If there’s one thing we can probably all agree on, it’s that no one can quite agree on DevOps.

DevOps is unquestionably a great movement for organisations looking to be more efficient, with a solid focus on faster and more reliable delivery. More and more organisations of all shapes and sizes are embracing the movement, with overall DevOps adoption rising from 66 to 74 percent, and enterprises reaching 81 percent*.

However, delve more deeply into what good DevOps really looks like (and what it doesn’t), and you’ll get a wide spread of answers.

With this in mind, we asked some of the leading lights in the community to talk through their experiences so far. We look at some of the challenges that early adopters have seen, from creating and nurturing the right culture and effectively using the vast array of tools available, through to hiring and developing the best DevOps talent.

While we now have the benefit of some hindsight, in the grand scheme of things, DevOps is still in its relative infancy, and there is a lot to look forward to.

If you’re undertaking a digital transformation, or are passionate about the power of DevOps, we hope this eBook will give you some interesting insights to take away.

Tony Chapman,
LinuxRecruit

* RightScale State of the Cloud Survey 2016
<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining DevOps with Hindsight</td>
<td>4</td>
</tr>
<tr>
<td>DevOps, a Retrospective</td>
<td>7</td>
</tr>
<tr>
<td>The current state of DevOps</td>
<td>13</td>
</tr>
<tr>
<td>DevOps as Terminology</td>
<td>19</td>
</tr>
<tr>
<td>The future of DevOps</td>
<td>22</td>
</tr>
<tr>
<td>Contributors</td>
<td>26</td>
</tr>
<tr>
<td>Next Steps</td>
<td>30</td>
</tr>
</tbody>
</table>
Defining DevOps
With Hindsight
The DevOps community is at an exciting crossroads. As Gareth Workman, Head of Web Operations at Kainos, puts it, “DevOps is no longer just a shiny fad that will pass by. It is solving real problems for businesses”. Antonio Terreno, Partner at Equal Experts, acknowledges that DevOps has become a “highly hyped term these days, but it has worked well to get traction, from startups to more enterprise companies”.

However, “DevOps is still a relatively new term and the meaning behind it changes depending on who you talk to and what day of the week it is”, notes Matt Smith of Net-A-Porter. “I think it’s always important to look back to what triggered DevOps, and it was the barrier between creation and delivery, the creation of silos of responsibility. So for me, DevOps is about breaking down those barriers, be it between Development and Operations, or between Marketing and Sales. Applying it holistically as a principle, a working practice to take into every encounter; not just Development and Operations”.

Looking backward is vital in understanding what direction to move forward, and as DevOps becomes more widely engaged with, early adopters now have the benefit of hindsight to delve into what DevOps really means and what it looks like action. Matt Smith, Cloud Adoption Consultant at Net-A-Porter, is excited by the fact that “there are a lot of people about to take the journey with hundreds of questions about it. How will it work? What will it look like? Having been doing this for fifteen years it is great to just have the answers and to share those experiences”.

Chris Jackson, Director of Cloud Product Engineering at Pearson, sums things up simply. “DevOps means be good to one another. The heart of embracing this for me is about being able to flip something around and look at it from someone else’s perspective. We get hung up on just developers and operations, but really, for me, it’s about being able to put yourself in anyone else’s shoes to generate some empathy about how others might feel regarding something you are working with them on. If you extend that respect and care to others in your work they will reciprocate it and the entire organisation will benefit”.

For Jon Topper, Principal Consultant at The Scale Factory, DevOps is a mindset, rather than a set of tools. “Specifically, that mindset is one of shared understanding that in any business, you need both change and stability to succeed. Everyone – not just developers and operations, but managers, product owners, info-sec practitioners, and designers – shares responsibility for delivering those changes as quickly as required, and for keeping the platform as available, secure and performant as the business needs”.

Antonio Terreno seconds this, saying that it’s a culture more than anything else: “a subset of the practice of working as a cross-functional team, with developers learning and acting more and more like operations people and vice versa: learning to code and crafting solutions together with the team”.

It is interesting to see the consistent language of collaboration and mindset, and certainly significant to note a distinct lack of technical terminology seen in these definitions.
Though the official definition of DevOps might still be up for debate, for experienced practitioners, it seems the most useful interpretation is to see DevOps as a way of working, rather than a collection of any particular tools or technologies. Though of course, it would be simplistic to disregard the technical ideology entirely, ‘good DevOps’ starts from setting up the right culture rather than any technological imperative.

While the people that ‘get it’ are seeing great success with DevOps, there still exists a barrier to entry for many looking to get involved. Some companies are concentrating on the wrong things, seeing DevOps purely as a synonym for automation, or adding in another siloed team – there’s still focus on observed practices rather than a deep understanding of what good DevOps looks like in action. Perhaps this is because it is easier to talk about than to actually do.

Chris O’Dell is wary that DevOps is often seen as a ‘cure-all’ for an organisation’s ills and so may be applied haphazardly without much consideration for the more difficult cultural aspects which underpin its success. As with previous ‘cure-alls’ applied in this manner, they tend to fail and leave many people feeling cheated. As such, I’m not a fan of many whole-scale ‘DevOps transformation’ initiatives”. As Gareth Workman puts it, “you can’t buy culture, although you can bring in people that can help adoption but DevOps takes an organisation wide shift to adopt”. Every team and every individual are unique, which makes adopting DevOps as a culture that much more challenging.

