BizDevOps Done Right!

An excerpt from About:Performance blog

I was recently moderating a webinar called "BizDevOps Done Right!". In the session, two of my colleagues, Andi and Klaus, were discussing BizDevOps and how the concept can help organizations quickly address demanding innovation cycles, and dramatically reduce time-to-market for new features and better end-user experiences. Let's take a look at 3 common problems and how monitoring redefined can help you resolve them.

About the Author

Jilene has been working in global IT infrastructure, software, cloud and mobile technologies and apps for 20+ years. She is a regular moderator, writer, and creative communicator supporting the DevOps Activists at Dynatrace from a global marketing perspective.

@jilenethomas

For the complete version of this blog, go HERE

Best Practices

The 2016 State of DevOps Report cites that high performers are 2X more likely to exceed profitability, market share and productivity goals. This is the holy grail of how companies can truly use digital channels be more competitive.

The 10,000 foot view of BizDevOps

Let's take a look at a 10,000 foot view of BizDevOps. In DevOps, you are building across the pipeline from Dev to Ops using automation and quality feedback loops, and pushing features from left to right faster to meet business demands.
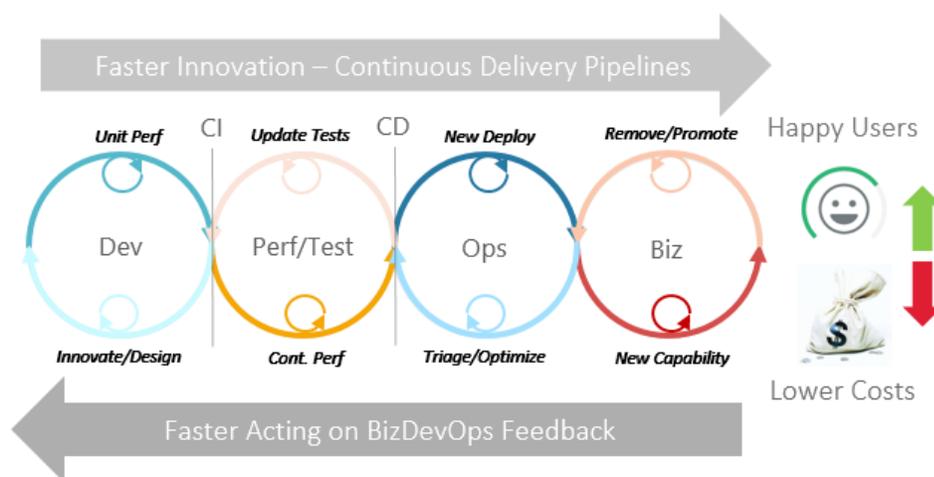
But what if end-users are not using what we are building? Not only is it a big cost factor, it does not help drive business success. For years it was to give R&D the right feedback to make improvements or enhancements, but dev, ops and the business were in tight silos and that didn't happen. The concept of BizDevOps breaks down these silos and quickly connects important end-user customer data into the development feedback loop. In addition to happier end-users, this increases opportunities for innovation, new revenue growth, and potentially more brand exposure beyond local markets.

Fun Fact, too: Andi Grabner, DevOps Performance Expert, and speaker on the session, gave a great example of this point from a discussion he had with Facebook. He learned that every time they push out a new feature, they also define a success criteria. If the feature is not used by, for example, 10% of the Facebook population within a certain time-period, it is not considered successful and therefore it, and associated costs, are removed.

Quality suffering with speed

While dev teams try to meet the speed of business, the Puppet Labs and DORA "2017 State of DevOps Report" notes that when dev teams try to meet the speed-of-business changes, quality suffers. This is a DevOps hurdle for low performers who want to become high performers.

Table 1: Changes in IT performance of high performers, 2016 to 2017

IT performance metrics	2016	2017
Deployment frequency	200x more frequent	46x more frequent
Lead time for changes	2,555x faster	440x faster
Mean time to recover (MTTR)	24x faster	96x faster
Change failure rate	3x lower (1/3 as likely)	5x lower (1/5 as likely)

In the session, Andi explained three primary reasons for this:

- DevOps promotes choice: you can pick the best stack for your problem, or in reverse, don't let your stack dictate which problem you can solve and how. This element of choice increases monitoring and managing technical complexity.

