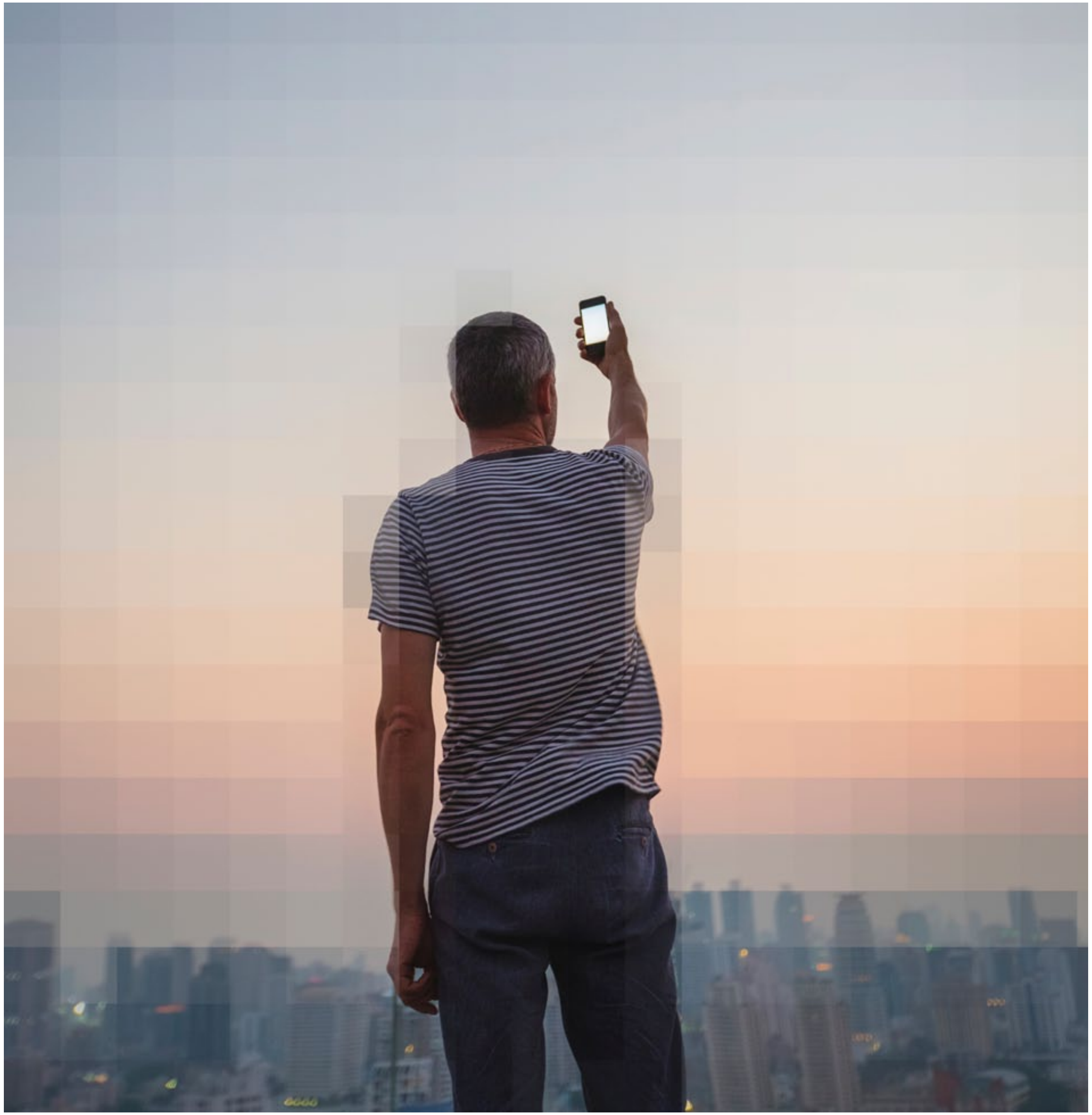
ca technologies

DevOps Perspectives 2

Ideas, insight and direction from DevOps leaders

CONTENTS



DevOps has a problem. In many ways it has become a victim of its own success, and the industry might already be accused of 'over-talking' it in much the same way as Cloud hype. It's like we've been waiting for some new unification paradigm to span the two sister worlds of software application development and operations ... and now it's here, and now we need to get to grips with it fast—but we're still drinking from a firehose of innovation.

This need to grasp the DevOps opportunity is an immediate imperative for all technologists if we accept that we are in the middle of what we now call the application economy – a place where customers are far more likely to experience your brand and interact with your enterprise through a software application than a living person. According to the results of the most recent CA Technologies study* on the application economy and the role of DevOps, 88 percent of 1,425 IT and Line of Business (LoB) executives already have or plan to adopt DevOps sometime within the next five years.

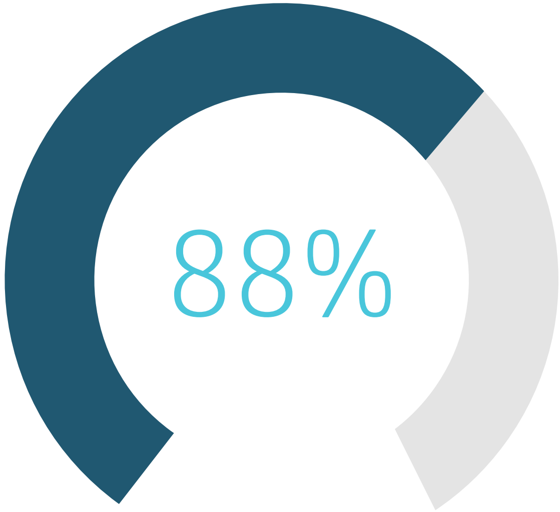
Should it really have been called OpsDev anyway?

But there are questions still to be answered. Should it really have been called OpsDev anyway? Is the balance in the very DNA of DevOps imperfect in some ways? Is DevOps evolving with enough LoB relevance to ensure productive implementation in all environments? Do we even understand that DevOps is a workplace culture rather than some packaged product that might have come shipped in a DVD-ROM box were we still back in the 1990s?

