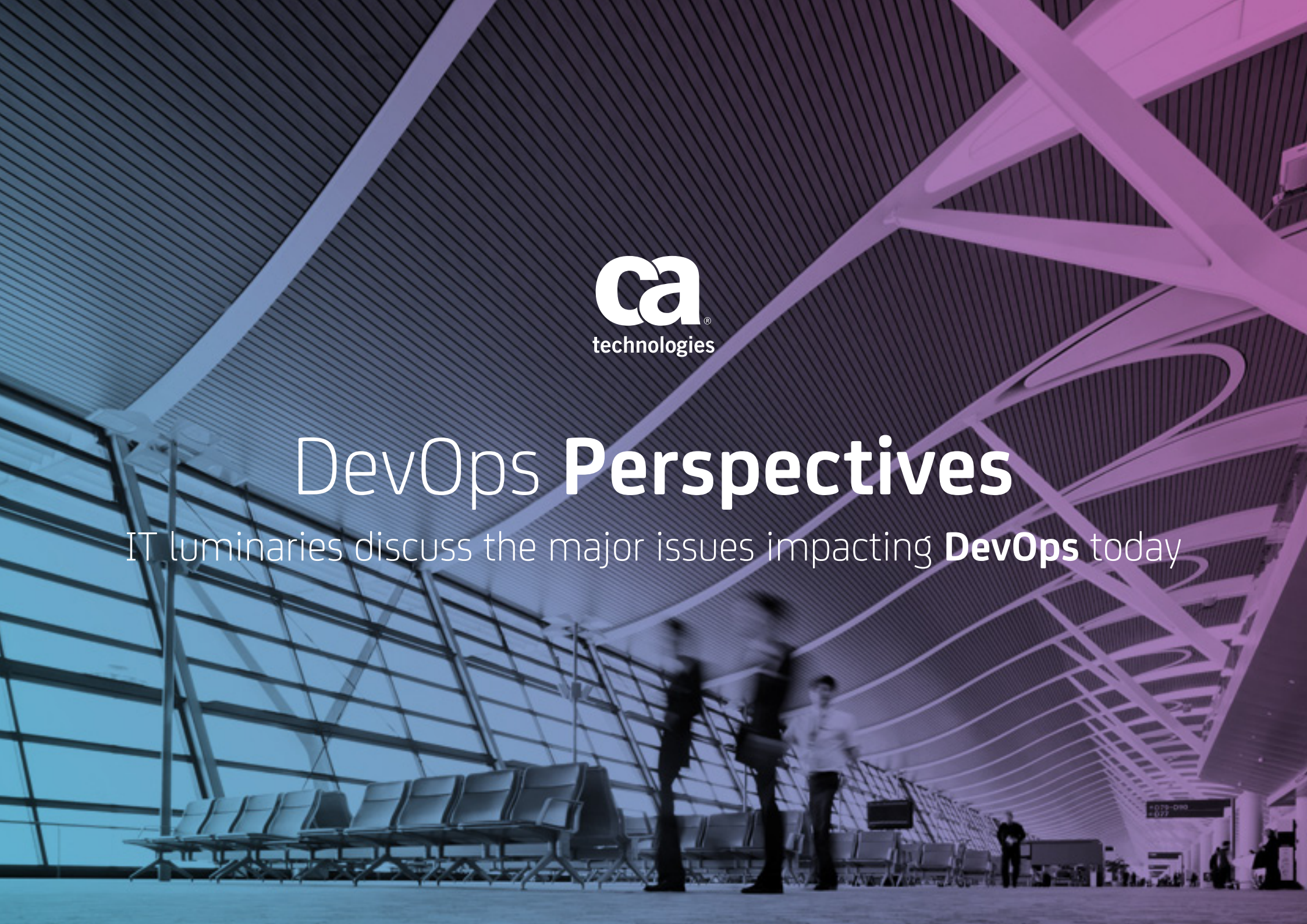


ca technologies

DevOps **Perspectives**

IT luminaries discuss the major issues impacting **DevOps** today

DevOps **Perspectives**

DevOps: a portmanteau for practicality

As a company that works with real world software application development and operations team practitioners at every level on an everyday basis, CA understands that this 'new term' DevOps needs clarification, analysis and examination.