As the community now benefits from the hindsight of implementing DevOps, definitions are becoming more technology agnostic. However, there are still challenges in enacting DevOps in the right way. Because of this, the experiences of different kinds of company, and of engineers with different technical backgrounds who are already undertaking a transformation can help demystify the journey that new adopters need to take to see success.

This is where the deep roots of DevOps within the Open Source world can really help, creating an impressive global community that shares its technical and cultural expertise, even in DevOps’ relative infancy. The technology world may not have got it all down to a tee just yet, but it’s definitely an exciting place to be.
The first time experiences of DevOps make for interesting reading, showcasing what needs to happen for a DevOps culture to truly proliferate.

“Although my background is in software development, I have always been close to the operations side of things in the systems I have built, so what we now call DevOps feels very natural (and 'no-brainer') to me. In 2007 when I was part of a team building systems for a large financial institution, we all were involved in regular operational testing, spending many hours together with networks teams, storage teams, deployment specialists, architects, and developers. We effectively designed good parts of the system together with both Dev people and Ops people playing equal and crucial parts.”

Matthew Skelton

“This was quite a while ago now! My first experience was refreshing, it was great to work as part of a sharing multi-disciplinary team working together in a collaborative way. My job simply became more enjoyable as a result.”

Gareth Workman

“You could say I've been practising DevOps since before anyone was calling it that! I worked as a sysadmin at a mobile startup from 2005, managing environments and releases, working closely with a team of Perl developers. I'd worked as a software developer for a number of years myself by that point, and some of the dev team had worked in operations before, so we already understood each others' challenges pretty well. I'd always worked in small companies, so it was natural to me that all the engineers would work closely with each other - it didn't cross my mind that there could be any other way to structure a delivery team.”

Jon Topper

“I can't point at a moment in time and say that that was when I first experienced DevOps. I worked at a startup for many years. Communication was high, as was collaboration between all employees, and there was a culture of ownership within the development team, so although the Operations team were on-call, the development team would regularly be involved in diagnosing and resolving any production issues. As a joint and ongoing effort between the Operations and Dev teams, we added metrics and centralised logging to our applications for greater visibility and used Configuration Management to automate the configuration of our servers. DevOps was something we gradually moved towards, although we didn't realise it, as each step we took seemed like a logical progression from the previous one.”

Chris O'Dell

“About 12 years ago the company I worked for built an application support team. It hired 'support engineers' who had not worked with the code, and the effort completely failed. This team was disbanded and replaced by experienced developers, who could code solutions and tooling to improve the service – similar to the SRE model at Google. I joined this team and ended up leading the DevOps effort when we rebranded as such. Having engineers who knew how the code worked, when and how to fix it, and when to push back to the development specialists improved relationships at work and with customers immensely.”

Ian Miell
“The moment for me was when my team diffused a hostile meeting with a group of developers by turning around and asking, ‘what is it that is difficult about your job and how can we help?’ The body language and direction resulting from that meeting set us on a pathway that we’ve never looked back from.”

Chris Jackson

“My first experience was when I worked on a dating website - where the developers had an understanding of where their code ran, and the ops teams had access to the developers’ code. Devs were involved in releases and Ops were involved in the product development. At the time, I thought that the CTO was crazy and unfocused, but it became a huge business differentiator when we started delivering hundreds of releases a year.”

Matt Saunders

“In 2007 I was at a startup, and the dev and sysadmin teams were next to each other. That helped us learn about each others’ domains through osmosis which led to better systems and software. It was a great way to work... I only wish we’d coined a catchy name for it!”

Tom Clark

What’s interesting to pick out from these experiences is how natural they seem. Chris Jackson’s offer of help to developers in a meeting is a casual gesture, but as he says, it transformed the relationship between the team for the better. Chris O’Dell talks about it being simply a ‘logical progression’.

Tom Clark began learning more about different domains through ‘osmosis’. Co-location is an agile stalwart and enabled the team to work to mutual advantage by removing a barrier to collaboration – both literally and figuratively. It’s a recurrent theme that people were practicing some manner of collaborative working “before anyone was calling it DevOps”, as Jon Topper puts it. This sense of collaboration is what is really crucial to a true DevOps environment really – whether you undertake a branding exercise around ‘DevOps’ or not.

Matt Saunders had a similar experience, joining a team that had developers involved in releases, and the ops team involved in product development. At the time he admits he thought his CTO was ‘crazy and unfocused’ but it meant they could move to hundreds of releases and really differentiate their business.
What is interesting about this comment is the discord between a ‘top down’ implementation – the push for faster releases through cross-functional work was ‘crazy’ – until it began to work. This highlights another challenge for an organisation looking to undertake a DevOps transformation. Unless the top-down implementation is particularly effective, it can be rough sailing, as Ian Miell experienced, coming in to overhaul a failed effort. That being said, failure can be very effective way to realise what you need to change to see success.

Gareth Workman touches on another interesting point – that his job ‘simply became more enjoyable’ as part of a sharing multi-disciplinary team. Whether from an engineering or development background, technologists are always keen to learn new things and improve. This is especially significant when you compare other professions, where it’s much less common to meet people who leave the office and go home to fire up their own projects, or attend meetups and other learning events with the same appetite as the open source world.

In the context of a DevOps culture, this appetite can benefit each individual and the team – a win-win if done in the right way. The balance to this is creating, and maintaining, a feeling of equality to ensure everyone is brought in. Chris Jackson was able to initiate a more collaborative atmosphere by reacting positively to team members who did not feel like they had equal footing in issues of workload and reward. Matthew Skelton highlights that both the Developers and Operations people played “equal and crucial parts” in his team’s success.

