

THE COMPLETE GUIDE TO: **DevOps**



Credit: iStock



Credit: iStock

Development cycle



In our quest to break down the biggest enterprise technology trends this month we are tackling DevOps, the development practice that has been sweeping the enterprise for some time now.

DevOps is a combination of, you guessed it, software development and operations. So instead of having separate teams building software and another running it everything is done continuously to shorten development cycles. The need for speed with modern development has led to this practice of continuous delivery, where software is built and provisioned to be constantly tweaked, without any downtime for users.

The term DevOps came to prominence around 2008 off the back of the rise of Agile software development.

The key to this practice has been the rise of ubiquitous cloud computing, which is charged on-demand, and advancements in automation and monitoring technologies. However, it is not just the technology that needs to change when adopting DevOps, especially at an enterprise with lots of legacy technology and ancient business processes, it is a cultural change which needs to be fully adopted to really see the benefits.

Here we run through some key practices to get up and running with DevOps, some help with key tooling, as well as case studies for best practices, including how businesses are already reaping the benefits of shifting to a DevOps culture. [Scott Carey](#)

Contents

- 4 Why culture is key to DevOps success
- 8 How to get started in DevOps
- 15 Alternatives to popular DevOps tool Jenkins
- 21 Capital One expands DevOps enterprise-wide
- 25 ITV embraces DevOps for product development
- 29 Sky Bet on doing DevOps right: learn from the best



Credit: iStock

Why culture is key to DevOps success

The term ‘DevOps’ has been bandied about for years, but is it a dated buzz phrase or is it actually new best practice for business?

It seems the word on the lips of every enterprise going into 2018 is ‘DevOps’. But what exactly is it? Should your organization be paying attention, and how can its near-buzzword status be unpicked?

As the name suggests, it’s a marriage of ‘operations’ and ‘development’, but in practice the term is an umbrella promoting collaboration, automation, and

potentially radical changes to company culture that seek to unify different departments behind common goals.

“At its simplest, DevOps connects people, processes and tools – people being the key component,” explains Gunnar Menzel, vice president and chief architect at Capgemini Infrastructure Services. “It’s crucial to be open to change and have the correct internal environment to allow change to happen. This is not just in IT but throughout the business. This is why it is so important that DevOps has the correct senior backing and buy-in, otherwise it will fail.”

What are the benefits?

The goal of DevOps is to deliver software quickly and effectively by fostering collaboration between different departments. This is supported by automation, and by analysing organization-wide metrics to see what’s going right and what’s going wrong.

A successful DevOps strategy also enables businesses to pay close attention to customer feedback and testing in real time, which can be responded to with swift deployments and updates. And by automating laborious tasks such as testing, DevOps lifts a burden from development teams, which are then able to focus on more productive endeavours.

DevOps is an entire approach in itself, and so by its nature should not be considered another department. Introducing a DevOps team, for example, would contradict its core ethos by fencing off staff rather than pressing for collaboration.

It’s not just early pioneers such as craft sales website Etsy or Amazon that are proselytizing for DevOps. Large, established organizations are beginning to take notice.

Even the British government has an online service manual that explores its fundamentals.

Faster application updates to meet customer demands

Consumer expectations are partly behind the demand for instant updates and deployments. Rolling out updates at seemingly arbitrary times is no longer acceptable, and it's unhelpful for developers to expect code, once finished, to be another team's problem.

Brett Hofer of application performance management company Dynatrace agrees: "Consumers have become more demanding and industry rivalry has grown more intense, so the ability to introduce and update digital services quickly has become critical to success," he argues. "Many businesses have come to realise that development and operations teams can no longer work in departmental silos [to achieve this]."

Effective communication throughout a company, then, is vital – both in the management of the workforce and in sharing systems. "This cooperation is best fostered through the use of shared systems and processes that bring down walls that previously existed between these silos," he adds.

Another advantage of DevOps, according to Hofer, is being able to make the most of that often-repeated start-up mantra that it is okay to fail, encouraging innovation while building a culture that understands and accepts there will be failure. "The art is in how and when we test, monitor, learn, pinpoint and respond to what is working, isn't working or could work better."