In a programme we have called CA DevOps **Perspectives**, we have spent time talking to leading community figures to discuss the major issues impacting DevOps today.

Voices for IT change and transformation

We are pleased to have contributions from some of the most knowledgeable and experienced voices in the DevOps community in this eBooklet. We focus on several key questions such as the measurement of DevOps progress, how to obtain critical buy-in from the business function and major "do's and "do not's" of DevOps.

I hope you will find these insights valuable and thought provoking. If you would like to participate in a subsequent edition of **DevOps Perspectives** please get in touch at ca.com/contact.

Justin Vaughan-Brown
Senior DevOps Market Strategist EMEA
CA Technologies

Over the past few years at CA Technologies we have come to a number of DevOps observations as well, perhaps best summarised as:

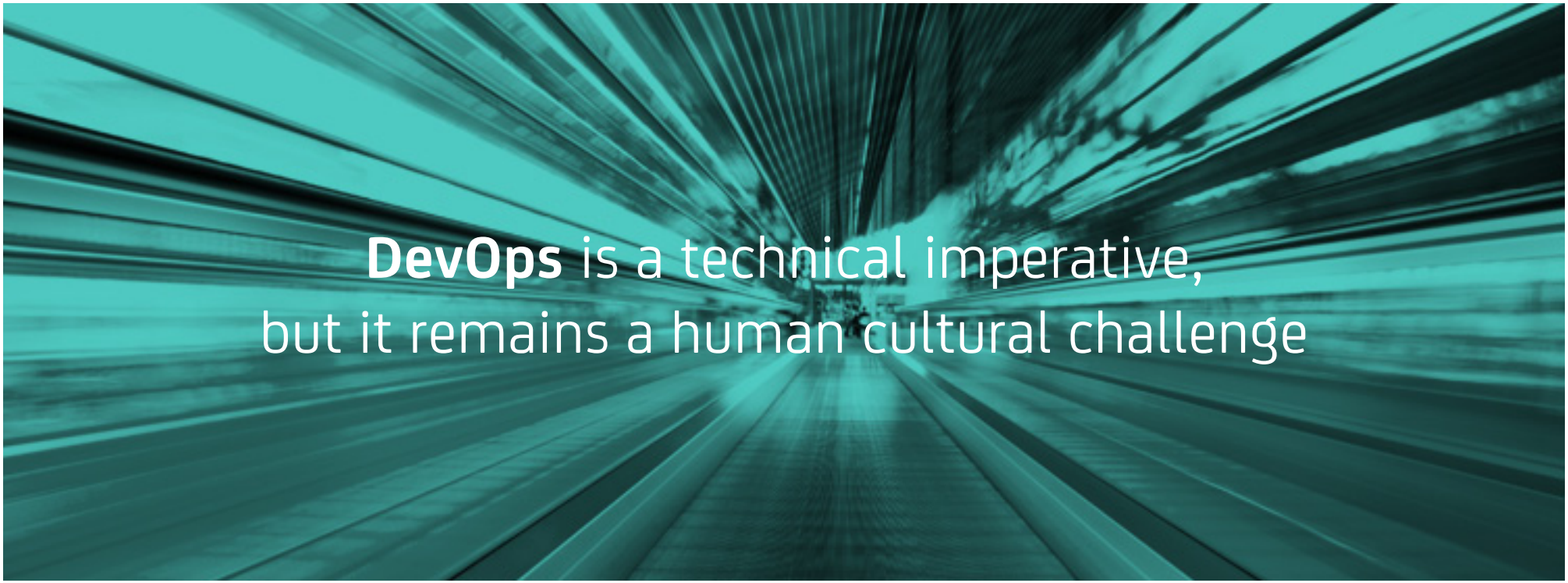
Automate wherever possible — taking manual approaches to routine tasks risks errors that turn into defects in production, so we should always view automation as an opportunity, not as a threat.