- DevOps is driven by business needs and speed: when your urgency is faster-to-market versus quality-to-market, you can end up with bad data and code.

- DevOps promotes small and agile — by putting services into containers we have small teams delivering small services — it feels like building little silos — and this chain of service interactions creates difficulty in analyzing problems across the total app delivery pipeline.

Klaus Enzenhofer, DX Expert, put this into a great visual story that illustrated end-user touch points using a "real world" frequent traveler he called Max Platinum. Max is booking a flight, checking in, going to the lounge, and then going in-flight.
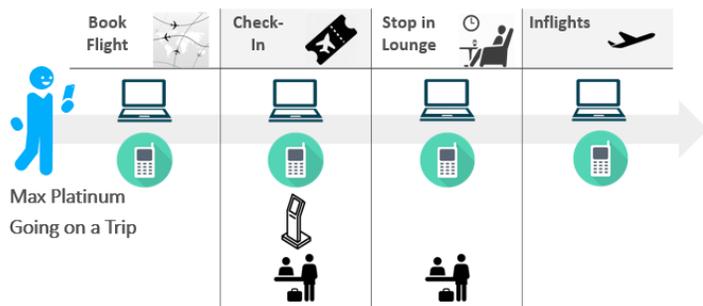

From a technical point of view, there are a lot of digital touchpoints interacting with Max. Now let's lay-over Andi's complexity view, the technical side, to these same transactions by example:

In Andi's overlay, we have small teams, working with the stack of their choice. There may be a geo-service using Node.JS delivering weekly; there may be a product team delivering late summer; and on the other side of the speed spectrum, a card service that is releasing daily. Add in check-in service on demand, and mobile teams doing things separately with their own development cycles. Here are the little silos Andi mentioned earlier.

If Max can't check in, troubleshooting this problem is a big challenge. From a support call perspective, what is happening and where? All channels need to be working optimally to ensure a good customer experience with your service and your brand.
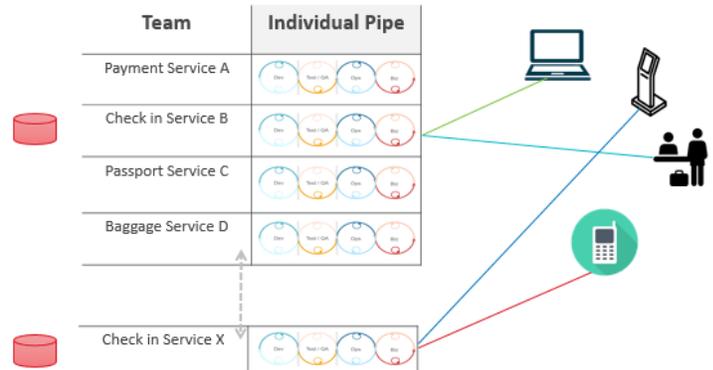
Breaking down silos and increasing visibility at the end-user level

Through the lens of BizDevOps, monitoring needs to be a continually feedback loop across the whole delivery pipeline — the 10,000 foot view we looked at earlier — with the ability to see every single user, every single app, every single touch point

There are a lot of monitoring tools out there that say they can help with this, alert on exceptions, slowdowns, but they miss the actual human being out there, the customer, that is being impacted. You need to be able to fix problems that matter most to the customer, to your business, first.

dynatrace

Let's take a look at how simply this might work using Dynatrace's AI-Powered monitoring for BizDevOps.

On the business side, in #1, Dynatrace shows a problem card with the end-user and business impact defined: you can see how many real customers are being impacted and from which applications, but this is only a $5.00 problem based on priorities. #2 shows a multi-million-dollar problem making it much more critical to take a deeper look. Where are people bouncing off? Where are we losing them? What is wrong with conversion rate? #3 shows conversion rates going up but your completions are going down. What is the impact of this? Is it because of different user behavior, what is that? Are there technical issues?