Finding ourselves inside this maelstrom of discussion is healthy in some ways; i.e. these are the teething pains that any new and emerging technology should go through on its way to adolescence and beyond. But long-term confusion will be unhealthy and disruptive forces will turn negative, as stated in CA's paper 'DevOps: The Worst-Kept Secret to Winning in the application economy'* – those experienced with DevOps are already seeing measurable results (15 to 21 percent improvements) in quality and performance of applications; enhanced end customer experiences; and cross-platform simultaneous software deployment.

Of course DevOps is already beyond its initial adolescence and is seen played out and fully executed by the types of firms we feature here in this eBook. What matters now are the more real-world mechanics and minutiae of the way we bring DevOps to bear in current working environments, which are so heavily driven by the need for Continuous Delivery and Continuous Integration.

> "Implementing **DevOps** properly will be a political, economic, technical and personal issue."



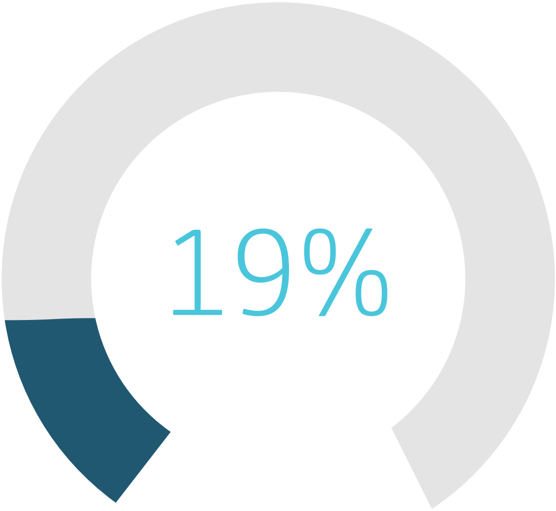
of all enterprises plan to adopt DevOps

*DevOps: The Worst-Kept Secret to Winning in the application economy, a study commissioned by CA and conducted by Vanson Bourne between July-August 2014

Mind your manners

As we now work to 'map' the scope of our own technology ecosystems onto DevOps-empowered workflows, we will need to step back and almost deconstruct each individual approach to software development and delivery. DevOps has to now eat at the same dinner table as lifecycle process management and collaboration tools and it will need to pick up some good manners very quickly if it doesn't want to come across as some raucous upstart.

Implementing DevOps properly will be a political, economic, technical and personal issue. Given this sensitivity (yet huge potential for success) it is crucial that we form working groups and discussion forums to do thing right. As many as half of all respondents in a Vanson Bourne survey for CA reported that their industry is being 'very' or 'highly' disrupted by the arrival of the application economy – but this can be a positive disruption if the opportunities are grasped with strategic insight. If DevOps has any problems, we are all stronger if we 'share with the group' and flesh these issues out now. Tomorrow should look brighter, continuously.



improvement in quality and performance

On enterprise **DevOps** transformation

Benjamin Wootton, Co-Founder and Principal Consultant of DevOps Consultancy, Contino assesses the place of DevOps in an enterprise culture.

DevOps started life among the 'born-on-the-web' companies, such as Facebook, Spotify and Google, a fact that has led some in the DevOps community and beyond to question whether it is suited to the more traditional large enterprise companies.

Many of those enterprises have been stung over the years by technology vendor hype, so if there is sometimes an underlying fear that those providers and consultants talking about Enterprise DevOps are just trying to repurpose and resell what is already there, that may be a forgivable assumption.

But having spent time looking at how to apply DevOps in an enterprise setting, my conclusion is that DevOps in an enterprise environment deserves its status as a niche within a niche and is in many ways a more potent tool in that environment.

This is reflected in the increased number of questions I find myself being asked by CxOs within large enterprises about enterprise DevOps transformation and how to achieve it.

When attempting a DevOps transformation within the enterprise we need to consider three things – people, process and technology. If all three are not in line, the initiative will likely not reach its full potential.

People

DevOps practitioners talk a lot about culture, but that is a hard thing to transform. How do you measure culture and how do we change and influence it? I think we need to get a lot more rigorous than that sort of question to achieve transformation of an enterprise towards DevOps ways of working.

One of the first things we need to consider is organisational design. Here we are interested in the departmental structures, the job roles and responsibilities, the reporting lines and the way people are incentivised and so on. This involves looking at the job roles within the enterprise, their function, what and how much you are asking people to do on a day-to-day basis and what rewards and incentives are being offered to achieve all this.

> “As an industry we do a bad job in bringing juniors into IT.”

This is all about getting everyone pointing in the same direction. The importance of this step means that you need to have a top-down approach with senior-level buy-in to make it work, but you also need to look at wherever responsibility sits in the organisation in order to drive this.

Then you need to think about your skills in the IT organisation. Specifically, do you have the necessary DevOps talent on the books and if not, where are you going to find them? Are you going to go out to the market to source them or will you ‘grow your own’ internally?

The latter takes time of course, but the former is not necessarily an easier option. It is frankly hard to find people with the requisite skill set. Industries and education have been set up around siloed thinking. If we're honest, as an industry we do a bad job in bringing juniors into IT. We need to look again at how we do that. It's a really common challenge that enterprises face, that they can't find the profiles they need out there in the market. Some enterprises I know have been trying to hire DevOps people for a long, long time.

It really is a candidate-driven market and I can't see that changing any time soon. The universities are still very 'old world' in their thinking and companies don't seem willing to invest in bringing new talent through. So what happens is that you end up with the senior engineers who are already there getting paid more and more and becoming more critical to the enterprise while the juniors aren't coming up behind them.

Assuming that you are going to try to recruit, you need to think in terms of incentives and culture. What needs to be changed or be in place not only to attract new recruits of the right calibre, but to retain them once they are on board. You don't want to train someone up, only to see that person disappear to a competitor that offers a more conducive work environment.

Remember that while financial recompense is clearly a factor, DevOps people also tend to be incentivised by delivery. They want to see what they're doing getting out there into the market and being used. They're looking for flexible working and they want to use the latest and greatest tools to do the job. You need to factor all this in before making someone an offer if you hope to lure them in.