Starting small is OK — no firm needs a company-wide committee to investigate introducing DevOps and then make recommendations or an implementation plan six months later. It's often easier to establish a pilot project, which is classified as 'important' but not mission critical.

Buy-in from the top matters — sell the business value and the need to stay ahead of the competition, measure the success of early initiatives and prove the difference a DevOps-centric approach is making.

Change each application's development and deployment from a project to a process — make each application release and subsequent update an automated repeat of an agreed, transparent and defined process. Don't reinvent the wheel each time a new product or service has to be brought to market.

Don't underestimate the people factor — the personality of each key stakeholder (and how they react to change) will be key to success or otherwise. Does your organisation have the right composition for successful collaboration and adoption of new ways of working?



Executive Summary

It's easy to overlook the human factor when it comes to DevOps; let's not forget that Dev + Ops is a portmanteau and coming together of two professional designations, so it's all about relationships. If a firm cannot engender a culture of positive development around its DevOps team (or division), then poor staff retention will ensue... and this leads directly onwards to poor DevOps.

Our contributors here correctly note that "Ops sometimes see Dev as the Wild West" and this age-old axiom is still true i.e. developers still take the software build and "throw it over the wall" to operations without looking back. If the birth of DevOps does nothing else, it gets everyone around a single table with no wall in between — or perhaps a smaller wall at least. The separate elements of DevOps should unite as comrades over a common appreciation for the fact that that the business function might not care (enough) about either of the two disciplines.

Coming back to the human element, it is important to test whether an unhealthy 'Us and Them' attitude exists across development and operations. DevOps needs to be self-aware of both sides of the task and also equally aware of who the customer is; without this harmony a customer risk factor is guaranteed. Successful DevOps will most likely be a case of restructuring to create an environment where 'continual engagement' flourishes and information-share is maximised across and end-to-end in the system.

Although cultural and organisational restructuring will be needed, bringing DevOps online need not be a case of total 'rip and replace'. Smaller incremental project elements may often be a more prudent means of introducing DevOps to the system coalface. This way, the entire business gets to see what DevOps is (and what it is not) and where crucial lines relating to accountability and ownership should be drawn.

> "DevOps is a technical imperative, but it remains a human cultural challenge. Crossing this chasm will be tough, but it need not be perilous if we go one unified collaborative step at a time."

There are pitfalls here of course and it is easy to get DevOps wrong. Building a DevOps function as a new and separate 'third team', from the (management) top down is a fatal mistake; DevOps should be drawn from existing knowledge bases and often from the most senior team members available. There is an identifiable skills gap for professionals with a broad understanding of both build AND deployment — and this is what DevOps seeks to address.

Let's remember again that at this stage, DevOps is all about breaking down previous silos and being experimental. The best companies doing this are agile companies not necessarily agile software application development organisations. It goes to the roots of the company and if you have a large legacy code base then the first steps can be painful. Once again, start slowly and methodically with a defined strategic goal clearly laid down.

DevOps as a formalised working practice and as an element of modern data management frameworks is indeed on the rise. This means that today, more than ever, we need to understand and appreciate the need to embed the wider principles of DevOps culture (leanness, automation, measurement and sharing) into our total approach to IT architecture.

DevOps is a technical imperative, but it remains a human cultural challenge. Crossing this chasm will be tough, but it need not be perilous if we go one unified collaborative step at a time.

Paul Speers, CEO, Speerhead Group

> "Don't forget the human element! A simple key to successful DevOps teams is retention of key permanent talent and staff."

What does good DevOps look like?

Good DevOps is all about simple measures that IT and business can use together. Processes such as mean Time To Detect (TTD) to provide metrics on an incident and measure its frequency, size and impact can make a big impact. Following directly on from there, we look at mean Time To make a Change (TTC) and start to look at repetition rates to see if the same problem keeps happening.

Any firm can use a use a simple service management system to collate, track and report. Only then can you start to benchmark and then see a tangible and measurable DevOps service improvement over time. But don't forget the human element! A simple key to successful DevOps teams is retention of key permanent talent and staff. High turnover leads to high failure rate of the DevOps process and, subsequently, the wider project.