Maintaining this balance in a fast changing environment is never easy, but ingratiating each member of the team and acknowledging their value is always a good place to start. The early experiences of adopting DevOps practices make for interesting reading, and there is a lot to delve into to see how these first experiences fit with ‘good’ DevOps. However, hindsight always brings some reflection and self-criticism, which is especially useful reading for those who may be undertaking a transformation.

**TOOLING UP**

The technical skills you need to develop will, of course, vary depending on your existing skillset, your team, and what you’re trying to achieve, but you’ll be chasing your tail if you focus too much on the latest ‘bleeding edge’ tool, as invariably newer, shinier tools will come out. Instead, it’s better to think big, rather than necessarily focusing on the intricacies of one programming language or configuration management tool. Antonio Terreno suggests that “developers get out the IDE and start hacking bash and Unix command line tools and from there learn how to automate everything you just learnt”.

Feidhlim O’Neill, Head of Technical Operations at Wonga, reflects that, if he could, he would have told his younger self to stay confident in his operational knowledge of what works and what products need to have – “that Ops is a partner of product engineering”. Matt Saunders seconds this, preaching that you need to “know that understanding the product you are building is as important as knowing the technology”.

Chris Jackson advises staying out of tool politics to avoid getting derailed. “I spent too long getting sucked into holy wars about the tools folks wanted to use to ‘do DevOps’. Afterwards, I realised that everyone had missed the point and that we were just substituting one kind of conflict for another”.

---

**DEVOPS: A RETROSPECTIVE**
“Code less, think more.”

“...better off stepping up a couple of levels, all agreeing on the higher-order requirements and value, and then agreeing which part of our group was going to go solve this for us”.

With an ever-growing collection of different technologies and tools out there, it’s vital to stay focused on the product you’re making and what that product needs to do to be considered successful. How you define and measure that isn’t always straightforward, but benchmarking what continuous delivery looks like for your product, and the impediments you are facing currently can be a good way to approach transformations to ensure they will be fit for your purpose. As Gregory Becker puts it: “code less, think more”.

CULTURAL SHIFT

For Gareth Workman, “culture is the most challenging thing to change, yet in my opinion, the most important. Therefore, focus on the people and the good behaviours you want to achieve, and don’t get trapped in the space of trying to fix things with technology”. Maintaining a culture of collaboration can be difficult and requires special efforts and attention when teams grow, says Gregory Becker, “time must be taken by the team to share knowledge, which is usually not planned for”.

The biggest challenge of any DevOps transformation is the people. You have to not only get teams brought in to the idea, but get them to be advocates. You can automate deployments, you can get everyone using containers, but you can’t ‘do DevOps’ to a team. For DevOps to be successful, everyone involved needs to have a deep understanding of what’s happening, and how and why it’s happening.

When I started out in Operations, I was the guy that wanted to do everything as perfectly as possible. This meant I often worked long hours to get the job done on time. From my experience, I believe the best thing is to have a pragmatic view. Whatever it is you are tasked to do, the first step is to make it work. If that means you hack it together with bash scripts and shoe strings then so be it. Make it work.

The next phase is then to make it better. Maybe replace some bash with configuration management or re-write into python. I’m not saying you start by cutting corners, but you need to be pragmatic: time planning is a big issue for everyone. Almost everyone underestimates the amount of work it will actually take significantly as they assume there are no interruptions, but when you start you get distracted, and you end up at your deadline with nothing working to show for it.

So the advice I would give is this: work out how long you think something will take, double it and add another 20%. Once you have your inflated deadline, plan a solution you can complete in half the time that is ‘okay’ and will work. Have a backup solution that you can implement in a quarter of the time.

From the halfway point continually evaluate how close you are to finishing. This will enable you to get to a working solution early and then iterate on it to make it better. The absolute worse case is you deliver on time a solution that works but you may need to invest more time in later to make better. Another way to think of this is, Make it work, Make it better. No one will be angry if you deliver early...
If you are relying on implementing tools and other observed practices, you can only take DevOps culture so far.

**TAKE ACTION**

While deep thinking is important, DevOps is a practical way of working, not just a theory. Despite complexities around what DevOps means and how it should work, focusing on making changes is the first, and possibly the most important step. Ian Miell stresses that “if you see a solution to a problem or an improvement to a process, implement it. Even if you fail, you will learn heaps. If you have a solution to a problem (however limited it is), you are ahead of 99% of people that turn up to meetings. This is true in startups and in enormous enterprises. Cultivate good relationships with those that do instead of talk about doing”.

As anyone who has engaged with DevOps before will know, you’ll see opposition to it. If you wait for everyone to get onboard, you’ll be waiting a long time. Tom Clark suggests that you “start small, think big. Run an MVP that demonstrates the benefits quickly, then use the data from that experiment to convince more people to give it a go. Eventually, you’ll build enough momentum that even the naysayers won’t be able to resist”.

**COMMUNITY SPIRIT**

It’s also important to work at instilling a real community spirit within your team, and treat this seriously. Matt Saunders says “respect no limits to how widely you should cast the communications net - talk and collaborate every day”.