Process

Here you need to consider the organisation and its business processes, ranging from requirements capture and prioritisation through software releases and operational processes such as change and incident management.

In a DevOps world, these processes should change so the knowledge and responsibilities sit with the right people. At the moment in most enterprises, there is a really process-driven approach to IT. There are people who do this, people who do that, it's all very process-centric.

People aren't robots. You need to move the culture towards one of collaboration and personal responsibility. These are intelligent and highly skilled people. You need to let them make their own decisions and to work with one another.

> "You need to move the culture towards one of collaboration and personal responsibility."

That typically means putting the right culture in place on the ground. Earlier I talked about the need for top-down support and how that is essential, but cultures also change from the ground up. Top-down, ground-up and hope they meet in the middle.

Of course there are always likely to be incumbent processes, particularly in larger, more established enterprises. Things have worked in a certain way for a long time and got the business, so how do you start changing these long-established working practices? There's human nature coming into play here as well. Even people who are change-friendly can become set in their ways.

We can see across industries that even the simplest of organisational changes can be difficult. This is DevOps, this is something complex that's touching people, processes and technology. The key message here: don't underestimate the challenge of process transformation.

Technology

The third element is technology and here we see a technology landscape within established enterprise organisations that is much more complex and complicated than that of a start-up. You have baggage. You've invested in CRM and ERP and SFA and HCM. You've got core systems on which your business has been built and you've layered more and more on top of those over the years.

Now you find yourself with a spaghetti-mound of systems with multiple integration points between them. There's a massive diversity of technologies, platforms and providers. And because of the way this stuff went in over the years, there's limited automation in place, you struggle to keep up with updates and new releases and frankly there's a backlog of investment needed that you probably can't afford.

> "Applications in production today are often very monolithic..."

So just as you need to transform the organisation and business processes, you need to transition your technology platforms. You need to look at new delivery models and technologies, such as cloud and virtualisation, to make IT easier to manage and to inject greater agility into the infrastructure.

But sadly I don't see people taking full advantage of the potential of, for example, virtualisation, where server provisioning still takes weeks if not months, environments are static and not dynamic or transient and not enough automation is in place to give consistent configuration-managed base images.

Applications in production today are often very monolithic and need to be broken up to make them easier to release, manage, maintain and change. Developers are frequently held back waiting on changes to servers by very siloed infrastructure teams when they should probably be owning more of the layers of that infrastructure themselves.

These situations all have to change for successful enterprise DevOps transformation.

Conclusion

I don't think we always help ourselves as DevOps professionals. Some of us are actively hostile to the idea of Enterprise DevOps and don't want DevOps escaping its ivory tower and entering the mainstream.

> "We need to embrace the fact that DevOps is growing in popularity..."

To my mind, we need to embrace the fact that DevOps is growing in popularity and help the organisations who are trying to do DevOps at enterprise scale by giving them a more detailed and rigorous roadmap for the transformation. We know that DevOps is about culture and empathy at its heart, but I think we need much more detail about how we move enterprise organisations from A to B in terms of concrete activities and process changes.

Enterprise DevOps is an underserved niche and working with it rather than against it will be key in growing DevOps adoption.

DevOps in an age of big data

This is the age of big data, Hadoop and Cassandra clusters, which provides organisations with more opportunity to mine and gather DevOps metrics than ever before. Opinion is divided on whether this is a good thing ...

While metrics are clearly essential building blocks of any DevOps strategy, the sheer availability of massive amounts of information can lead to the risk of curating too many data points, resulting in a kind of data blindness.

> "Useful metrics should either inform or validate."

"More is not better," says Robert Benefield, CTO, Evolve Beyond, bluntly. "Metrics and data that are not clearly useful create noise that obfuscates and slows down the delivery of those that are. When in doubt, don't collect it. Collecting data with no clear motive beyond a hope that it might someday be useful for something is not only wasteful but can bury or crowd out more immediately useful metrics in a sea of noise."

So how do you determine which data points are useful and which are not? Start by getting back to first principles, argues Benefield. "Useful metrics should either inform or validate," he states. "Where they inform, they indicate anomalies or hazardous trends that need to be actively investigated and resolved quickly." This would include such things as a flurry of unexpected application restarts, dropped sessions, or dangerously increasing resource utilisation, all of which are anomalous and require action to be taken.

"Where metrics validate, they are providing proof against a hypothesis or event," he adds. "This includes proving the validity of expected usage patterns and behaviours, as well as things such as metering for billing. The audience is typically trying to verify business value, confirm user experience and engagement patterns, or validate technical and architectural behaviours. Like useful informative metrics, validation metrics also determine a course of action, even if that action is 'stay on the current course'.

"When determining what metrics to gather, know what is being looked for, by whom and why. If you can't answer any one of those, or the answer does not indicate that there is anything actionable that the person consuming them can reasonably take, the chances are that the metric is either flawed or not sufficiently useful."

Not just a lot of data

In order for DevOps metrics to deliver validated and informative data, rather than just a lot of data, it's useful to question whether the five Vs of big data can be applied: volume, velocity, variety, value and veracity. The answer to this is yes, up to a point, according to Peter Matthews, Research Staff Member, CA Technologies Labs, who argues that the first three categories are easily resolved in a DevOps context.

"Variety depends on the scope of the metric data collection," he posits. "If you are just looking at change logs, error rates and time to resolution then the variety is low. However, if you are measuring inception to retirement of software, there is most likely to be a wide variety of data ranging from configurations, project plans, user stories etc.

"Volume depends on the size and scope, but it is easy to see large volumes of data being collected in busy IT shops. Finally, velocity is not in doubt when there is a major incident that is being tracked and managed."

That leaves two categories outstanding – value and veracity. The first principle here is that any metrics you gather need to have a value, but organisations mustn't fall into the trap of assuming that all metrics automatically carry value.