Gordon Haff, DevOps expert at Red Hat, concurs: "One of the points of DevOps is to enable better, easier,

quicker, experimentation. More or less by definition, though, some experiments fail. If it doesn't work, move on without repercussions. But fail quickly."

There is no 'correct' approach

So how do you do it? Well, there's no textbook method – each organization will have to examine the lay of the land and plot out a course towards where they want to be. That means starting with a business objective, such as reducing the time it takes to develop software, and then defining the end-to-end process required to do so, advises André Pino, vice president of marketing at DevOps specialists CloudBees. It won't happen overnight. Transforming into a DevOps-oriented organization is a "journey that will take time to accomplish".

But it's a journey that businesses may have to take.

It's tough to disagree with Microsoft CEO Satya Nadella's declaration that every business will become a software business. DevOps advocates assert a joined-up philosophy is completely essential – and that if businesses aren't able to develop continuously, then they run the risk of being left behind.

"The adoption of DevOps practices will become more mainstream," argues Georg von Sperling, senior director at CA Technologies. "Eventually, all stakeholders will have a holistic view on the business value of just-in-time, high-quality and analytical applications, and services for their market."

In fact, according to Pino, the term is likely to fade away completely – as DevOps "becomes the standard way software is developed and delivered." [Tamlin Magee](#)



Credit: iStock

How to get started in DevOps

We speak to DevOps professionals about practical advice your team can take to start changing the culture of your organization

Over the past couple of years, larger enterprises started to see the value in DevOps, helping to shift the movement into popular business discourse. If every company is now a software company, as Microsoft's Satya Nadella famously claimed, then being able to fail fast and push releases out in hours not weeks is critical. But what are some practical steps that can be taken?

First, suggests Accenture's MD for DevOps Martin Croker, it's worth defining what DevOps is likely to mean for your organization. It's a broad term, he says, generally meaning a collection of behaviours and tools – encouraging collaboration between departments, automating processes, and so forth.

“But when we talk about an organization being friendly to DevOps, we're probably more likely to be talking around the processes of automated build and delivery,” Croker explains.

“The processes of branching strategy for configuration management, and really that drive of trying to get the development and operations teams harmonized around delivering their business outcome, and trying to drive continuous experimentation.”

In real terms, Croker says this translates to a huge focus on cycle time – how quickly a business can go from having an idea to making an idea live.

“There's a very IT component to that, about how quickly the development team is able to produce and add it to the code base,” he says. “But also, how quickly the operations team is able to accept it, operationalize it, and get it to the point it adds business value.”

With this in mind, it's worth any organization remembering that there are roadblocks to the practical aspects of implementation. “I like to think of it that there is a set of things an organization can do to inhibit DevOps,” Croker says.

“That could be a governance process that allocates funding on an annual basis – poor flexibility is not going to help an agile delivery. Enterprise companies are finding they have their legacy structures, infrastructure, teams and processes, so getting to DevOps is slightly

harder,” says Hewlett Packard Enterprise’s (HPE) senior director of strategy for DevOps, Ashish Kuthiala. “There might be pockets of team that are doing DevOps successfully or across a couple of teams, but to scale this for an enterprise is an interesting challenge.”

Kuthiala says there are a few things that he and HPE have seen work well. “You need to have a mindset of working together, and understand we’re going to have some setbacks as we’re going on – there’s no prescribed formula to get from point A to point B in DevOps in X number of days,” Kuthiala explains. “We have huge teams that figured out, collectively, the common goal for us between the teams – and the kinds of metrics we want to keep improving over time. Let’s figure out what works and what doesn’t work for our organization in these loose boundaries – as long as we are making progress towards continuous improvement, and those metrics look good, and we’re trying to deliver better results to the business, it starts to work.”

Get the buy-in

It’s also critically important, Kuthiala says, to make sure executive management and leadership are supporting your efforts – they need to be clear on the concepts of failing fast and experimentation.

