevOps and Continuous Delivery transformation initiatives, helping enterprises deliver better software, earlier and more often. Prior to this, Benjamin built over a decade of experience as a software engineer, team lead, architect and agile transformation coach.

Steve Thair

co-founded DevOpsGuys in 2013 after a 20 plus year career in IT infrastructure and operations. Steve simplifies the management of online applications by leveraging DevOps practices, improving the delivery, performance and time-to-market of the entire software development pipeline. Steve blogs extensively on DevOps at and has presented at numerous meetups, webinars and conference to evangelize the benefits of DevOps. He also started the London Web Performance meetup and launched WebPerfDays London in conjunction with VelocityConf to help "speed up the web." He has been on the organizing committee for the Web Performance track for Velocity EU for the past three years.

Matthew Skelton

has been building, deploying, and operating commercial software systems since 1998. Co-founder and Principal Consultant at Skelton Thatcher Consulting, he specializes in helping organizations adopt and sustain good practices for building and operating software systems: Continuous Delivery, DevOps, aspects of ITIL and software operability. Matthew founded and leads the 1000-member London Continuous Delivery meet-up group and instigated the first conference in Europe dedicated to Continuous Delivery, the PIPELINE Conference. He also co-facilitates the popular Experience DevOps workshop series and is co-editor of "Build Quality In," a book of Continuous Delivery and DevOps experience reports.

Question 1

Putting DevOps “unicorns” aside and looking at more traditional enterprise “horses,” do you foresee (or see) a change in the culture of operations teams?

The stereotype of IT operations is that of stability to the point of being risk adverse. Teams maintain largely static infrastructure, mandate onerous security and compliance edicts and are generally less flexible.

For those companies that do adopt agile development and deployment, how will the profile change of the people hired to lead and run these teams?

Question 1: ANSWERS

Wootton

"The people who thrive in their software development or IT Operations careers over the next five years will be those 'silo breakers.'"

I think this perception of IT operations as a risk-averse barrier to change is sometimes a little overplayed. IT operations have always been trying to bring valid operability concerns to the table and usually would work with development to get change successfully shipped in the end. If the stereotype is as bad as it is sometimes made out to be, these organizations would have gone out of business.

What has often happened is that IT operations were involved too late in the cycle and not given a seat at the table to begin with, so it felt as though they were putting on the brakes or pushing back on initiatives late in the day. Better and earlier engagement with IT operations could have avoided this situation and led to better service transition all along.

Fortunately, I am seeing this dynamic between IT operations and the rest of IT change. The need for reduced cycle times means that the relationship simply has to become more collaborative to meet the speed and quality standards required. Trends such as cloud and infrastructure as code are also pulling operations concerns left in the delivery lifecycle. From the other side, developers also need to take more interest in their full stack, and the production environment, as their application stacks become more complex. All of these trends are naturally changing the culture of IT operations teams and the way they interact with development.

Alongside the rest of IT, I think the skills and character profiles of people within IT operations have to evolve away from deep specialization and siloed thinking into a generalist approach—collaborating across the software development lifecycle (SDLC) to optimize the full delivery cycle rather than their local responsibilities. Neither world class software development nor IT operations counts for anything alone—the two need to come together. The people who thrive in their software development or IT operations careers over the next five years will be those "silo breakers" who optimize the whole.

Question 1: ANSWERS

Thair

"The culture of IT Operations has to change to being less risk-adverse."

The culture of IT operations has to change to being less risk-adverse because businesses have to change to be less risk-adverse. In 1960 the average lifespan of a company was 60 years. It's now 20. The time you have to grow and succeed is compressed, hence you need to take more risks in order to achieve your goals.

So the culture inside IT operations has to widen its focus on "how can we help the organization achieve its goals." We, as IT operations people, can't stay siloed in our thinking, our objectives and our culture. There is no point in having a Web application platform with five nine's availability if it's nearly impossible to deliver change into that environment, and it's even more pointless if your company goes bust because it can't keep up with the competition.

Luckily, I think that we are at a very exciting time in IT operations because we have amazing new toolsets that enable us to reduce risk, improve compliance AND simultaneously offer a better service to our organization by reducing the cycle times for change. It's a win-win.

