

Who needs DevOps? Some of us actually...

Platform.sh is a Killer App PaaS, that lets you implement best practice DevOps with your eyes shut

INTRODUCTION

DevOps are a set of principles and process management practices that aim to make software building, testing and release management fast and reliable, and when performed well they can give your company a competitive edge, as the Puppet Labs '2016 State of DevOps Report' clearly shows. And as long as these principles are understood and are being implemented through productivity tools and/or automation, you will be on your way.

Platform.sh provides automation that removes all problems associated with the administration of consistent environments, which as a result improves all related activities, including integration, testing, deployment, scaling etc. And it is the confidence that comes with this that allows you to think differently about your working practices and your service offerings, and possibly gain a competitive advantage.

An example of DevOps related to one simple change to live

When a developer pushes a change to QA, and says 'it's been tested so it will work', but doesn't tell operations they were using their own database and that caching was off, that's when the problems start, because the testers find a working feature, and duly push it to live out-of-hours. The production configuration also

happens to be slightly different to QA, and strange things have started to happen the next morning when the site starts getting busy. By late morning, and after many customer complaints and several tickets, operations narrow it down to a handful of culprits, they take the site down to reverse out all the changes, and the developers begin investigating the code to see what caused the incident. And so the process restarts all over again.

Better DevOps practices could have easily prevented the disruption and all that waste of time. By the way, the problem lay in the configuration difference between QA and live. Anyway, now multiply this incident up across a team of 25 deploying 80 changes a week to 3 business critical applications.

You're probably experiencing various issues and complications that you sense are down to irregularities within your general development & deployment processes. The big question is 'can you make a significant impact here without years of effort?'

The answer is 'YES', and Platform.sh makes this a great deal easier and quicker.

We've been here before

Organizations are always looking to speed up the delivery of value to their customers, whilst

improving service consistency and reducing cost, all of which are brought about by better processes. In today's world, information technology (IT) is present in nearly everything we deliver as a business, so the many different processes that exist between 'the build' (ie. developers) and 'the product/service' (ie. the live system) - namely development to operations, or DevOps - have a hugely important significance to the customer experience. And so, it stands to reason that the adoption or improvement of DevOps should be seen as critical to your competitive success. And because there is so much IT behind our product offerings these days, DevOps principles are actually as business-affecting as Lean principles were to manufacturing in the 1970's and 80's. Manufacturing may no longer be the dominant value creation process in today's world (IT is); however many of the processes improvement principles are the same.

Many of us probably do not know too much about Kaizen, Continuous Improvement and Just-in-Time, suffice it to say they were groundbreaking principles at a time before many of us we were born, have since been iterated to perfection, and are at the heart of most physical products we consume today. Think about it, a PC once ordered over the phone took months to deliver, now it's next day off the web, and all down to highly successful process improvement. DevOps is just one of the current names for a stream of process improvement within the busy arena of IT that we all have to work with.

Platform.sh takes DevOps process improvement a step-change further, automating a cluster of key processes that maintain consistency between environments, the result of which is massively faster delivery of value to customers, and at a much lower cost.

THE BUSINESS VALUE OF ADOPTING DEVOPS

Firstly let's frame this topic. Is DevOps seen as a 'capability' within your organisation, and therefore being the best at it makes it a competitive advantage? Or is it seen as an administrative burden - a hygiene factor - that just has to be done to make things work properly? The answer partly depends on whether IT is one of your dominant value creation processes, and is also an important (or even the primary) means of customer acquisition?

Whatever it is to you, there is a lot of evidence out there to suggest that doing it properly does create competitive advantage, giving you faster feature time-to-market, increasing customer satisfaction, market share, employee productivity etc.

One of the hallmarks of high performers in any field is that they always "accelerate from the rest of the herd". In other words, the best always get better. In contrast, organisations that take weeks or months to deploy software are at a significant disadvantage in today's marketplace.

This constant and relentless improvement in performance is happening in the DevOps space, too. In 2009, 10 deploys a day was considered fast. Now that is merely average. In 2012, Amazon went on record stating that they were doing on average, 23,000 deploys per day.

What are the 3 types of work that impact business value?

First, let's think about the 3 main types of work that drive your company's IT agility, defined as:



The queue of planned customer-facing application build projects, and internal IT improvement initiatives



...coming from the planned work above, usually through an issue management system



Unplanned work coming from 1 and 2 above, and almost always at the expense of planned work commitments

DevOps and automation usually has the end goal of enabling the fast flow of planned work into production (eg. hundreds of new features/changes per week) whilst preserving stability, reliability, availability, performance and security. But where there is less attention paid to putting the right DevOps practices in place, output from development teams will inevitably cause problems downstream in QA, operations and security. So, planned work put through poor DevOps processes results in unplanned work, and the more planned work that goes into the funnel, the more unplanned work comes out.

PUPPET LABS "STATE OF DEVOPS REPORT"

This 2016 report (i), benchmarks 4,600 IT professionals and organisations, with the goals of better understanding the health and habits of organisations at all stages of DevOps adoption.

The first surprise was how much the high performing organisations using DevOps practices were outperforming their non-high performing peers when comparing commonly recognised metrics, see below:

AGILITY METRICS

200x
more frequent code
deployments

2,555x
shorter feature
deployment lead times

RELIABILITY METRICS

3x
less failed
deployments

24x
faster MTTR (Mean Time
To Resolution)

In other words the high performers were much more agile. They were deploying code 200x more frequently, and the time required to go from "feature requested" to "successfully running in production" was 2,555x faster. High performers also had lead times measured in minutes or hours, whilst the poorer performers has lead times measured in weeks, months, or even quarters.