John Ediger, IT Strategist at HPE, says the company brought in a culture of continuous improvement that also got the management team involved in the concepts and benefits of DevOps. They focused on that, and every month each team would pick a small set of goals towards DevOps, and when they made that progress, they’d review their work, make adjustments, and set a goal for the next month.

“It all has to be based on what we’re trying to do in each group, on a set of organizational metrics and goals we’re trying to achieve, so it’s not just DevOps for DevOps’ sake,” Ediger argues.

“That cultural shift is the hardest part. We were absolutely finding it’s 80 percent practice and culture and 20 percent tools. So we started small and figured out how we can scale this, and then do the continuous improvement process. There’s three core things we’re driving to, and that’s collaboration, codification and automation.”

According to Mandi Walls, consulting director for EMEA at automation specialists Chef, there are a few steps organizations can take to address that cultural challenge. “Open brown-bag sessions – casual meetings over lunch – anything that helps share the kinds of work people are doing,” she explains.

“Open stand-up sessions, where anybody can join the team stand-up and learn about what each other is doing during the week. Regular reviews, and demos of things that they’re doing.

“Even for operations folk, who are like: ‘here’s the dashboard we built, here’s how the monitoring works’, those are the things that help put a face and a voice to the work that they’re doing. So they’re beginning to get an understanding of what the teams actually do, and to try to reach a more human conversation that there’s actually people involved in those teams.”

By persisting in bringing different teams together in pressure-light, low-stress situations, the hope is that people interact with each other in ways they might not be used to – think constructive conversations between teams rather than bureaucratic ticket-filing processes.

“You’ll have a more a more personal relationship, rather than silos represented by formal requests and all these things that are necessary when you don’t have a lot of interpersonal communication,” Walls adds. “The breakdown of that becomes much more person-to-person and conversational.”

DevOps 101

From a technical standpoint, Walls says the “baseline, 101-level of getting started” is bringing in system automation and the use of source code control mechanisms such as GitHub, where software components can be shared internally.

“It’s where you have this explicit documentation in those components, for everything that’s going on in the systems and all the source code is saved and available,” Walls reveals. “All the system configurations, similarly, are also saved and available – so you have a starting place where everything is shared and apparent to all of the stakeholders.

“As you become more proficient with those over time, most of the teams we work with end up with their goal being some kind of continuous deployment, continuous delivery situation – where they’re making small changes on a continuous basis and it runs through some kind of automated pipeline. That’s whether it’s our product, or Jenkins, or Go, or one of the others. To do all of the automated testing, provisioning and integration components, and then it just pushes out to production.”

Walls says this does take a certain level of technical competency to get the ball rolling. But provided that talent exists, is it perhaps worth bringing in outside consultants to put management and execs at ease?

“The term DevOps is a door opener at the moment, and is being used as such,” says Accenture’s Croker. “There are pitfalls and things to watch. DevOps is a change in which an organization behaves, so an external organization can act as a catalyst to help you get there and bring a new perspective – they can bring the tools and the tricks – but the really important thing is that they are leaving those behind, and leaving you substantially better than when they arrived.”

Croker adds that there are certain steps to take to ensure that the approach remains in place. “We make sure the teams understand what they are doing, we work through different ways of working, we speak about blame-free ‘postmortems’ and how we use them, and we support them for a period of time.”

Fixing bottlenecks

Specifically for having management on side, HPE’s Kuthiala says that even simple things like getting the VPs and directors from different teams together can be easier with a neutral third party to broker the conversation.

“I was in a customer meeting in which the VP of ops and the VP of apps had come together for the first time, to discuss how to optimize their end-to-end software delivery life cycle in the context of DevOps,” he says.