Do we need to work towards a common language to unite Dev and Ops in order to make DevOps work?

I was co-founder of Fox IT, one of the companies that developed and authored ITIL into the commercial world. Over 90% of global corporates even down to SMEs have implemented a service management framework like ITIL in their IT departments covering IT operations — so this should show you the importance of codified terminology and language.

The key is to understand how developers can interface into this world. Ops sometimes see Dev as the Wild West, however ITIL needs to lean out, hook into and in some ways optimise and leverage the DevOps principles of culture, lean, automation, measurement and sharing. ITIL is not moving fast enough and IT is still struggling! DevOps is just starting to catch the attention of the ITIL base — and the thousands of Ops departments who speak ITIL. Find the bridge and hook into the ITIL world they will love and embrace Dev forever more.

"Pick a project that can be started and finished using a new methodology."

How do we ensure the next generation arrive DevOps ready? What skills need to be developed?

The demand for DevOps skills is growing; it increased by 75% from Jan 2012 to Jan 2013. DevOps as a skill has increased on LinkedIn by 50% in the same period. As with ITIL, an open set of common training IP needs to be produced by the community for consumption by new talent.

We believe we can assist by placing junior engineers into the IT industry and training them over a six-month period whilst on secondment. After six months they have the skills to become a DevOps engineer.

The three disciplines we believe should be trained are:

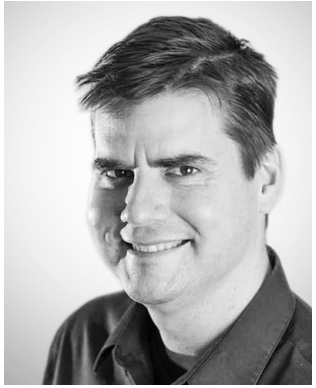
» People & communication skills
» Process re-engineering — an understanding of how to examine and re-engineer business and IT processes are essential skills for DevOps
» Experience with automation tool sets.

Does DevOps need to be sold to the business, or should we just make the magic happen? Does the business need to change to make DevOps work?

DevOps needs to be sold to both Dev and Ops. The business doesn't care; all they want is lightning fast and business changing deployments. The focus is with the IT folk entrenched in the silos. Pick a project that can be started and finished using a new methodology. Set the team goals as a pilot and set up a journal of the activity, issues and solutions.

What are the biggest barriers to starting DevOps?

Probably disbelief from those that would say: "Oh it's another service management framework" etc. Other big barriers include troublesome non-ratified standards (where they exist), IT silos (and the need to break them down) and no stakeholder buy-in from the CIO.



Robert Benefield, CTO, Evolve Beyond

"DevOps requires changes in the ways that teams work themselves and with each other."

What does good DevOps look like?

There are several things that I look for to determine how far an organization has progressed on the journey towards DevOps. I use a series of tests to evaluate or help an organisation moving to DevOps including the "Us vs. Them" test. This asks whether development, operations and the business work so closely together that they are capable of acting as one team, or are there signs across the organisation that they are very different teams with very different, if not competing, goals?

We also use what we call the "Service Awareness" test. This examines whether anyone in the company can tell you who a given customer is and how they use the service, what is important to them when and why, and how their work contributes to improving that experience. We look at whether the answers are consistent with the reality on the ground and award extra points if the person is in a technical role and has actually spent time with the customer to improve the service.

We also use the "Friction" test to see if there is an understanding of how long it takes to go from a concept or idea to it being launched into production, and why? The "Risk Awareness" test asks whether a customer could approach different people across the company, from the business to development to operations, and get the same answer for where the risk is, how severe it is, and how it is being mitigated.

What are the top five do's and don'ts for implementing a real DevOps environment?

1. Do not believe that there is a tool or technology that can be purchased that will give you instant DevOps. DevOps requires changes in the ways that teams work themselves and with each other.

2. Do look at your entire lifecycle... DevOps is not just about release agility. The fact that DevOps is a portmanteau of development and operations does not mean that it ends there.