Chris O’Dell highlights how important it is to not be afraid of asking and answering questions and sharing knowledge. “DevOps is about collaboration, everyone brings skills and knowledge which may seem obvious to yourself but is not so for others. Share this knowledge. At first, I was anxious about asking for more information on something which seemed basic to an Operations, Network or DBA person, but I’d only had cursory experience with in the past. Through asking these ‘basic’ questions, I learned much more about Operations and would feed this back into the way we developed our apps and so created a platform which was far more resilient and simpler to support. I’d have chats about the best way to structure a logging statement, how best to compose a graphite query to get the most useful graph, what happens in a failover situation and how we could mimic this in test, plus many other areas”.

This extends into the wider community. When Jon Topper started out, Meetup groups and similar communities were less available and he admits honestly that “I just didn’t go looking for them! The perspectives I get from other people working with the same technologies I use are invaluable, and the contacts I’ve made there have often come in handy whilst I’ve been building my business”. In short, take advantage of the Meetups, forums, books, repos and friendly twitter folk that exist and are experiencing the same challenges you are, you’ll see the benefits.

It’s also important to remember that a DevOps transformation shouldn’t be a competition. Milos Gajdos recalls the idea of horizontal hostility (a term coined by Judith White), which sees some teams declaring that “they’re doing DevOps ‘better’ than some other groups – whatever ‘doing DevOps’ means in this context. We need to get on the same board as a community too”.

---

**DEVOPS: A RETROSPECTIVE**

---

---

---

---

---
The Current State Of DevOps
THE CHANGING TOOLBOX

As you might expect, container management and orchestration is a hot topic. Ian Miell highlights Docker as “changing the way software is constructed, distributed and consumed”. Gregory Becker agrees that Docker represents “a pretty big change, and whilst it began a few years ago, it is still and will remain the main driver of change for the next couple of years. We have yet to see cloud platform supporting Docker environments natively. The current products are pretty basic and missing key features and solutions”.

Tom Clark is also looking forward to forthcoming developments in containers. While the ITV team is already using containers regularly in development, he’s looking forward to seeing it mature so they can use it in production via an orchestrator like Kubernetes.

Antonio Terreno highlights the “beautiful tools that Hashicorp are building”, which is an appreciation shared by Feidhlim O’Neill. For him, the maturing of the cloud and tools like Hashicorp “means less writing non-functional tools and more thinking about the product problems, and more time adding value, which is great”. Jon Topper highlights Terraform specifically for its ability to solve a real user need. “At The Scale Factory, we build our ReadyScale platform design for new customers all the time, so repeatability is really important to us. We’ve used our own tools for provisioning up to now, but this story is cleaner and more effective with Terraform involved. It’s by no means perfect yet (I’m tracking a handful of bugs that cause me some discomfort), but it’s improving all the time. I don’t get excited easily, but it’s difficult not to be in awe of such rapid development”.

Server-less architectures are also high on the agenda. For Matthew Skelton, despite the hype, this “is a good thing”. However, Jon Topper takes a contrary position. “The hype train is currently going full speed ahead on what we’re inaccurately referring to as ‘server-less’. Really this is just another example of commodification, just like the other thing-as-a-service plays, enabling end-user organisations to spend more time worrying about their product instead of on servers, networks, and operating systems”.

“It’s difficult not to be in awe of such rapid development.”
Chris O’Dell points out that server-less architectures are calling into question what we would consider an application. It seems this is an important tipping point as advances in technology are requiring that people go ‘back to basics’ and reassess things with an open mind and consider how they should function and interrelate in the face of these incoming technology advancements.

The same is true, if not perhaps more so, when looking at the teams that own and organise this technology, with consideration needed around how people’s roles and responsibilities will change and converge. It’s also important to think about this in reverse – how technological change could help overcome inadequacies in capacity.

For example, Matt Smith believes that the next big thing will be a move to more Heroku or Lambda style services, as a reaction to the issues that people are seeing with the management of virtual machines and containers, which will force a push to move applications to become heavily focused on micro-services and pushed into a Lambda server-less style environment. For him, this move is ultimately happening because, “currently, DevOps are not fulfilling their job roles properly, and as a result, development teams will start managing much more complex environments purely as code to remove the risks around containers and VM management, as it adds an unnecessary layer in between”.

Antonio Terreno builds on this idea of an imbalance between development and operations, specifically looking at working with micro-services. He argues that, “for some reason, programming has always been a bit ahead of operations, it still takes too much time to craft a resilient infrastructure to support micro-services with features such as auto discovery, auto scaling, auto healing, proper distributed monitoring, tracing and so on”.

While the latest tools are allowing for great strides within continuous integration, continuous delivery and automation, the arrival of this new ‘stack’ of containers, micro-services, and server-less environments require some significant thought to fully conceive how these should fit into an organisation, and what pressures these advances can create and alleviate, both from a technological and a people perspective. It’s vital that due attention is given to the abilities, and capabilities of the labour that will be needed to optimise and manage this growing bank of shiny things that the community has now created.

As we’ve seen, tooling is improving extremely quickly, allowing technologists to make great strides in recent years. However, with things changing at such a rate, getting to grips with new tools and processes, establishing what they can and can’t do, and how they can (and can’t) scale and adapt, is often a challenge in such uncharted territory.

While containers are certainly taking the software world by storm, as Milos Gajdos points out, “we have hardly got past the initial phase trying to figure out what the containers are and how to use them effectively as opposed to ‘fight’ them”. He observes that Jevons paradox has started taking its toll, so we are now slowly moving to the phase of learning how to manage them efficiently on scale. “We have finally started moving slowly from server space to application and service space, which is super interesting and exciting”.