“The first question I asked both of them was: what drives your objectives? And the VP of apps said, well I get paid to incentivize rapid innovation in the marketplace, more frequent software releases. The VP of ops smiled at me and said: ‘My job is keeping the system stable. It can go down. The more changes you make, the more unstable my system can be, and I’m always scared you’re going to break something.’ They were completely

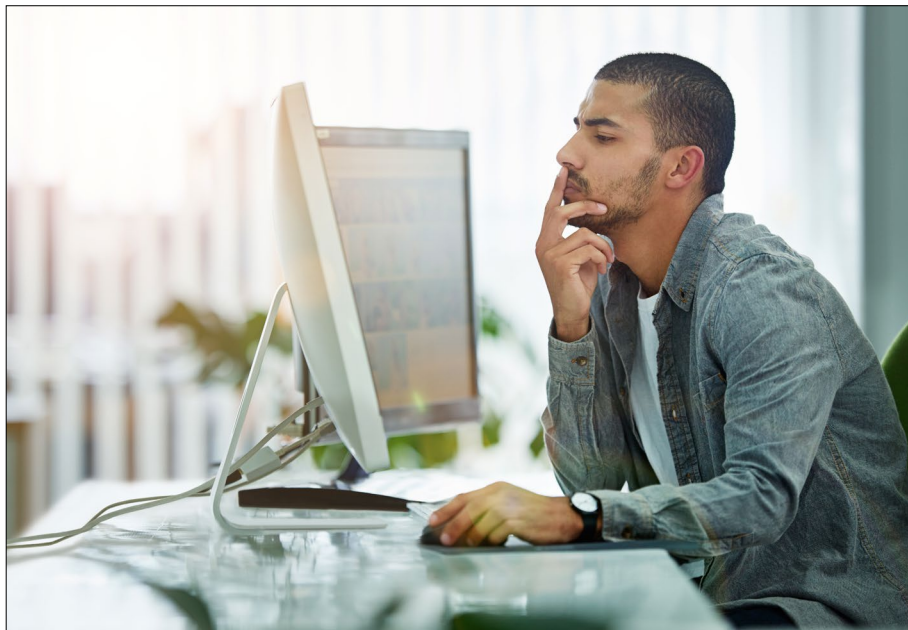
opposite metrics. Being in the same room can move that on to talking about how to get to a common goal. It's a big cultural thing, and that dialogue was not facilitated internally. They were at loggerheads."

And HPE's Ediger stresses the importance of 'rapid feedback loops' – the ability to immediately gain feedback on a new release and fix an issue, not just in development and operations but across many different areas of a business. "The feedback loop concept is fundamental whether you're in planning, or doing any kind of marketing or business venture," he says.

"If you tipped your mindset to: 'let's not create a long lead time' but instead shrink that down just like you do with a DevOps continuous delivery pipeline. Try to roll that out in small batches, get the immediate feedback. You can apply that pattern and principle to most business areas, and I think that's fundamental."

Kuthiala equates DevOps to physical manufacturing, where the concept has its roots. It's a useful metaphor for introducing agile or continuous development processes into a workplace. "Think about the car, where a team works on one thing and hands it over to the next person. At the end a car comes out. Doors are assembled, an engine is put in – software delivery life cycles are really not very different.

"If putting the engine in takes six hours, it doesn't matter how fast the chain before or after that goes, that's your bottleneck. In software, you can identify those bottlenecks in the same way. Where are the biggest bottlenecks, where you wait for testing for three months? If that's your biggest bottleneck, let's go solve that first, and then look at the next biggest bottleneck." [Tamlin Magee](#)



Credit: iStock

Alternatives to popular DevOps tool Jenkins

We round up your best options, both open source and proprietary

As an increasing number of organizations move towards a DevOps culture, Jenkins (jenkins-ci.org) has proved itself to be popular continuous integration software for developers since being spun out of the Hudson project at Sun Microsystems in 2011. However, it can sometimes prove unwieldy, and a new breed of tools are starting to emerge to challenge the incumbent in the space.

In short, continuous integration software allows developers to find and solve defects in a code base and automate some testing to speed up deployments.

There are plenty of alternatives to the open-source Jenkins software, including some proprietary options as well as alternatives on open-source licenses.

Our top picks are:

- CircleCI
- Atlassian Bamboo
- Harness

1. CircleCI

CircleCI (circleci.com) is a commercial provider of both an orchestration layer and a workflow tool for automating code changes for DevOps teams. Pricing is based on the number of containers and concurrent builds being run, starting at \$50 (about £37.50) per month for two, with unlimited users and builds included.