"Some metrics are of little value," suggests Matthews. "Terabytes of metrics telling you that the system is operating within normal parameters are likely to be needed during the early days of use when no-one is sure what is normal operation. After the system has stabilised, some of those metrics should not be kept, but discarded or ignored. It is important to establish the value of any measure, what it tells you in its own right and what it allows you to infer when allied to other measures."

> "It's useful to question whether the five Vs of big data can be applied: volume, velocity, variety, value and veracity."

When it comes to veracity of metrics, this is a familiar issue. "The accuracy of data is always a concern when analytics are applied," notes Matthews, who argues that a major criticism of big data is the view that all data has value if only we knew how to analyse it. DevOps metrics fall into that group, he notes.

"There are suggestions that data collection regarding DevOps metrics should discriminate in favour of data that brings value and that the collection of large swathes of data creates noise," he explains. "This is to ignore the output of techniques such as data triangulation. Data triangulation is the technique of using multiple, potentially related data sources to validate or create an unknown data point.

"It is taken from the concept of navigation and is much used in social sciences but can take advantage of all the data that is retained as part of a big data strategy to triangulate fixes for broken data. Once large volumes of data are used for improving DevOps metrics for analysis and prediction, the value of that data can be more easily established.

"More data gives the opportunity to discover new truths about the behaviour of the DevOps systems and predict problems and divergences before they occur. It is worth noting that a 'good enough data' approach can be complemented by a 'good enough' prediction to enable trends and patterns to be established more consistently."

Other considerations

Audience

It is important to be aware of the fact that there often is more than one audience who could benefit from the metrics gathered, states Benefield. “Metrics traditionally gathered by operations staff, such as faults, application and infrastructure errors and conditions, as well as resource utilisation can provide important insight to help development and testing efforts.

“Equally, code quality, build and testing metrics can improve operational understanding of potential risk hot spots to watch out for. Usage information and customer experience expectations can help guide design, configuration and maintenance decisions being made by both business as well as technical groups.”

Be careful how DevOps metrics end up being used, adds Matthew Skelton, Principal Consultant at Skelton Thatcher Consulting: “When collecting metrics in a DevOps context, we need to be very careful not to let those metrics be used in a way which induces negative behaviour in teams,” he says. “For instance, rating product teams based on the number of deployment failures or rewarding teams based on the speed of blue-green switchover would be counterproductive. Metrics should always be there to empower and support teams in improving their work, not to punish teams for ‘getting it wrong’.”

Timing

It is important how time sensitive the data is. “Service and infrastructure health metrics are typically far more time sensitive than analytics captured for strategic business intelligence,” argues Benefield. “Some need to be captured and relayed quickly to the right place with enough context that the right action can be taken, while others can be batched off and processed offline.”

That demands having the necessary skills in place to make the necessary choices around identifying what data is important and when. “The purpose and real meaning of metrics must be clear,” says Skelton. “It is important to have staff who really understand mathematical correlation and how to interpret data properly.”

Don’t ignore the service ecosystem

Another factor to consider is the scope of your service ecosystem. The more effective you are at authoritatively capturing, eliminating and preventing discrepancies from entering parts of your ecosystem, the more consistently representative and sophisticated your metrics can be.

“One of the most important and neglected factors that affects the efficacy of metrics is the level of hygiene present in the service ecosystem,” says Benefield. “Slight differences in configurations, handling and other environmental factors can obfuscate or interfere with the capture, delivery and understanding of metrics data.”

Nothing stays the same

The value of metrics is not a constant. “The usefulness of metrics waxes and wanes over time, requiring that metrics be regularly assessed for their value and actively pruned and managed,” concludes Benefield. “Friction with curation greatly increases the amount of effort required, reducing both the responsiveness and will for the organisation to change what is captured and tracked to adapt to changing conditions.”

Shadow IT: opportunity or threat to **DevOps**?

CIO Magazine's 2014 State of the CIO Survey results revealed four out of five IT leaders feel IT projects done without IT involvement create problems. CA Digital Transformation Lead, Justin Vaughan Brown, asks how DevOps should deal with the rising spectre of Shadow IT.

The velocity of application delivery cycles has increased massively over the past few years. This has placed pressure on development, test, release and operations teams, who face demands for more agile technology delivered more quickly and more responsively.

It also has an impact on Line of Business (LoB) heads, directly tasked with introducing new products and services to grow the bottom line. All too often those business users vent frustration on not having the 'right' IT to enable and support them, but which is instead blamed for holding them back.

This in turn has given rise to the spectre of Shadow IT. This can be defined as any team that builds or deploys a service using infrastructure or environments not approved by the IT department.

Or to put it in another way, it's LoB people going 'rogue' to address a business imperative because the traditional routes are too expensive, too complicated, too frustrating, but most of all, too slow.

Cloud, SaaS, PaaS, etc. have all made the options easy to order and pay for – and in a language the business understands. This is a generation used to swiping a credit card to get browser-based Business-to-Consumer application functionality. What's so different in the workplace?

Well, quite a bit of course in terms of governance, compliance, security, not to mention the dangers of multiple-track IT architecture strategies emerging.

How does DevOps approach this symptom of the age we work in? Does DevOps ignore it, welcome it or fight it?

Like many things in life, the situation isn't totally black and white. Overleaf in more detail are a few reasons why understandably Shadow IT causes some concerns for DevOps.

Dark shadows

Parallel worlds

When DevOps is trying to standardise the production line, Shadow IT creates another 'factory' that creates another silo of activity running parallel. Which one dominates?

Continuous Delivery transparency

How will the Shadow IT project map to other applications? What visibility is there of the relative release stage?

Security

What are the risks of API exposure to threats? Does the business understand the dangers of opening the gates to unforeseen attacks when it builds a third-party integration? (Answer: almost certainly not, why should they? That's your job.)

Governance

What reports are created that empirically demonstrate which team or individual approved deployment to the next environment, or most importantly into production? Certain industries such as financial services and pharmaceuticals have very strict regulations around the creation of approval or audit trail.

Reuse

The business got what it wanted and delivered the application, but what next? How will Shadow IT approach the next release? In the DevOps world, application development and deployment needs to be a repeatable process not a project. Shadow IT threatens this by looking at each requirement from a piecemeal, departmental and one-off viewpoint.