For Milos, this Cambrian Explosion, which is not only relevant to the open source software space, “has led us to a paradoxical situation where the choice of the software can become a crucial decision for the success of the organizations.”
Milos points out that “It’s easy to break down under the pressure of software vendors and consultants, but it’s challenging to choose the technology, lots of which are available as open source, which will really help the business. Organizations are finally getting past the virtualization phase, we have a much better understanding of how to collect metrics and monitor software, while micro-services allow self-contained teams to move faster. All of this often depends on particular technologies often chosen prematurely or without considering the big picture”.

Jon Topper sees a challenge in remaining current in this climate of constant innovation. “As well as the gradual evolution of parts of our stack (the ELK components, for example, have all moved on a lot in the last year or two), there are other new tools to keep up with. The container ecosystem gives us new things to try all the time, and as a consultancy, it’s important that we familiarise ourselves with this stuff so that we’re well informed when our customers ask about it. Ultimately, if there’s a demand to use something new, we want to be able to embrace it, but only when we can do so without it becoming a support burden, so we’re always evaluating new things with production-readiness in mind”.

COMING FROM BEHIND

As DevOps matures, it’s great to see some areas that have been underrepresented in the past, such as metrics and security, starting to come into focus, but there is still a way to go. Chris O’Dell sees an opportunity for improved pipeline tooling. “We have tools for CI which competitently manage the building and testing of both code and infrastructure. We also have tools for deployment which simplify the process of getting changes out to QA and production environments. However, I feel there is a gap for a tool which lives above these stages that manages and visualises the flow of a change from commit to pre-production environments, automated testing, to deployment in a seamless flow, promoting regular change”.

Matthew Skelton draws attention to the recent maturity of logging and metrics tools, which is really helping many organisations to get up to speed with modern operational techniques. We have found that SaaS tools like LogEntries, HostedGraphite, and PagerDuty can really open people’s eyes to the possibilities and advantages of a metrics-led approach. It is great to be able to help organisations raise their game in this way”.

Tom Clark highlights the issues of security. “We’re moving faster and faster, and if the security world doesn’t keep up we’re going to be in a bad place. DevSecOps is a great response to this problem”. Milos Gajdos seconds this, believing that there is still a lot of room for improvement, especially in “including more diverse groups into the mix: security, storage, and network engineers. Finally, let’s not forget the legacy solutions that are lurking at every mature organisation. The challenging question is not only how to deal with it not only from the technological point of view but also from the organizational”.

“ If the security world doesn’t keep up we’re going to be in a bad place. “
HIRING CHALLENGES

DevOps calls for a new breed of technologist that the industry has struggled to attract, recruit and retain. It’s a great environment for talented individuals, but a challenge for businesses.

For Jon Topper at The Scale Factory, one of his main challenges is hiring. “Because we’re a consultancy, we hire for good people skills as well as good technical skills, and that can shrink the candidate pool pretty considerably. On top of that, it seems to be the case that beyond a certain level of experience, a lot of suitable engineers put themselves on the contract market”.

“Add the fact that universities don’t seem to be teaching relevant DevOps skills, and it’s clear why there’s such a challenge even before we take into account our industry’s disgraceful structural problems as regards gender and race discrimination”.

Tom Clark agrees that “there’s huge demand with little supply at the moment and money is not enough – you’ve got to tick all the boxes and keep ticking them. We need a ‘Makers Academy’ for DevOps to get smart and kind people who want to enter our industry up to junior level quickly – we can take it from there”.

Matt Smith touches on issues around being too focused on technology, with contractors focusing too much on the question of ‘what will make my CV look better?’.

“Unfortunately, the management guiding these teams are not experienced enough to see that the wool is being pulled over their eyes and they are taking unnecessary business risks in the hope that they will be considered to be doing ‘DevOps’”.

Tony Chapman of LinuxRecruit, talks about when he first started out, and “working with Linux System Administrators was comparatively easy. There was a clearly defined set of required skills, and everyone was generally in agreement in relation to the general scope of the role. By contrast, the rise of the ‘DevOps Engineer’ was a mine field”.

“We now needed polyglot engineers, people who could work with nontechnical people on technical issues. People who would be involved in the build process and development life cycle, rather than just being the other side of the wall waiting for the developers to throw things over. The requirements of organisations are definitely more bespoke these days”.

THE STATE OF DEVOPS
Tony recalls an early DevOps role that he worked on, placing System Integrators into The Guardian with the very basic qualification criteria being “an understanding of Configuration Management – ‘do you know Puppet or Chef? Great, you’re in!’”. Now, as we enter the age of the application economy, the requirements of the companies now are getting more bespoke, with new tools and a growing diversity in the types of organisations looking for ‘DevOps people’.

However, Tony is unsure if he is “a firm believer of the well documented ‘skills shortage’ the media and social commentators will have us believe. I’m more inclined to suggest there is a fundamental issue with hiring approach in this area. There isn’t a shortage of people within a growing population, and the numbers being involved in Computer Science fundamentals are ever increasing”.

In his view, the mindset around hiring needs to shift substantially, with “a more serious consideration of what we should actually be looking to identify when hiring within the DevOps space. There remains too much of a focus on the latest buzzword. Instead, companies need to pay more attention to the key interpersonal and cultural aspects required to be a great DevOps Engineer. After all, today’s buzzword is tomorrow’s legacy”.

The ability to learn the next big thing and the ability to work in a truly collaborative way is significantly more important than an individual’s current skill set – but this is harder to measure. Many companies are still struggling to identify and test engineers effectively. By adding additional planes of assessment in learning and collaboration, the situation looks challenging, to say the least.