The company was founded in 2011 to provide continuous integration as-a-service for customers wanting to replace Jenkins if they didn't want to manage their own servers. It has since expanded to be deployed on-premise or in a private cloud.

Where CircleCI claims to stand out is in environments running at large scale, easing the burden on DevOps teams that they would encounter with Jenkins.

2. Atlassian Bamboo

The well-known vendor Atlassian (tinyurl.com/mtLkxuu) has a continuous integration tool called Bamboo. The tool can be deployed across Docker containers or for running applications in the public cloud with the likes

of Amazon Web Services. The tool gives developers code visibility and easily integrates with fellow Atlassian tools like Bitbucket Git code management solution, Jira for project management, and HipChat. Again, Atlassian claims that Bamboo is a more scalable option than Jenkins.

3. Harness

The newest entrant is Harness (harness.io), which came out of nowhere in October 2017. The company was established by Jyoti Bansal, the founder of application monitoring company AppDynamics, which he sold to Cisco for \$3.7 billion.

Harness essentially automates monitoring by leveraging machine learning to ease the deployment of software. To start with, it will seek to identify an application's baseline performance and automate rollbacks when things aren't running as they should be. The platform comes with analytics for DevOps teams to see what's happening across deployments and has integrated security and role-based access control.

The professional version costs \$25 (about £16.75)/month per service instance.

4. Shippable

Shippable (tinyurl.com/oy3433w) has developed continuous integration software, which can be run as-a-service. The software is declarative, meaning developers don't need to write as much code, they can aim the tool where they need it. The firm made its intentions towards the enterprise clear when it released Shippable Server in June 2017, though the product lacks the sort of granular security controls many enterprise customers will want.

5. Spinnaker

Netflix open sourced its continuous delivery tool Spinnaker (spinnaker.io) in 2015 and has since seen heavyweights including Google, Oracle and Microsoft join its community. Spinnaker runs on all of the major public cloud providers, as well as Kubernetes and OpenStack. The product is only in version 1.0 and may not be exactly enterprise-ready at this stage though, and it lacks the depth of integration complex enterprise customers may require.

6. TeamCity

TeamCity (tinyurl.com/Lerbqd6) is a proprietary commercial continuous integration tool from JetBrains. While it is quick and easy to set up, it can be a pain to configure and maintain, especially compared to some of the other as-a-service continuous integration tools on this list. It is a highly flexible tool, with tonnes of integrations, which can also bring with it added complexity, so may not be the best option for teams that aren't highly refined. The professional version of TeamCity is free for up to 20 build configurations and open-source projects can apply for a free licence.

7. Codeship

Codeship (codeship.com) offers continuous integration tools as-a-service, which can be set up in a matter of minutes if you are using Bitbucket or GitHub.

It's a rebrand of Railsonfire, which was founded in 2010 by Moritz Plassnig, Manuel Weiss, and Florian Motlik for Rails developers looking for continuous integration software that wasn't Jenkins. The company got its first customer in 2012 and subsequently moved to

Boston and rebranded. It is generally well reviewed over at G2 Crowd, with an average rating of 4.4 out of 5 stars, with many customers praising the vendor for its setup speed and responsive support across channels. The free tier is limited to 100 builds per month, though.

8. CruiseControl

One of the oldest continuous integration tools is CruiseControl (tinyurl.com/38bch), which started life as an internal project at ThoughtWorks more than 15 years ago. It has since struck out on its own on an open-source licence, and has a host of integrations out of the box.

9. Travis

Open-source continuous integration testing tool Travis CI (travis-ci.org) was born out of Germany in 2011 to automate testing for projects hosted on GitHub. It can be configured to run tests on both master developing branches as well as feature branches and even Pull Requests. It's limited to just GitHub repositories, though.

10. GitLab CI/CD

Open-source GitLab CI/CD (tinyurl.com/ybej35pq) can be run either as-a-service or behind your enterprise firewall on-premise or in a private cloud. It is multi-platform and multi-language and can scale automatically.