On the operations side, AI automatically analyzes the problem for me. I don't have to look at Node.JS data, AWS data, Google Cloud data, Dynatrace automatically does this for me. I will get the same problem ticket with the technical details already identified from Dynatrace SmartScape, full stack monitoring, base-lining and artificial intelligence. This gets the problem solved faster, saving time and resources, as well as avoiding possible negative business impact.
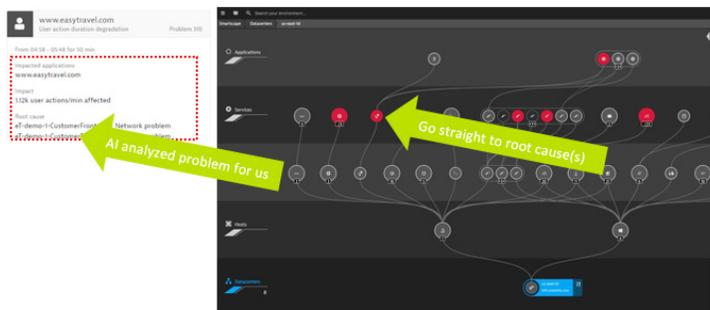
Let's circle back to the developer. We have all the data available that allows developers to gain precision and clarity around issues. Is this a JavaScript problem? Is it a third-party issue with a third-party API, or is it something in my code? So, no thinking — is it business, or operations or dev? It's clear and the details are spelled out for the dev team.

And from a DevOps perspective, does your developer want to be woken up at 2:00 a.m. if 300 people are impacted? Yes, if there are only 400 on the website, but if I have 5,000,000 and 300 are impacted, it is probably not that critical.
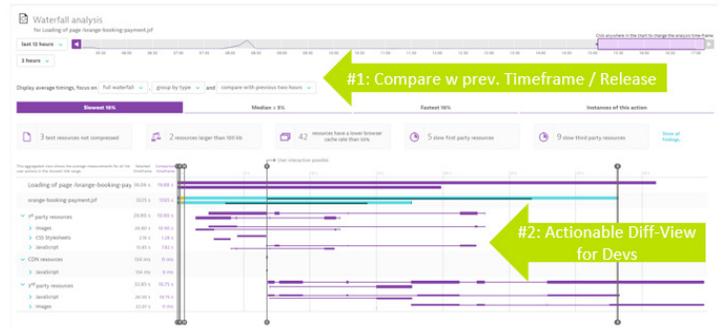
So what we've looked at are three problems and resolutions.