Light shadows

Equally, there are a few reasons why Shadow IT can also be a good thing for DevOps:

Casting light

The existence of Shadow IT reveals where current IT practices are not matching the desire velocity of the business. That sounds like the perfect moment for a 'Let me explain DevOps to you' conversation, and a chance to win parts of the business over to the cause.

DevOps project alert

Shadow IT can also highlight a potential testing ground for an initial DevOps project. Here you need to find a situation where the business has a compelling need and is willing to consider non-traditional delivery methods. Then, step up to the mark.

The Cloud

With so much of Shadow IT utilising the availability of cheaply and easily accessible Cloud-based applications and platforms, there is potential to explain how DevOps is geared up for fast delivery/provisioning in public, private and hybrid cloud environments.

Action plan

So perhaps the best course of action for anyone involved in a DevOps programme when confronted by Shadow IT is to:

» Understand what has led to traditional IT being bypassed and what the genuine concerns/grievances/frustrations of the business users are.

» Establish how (if possible) the requirement can be adopted into a current or future DevOps-centric project.

» Meet with the LoB heads who have considered going their own way and outline the collaborative DevOps vision. Remember – few employees will actively seek to be a maverick in this context.

» Maintain contact with the Shadow IT advocates and continue to illustrate how DevOps can achieve Continuous Delivery but with security and quality still paramount.

Shadow IT is a sign of the times. However DevOps enthusiasts view the topic or choose to react, it is simply too important to be ignored.

ITIL – help or hindrance?

The ITIL 2011 framework provides guidance on the process and functions required to deliver quality IT services that are aligned with the business goals and objectives.

It's widely respected as a service management framework, but is it necessarily as applicable to effective DevOps as some of its more evangelical advocates would insist?

Let's start with the basics – what is ITIL in reality?

"ITIL is a library of good practices for IT service management, not a prescriptive framework," argues Matthew Skelton, Principal Consultant at Skelton Thatcher Consulting. "It also emphasises continual improvement of ongoing services, not 'finished' software. A key aspect of a DevOps approach is that handoffs are to be avoided if speed and quality of delivery (and effectiveness of operation) are to be maximised.

"Agile, ITIL, and DevOps all have a shared focus on working software, and emphasise iterative improvement and collaboration. Agile methods favour iterative, early delivery, collaboration with stakeholders, responding to change, and working software rather than extensive documentation and planning."

So there is certainly some commonality of purpose at play, but expectations do need to be kept in check, argues Harry Vazanias, Principal, The North Highland Company. "The question of whether ITIL helps or hinders DevOps is a common one," he says. "This is the wrong question, and actually distracts from the right question, which is 'What governance do you need to nurture the right DevOps environment?'. This may or may not heavily draw on ITIL."

> "While ITIL ticks off Service Strategy and Design, it doesn't really cover the 'Build' part of IT but rather goes straight to Transition."

Vazanias's contention is that ITIL itself doesn't notably hinder or drive DevOps because it's not set up to do either. "ITIL is a framework so if any element proves too bureaucratic and siloed for DevOps, you have the flexibility to ignore it," he elaborates. "Hence ITIL only gets in the way if that is how you tailored it."

"ITIL is not designed to comment on or support Agile or DevOps," he adds. "Equally, its guidelines contain little to challenge DevOps."

He also notes that while ITIL ticks off Service Strategy and Design, it doesn't really cover the 'Build' part of IT but rather goes straight to Transition. "The build processes are missing," he states. "DevOps, a movement born heavily from the Dev world, is not addressed. By not addressing it in any real way, ITIL doesn't do much to support or conflict with DevOps."

There's also the danger that ITIL in its present form hasn't really kept up to date with current tech developments. "ITIL 2011 is the latest version and really only a minor update of V3 from 2006/7," says Vazanias. "In other words, ITIL core thinking comes from a time pre-Smartphone apps when Agile was only beginning to get into large, long-standing corporates."

An opposing view

But Patrick Hyland, DevOps Engineering Manager, Pearson, argues that DevOps can be highly effective in the transition and operations stages of an ITIL V3 lifecycle-enabled service provider, which is being managed using lean.

"Using Kanban to interconnect the lifecycle processes that deliver the service creates visibility of the service provider's entire organisational system, leading to a more profound understanding of the system," he argues. "This is akin to Gene Kim's 1st way 'Systems Thinking' but with ITIL V3 strategy and design connecting into development and operations.

"Furthermore a feedback loop from problem management in operations can extend all the way back into all parts of the lifecycle, not only the transition phase which is the typical DevOps framed area of the lifecycle. This can take the feedback deep into the early lifecycle phases."

As an example, Hyland cities the instance of operations doing problem management on a service that is performing poorly, then being able to tell strategy that the design capacity has not been regulated to meet the strategic demand.

"The service provider can then use Theory of Constraints to see why this may be happening," he explains. "Perhaps capacity management is a bottleneck and needs to be appropriately exploited? Or perhaps capacity management is in fact overproducing and the consequences are being felt in service validation and testing? What does the WIP (Work In Progress) inventory say?"

Hyland argues that ITIL can be very effective if it is considered in this lifecycle sense. "ITIL V3 provides an impeccably sound map for left to right service delivery," he says. "In the transition phase, application development, service validation and testing, change management and release and deployment management are the areas directly relevant, ripe for automation and synthesisable with continuous delivery which forms the backbone of the DevOps value stream.

"Applying lean to ITIL V3 interconnects all the lifecycle phases, casting the collaboration net wide, from Strategy through to Operations, not limiting it to collaboration between Development and Operations."

But all of this does take time and consideration to get results; it's not a miracle cure for anything. As Vazanias concludes: "People keep looking for a silver bullet to make DevOps work, and expect ITIL to do this 'out of the box' and when it doesn't, claim it's not fit for purpose."

Governance questions to ask of ITIL

Harry Vazanias

» How does authorisation for dev to production work, especially when delineation of responsibilities is required and single DevOps roles aren't feasible?