GitLab CI is part of the GitLab code-hosting platform, which means it naturally integrates well with projects there, allowing users to set up testing easily. Testing is run concurrently on GitLab Runner.

Both CI and CD are included with both the open-source GitLab Community Edition and the proprietary GitLab Enterprise Edition.

11. ThoughtWorks Snap

ThoughtWorks Snap (snap-ci.com) has been built specifically for cloud-based continuous integration and delivery, aimed at teams working exclusively in the cloud who don't want to manage any infrastructure. Pipelines can be set up from cloud platforms including Amazon Web Services, Heroku and GitHub. Snap is free for use with public repositories on GitHub, but will cost you if used with private repositories. [Scott Carey](#)



Credit: iStock

Capital One expands DevOps enterprise-wide

How a small team of developers proved the value of DevOps at one of America's biggest banks

Five years ago, a small team of developers at American financial services firm Capital One began experimenting with becoming more agile. Now, in an enormous leap, the bank has rolled out DevOps practices to all of its branches worldwide.

Speaking at the DevOps Enterprise Summit in London, Tapabrata 'Topo' Pal, senior director of agile

at the firm, explains just how significant this has been for the organization.

“The whole purpose was to remove manual steps,” says Pal. “So it started with a small team with small scope – all I want is to automate my build, and a few test automations in the pipeline.

“But success spread on our good news and reached the leadership, and at that point in Capital One there was a lot of transformation ready to take off.”

That transformation included changing the organization from one that outsourced its IT to working internally. There was a lot of noise about agile and working in a more lean fashion.

“It was good timing to put all these things together – the leadership had already bought into these practices from a conceptual perspective, but when they saw these things happen they bought in and the push came from the top: thou shalt go agile and DevOps.”

According to Pal, it took about three and a half years for DevOps practices to spread out from that one team to the whole enterprise. Now, aside from a few areas, most of Capital One is in-sourced and it’s pushing for an all-cloud environment by the end of 2018.

“About 40 percent of our production workload is on AWS now,” Pal says. “At 40 percent we are larger than Netflix on the AWS footprint, it’s huge. We are not running a hybrid model – our focus is everything on cloud. There are data centre-based applications being transformed, rewritten, re-engineered, thrown out, to get to cloud.”

Now Capital One is mostly an open-source tooling shop and automates what it can. The company found that using open-source tools allowed the teams to

contribute fixes quicker than closed-off solutions, and this offers clear benefits in shutting down bugs while also going to production as fast as possible.

It is, of course, phenomenally useful for enterprise-wide transformation if it's got that leadership buy-in. And the principles of DevOps really clicked with that leadership: speed, repeatability, the fact that nobody had to wait for source code to build and deploy, and the added trust of using tooling rather than humans.

"There were pockets of resistance but since we got leadership buy-in those resistances didn't last long," explains Pal.

To illustrate the time benefits of moving to the cloud, Pal mentions how long it used to take to get a server up and running.

"We had a manual process for building a server in production and I think we listed 62 steps, and it needed people from all different organizations," Pal explains. "They all had their own checklist, so we found that for a developer, to get a server I needed to complete these 62 steps and none of this is owned by me.

"It took anywhere between two to six months to get a server. Now, developers are just clicking a button or running a command line tool, the ROI is right there – and if I don't need that server, I just shut it down, as opposed to if I had a server before I owned it for a lifetime. To maintain that, manage that, or if the server is not up to date with patches I get nasty emails. I don't get that anymore, if I don't need a server I kill it."

Pal adds that there were challenges in achieving the cultural mindset across the organization – of course, this is to be expected with any global company that has thousands on the payroll. There were a lot of silos

that needed to be broken down and handing over projects between teams, but in addition to this, a lot of retraining was needed, too.