1. Technical complexity challenge — Dynatrace OneAgent Full Stack monitoring solves this problem. It breaks down the silo's that were introduced and give deep visibility across the whole application delivery pipeline. So if you have a multi-technology environment, there is only ONE AGENT that needs to be installed on your host. And whether it is Node.JS, or whether it is Java, .NET, PHP, or things outside of your data center, mobile applications, websites, whatever it is – Dynatrace can provide visibility across the whole environment.
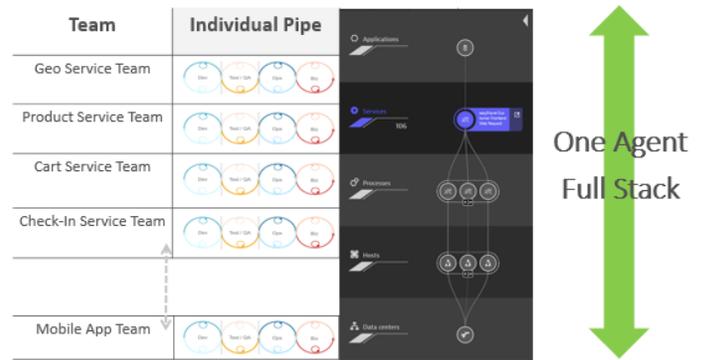
2. Bad code quality — in your DevOps tool chain, Dynatrace can help you to Shift Left and start to detect all potential performance, scalability and architectural problems early, with the right information and all the details down to the root cause. If you've already deployed, Dynatrace can help with auto-healing. All of this gets you closer to building resilient pipelines and systems by baking monitoring across the end-to-end application delivery chain.
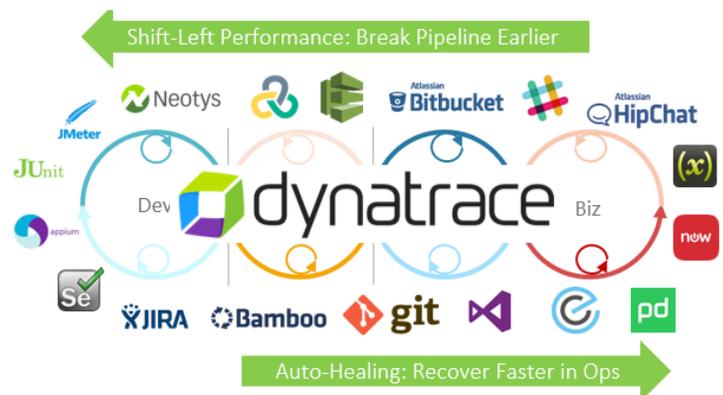
3. Data silo challenges — just like the earlier focus on Max Platinum, the premier travel customer, we can now look across all these services, across all the channels and devices, and get a view into the business that allows us to analyze the true user behavior — this creates insights as well as gives the technical folks information to remedy problems with user experience.
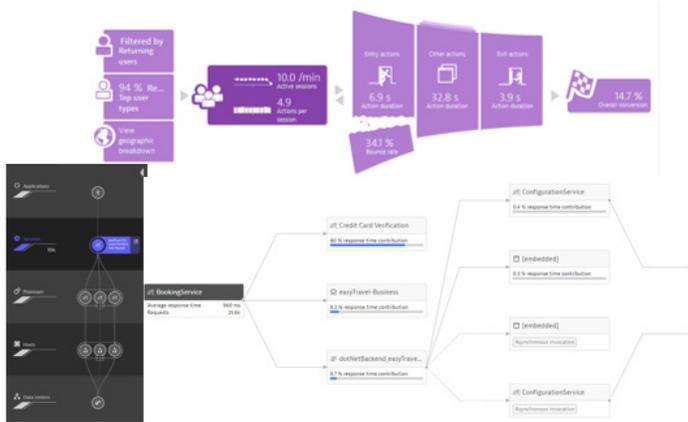
This is what is needed to be baked into every application delivery chain.

Resolving these 3 issues leads to the ultimate goal of development, operations and business teams working together on a single data cube. WIth Dynatrace AI-powered analysis, avoid a lot of data overload about the performance of your application delivery chain and get only the data that you need to refine and continue to innovate.

You can still choose your technology stack, your pipeline — and if you deploy once a week, once a month, what works best for you — all of the data is captured in one product. The AI technology alerts you if it is a big problem, or a small problem, and where it is, and what is the impact. So in the end, you are building stuff that makes users happy and doesn't explode costs, yay!
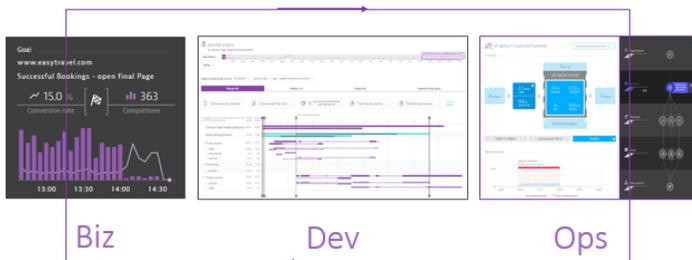
To learn more about BizDevOps, please watch the webinar replay "BizDevOps Done Right", or check out our DevOps Journey Series, or visit our web page, Scaling DevOps to Deliver Better Software Faster.

Dynatrace Digital Performance Platform — it's digital business...transformed.

Successfully improve your user experiences, launch new initiatives with confidence, reduce operational complexity and go to market faster than your competition. With the world's most complete, powerful and flexible digital performance platform for today's digital enterprises, Dynatrace has you covered.

Learn more at dynatrace.com

@Dynatrace   fb.com/dynatrace

dynatrace