However, the assessment, recruitment and retention of great people is arguably the cornerstone of any DevOps transformation. As Tony puts it, “good companies are missing great people and making poor hiring decisions based on what they believe they require”. Companies who have the right hiring mindset are flourishing, and those who don’t are beginning to feel the pinch.

It’s vital that the DevOps community doesn’t get too distracted by the ‘bleeding edge tools’. Firstly, things are changing so fast that you’ll be chasing a moving target. Secondly, by hiring for technical ability in particular languages or technologies, you’re continuing to focus too much on observed DevOps practices. Look instead for talented engineers that understand and embrace a collaborative culture, can think well, and are willing and able to pick up new skills. Or, said another way, hire for mindset, not skillset.
DevOps As Terminology
For Feidhlim O’Neill, while there are lots exciting developments that he is looking forward to, “what excites me the most is the acceptance that DevOps is now the norm”.

However, there’s still work to be done on what DevOps as a term actually does mean in practice. Matthew Skelton argues that “too many people think that DevOps is just about infrastructure automation & configuration – modern-day sysadmins who can use Git. This limits people’s awareness of what a wider approach to DevOps can bring to an organisation: a transformative capability rather than just some scripted automation”.

As DevOps continues to gain traction, it’s definitely important that people are talking about it – even if they don’t exactly know what ‘it’ is just yet – as they work to unpick what is possible and how it can work. However, as the DevOps gold rush continues, it seems it will have a future as an industry standard. What does that mean for the term itself? Will it just cease to exist? Will it just be the way people work, the norm?

Tony Chapman goes into more detail about how requirements are changing and converging. “At LinuxRecruit, we’re already seeing the amalgamation of skillsets, developers are now all polyglots, being able to turn their hand to any modern language. Infrastructure Engineers now all understand Infrastructure as Code and understand how to build CD pipelines. The lines are being blurred between traditional skillsets. In five years if you’re unable to continually adapt, and work with a DevOps mentality, you will constantly be playing catch up”.

Antonio Terreno thinks the term should disappear. “Just like we don’t talk about Agile any more but we just do it, everybody should do DevOps and everything should be simpler and faster to implement”. Chris Jackson agrees that “DevOps at present is a buzzword, and will at some point become normal. We’ve seen it with SaaS and Cloud and we will see it with DevOps. The interesting thing will be whether the enthusiasm for the movement continues when the ‘new and shiny’ veneer wears off. This will be fun to watch because I really don’t see many of the elements of DevOps as anything other than good common sense…will people steer away from common sense if it’s not shiny!?”. Matt Smith looks at the wider possibilities that DevOps can bring into an organisation. “DevOps is about breaking down barriers, be it between Development and Operations, or between Marketing and Sales. Applying it holistically as a principle, a working practice to take into every encounter; not just Development and Operations”.

Once IT change stops being the bottleneck, other parts of the business will need to get slicker and leaner too, or they will be seen as the impediment to change.
This is echoed by Jon Topper, who expects to see “some of the things we’ve learned in DevOps and agile transformations of technology departments start to spread into wider business contexts. Once IT change stops being the bottleneck, other parts of the business will need to get slicker and leaner too, or they will be seen as the impediment to change”.

However, Chris Jackson takes a contrary position on thinking about how far DevOps should reach into an organisation as a term. “I wonder where you stop with DevOps. Do you really want your C-level exec’s sponsoring ‘DevOps’? In my opinion, when you hit the wider business outside of technology, you’re actually asking people to support a culture of innovation. We need to be careful that we stop stretching the boundary of definition for DevOps and recognise when we’re fundamentally asking a company to operate differently. If you bundle all of that up into DevOps, you’re likely to miss key things or worse, assume it’s too difficult and ignore it”.

It would seem that there is agreement that the term ‘DevOps’ will be heard less in the future as it becomes more normalised. That’s certainly not to say DevOps won’t remain an important concept in the technology space, but perhaps the community needs to consider what place it has in other areas of an organisation. Should it be a trendsetter for other areas to become faster and leaner, or would this make things too unwieldy? By using the term DevOps in the context of other areas of an organisation, do we really mean innovation? Does stretching the term DevOps damage its definition?

Assessing how far DevOps can go is perhaps something that won’t be fully possible until it’s already happening, but it’s certainly an interesting concept to think about. Only a few years ago, the IT department was often seen as a slightly dusty cost centre that sat behind the curve of innovation. Now, what IT teams are doing is being heralded as the act to follow.

For example, we’ve seen terms like ‘agile marketing’ and ‘agile accounting’ pop up – the wider business world is seeing what technology teams are doing, and they like it. What we will have to wait to find out is whether DevOps as a way of working could proliferate outside technology teams, and what that might look like.
The Future Of DevOps
It’s clear that, if it is done well, the increased efficiency and productivity available through DevOps has the capacity to save companies millions and pave the way for ever faster innovation. This innovation is already happening, and it’s certainly exciting. It’s perhaps even more exciting to imagine what might be happening in five years time.

Jon Topper opines that there will be some substantial developments within the infrastructure space, predicting that “AWS will continue to stay out way ahead of the competition. They’ll keep adding services until most new users won’t need to run operating system instances at all, but there’ll still be a need to employ people who understand how the pieces fit together”.

Jon also thinks that “Microsoft will continue to surprise us. They won’t stop with Linux versions of SQL Server and .Net runtime, we’ll see more announcements along those lines. They’ll release more open source, and integrate with more and more non-Microsoft applications. Azure will become a promising alternative to AWS for some specific types of workload”.