» How do you streamline (lean) your dev to production processes for faster deployment?

» How do you run a change process that allows for multiple deployments a day?

» How do you perform transition planning when using Agile?

» How and where do you utilise DevOps teams in conjunction with more traditional Dev and Ops silos?

» How do you reward failure in the right way?

Practical steps for joining up Agile + ITIL

Matthew Skelton

There are several ways in which Dev and Ops teams can be brought together within Agile and ITIL® contexts to build a collaborative DevOps approach:

» Run Book collaboration where Dev teams write a draft Run Book, seeking help from Ops teams on the details

» Choose tools that encourage collaboration and avoid expensive Production-only ones

» Test early for operational readiness, using network emulators like NE-ONE from iTrinegy, network fault injectors like Saboteur, and security test frameworks like Gauntlt

» Single product backlog, avoiding the term 'NFRs', with user-visible and operational features in the same backlog

» Keep changes small

» Rotate people through Dev and Ops teams

» Ensure that the right kind of collaboration is set up between Dev and Ops teams, and shared tools are used effectively in order to reduce Continual Service Improvement feedback loops timings to minutes or hours, rather than days or weeks. How do you reward failure in the right way?

Six habits of the effective **DevOps** practitioner

What makes a highly effective DevOps practitioner?

Clearly this will vary from person to person and organisation to organisation. But the following list is a curated view of the views of three experts: Matthew Skelton, chair of the DevOps Summit London 2013; Dave Farley, author of 'Continuous Delivery'; and Benjamin Wootton, Co-Founder and Principal Consultant of DevOps Consultancy Contino, who also provides commentary on the recommendations.

1

Look up from the keyboard.

Don’t focus all attention on the computer screen in front of you but get involved with the wider context. This means having an ongoing dialogue with many teams across the organisation, even if the dialogue is awkward.

Wootton says: “Go and sit with your other colleagues across the technology organisation – the developers, the architects, the testers and the operations teams. Learn what constitutes good software development, good IT operations, good testing and the challenges inherent to those activities.”

2

Collaborate, collaborate, collaborate.

This is not a personal mission; it’s a team effort and that means collaborating and sharing information and best practice.

Wootton says: “Make an effort to get involved in activities across the software development lifecycle in order to put your own perspective across in a collaborative way. It’s easy to say ‘break down silos’, but the power to start that process is generally in individuals’ hands even in the most bureaucratic, process-driven, stilted organisation imaginable!”

3

Be customer and product focused and use feedback to refine and adapt, at all levels of detail.

What is it that your customer wants from you and your products? And do those products reflect those desires and demands? Are the feedback processes in place to facilitate this level of understanding of customer needs?

Wootton says: “I am increasingly seeing a customer and product focus from effective DevOps practitioners. They understand what customers are looking from the systems they support and have good collaborative relationships with their users. They want to enable the business to deliver the features in short cycle times, primarily in order to satisfy user demand. They care about user experience above all else – more than their day-to-day role as developers, testers or operations engineers.”

4

Help colleagues to focus on the purpose of their work ('the goal') and work on the most valuable things for the business

Again this is about the wider mission. What are the business objectives that you and your colleagues are working towards or working to support? Are you able to assist and support others in their understanding of those objectives so that you are all on a common journey?

Wootton says: "Share your learning with colleagues in your own teams and make sure that requirements that will make everyone's lives easier are captured and reflected in what you do. When technologists are aligned with the business around customer experience or product, a lot of DevOps-like principles emerge naturally."

5

Use computers, but don't forget the human beings

This means allowing computers to do what they are good at and people to do what they are good at, and automate everything, but allow people to make decisions based on information, rather than guesswork.

Wootton says: "DevOps practitioners know that transformation is about people as much as technology, but I think we need to realise that the people elements begin at home in our own daily job roles and activities."

6

Eliminate waste and optimise the whole process, not just parts of it

Having a lean and agile organisational structure and processes in place is essential if the DevOps team is to service the business at maximum capacity. This means stripping away all surplus weight and focusing on the essentials needed for the DevOps practitioner to do his or her job in the most productive fashion.

Wootton comments: "DevOps practitioners want to capture data about how their product features are being used and have that data back with the product managers to inform the next iteration and improve the product."

DevOps – bending to the needs of the software testers?

A knock-on consequence of the rise of the Application Economy is the question of how does DevOps bend to accommodate the needs of software testers in a mobile world driven by web/cloud-based applications under strain to roll out at the speed of Continuous Delivery?

With software development and delivery an increasingly fast-paced business, especially in the world of mobile, project durations are typically short and solutions need to be delivered to a potentially large user base that, through the use of application stores, have the ability to quickly give very visible feedback. In these situations, testing can frequently, and incorrectly, be seen as a bottleneck in the delivery process.

But DevOps doesn't necessarily need to bend to accommodate the needs of software testers, says Steven Janaway, Test Coach and Trainer at The Net-A-Porter Group, arguing that one major advantage of the DevOps movement is that it places operations skills into the cross-functional delivery team and away from a siloed and separate team.

"The efficiencies of such an approach are already well understood within testing," he suggests. "Moving all the required competencies to design, build and deliver software into a single responsible team just makes sense. A single responsible team takes responsibility for quality and whatever is required in order to understand it."

The move towards DevOps should be seen as another opportunity for teams to own quality, Janaway advocates. "Just as testing should be seen as a team activity then so should the ability to deploy to, and maintain, live systems," he says. "A user story should not be considered 'done' unless it has been running for a certain time in production, with the required metrics indicating it meets customer demands. This feedback can supplement the feedback that testing can give to the team."

It's also the case that DevOps can help solve the issues that testers have traditionally had when obtaining and maintaining production-like test environments. "Testing on a production-like environment, or production itself, is becoming increasingly important as we move towards faster release cycles," says Janaway. "Testers can learn from those responsible for operations and moving those people into the team or moving the aspects of the role onto existing developers gives further opportunity to the software tester. A team that understand their production environments is able to assess where their testing effort is best spent, whether that be pre or post deploy."