“Not only organizational change but also to retrain people – right now we do not have a Unix admin because there’s no Unix admin needed in the cloud,” explains Pal. “What do you do with those people? There’s no admin as such, so a lot of retraining, education, certification, pushing people to get up to date. It’s a lot of these kinds of silos and how to get people embedded into the team that requires a lot of work from a people-management perspective.” [Tamlin Magee](#)



Credit: iStock

ITV embraces DevOps for product development

ITV head of Common Platform, Tom Clark, reveals how VM provisioning time has been reduced from days to minutes

ITV has embraced DevOps across its organization to support development, enabling the broadcaster to create digital products more rapidly and with greater reliability. The DevOps work centres around the creation of what ITV calls its Common Platform. This is a “standardized coding toolkit that gives teams across [ITV’s Technology division] a common way of doing

things like: logging, monitoring, alerting, deployment pipelines, configuration management”.

Starting off as a pilot project to create a version of its ITV Player online service for connected TVs three years ago, the DevOps team has since grown from three to 15 engineers, with 16 products now reliant on the shared Common Platform.

The benefits of the DevOps approach are clear, according to Tom Clark, head of ITV’s Common Platform.

“We are doing more changes in parallel than ever before – VMs in minutes not weeks, and greenfield hosting environments in days not months,” he says. “I have one engineer who started one Monday and by the time he joined the product team the following week, he had built an entire hosting platform and he literally started and had never seen that tooling before.”

Clark adds: “Performance has improved with more eyes on a problem, and reliability is up as well.”

From cost-cutting to modernization

This was not always the situation for the company. Back in 2010, ITV was in financial trouble, and a new CEO was brought in to cut costs. “One of the things he did was to outsource our infrastructure to a managed service provider, which saved a lot of money and helped save the company,” recalls Clark.

The outsourcing arrangement had drawbacks, however, and meant that virtual machines took ‘weeks’ to provision. “That was 2010: terrible times,” he adds.

ITV initially began adopting a DevOps approach in 2014 to support a project to create software for connected TVs. “When we did the big cost-cutting we pressed pause on all of our internal development,”

says Clark. “After a while we realised that actually we were slipping behind the competition. They had cool new fancy systems and ours kind of stopped five years ago. So we did a modernization programme.”

The project, led by another member of ITV’s IT team, served to highlight the barriers the company faced in order to operate at speed.

“He got some devs and he got some ops, and lumped together,” Clark explains. “He requested some VMs and he waited and waited and he waited some more, and eventually he got so annoyed with waiting for these VMs from the MSP he went on to Amazon, got his credit card out and built the entire stack in about four weeks.”

Automation and standardization

While the developers had been moving from waterfall-based development to agile development, the delivery of infrastructure wasn’t fast enough. “We still had a very waterfall infrastructure, and we realised it wasn’t really going to work,” Clark says.

ITV grew its Common Platform approach to address this. It was built on a few key principles, and one is the use of automation tools.

“We want to automate the boring and repetitive stuff, take the humans out of the equation, because when you make smart humans do boring repetitive stuff, they get bored and they leave or they make mistakes,” Clark says. “It also means that they can concentrate on the interesting fun work that makes them happy.” He adds: “The great thing about automation is that the more you automate, you have more time, which means you have more time to automate, which means you have more time. So you get this virtuous infinite loop.”

Standardization was also key to greater efficiency.

“Standardization allows you to make assumptions,” he says. “If you all agree that you are going to make square pegs of a certain size, when someone else makes a square peg you know can go into their system and take their square peg and fit it into your square hole because you have all agreed on that standard.

“What that means is you have multiple teams developing stuff, you can cherry-pick from all of their systems and you basically have all of their work for free. It’s kind of like a rising tide lifts all boats.”

Since the introduction of the DevOps approach, the methodology has gradually become more deeply embedded in the IT team. Alongside a ‘core’ team of engineers – the ‘first responders’ who are responsible for maintaining the common platform, as well as other areas such as R&D – there are engineers in each of the five product development teams.

There are also plans to extend the Common Platform across the rest of the IT estate at ITV, specifically its Windows-based systems.