Gregory Becker thinks that “Docker adoption will increase and change the way we use clouds. The need for configuration management will decrease, will be simple and cloud infrastructures will become more immutable”. Feidhlim O’Neill also talks about infrastructure changes, predicting that micro-services will become more ubiquitous. He also believes that we will see a “maturing of multi-cloud tools and the continued rise of the API both in the cloud in our data centres to build and manage infrastructure”.

Matt Saunders also looks ahead “after several false starts” to a continued shift within the software delivery world. “Containerisation will revolutionise the DevOps space, giving previously unheard of power to developers and gaining everyone an understanding of the importance of feedback loops. Organisations acknowledging that DevOps combined with Lean and Agile can make dramatic differences in software delivery. Tooling will get better and better, and big enterprises will continue to realise that buying software solutions off-the-shelf to solve their DevOps problem won’t get them where they need to get to”.

I predict a spread of the concept of the site reliability engineer, consolidation in the automation space to a few big players, and the slow death of the OS as a monolithic software deployment. And that these predictions are likely to be wrong.

Ian Miell
More organisations are waking up to the idea of DevOps, which is something also mentioned by Feidhlim who believes that “there will be more acceptance of DevOps in mainstream companies.”

Chris O’Dell also talks about the increased acceptance of DevOps, saying that “I imagine more companies will embrace the tenants of DevOps as the benefits continue to be understood. The adoption of technologies such as containers and server-less will require greater collaboration between all skillsets to be successful, particularly when it comes to monitoring, logging, security and performance. I see the lines blurring between traditionally distinct roles into cross-functional teams, consisting of people with both a deep knowledge in focused areas complemented with a wide breadth of other skills”.

Matt Saunders also refers to containerisation as an opportunity to give more ‘power to developers’. This is something supported by Matthew Skelton, who expounds on the changing role of the developer. He predicts that “we will see changes in the types and extents of collaboration between Dev and Ops teams as different kinds of technologies mature. In particular, different approaches are suitable for Docker, containers, managed container stacks like Kubernetes, and server-less architectures like AWS Lambda or Azure Functions”.

Changing attitudes to DevOps open the door to improving collaboration between Dev and Ops teams, becoming, as Chris O’Dell puts it, increasingly ‘cross-functional’. Chris Jackson expects that we will see a “diversification of how and where DevOps is implemented, from how different systems that are capable of generating code collaborate with each other autonomously, to how we continue to carry legacy systems along due to their unique purpose or need”.

Matthew Skelton also thinks that there will be an increased appreciation of different team topologies to suit different technologies and their responsibility boundaries. As iterations of DevOps diversify and change over time, it seems likely that there will emerge a set of frameworks to support different organisations in their transformations.

I think building delivery pipelines and platforms might become the new normal. I hope the attention and focus will move higher up the application stack. We have been talking about this for a while, but the move is happening quite slowly as people have this often unnecessary obsession with owning every piece of their application stack.

People will hopefully realize the real value of the company lies way higher than the OS level. We are already living in a much better automated world than a few years ago and we will be pushing this even further in next couple of years, maybe even with a bit of server-less magic sprinkled all over. Machine Learning might also play an interesting role in software delivery in some way.

The future feels intriguing. Let’s work hard together and make it awesome.

Milos Gajdos
Jon Topper also considers a future with larger, slower enterprises increasing their adoption of DevOps practices, but is perhaps more pessimistic in what the reality might look like, commenting on the role of “the big integrators who got them into this slow-moving mess by encouraging them to outsource their operations in the first place will land enormous contracts to ‘implement DevOps’. This will look like every other previous engagement, only the graduates tasked with this work will show up in polo shirts instead of suits”.

From a technological standpoint, the future of DevOps looks exciting. Even today, as Jon Topper puts it ‘it’s difficult not to be in awe of such rapid development’, and it would seem to be a sensible prediction that we’ll see even greater strides within tooling and areas like containers and server-less environments.

However, it perhaps will not be completely smooth sailing, and there remains some challenges ahead around dismantling the ‘cargo cult’ of DevOps, especially in larger enterprises, and in successfully upskilling the next generation of DevOps adopters to be able to keep up with the pace of change.

Chris Jackson makes an interesting point, that for him, “the bigger transformation will be around what people will be able to do with their time when they are not fighting fires and performing repetitive work. The investment they can put into building meaningful relationships in both internal and external communities will give us a new and much faster wave of innovation in tooling and implementation for others to pick up on”.

As the DevOps community works to overcome more of the remaining legacy of inefficiency and unplanned work and continues to advance forward in areas like logging, metrics, and security, it has the potential to induce somewhat of a butterfly effect, producing a ‘new and much faster wave of innovation’ as more time is freed up.

It would seem that perhaps the largest barrier to this innovation is in ensuring that we can develop, upskill and nurture the next generation of DevOps adopters to lead this wave to its full potential.
Antonio Terreno  
@javame  
I’ve spent most of my career as a consultant (plus stints in start-ups and R&D). I am a hands-on software architect and tech leader, with more than fifteen years’ experience of successfully delivering solutions around the world using agile methodologies. In 2015 I’ve joined the Leadership team of Equal Experts as Partner.