3. Do engage continually... There needs to be effective and continual engagement between the technical side and the business to improve understanding and responsiveness while reducing ambiguity.

4. Do review and improve development practices... from how work is structured to code repository setup, build and integration systems, packaging, and testing to increase visibility and feedback while continuously reducing complexity, manual changes and bottlenecks.

5. Do review, restructure and improve operational practices... from infrastructure and environment management through to monitoring, change and incident management across the same lines. The process is continual, and visibility and feedback needs to extend beyond the individual team so that it can help improve situational awareness and collaboration across the organisation.

How do we ensure the next generation arrive DevOps ready? What skills need to be developed?

The biggest failure that has happened within the software industry has been the lack of emphasis on systems thinking, to the point that I feel it has been losing sight of the end-to-end system concept as a whole. In order to reduce complexity, there has been an incessant drive to put in place various abstraction layers to compartmentalize and simplify, often at the cost of becoming disconnected from what is really happening across the ecosystem and their role in it.

More and more I run into developers and operations staff that can quickly assemble software components like so many cogs to a machine and run it without grasping what may or may not be happening within the internals of the software stack. When this happens, not only is there a danger of people unintentionally making the wrong choices, but they tend to lose accountability for solving problems that they may in fact have caused.

The business is just as much at fault here as the technical staff, often seeing IT as nothing more than a cost centre that produces technical widgets that can be assembled and run by anyone. They lose sight that, like electricity or telephony, IT has moved to the core of the business, if not the actual product that provides the value to the business itself. We need to find ways to bring back thinking about the interconnected systems of the world.

Does DevOps need to be sold to the business, or should we just make the magic happen? Does the business need to change to make DevOps work?

The business ultimately needs to be brought along on the journey to both understand what DevOps is and is not, as well as to understand and take full advantage of what value it can bring to their organisation. The changes that they ultimately will need to take will greatly benefit their ability to more quickly recognize inherent risks as well as help them more thoroughly exploit capabilities to help them outmanoeuvre the competition. However, when they need to be sold is heavily dependent upon the structure and nature of the organisation.

Often the business needs to understand that there is a situation that needs improving, one that will provide real value to back to the company. Any proposed approach around DevOps needs to show that it is not only a likely solution, but also has a reasonable chance of working in their particular environment. Some organisations are fortunate to have visionary thought leaders who are willing to take a change and put in the time and effort to champion the transformation. Others tend to need more help and convincing to see the value.

The most successful implementations that I have seen started out with a small yet meaningful trial within the organisation. These were often started by some passionate employees on the ground that saw a way to overcome problems they are having that prevent them from greatly improving the responsiveness and quality of the solutions they are delivering.

“The most successful implementations that I have seen started out with a small yet meaningful trial within the organisation.”

While automation may have been part of the picture, the drive was focused far more on making steady and continual improvements over any particular tool or technology choices. They often had at least some level of managerial air cover or buy-in, enough to allow the team to experiment to both better understand what DevOps may mean for the company as well as to come up with a potential roadmap for others to follow. Success for these teams was not measured by how many deployments one can get out on any given day, but by interest from other teams wanting to adopt the same changes themselves.

What are the biggest barriers to starting DevOps?

There are many barriers. Some of the biggest ones that I have seen include lack of accountability and ownership for delivering quality production services quickly in a way that provides value to the customer. In these instances you usually see developers wanting deployment tools and access to production to release software more quickly with little regard for the running and supportability of the service. Operational staff put in place processes that make it increasingly difficult to change environments. Business people view IT as a cost centre that needs to be further squeezed to remove expense, with investment focused on features over "non-functionals".

Management by command and control is also a major issue. In these micromanaged environments the teams are given little leeway to self-organise, with communication paths up and down the hierarchy preferred over cross team ones. There also tends to be a somewhat implicit lack of a sense of ownership and helplessness towards improvement. This makes for very sterile ground for DevOps, no matter how talented the team or how much investment is made in tools and training.