Question 1: ANSWERS

Skelton

"...the Agile Ops leader will provide a risk-aware 'change enablement' based on measurement and metrics."

The Agile Ops leader for a 21st century organization will understand the need to balance security and stability with release, deployment and the development of new features, relating these to business needs. In particular, they will distinguish different risk profiles for different software applications, avoiding a "one-size-fits-all" approach for changes.

In fact, rather than "change management," the Agile Ops leader will provide a risk-aware "change enablement" based on measurement and metrics.

Question 1: ANSWERS

Kim

"The person leading the DevOps transformation is having to overcome deeply entrenched command and control bureaucracies and low trust cultures."

What makes me so excited is the demonstration that DevOps principles are being applied in large, complex organizations, often with decades of legacy applications and infrastructure. They're showing that DevOps principles transcend technology and apply to almost all types of organizations reliant on technology to help the organization win.

I've reviewed the practices of organizations such as Disney, Nordstrom, CSG, Target, Blackboard, Raytheon and US Citizen and Immigration Services. In my opinion, the stories being told are as heroic and awe inspiring as any presentation you'd hear from Google, Amazon, Twitter and Etsy.

Of particular interest to me is that the person leading the DevOps transformation is having to overcome deeply entrenched command and control bureaucracies and low trust cultures.[1]

What I've learned is that there is a higher pinnacle which is for technology to be a part of the team working together with everyone in the value stream (for example, development, test, operations, information security and so on) to help the organization win in the marketplace. To achieve that, we need to change the culture, especially in IT operations, from one of viewing ourselves as "order takers"—only performing the work that is requested—to one of being full team members, working together to achieve the organizational objectives.[2]

(1) Source - http://www.infoq.com/articles/interview-gene-kim-devops-enterprise
(2) Source - http://rewrite.ca.com/us/articles/devops/itil-and-devops-better-together.aspx

Question 2

The book “The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win” introduces the three ways of DevOps thinking. The **“First Way”** emphasizes the value of the entire system, with a focus on the flow of business value enabled by the DevOps processes, procedures and practices.

With IT traditionally organized via technology function, what changes do you view as necessary to support this more “lean” and systems way of thinking?

Question 2: ANSWERS

Wootton

"Effective collaboration, at the heart of DevOps, is how we achieve better and leaner handovers."

As you said, IT has traditionally been organized around technologies or areas of the architecture, e.g. a front-end, a middle tier and a database team. We are seeing a lot of businesses reorganize into cross functional teams of developers, testers and operations engineers who are aligned with particular products or business capabilities. This cross functional collaboration reduces the handovers, increases the ownership and enables the team to deliver software faster with more responsiveness to the customer. This is the way that leading digital businesses such as Amazon and Netflix tend to operate, and the case for making this change is often compelling.

Beyond this, I tend to find that the key to improving cycle time is often about targeting the handovers between development, test and operations. It's one thing to work more efficiently within a phase, but lean thinking shows that the information loss and the delays and overhead in the handovers are where the waste lies. Effective collaboration, at the heart of DevOps, is how we achieve better and leaner handovers.

Automation is also key to optimizing flow from left to right. Automated builds, integration, testing and deployment will accelerate delivery, increase quality and make the software delivery process more efficient. Of course, automation isn't the whole story, but we often see it as low hanging fruit for reducing cycle time without requiring significant disruptive changes to the ways of working.

Stepping back a little, I think IT generally has had a bias towards providing rigor, stability, robustness and predictability—at the expense of speed and agility. I think we need to move the needle to reprioritize speed, hopefully without throwing the baby out with the bathwater and reducing quality at the same time. Only by doing this will we meet the growing demands being placed on IT functions today.

Question 2: ANSWERS

Thair

"Many of us, sadly, still talk about 'IT' and 'The Business' like they are two separate organizations."

There are two main things that block "flow" inside traditional IT organizations—silos (the organizational structure) and the project-based finance model. Both of these have to change before you can really start to deliver the business value benefits of DevOps.

The first silo is that many of us, sadly, still talk about "IT" and "the business" like they are two separate organizations. I suspect that distinction arose very early on, back in the 50s, when IT people literally walked around in white coats working on room-sized computers. Computers were mysterious, esoteric and, to many people, frightening. They didn't understand them so they wanted to keep them at an arm's length, and the "High Priests of IT" wanted to promote that mystique because it gave them prestige.