All of this is contingent on companies learning to adapt to new ways of working though, cautions Janaway. "Just as the move towards DevOps required a mindset shift, the movement towards single team ownership and testing in production requires the same," he concludes. "Moving some testing activities to post deployment requires a change in the perception of risk. The risk appetite of a company needs to be understood and potentially changed, to allow companies to get the maximum benefit."

DevOps talent – buy in or grow your own?

Paul Speers, CEO of Speerhead, a leading DevOps search and recruitment practice offers five tips for finding DevOps talent.

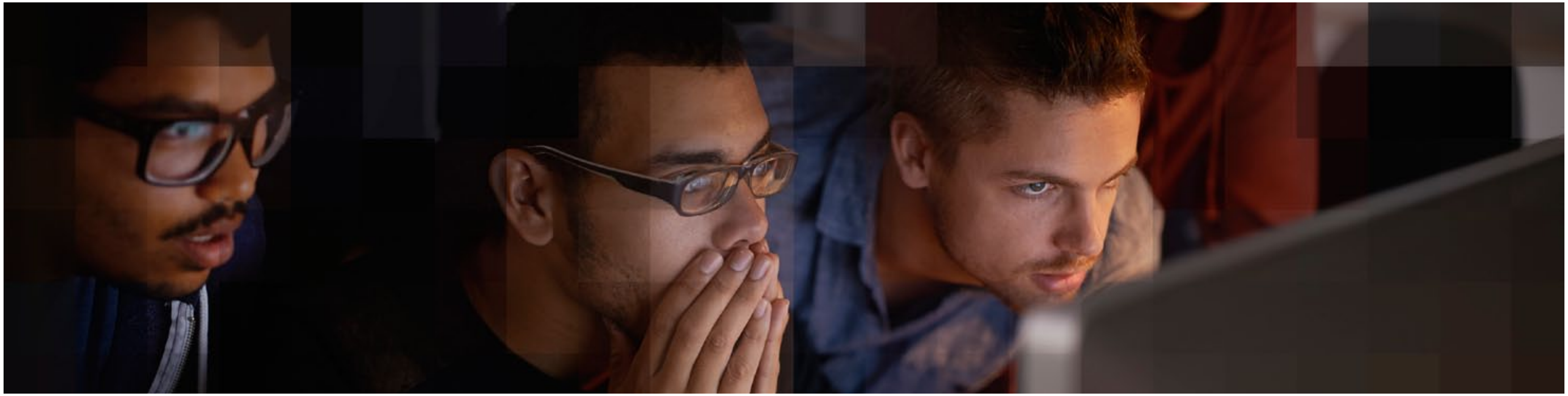
Hiring talented DevOps rock stars is the foundation of a strong DevOps culture. Rapid adoption of open source tooling and immediate and (sometimes) cheap access to computing platforms like Amazon Web Services (AWS) have significantly empowered individual engineers to build highly flexible and automated systems.

But with empowerment comes accountability. I have never known a time in our industry when businesses have been able to accomplish so much with so few individual IT resources. This empowerment comes with an even greater accountability to create a quality workplace and environment for excellent engineers.

With many IT departments adopting a DevOps culture now, a single engineer can make the difference between a high-functioning team contributing boundless work in complete alignment with the business goals or a stymied mismatch of inefficient engineers fraught with tech politics and personal agendas.

Making a considered effort to develop domain talent and attract and hire engineers who appreciate the implications of the DevOps way will enable organisations with the competitive advantage they seek.

At my own firm Speerhead, we have hired hundreds of DevOps engineers with our clients across all our European offices. Over the last three years, our experience has led us to come up with our own Five Tips for Success:



Sell yourself

The reality is that this is a candidate-driven market with top talent hard to find and seemingly everyone working. We see hundreds of dull and boring and not thought-through job specs. Those are not going to attract the right people to you.

You need to create an engaging set of materials. This is a chance to sell the company and sell the great IT team and work ethos you have.

Generate a clear understanding of the technical transition the business is going through and call out the great projects and the technologies that the team will use. An outline of where you are in the DevOps journey or your path to Continuous Delivery is essential.

Who's the Daddy (or Mummy)?

You need a company mentor who can inspire and engage with the new engineers. You need to create the magic and make first impressions last through the hiring process. Do you know who that person is inside your organisation?

Personality, personality, personality

There are many profiles on the internet of DevOps skills. Half of these should be soft skills and communication skills, which are essential alongside tech ability.

Benchmark your existing engineers using Speerhead's online tech profiler. Once complete, you have an accurate profile of the engineer's personality traits, then match the personality sets with all new candidates as they apply. We provide this as part of the CV when selecting engineers.

Tick tock, tick tock

You need to move fast. Most DevOps engineers are snapped up very quickly in the market. So many companies lose top talent because they are on their 'seventh interview'.

Your process has to be aligned with the market and most companies are not. We're talking about a two-week window for you to find your mentor, excite the candidate by selling the role and company and complete the hire.

Grow your own

Imagine a service that hires top development graduates or talent from colleges, then places them with you for six months. Over that six months Speerhead can train the junior engineer on the relevant skills to become YOUR DevOps engineer, and at the end they transfer to your business full-time.

They will undergo soft skills training, Process re-engineering, IT automation and tooling and Continuous Delivery principles and practices. At the end of the six months whilst they have been working in your domain and business they are fully equipped to add their skills to the team.

So why is this such a great idea? Salaries are rising and the contract market is booming for DevOps engineers, and this has led to a shortage of experienced engineers with exposure to the tech and practices to make it a reality in your organisation.

Bringing on new talent under a defined training programme with an understanding of your domain and then backed by a solid on-the-job training programme will help you develop your own loyal and trusted engineers.

The result? They will grasp the principles and all the cultural attributes of the engineering role whilst being trained on the specific tooling and services that can transform infrastructures. Whilst they are technically competent, they would be shown the direction in which to develop with strong emphasis on soft skills like presentations, meeting, leadership, planning and organisation.

Conclusion

IT and HR teams need to think hard and significantly alter their hiring practices and processes to attract and keep talented engineers in their business.