Chris Jackson  
@chriswiggy  
Chris Jackson is the Director of Cloud Product Engineering at Pearson. He leads a team into a brave new world of containerised platforms. His passion for building valuable technology combined with his amateur skills in all things code inevitably led him to a leadership role where he can help others achieve their goals and engage them in some of the most exciting technology spaces available. He is a thought-leader who will over a beer tell you that many of the world’s problems could be solved with a bit of DevOps...

Chris O’Dell  
@ChrisAnnODell  
I have been developing software with Microsoft technologies for over 10 years. Between 2012 and 2014, I led the API team at 7digital, a leading provider of music streaming services worldwide. I currently work on the Platform Engineering team at JUST EAT in London. In all of my roles I have promoted practices we now know as Continuous Delivery, including TDD, version control, and Continuous Integration. I am co-author of the ebook Continuous Delivery with Windows and .Net, and a contributor to the book Build Quality In.
Feidhlim O’Neill  
@feidhlimoneill  
With broad technical operational experience in a variety of companies on both sides of Atlantic  
Feidhlim is currently group head of platform engineering and technical operations at Wonga after  
spending 2 years in a similar role at Hailo. Building teams, infrastructure automation, engineering  
productivity and non-functional requirements are among the areas that interest Feidhlim.

Gareth Workman  
@gjworkman  
Gareth has been with Kainos for 10 years and is Head of the Web Operations capability. For the  
past 5 years Gareth has been working within Central Government Departments in the delivery of  
programmes that aligns to their Digital by Default strategy. Gareth’s focus has been in the field of  
Web Operations ensuring the delivery of robust, secure and scalable platforms for citizens to  
interact with Government.

Gregory Becker  
Gregory Becker has spent the past 9 years at Notonthehighstreet starting as a web developer and  
system engineer, supporting the company from its start-up days to being a well established tech  
company and is now leading the DevOps team to build and maintain our cloud & Docker  
infrastructures.

Ian Miell  
@ianmiell  
Ian is an experienced software professional with over fifteen years’ experience working with FTSE-100 companies  
delivering mission-critical and performant software and services in technical, managerial and leadership capacities. A true  
all-rounder, he is comfortable dealing with everything from operational management issues to system architecture analysis  
right down to line-by-line code. He is the Author of Docker in Practice (Book, Manning Publications, and runs a great blog on  
DevOps and Docker at zwischenzugs.wordpress.com.
Jon Topper
@jtopper
Jon Topper has worked in infrastructure and operations for 15 years, and has the facial hair to prove it. His consultancy, The Scale Factory, are a team of DevOps specialists, helping businesses design, build, operate and scale their infrastructure.

Matt Saunders
@cm6051
Matt Saunders is a DevOps Consultant for Contino, and co-organiser of the London DevOps Meetup group. With an Operations background, Matt specialises in helping and building cross-functional teams deliver good software quickly and effectively.

Matthew Skelton
@matthewpskelton
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 curates the well-known DevOps team topologies patterns and is co-author of the books Database Lifecycle Management (Redgate) and Continuous Delivery with Windows and .NET (O’Reilly).
**Matt Smith**  
@solmafreak
Matt is an experienced hands on DevOps manager who has a focus on automation and tooling to provide continuous benefit and improvements back to the business on key strategic and tactical projects – and is an AWS expert. Matt has also worked for Alfresco Software, British Gas Connected Homes, Defra and Dataloop, and is currently working with Net-A-Porter as a Cloud Adoption Consultant.

**Milos Gajdos**  
@milosgajdos
Graduating from Czech Technical University in Prague with a Masters degree in Technical Cybernetics, I am a Developer/Operations Engineer based in London. I help to organizer Kubernetes London Meetup. I like automation, building software in various programming languages and have recently rediscovered my passion for Machine Learning and Functional Programming.

**Tony Chapman**  
@linuxrecruiter
Founder and Managing Director of specialist Open Source consultancy LinuxRecruit, who work with organisations across the UK, designing their DevOps recruitment strategy and staffing their DevOps Engineering teams. Tony is also co-organiser of the world’s biggest monthly DevOps meetup, the DevOps Exchange in London.

**Tom Clark**  
@tomonocle
I've spent 15 years working with technology across a number of sectors, from Jaguar Cars to the BBC, Global Radio and most recently ITV. I've held senior/lead positions in several disciplines, including engineering, development and architecture. I'm now Head of Common Platform at ITV, responsible for growing the next-generation cloud hosting platform that underpins all of our key applications. I'll always be technical, but now my day job is managing and developing the team of "smart and kind" engineers that build, maintain and evolve the platform.
We put this eBook together as a new way to engage with some of the thought leaders within DevOps in a way that is accessible and (hopefully) interesting to anyone within the DevOps community. If you’d like to be involved in future editions, or have anything you’d like to add, please do get in touch with us, we’re always up for a chat about anything DevOps.

**About LinuxRecruit**

We work with organisations across the UK, designing their DevOps recruitment strategy and staffing their DevOps Engineering teams. LinuxRecruit work roles across Infrastructure, Development and Architecture across the Open Source sphere as well as having a significant niche within DevOps Engineering. Founder Tony Chapman has been working in the Open Source community for 10 years and is co-organiser of the world’s biggest monthly DevOps meetup, the DevOps Exchange in London.

If you are looking for a new role, a new member of your team, or even looking build a team from scratch - we can help you through every stage of the process, and offer you honest, consultative advice.

---

**The Next Steps**

Join the DevOps Exchange

info@linuxrecruit.co.uk

020 379 52605

LinuxRecruit

@LinuxRecruit

Join the DevOps Exchange