But now computers are ubiquitous. The CEO of GE, Jeff Immelt, said it perfectly, "If you went to bed last night as an industrial company, you're going to wake up today as a software company." We need to stop thinking of IT as separate from "The business." We don't think of sales or marketing or finance as being separate from "The business," so why use that language?

The second silos are the ones we created for ourselves by basing our organizational structure on technology function, generally because of our search for "efficiencies" and "economies of scale."

We've now learned that those silos destroyed productivity, collaboration and flow in a way that more than offsets the "economies of scale." The (many, many) slow, formal handovers between silos create so much friction in the process that it's amazing we get anything done. It sometimes feels like 40 percent of an organization spends its time stopping the other 40 percent getting anything done, leaving 20 percent of us to actually do the work.

Continued

"In short, fund teams, not projects."

Thair

"If your employees can't link what they do to a 'bigger picture,' they switch off."

Silos also have an impact on staff morale and motivation. The French historian Alexis de Tocqueville said, "Nothing tends to materialize man, and to deprive his work of the faintest trace of mind, more than extreme division of labor." Economist Adam Smith warned that workers "become ignorant and insular as their working lives are confined to a single repetitive task."

If your employees can't link what they do to a "bigger picture," they switch off. This is why we feel that multi-disciplinary DevOps teams, organized around core customer products or services, are the solution. Teams that cut across the business/IT divide, as well as across the silos within IT, enable people to link their work to the success of the product—to the delivery of value for the business as a whole, and its customers.

The second blocker to flow is the project-based funding model. The project-based model has three fundamental flaws. Firstly, in most organizations in order to justify "the project," requirements need to be defined up front, costs and durations estimated and ROI "guesstimated." This results in the lumpy flow of work through the system, as these large units of work block the flow through the SDLC. Secondly, these estimates, durations, plans, etc. soon become their own raison d'etre. You start delivering "the plan" and not the business value.

James Smith, co-founder of @DevOpsGuys, summed it up perfectly after a visit to one customer site when he said, "I see a lot of PLANS on the wall, but not a lot of WORK." Lots of walls are covered with Gantt charts showing tracking against plans, but little visibility of the "work in progress." Who is doing what, now, and how is it delivering value to the organization is what's important. The first step in any lean (or DevOps) transformation is making the work visible, and understanding what work is adding value and what work is just pointless paper shuffling.

Thirdly, projects are, by definition, temporary. You build a team, incur the overheads of getting that team working smoothly and then you throw it away to start all over again. It's an incredibly wasteful cycle. But if you focus on building stable, product based teams, optimized for flow, where members can see the fruits of their labors (a feedback loop) then you can reduce the waste inherent in the project model.

In short, fund teams, not projects, and feed work into those teams in bite-sized chunks following lean principles that are prioritized by expected business benefit.

Question 2: ANSWERS

Skelton

"Arrange teams and their responsibilities to match the flow of product and service changes."

The most important organizational change for effective DevOps is to arrange teams and their responsibilities to match the flow of product and service changes such as "service teams" or "product-aligned teams."

This in turn drives changes in software and systems architecture to decouple the software of one team from that of a second team; aside from some key coordinating data stores (an artefact repository, a message queue, log aggregation, metrics), almost all the infrastructure for one team is separate in some way from that of other teams. Taking account of Conway's Law, software from different teams is not mixed onto the same machine or even group of machines, but isolated by team in order to clarify system boundaries.

Question 2: ANSWERS

Kim

"The organizational culture leads to what gets delivered."

Conway's Law states that the organization dictates how work is done because it produces designs that copy the organization's communication structure. So if the organization says we're going to do things in a waterfall way, then we're going to have a development department who hands the code off to a test department, who hands it off to the operations department. You end up with very large releases combined with low ability to find and fix defects quickly which leads to horrendous deployment outcomes. The organizational culture leads to what gets delivered.

In DevOps organizations you end up with much smaller teams, where typically you will have dev and ops working side by side, and if not integrated into the same team, at least working far more closely together, collaborating to not just facilitate getting the feature done but making sure that it's safely and continuously integrated and deployed into the production environment. You end up with a different looking organization.