Where deep organisational silos exist with few shared objectives, DevOps has a hard time flourishing. We also see problems where there is a lack of shared understanding of the importance and value of metrics and situational awareness, coupled with a lack of interest in continuous improvement.

> "A lack of understanding or desire for IT service agility is also extremely problematic."

Some teams are perfectly content with the status quo, seeing no need to understand what is going on or improve. They often feel that they have the "One Right Way" of doing things, and that any problems or disagreements they have with the customer are not of their concern.

A lack of understanding or desire for IT service agility is also extremely problematic. There are (surprisingly) still a fair number of businesses out there that simply do not see value in IT service agility, either because their industry still moves relatively slowly, there are still monopolistic elements or limited industry competition limiting customer choice, or there is enough friction that creates a high cost for customers to change suppliers. As innovation and globalisation go on to remove the barriers to entry the numbers will continue to shrink.

Benjamin Wootton, Co-Founder, Contino

What does good DevOps look like?

Being from the development side of the fence, I tend to view good DevOps from that perspective. At its heart, this involves designing and developing software that places a high degree of emphasis on what have traditionally been operations concerns and requirements.

For instance, developers should be asking themselves how their software will be released safely, rolled back if necessary, monitored, upgraded, maintained, scaled, and supported on a day to day basis. Designs and architectures should be put into place to support all of these non-functional requirements and baked into the software through everything the development team does.

> "DevOps has to be a bottom up initiative if it is to succeed."

Developers should also have a strong awareness of their production infrastructure, understanding the environments that they are releasing into and how they will intersect with their application. Good use of virtualisation and infrastructure automation will empower the development teams to gain understanding, visibility, control and ownership over the infrastructure that supports their software.

There is also a huge cultural issue here, in that development teams should work in partnership with operations and care about delivered production software rather than throwing code "over the wall" to testers or operations. Developers working with a DevOps mindset will never see their work as done until software is in production, or indeed even retired from production after it has served its useful life. This change in perspective with regards to ownership is subtle, but hugely important.

Operations teams obviously have a journey to make towards a good DevOps approach, but I do think that we as developers have slightly further to go.

Does DevOps need to be sold to the business, or should we just make the magic happen? Does the business need to change to make DevOps work?

To move towards DevOps, a number of things need to fall into line; cultural, organisational, process and technical tooling changes all have to be put in place to fully realise the benefits.

I think there is a lot of value in passionate individuals or teams moving towards a DevOps way of working locally. Putting configuration management in place or suggesting tweaks to business processes can certainly deliver value and improve things locally. This is all great as I think that ultimately DevOps has to be a bottom up initiative if it is to succeed.

That said, to really excel and lock in the benefits of a DevOps approach, we have to go deeper than this, and some of the challenges can only be overcome with management buy in. Culture needs to change with regards to ownership and collaboration. Incentive structures have to be changed so that people are pulling in the same direction. Organisational structures have to be changed to break down silos and allow pragmatic collaboration. Only relatively senior management have this level of influence.

All in all, the best situation is probably for management putting into place an environment where DevOps is allowed to flourish from the bottom up.

What are the top five do's and don'ts for implementing a real DevOps environment?

Implementing a separate DevOps team is the most common mistake I have seen. This simply introduces a third silo, which tends to focus on tooling, with no mandate or attempt to deliver cultural or organisational change.

I have seen one instance of a DevOps team that was actually harmful to the organisation's performance and collaboration. I think you are generally much better to embed people with the right skills, approach and mindset directly into your development and operations teams.

Rebadging a Systems Administration or Support Team as DevOps is also missing the point in a similar way - we might shift perspective with regards to tooling, but it doesn't go nearly far enough.

Pushing a DevOps way of working down into an organisation from a top down perspective is also doomed to failure. People can be told to attend meetings, use certain tools, or work in a certain way, but if they still retain a siloed mentality then the initiative is doomed to failure.

In terms of things to do, I think the most effective thing you can do is push decisions and responsibility down to the people on the ground. Allow your teams to make choices about the tools they use and how they work, and usually the best, most pragmatic and collaborative solutions will emerge.