And not only did the higher performers get more work done overall, they had far better outcomes:

- when they deployed code changes and new features, they were 3 times as likely to be completed successfully (ie. without causing a production outage or service impairment), and
- when the change failed and resulted in an incident, the time required to resolve the incident was 24x faster.

So it's easy to see from some very strong evidence, that great DevOps most definitely should be considered a competitive advantage.

In addition, the report claims that high-performing IT teams spend 50% less time remediating security issues, as well as spending 22% less time on unplanned work and rework....compelling stuff!

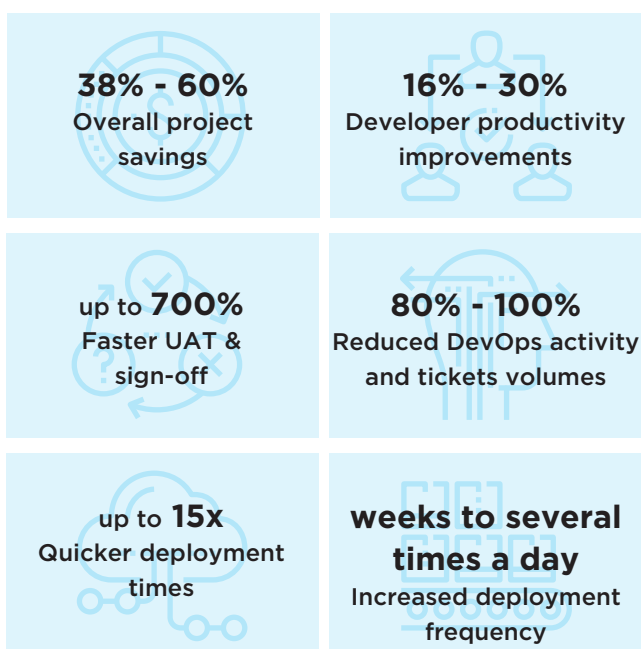
The Platform.sh effect on planned and unplanned work

Platform.sh is a Platform as a Service (PaaS) hosting offering. It introduces unsurpassable consistency within the

(i) Puppet Labs "2016 State of DevOps Report"
<https://puppet.com/resources/white-paper/2016-state-of-devops-report>

development-to-production workflow process, for any and all technologies supported. This reliability elevates confidence and trust to such levels that Continuous Integration (CI) processes can be simply extended to implement truly seamless Continuous Deployment (CD), accelerating the flow of planned work to a constant stream of change into the live service.

Platform.sh is seeing tremendous results, with many hundreds of customers citing huge productivity and performance improvements, as follows:



When a technology affects your IT organisation in such an impactful way, it allows you to change the way you think about your product offering, and therein lies your potential competitive advantage. As an aside, these Platform.sh productivity improvements elevate the organisation from their current position in the Puppet ranking, and up towards the 'high performing organisation' end of the scale.

How does Platform.sh achieve this ?

Firstly, our snapshot technology allows your cloud to create identical copies of running customer applications within seconds. It keeps the code, files, services and data in lockstep,

thus maintaining perfect consistency between environments. This is valuable for a live service, but even more so for development teams who can create and test every change and new feature in its own exact copy of the production environment. And in order to build maintainable test suites, it's essential to have an effective strategy for creating and maintaining data sets for these tests. Many customers have told us that managing test data is a major source of pain, and Platform.sh is the only system that fully automates this, completely removing this pain-point. So once something has been successfully tested in development, you know with certainty that it will behave exactly as expected in production.

And it is only when we have a high degree of confidence that a new feature is working as designed, do we roll it out to customers, and only when we create a repeatable process that makes the testing and deployment routine, are we able to achieve Continuous Deployment (CD). Platform.sh elevates confidence and certainty in the workflow to the point where stakeholders can now put changes into the process - regardless of priority or complexity - and expect them to appear in live in a very short space of time.

The high level of trust gained by using this new approach comes from repeated and frequent upstream deployments to say staging/QA, ie. your copy of live. So when the code actually reaches live, the process has been repeated so often you know it's going to work as expected, as opposed to relying on our product marketing being true. And this is because with Platform.sh, every time you test a new feature, you also test its deployment, so you are not only assured the code works as expected but also that applying the change happens as it should. So now, even critical or complex changes become routine, and your new approach to CD allows many small changes to be constantly implemented into live throughout the day, as opposed to everything being saved up for the periodic big

push. Nothing should now go wrong, and zero time should be spent diagnosing which changes caused a problem.

Other important considerations - aside from hugely improving what you do now, and achieving Continuous Deployment

1. Platform.sh offer Systems Administrators the ability to set the rules and permissions on how things should work, and then leave the developers to the system. They still get to influence best practice DevOps!
2. Platform.sh also enables Continuous Delivery allowing you to implement features in advance of them actually being available to the customer. Coupled with implementing feature toggling and A/B testing this gives project teams the full array of options over the development cycle from lean growth-hacking to a planned release schedule.
3. Without automation, DevOps is a combination of process, tools and culture sitting across the development, QA, IT Operations and InfoSec functions. Making this work smoothly is complex and difficult, so replacing a large process block within the overall system with Platform.sh considerably reduces the total cost & effort required to become one of the high performing organisations in the Puppet report.
4. In the chaotic world of the low performing organisation, deployments may be regarded by deployment managers as an exciting / stressful / risky period activity, and conducted outside normal hours with lots of people on standby. In the Platform.sh world, code deployments and infrastructure changes, upgrade and scaling become routine, happening during the working day and without incident.
5. Platform.sh refocuses the process of continual improvement across many activities away from the administration of infrastructure, and onto development and enhancing the relationship between features and business value.
6. Platform.sh gives you all the benefits of a full DevOps chain, as a service, with zero upfront cost, and no operational overhead to maintain your custom-built toolchain and infrastructure.