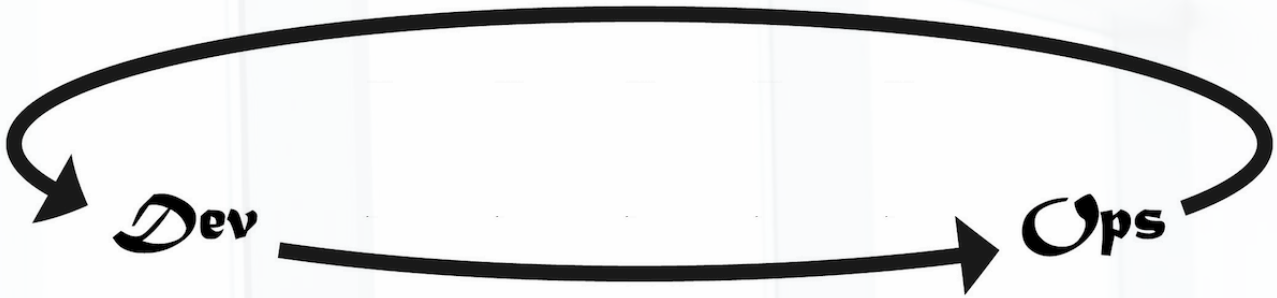
And a very real question related to that point is, "What does the next generation of centralized operations look like in large, complex organizations?" One model is this notion of banding operations engineers together in the same group as the feature or service team—which is great because now the operations people have the same goals as the service team. Another model is that operations is less in the business of doing work that comes out of the ticketing systems. Instead they're building the automated tools that allow other people to do their work in self-service mode. The concrete example is this notion of a shared group that's all about increasing development productivity. How do we make environments available on-demand to any feature team or service team who wants to use one, and what are the deployment mechanisms they can use to get into the production environment? Basically, make the easiest route for any development team to become productive and become more reliable.

https://enterprisersproject.com/article/2015/4/conversation-gene-kim-devops-waterfall-development-and-containers

Question 3

The **"Second Way"** suggests that organizations "amplify feedback loops" and, in particular, right to left feedback from ops to dev. What process changes are necessary to create or optimize these feedback loops?

The Second Way: Feedback



What in particular can IT operations teams do differently to facilitate right to left feedback?

Question 3: ANSWERS

Wootton

"IT operations should have a voice and a path to production for their changes."

I have often encountered IT operations teams who don't appear to have (or know of) any process for getting their requirements on the book of work. In the past they may have tried, but nothing seemed to happen, so sometimes they have given up even trying to get their voice heard. This is a sorry state of affairs as IT operations are closest to the system and the users. They should definitely have a voice and a path to production for their changes. Simply putting this process into place is a good and actionable first step in achieving this.