> "There are no shortage of people with the technical skills, but few people who understand the big picture"

How do we ensure the next generation arrive DevOps ready? What skills need to be developed?

There is a big gap at the moment in terms of infrastructure coders. These are people with the ability to use infrastructure automation tools and a developer's approach to configuring servers, infrastructure and tooling via APIs.

With the cloud model taking off, people are scaling their server farms to very large volumes of commodity cloud servers, and old techniques for managing all of this will simply not scale. The whole software operations field simply has to move in this direction.

There is also a skills gap for people with really good understanding of build and deployment. This is not always a sexy area to work in, but getting it right can hugely empower development organisations to move faster and with higher quality, creating real competitive advantage. The Continuous Delivery movement is really giving this visibility.

Finally, there are not currently enough people working in the industry with the right DevOps mindset. I speak to a lot of engineers through my DevOps consultancy, Contino, and I find that there are no shortage of people with the technical skills, but few people who understand the big picture and are able to take the organisation on the journey through cultural, organisational, process and technical change.

Dave Farley
Co-author of *Continuous Delivery*, Architect, KCG Ltd

What does good DevOps look like?

Good DevOps requires the application of 'scientific method' to software development. We are trying get to a verifiable and repeatable process where we establish effective feedback loops. Form a hypothesis, design an experiment to test it, carry out that experiment, reflect on the results and move on.

The scientific method as described above is simply the best approach to problem solving that humankind has found. That is why these techniques work; applying what is our most effective problem solving technique to what is a hard problem - software development. An important attribute of this approach is that we complete this cycle frequently. This rapid iteration limits the damage of us making incorrect guesses and allows us to deliver value very quickly. Cycle-time is a key measure of the effectiveness of your DevOps approach. Think about the time it would take to get the smallest possible change through your existing regular series of verifications, and whatever that duration, aim to reduce it.

The shortness of this feedback cycle is a key metric for success. There are actually a series of interlocking feedback cycles. At the fine-grained level there is the TDD (Test Driven Development) feedback cycle, you want that to be small and efficient. That means you need to write small tests, see them fail then write the code that makes them pass, look at result and move on. At the outside is the business cycle that we all really care about, have an idea, get that idea into the hands of your users or customers, see if it works, iterate!

Good DevOps needs to work on tightening the business feedback cycle too, the business and technical team need to become experimenters. Developing the experimenter mentality and capability is an important enabling step for businesses not just development teams. It is however a big change, they are not used to working this way. The tough stuff is all around cultural change. There is what I call an 'illusion of rigour'; it feels like a rigorous process to have huge amounts of up front thinking, planning and preparation. In reality we humans are so bad at guessing that this is not really true. You are being more rigorous if you are doing fine-grained iteration.

Do we need to work towards a common language to unite Dev and Ops in order to make DevOps work?

Many ideas move on and spread when someone or a group of people come up with a way of describing things that people are already doing and that description becomes adopted. However I am not sure there is an issue around differences in terminology or language between the Dev and Ops roles. If there is a communication issue, you need get people in the same room to break down barriers. Lots of stuff gets lost in translation, however when you are actually working together, if you do not understand what people are saying it is evident from expressions when you are face-to-face.

> "Cycle-time is a key measure of the effectiveness of your DevOps approach."

What are the top five do's and don'ts for implementing a real DevOps environment?

Fundamentally you need to remove silos and ensure people are working towards common goals. This has to be applied to the way that teams are organised, the way that work is organized and the way that goals are organised. Whatever people are working on they need to share the ultimate goal of getting changes out into production — compared to that, everything else pales into insignificance.

Then, however the path to the goal is defined, the processes created must be made repeatable and reliable, that means there must be a lot of automation. The mechanics of validating, verifying and releasing your software into production shouldn't depend on human decision making, because any process that does won't be repeatable and reliable. Use people for what they are good at, complex subjective decision making, use machines/automation for what they are good at — repeatability and reliability.