Overall, Clark said there have been benefits for staff and for the wider business. “The result: happy people, happy engineers, happy devs, happy products owners, happy directors, users and customers,” he enthuses. [Matthew Finnegan](#)



Credit: iStock

Sky Bet on doing DevOps right: learn from the best

Why there's absolutely no shame in getting stuck into the community and learning from the best examples that others set

A top-down reorganization of Sky's gambling wing, Sky Bet, completely transformed the way it operated – moving from a kind of gaming-centric hosting service into a DevOps-friendly developer house intent on releasing as fast as possible and open to experimentation. “We ripped Spotify off,” says Gavin Harris, engineer manager for platform services at Sky

Betting & Gaming. “We copied them, and that’s kind of the model we’re still using now – small, autonomous teams that have ownership of services and do what they will without outside interference.”

It was quite a drastic change. The company had roughly 100 employees and did ‘everything’ manually – and its model was to take software releases from third parties, deploy these, and run them.

“We wanted to move to a more DevOps and agile approach because we wanted to be first to market with features, more quickly than our partners,” Harris explains. “So we started creating internal development teams. The first one that we wanted to do was bringing the betting website in-house – the bet website was where the first in-house code was written, and it allowed us to start iterating our site much faster.”

Sky Bet began emulating that approach across all the other divisions in the business, including bringing the application front-ends in house, and has started to fold the middleware and APIs in too. “Really these teams create services and have other teams using them,” Harris says.

The configuration management suite from Chef is “central to everything” the company does.

“We use Chef for our infrastructure automation, for building applications, for driving APIs – what we want to do is give a developer a toolkit that has everything that they need to get started,” Harris reveals.

“It really starts with code generation – we supply Chef’s cookbook generator to our developers and they can get started on an application very quickly, with all the compliance and everything built in from the get go. They can just start with their application, they don’t

need to start with: ‘How do I get a server?’ or: ‘How do I apply this shared configuration that everybody needs to run on and add service to it?’ It’s just there by default because of the toolset we use, and because of Chef.”

Selecting DevOps tools

When the business began thinking about practising DevOps, it looked at all the configuration management suites available at the time, including Puppet, Ansible, Chef, Saltstack, and CFEngine.

“We had a look to see what capabilities they gave us, and which was the most appropriate for our fit,” Harris explains. “We went with Chef because it’s very extensible, and very powerful – you can basically get it to do whatever you want, and extend it to do whatever you want. That extensibility is very important to us, so that’s why we went for Chef.”

He adds that the community was a factor, too – at the time Chef’s contained the largest active community and contributor network, and it probably still does. “There’s a lot of sharing and collaboration that goes on,” Harris adds.

“Previously, you wouldn’t really get events like this, where you get different companies telling their stories, and even horror stories. That’s a bit different to how it was 10 years ago – you were inside big enterprise and you did things the way enterprise told you.

“I think things have changed a lot,” he continues. “And practising DevOps has been absolutely critical for keeping people invested in services and passionate about them. It gives people ownership of the things they’re working on, and it lets people be more passionate about it. Certainly for us, because they’ve

got ownership of those products or whatever, naturally people feel more close to it – it's their thing they're looking after, they care about it.”

It can be easy when you're elbow-deep in the DevOps community to assume this relatively new way of working is more widespread than it is. Harris offers a few morsels of advice for businesses like SMBs which have seen the noise surrounding DevOps and wish to explore it further. “Look at what other companies are doing and rip them off,” he says. “It's what we did five years ago, and it's not exactly the same as it was five years ago, we've tweaked the model to suit how we want to do things. But look at how other people are doing things and use their experience to drive your own DevOps revolution internally.”

But just how would a business find out which of their competitors to rip off? Unusually, the DevOps community is incredibly open, and it's easy to get involved. “Come to things like Chef Summit, Chef Conf in Texas, go to conferences, go to your local DevOps meetups, there's usually one in each major city in each country,” he reveals. “Go to those things and talk to people, see what works and doesn't work, and which ideas you should or shouldn't copy. There are people out there who have tried to do these things, so take advantage of that.”

“Don't be ashamed to do it,” he insists. “Follow the example other people set.” [Tamlin Magee](#)



©2017 International Data Group.