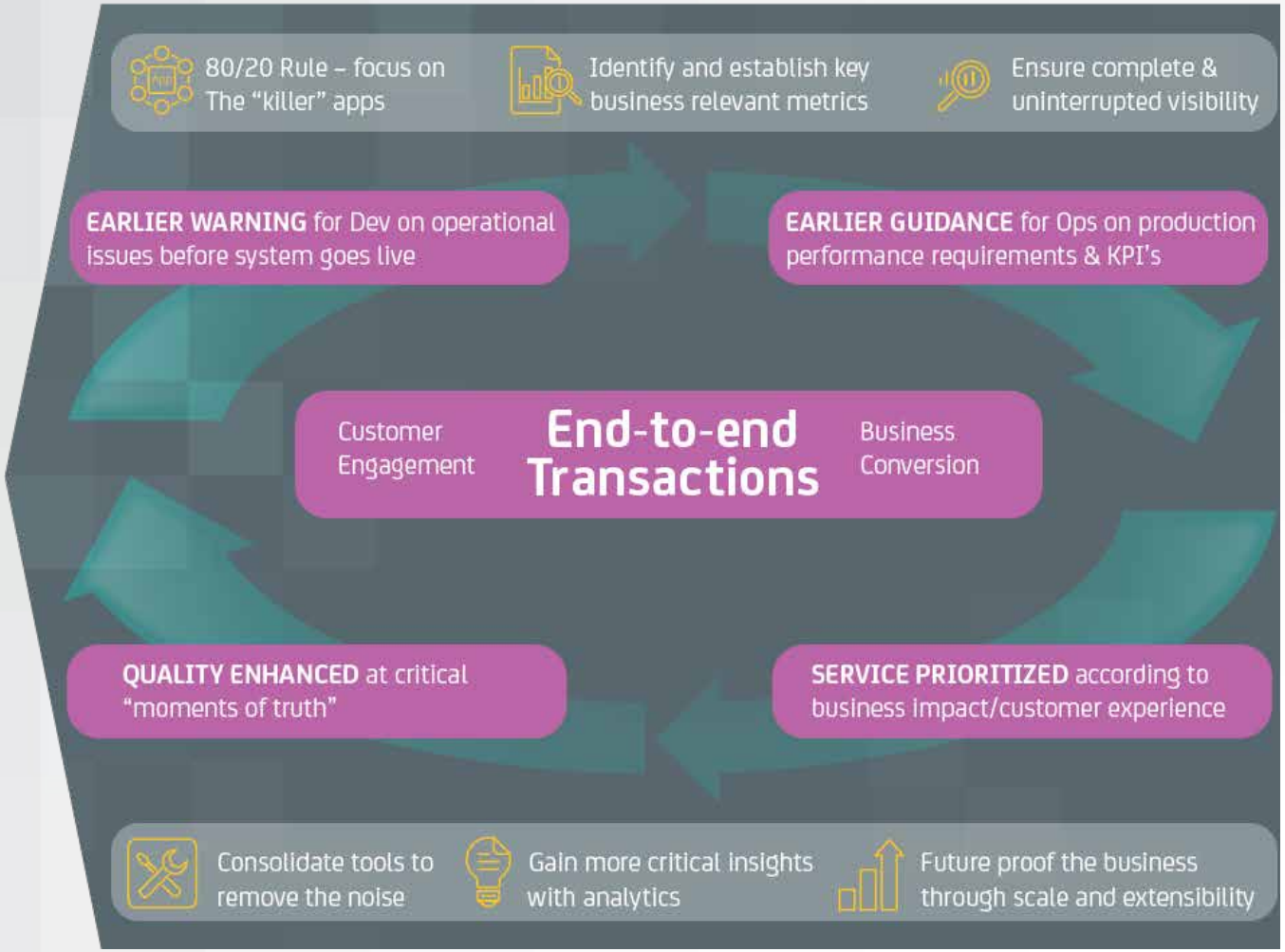
Where there is some loose process for getting operations requirements and feedback from right to left, I also see that IT operations concerns are sometimes lost in translation or not given high enough priority on the product backlog against new feature requests. New features constantly take priority as they have an immediate impact on revenue, while IT operations and nonfunctional requirements are deprioritized. The result of this is that, over time, the system degrades, technical debt creeps in and working practices within IT operations fall behind industry best practice.

As with all of the functional siloes, I also think IT operations has its own proactive role to play in improving collaboration across the software development lifecycle. Bring people into their process. Explain what you are working on and why, and publicize your metrics, your activities, your challenges and your wins. IT operations should reach out to the rest of the business to show what their role is all about. This will allow other engineers to build empathy with what IT operations does. Transparency wins if you want to get your voice represented.

Wootton

There is also a very specific example which we encourage to drive "right-to-left feedback," specifically around monitoring. Often, people have an enterprise class monitoring solution deployed only into production. We encourage our clients to bring the same monitoring solution back into test and development environments. This means that code is developed that is easier to monitor and that monitoring is tested before the application is promoted to production. As a result, real operations concerns are picked up earlier in the lifecycle and everyone becomes aligned around these. This is simple but effective.

Applying a "Shift Left" Monitoring Strategy



Question 3: ANSWERS

Thair

"Ensure that technical operations staff are involved in the product design from day one."

This one's actually very straightforward, from my (Ops) perspective:

1. Make sure that the "operational requirements" (formerly known as "nonfunctional requirements") are clearly articulated in the design phase and not treated as second-class citizens compared to the functional (a.k.a "business") requirements. Note that that language of separation between the business (requirements) and IT (requirements) crops up here yet again. IT is part of "The business," and the operational requirements are what are required to manage the product safely and securely in production. They are not something separate from, nor ancillary to, the delivery of the finished solution.