Long gone are the days of putting a job ad out and expecting a flood of responses. Hiring measures will also have to change around the impact these engineers make to the business rather than the normal hiring KPIs.

Connect and engage socially with the communities, go to the relevant meet-ups and open source vendor events and conferences. The best talent is out there, but you need to go and find it.

> So many companies lose top talent because they are on their ‘seventh interview’.

Benjamin Wootton

Co-Founder and Principal Consultant of DevOps Consultancy, Contino

Benjamin is the co-founder of Contino, a consultancy helping enterprise organisations adopt DevOps and Continuous Delivery tools, practices and approaches.

Benjamin has over a decade of experience as a hands on agile software developer and consultant. His experience ranges from hands on software development and IT operations roles through to agile transformation and organisational change with large organizations such as Goldman Sachs, UBS, Deutsche Bank and Oracle Consulting.

LinkedIn: Benjamin Wootton

Robert Benefield

CTO, Evolve Beyond

Robert Benefield has over 20 years of executive leadership experience building and leading world-class global lean and high-performance engineering and technical operations organisations in demanding high-uptime environments spanning industries such as investment banking, defence, telco, and internet service industries.

Robert has led transformations utilising Agile and Lean, and has developed and successfully implemented best-of-class Cloud and elastic computing techniques in a variety of complex environments. Robert enjoys solving complex problems, creatively using technology and organisational techniques to bring a new level of understanding and dynamism across businesses and their markets.

LinkedIn: Robert Benefield

Harry Vazanias

Head of Technology, The North Highland Company

Harry is a leading expert on technology strategies and transformations, heading up North Highland's thinking on the next generation of IT and digital. He has over 15 years' experience working with dev and operations teams, helping them to redefine their delivery models and value proposition to the rest of the business.

Harry has led the redesign of IT and digital functions at world-class brands across various industry sectors including retail, media and telco. His focus is now on helping companies to move towards enterprise agile and DevOps ways of working, with a focus on product development, not project.

The North Highland Company

Matthew Skelton

Co-founder and Principal Consultant, Skelton Thatcher Consulting Ltd

Matthew Skelton has been building, deploying, and operating commercial software systems since 1998. Co-founder and Principal Consultant at Skelton Thatcher Consulting, he specialises in helping organisations to 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, 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.

Skelton Thatcher Consulting

Patrick Hyland

Founder, DevOps Associates

Patrick Hyland is the founder of DevOps Associates, a London-based consultancy concerned with application engineering management. The consultancy applies a blend of agile methods, connected ITIL lifecycle processes and DevOps collaboration/engineering practices to help companies design, build, deliver and operate outstanding application services.

Patrick is an ITIL expert with 18 years of development and operations experience. He is particularly interested in management via Eli Goldratt's theory of constraints, applying a lean manufacturing mindset within an IT Service Management context.

DevOps Associates

Dave Farley

Co-author of Continuous Delivery, Architect at KCG Europe Ltd

Dave Farley is co-author of the Jolt Award-winning book Continuous Delivery. He has been having fun with computers for over 30 years. Over that period he has worked on most types of software. He has a wide range of experience leading the development of complex software in teams, large and small.

Dave was an early adopter of agile development techniques, employing iterative development, continuous integration and significant levels of automated testing on commercial projects from the early 1990s. More recently Dave has worked in the field of low-latency computing, developing high-performance software for the finance industry. Dave currently works for KCG Europe Ltd.

LinkedIn: Dave Farley

Stephen Janaway

Test Coach and Trainer at The Net-A-Porter Group

Stephen is a mobile and e-commerce Test Coach, Strategist and Manager. Over the last 15 years he's worked for companies such as Nokia, Ericsson, Motorola and The Net-a-Porter Group, as well as advising a number of mobile application companies on testing and delivery strategies. He has written and presented many times about testing, frequently with a focus on mobile devices and mobile applications. Stephen also provides training courses and coaching, focused on both mobile software testing and software testing in general.

Stephen loves talking to others about software testing, test techniques and the mobile device and application world in general.

stephenjanaway.co.uk

Paul Speers

CEO, Speerhead Group

Paul Speers is the CEO of Speerhead. He started the company over 5 years ago to direct Speerhead on the path to long-term growth, further its commercial success in the DevOps market and drive the launch of its revolutionary DevOps Recruitment franchise and play a leading role in the creation of DevOps Training and Certification IP.

Not only is Paul focused on leading the DevOps market, but also on building the successes of the company's global customer base and recruitment solutions for explosive growth. Paul is a vital conduit between Speerhead's customers, its global franchise partners and the DevOps industry as a whole.

Paul brings to Speerhead over 20 years' experience in sales and marketing within the IT industry, having held senior positions at Opsware – the first IT Automation vendor from Marc Andersson. He is also the co-founder of Fox IT, the Global ITIL vendor.

LinkedIn: Paul Speers

Justin Vaughan Brown

Global Digital Transformation Lead, CA

Justin Vaughan-Brown is Global Digital Transformation Lead, Product Marketing at CA Technologies. He is the author of 'The Digital Transformation Journey: Key Technology Considerations' paper, hosts the quarterly DevOps Influencer Dinners and is responsible for the DevOps Simulation Experience, an interactive online workshop that explains core DevOps principles.

Peter Matthews

VP and Research Scientist, CA

Peter Matthews is a Vice President and research scientist in CA Technologies. Over 35 years in IT Peter has worked in enterprise scale mainframe, Unix and PC environments. His current research covers cloud computing, privacy, security and analytics and the streamlining of the IT supply chain. His co-author of 'The Innovative CIO'.

Next Steps

Mainstream adoption of DevOps is here. Is your organization ready to seize all the business benefits and opportunities it presents? At CA Technologies, we have built a portfolio of products and solutions on our DevOps expertise.

Visit ca.com/contact to learn more about how CA can help you close the gap between your developers and your operations—and keep your competitive edge in the application economy.

For more information on DevOps solutions from CA Technologies, go to: ca.com/insights/devops

#BusinessReWrittenBySoftware

Contributions and comments were solicited following a discussion held in July 2014 and in subsequent email interviews.