The path to release needs to be automated so we can trust the process, so that deployment is not down to a matter of human judgment on whether your software is ready or not. The deployment decision should be down to having done everything we can possibly think of, on an automated basis, to show that the code is ready — that everything we know is telling us that nothing has failed, we are being scientific, so we can't prove that the code will work in production, you can never have enough tests, but if a test fails we know that the code is not fit for production and that is a big step forward.

> "Fundamentally you need to remove silos and ensure people are working towards common goals."

How do we ensure the next generation arrive DevOps ready? What skills need to be developed?

We need to teach science generally and also teach the science of computing. As an industry we are terrible for not having learned much from problems that have already been solved. We often fail to build on the shoulders of giants; we do not have a culture of that. Instead, problems solved decades ago by really smart people are approached as though they are new, often using quick folk-lore recipes rather than building a real understanding of the problem that we are trying to solve.

People need a grounding in the basics of computer science, but most importantly in the principles of science and the scientific method. We need to teach people to iterate, to approach ideas with a sceptical mind and to always doubt — even if it is your idea, to be unsure until you can assert, to design experiments and test the idea.

One of my favourite quotes is from Richard Feynman, Nobel Prize winning physicist, he said:

"It doesn't matter how intelligent you are, if you guess and that guess cannot be backed up by experimental evidence — then it is still a guess!"

Instead we all too often learn by rote. This eventually results in every project starting the same way: get the database established, set up the systems layer then decide on what language we are going to use — until eventually someone asks, "What was the problem again?"

Projects that I have started (particularly in high performance environments) by broadening out the search for solutions and starting with an open mind have often ended up with simpler cleaner solutions than we expected — just by starting fresh and thinking about the fundamental problem. Whether writing code or test this holds true, of course DevOps is not unique in needing this approach to problem solving.

Does DevOps need to be sold to the business, or should we just make the magic happen? Does the business need to change to make DevOps work?

Absolutely the business needs to change! Most organisations will benefit from a rational approach to development of all types. Questioning, "Is this really going to make some money, will it be valuable? — or is this just ticking off a feature people have asked for?" is essential.

It is about being experimental. The best companies doing this are agile companies not agile development organisations. It goes to the roots of the company, it becomes fundamental to the ways in which they make decisions. Getting to this state is a huge change for many organizations, but it needs to happen to get to the really high value that is possible from this approach to development.

> "Organisations who take this approach absolutely love it."

The big barrier is that many people have spent a huge chunk of their careers in industries and environments where, to be frank, development has not been very good, where it is not delivering the maximum value that should be possible. Software has become a bottleneck in many businesses because firstly it is now so important and secondly it is a difficult, creative thing. Traditional management is terrible for creativity, you would not make a movie by being prescriptive about everything up front, it is a creative process that needs a fusion of ideas; software is the same.

What are the biggest barriers to starting DevOps?

Technical barriers to DevOps can be real. If you have a large legacy code base, it can be tough to achieve sufficient test automation. These technical challenges are however tractable and vanishingly small compared to achieving the cultural shifts required.

Cultural change is the hardest to effect. It is hard to change the opinion of people who have built successful careers doing the wrong things. Recognising that you have been doing the wrong thing is unnerving, it's a tough problem, challenging personally emotionally and on a skills level. Designing good experiments is also hard, it takes time to learn that skill. But all of those organisations who take this approach absolutely love it. It is more productive, higher quality and lots more fun — difficult and challenging sometime, but more fun.

Biographies

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.

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 organizations in demanding high uptime environments spanning industries such as investment banking, defence, telco, and Internet service industries.

Robert has led transformations utilizing 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 organizational techniques to bring a new level of understanding and dynamism across businesses and their markets.

Benjamin Wootton

Co-Founder, Contino

Benjamin is the Co-Founder at Contino, a London based consultancy specialising in applying DevOps and Continuous Delivery practices to software delivery projects.

He runs the popular DevOps Friday newsletter and is currently authoring a book, *Flowing Software*, due for release in mid-2014. He has over a decade of experience within agile software development and operations teams across a broad range of industries.

Dave Farley

Co-author of *Continuous Delivery*, Architect, KCG 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 Ltd.

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: www.ca.com/insights/devops

#ReWrittenBySoftware