2. The easiest way to achieve number one above is to break down the silo between dev and ops and ensure that technical operations staff are involved in the product design from day one—and then keep them constantly involved in sprint planning, stand-ups, retrospectives and all the other phases of (scrum-based) agile delivery. This provides a constant right to left feedback loop from ops to dev.

3. Share the measurements. The Culture-Automation-Lean-Measurement-Sharing (C.A.L.M.S) model of DevOps highlights the important of measurement and metrics to track our improvement. But very often IT operations has a nasty habit of hiding this data away in tools and systems that only they can access. We need to share this data better (sharing being another key pillar of C.A.L.M.S). Give development access to that monitoring system, APM suite or log analysis tool; create dashboards; set up radiator screens; set up an API, etc. Give them access to the information and then go over to them and have a conversation about what it all means, what the key metrics are, and what direction you'd like them to trend. Give them the information they need to make better decisions and improve your software.

Question 3: ANSWERS

Skelton

"IT operations groups need to become more adept at articulating operational requirements in ways that are helpful to development teams."

IT operations groups need to become more adept at articulating operational requirements in ways that are helpful to development teams: least-privilege security, logging as a first-class concern, "tracer bullets" for critical code performance and diagnosability for instance. Agile Ops teams also see the value in agile/lean techniques such as retrospectives, attending stand-ups, limiting work in progress (WIP) and tracking throughput.

Question 3: ANSWERS

Kim

"The shorter the time, the better the outcomes."

One of the best predictors in manufacturing and DevOps is lead time (in terms of quality, custom satisfaction and employee happiness)—how quickly can you go from code committed to test it up, test run and successfully running in production. In general, the shorter the time, the better the outcomes.

One of the best things in terms of how operations can help before we hit production, is production monitoring before production, with the notion of realistic performance testing in a preproduction environment.

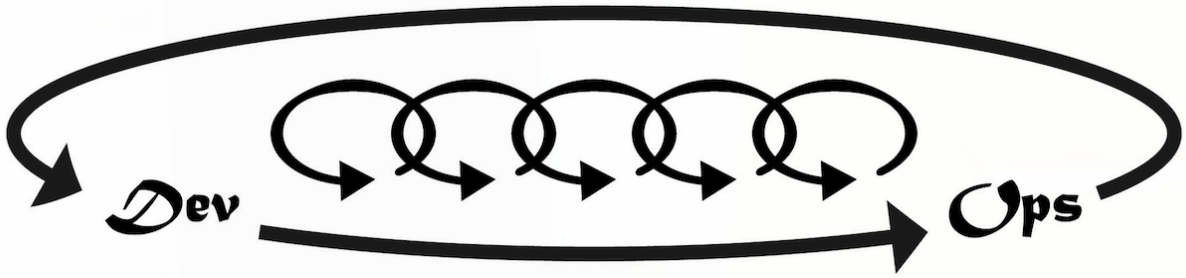
John Allspaw once said to me that if you take two teams—one just a bunch of developers and ops working in isolation and the second team working together where dev and ops are working towards a common objective, towards a common design requirement—the second team will outperform the first in every scenario.

https://channel9.msdn.com/Shows/Edge/Edge-Show-118-Gene-Kim-DevOps-Interview#time=11m54s
https://channel9.msdn.com/Shows/Edge/Edge-Show-118-Gene-Kim-DevOps-Interview#time=14m03s

Question 4

The **"Third Way"** introduces continual experimentation with a goal of ongoing optimization. Depending on the team, experimentation may be discouraged by operations or only found within development teams.

The Third Way: Continual Experimentation



Should IT operations teams be embracing the "Third Way" and what can/should they be doing differently?

Question 4: ANSWERS

Wootton

"An IT department is like a car—it needs maintenance and servicing if it is to keep running at maximum effectiveness."

The whole organization needs to adopt the "Third Way," not just IT operations, with more experimental in their approach, more innovation, more ongoing optimization of working processes. In today's modern, competitive, digital, software-driven world, it is essential to do this to stay up to date with industry best practices across IT. If not, you will very quickly fall behind your competition who are entering the market with no technology or process legacy.

An implication of this is that the entire IT function needs the time, space and resources to optimize their platforms, their tools and their ways of working. IT are often under sustained pressure by "the business" to ship new features, but an IT department is like a car—it needs maintenance and servicing if it is to keep running at maximum effectiveness. Smart organizations and leaders will accept this and invest in the platform and ways of working as well as the visible application stack.

IT operations has to be driving this activity as much as development. They are under pressure with more infrastructure to support, more complex application stacks and more complex change coming over the wall much more frequently. If they don't continue innovating and evolving, their job will become harder and harder till they eventually become a barrier to innovation. IT operations needs strong leadership making this clear to their stakeholders and then shouting from the rooftops when they do deliver value with these initiatives.

Question 4: ANSWERS

Thair

"IT operations needs to start embracing experimentation."

Traditionally the cost of experimentation has been high—you needed to provision infrastructure to run those experiments and the cost of deploying the experiments (and rolling it back) was high due the manual labor involved. But virtualization, cloud and DevOps automation tools have totally changed that equation. We can now rapidly and easily provision infrastructure and deploy applications at the click of a button, and we only have to pay for what we use on a consumption based cloud model.

Want to know how your application and monitoring systems perform when you have 200 front-end webservers? Spin them up in the cloud, simulate some load, do your experiment and shut it all down again. The total cost is virtually nothing compared to running that same experiment 10 years ago.

We're also starting to see some interesting stuff happening in the database virtualization space (instant database clones with data de-duplication so you can rapidly provision new test database instances complete with high-quality test data) and in service virtualization (so you can easily virtualize third party back-end web services to use in your experiments) which further improves the economics. We also have a vast array of new tools for measuring the application and underlying platform during experiments.

IT operations needs to embrace these new economics and start embracing experimentation. In his presentation at QCON London 2014, Dave Farley (co-author of the book "Continuous Delivery") likened this to the application of the scientific method. We need to start testing our opinions while using facts and hard data instead of opinions and guesses.

Question 4: ANSWERS

Skelton

"A culture of learning and guided experimentation is essential."

A culture of learning and guided experimentation within IT operations teams is essential if they are to provide real value to organizations. A strong sense of driving continual service improvement should inform priorities: the IT operations team sees itself as building and improving key parts of the software systems. The experimentation and improvement therefore needs to be joined up with those of the development teams.

Question 4: ANSWERS

Kim

"Continuous amplification of creating a culture of safety, experimentation and learning."

The "Third Way" is really continuous amplification of creating a culture of safety, experimentation and learning. And in the DevOps space, I think the best embodiment of that philosophy comes from Adrian Cockcroft. He's famous for many things, but, most recently, he's famous for his work at Netflix where they transform the Netflix architecture from a J2EE wrap running in data centers to running entirely in the cloud. And he said, "Our goal is to do painful things more frequently so that we can make then less painful in the future. So, even though we make life hell for developers sometimes, the answer that we get back from them is: Thank you. Thank you so much because we know that doing this will make life go far more smoothly in the future."

And the best evidence of the effectiveness of this philosophy and practice, in my mind, was the first Amazon EC2 outage that happened in 2011. So, for anyone using EC2 at the time, you will remember this day because everybody went down. Everybody went down except for one curious exception which was Netflix. So for weeks, everybody was asking, "What is Netflix doing so differently that resulted is such a different outcome for them?" And the answer was revealed some weeks later when they put out the seminal blog post that essentially exposed two things. The first was this decision they made very early on that said, "We can never depend upon Amazon for availability." In other words, Amazon will never be there when we need them most. The second was they decided that to survive failure, they were going to need to fail all the time. And that's when they unveiled this thing that we all now know as "Chaos Monkey."

As we can see, the impact of Agile Operations is significant. So let's get started.

Hear from industry experts and your peers on how adopting Agile Operations methods can improve the speed and scale of application deployment while ensuring high quality end user experiences.

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact, and communicate – across mobile, private, and public cloud, distributed and mainframe environments. Learn more at **ca.com**.

CS200-140656_0715

ca technologies