dynatrace

5 ways to use performance monitoring to make DevOps a success

A case study: How Prep Sportswear moved from monolith to microservices

© 2017 Dynatrace

Table of contents

Introduction 4

Section 1: The Prep Sportswear story 5

Section 2: 5 ways digital performance monitoring helped Prep Sportswear 9

Section 3: The way forward: Microservices 11

Your next steps 13

© 2017 Dynatrace

Our authors:

The DevOps experts

Richard Dominguez

Developer in Operations,
Prep Sportswear

Online: prepsportswear.com

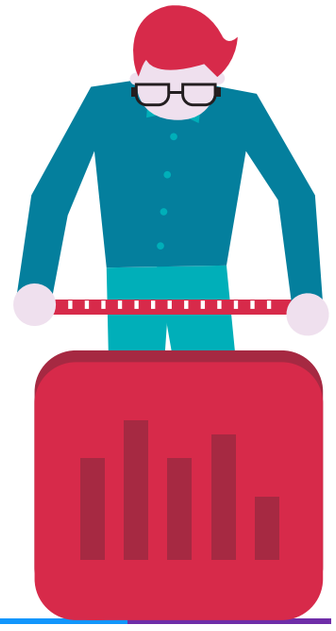
RD Richard has over nine years of experience as both the Systems Analyst and Software Developer in Test. He has worked on many high-profile projects in Microsoft, such as Hyper-V, Windows 7 Client Performance, and Windows phone services. Richard is now a DevOps Engineer at Prep Sportswear. His responsibilities include site reliability, external synthetic testing, release management, and overall site performance. To hear Richard's part of the story, look for the purple talk bubble with the initials RD.

Andi Grabner

Performance Advocate,
Center of Excellence,
Dynatrace

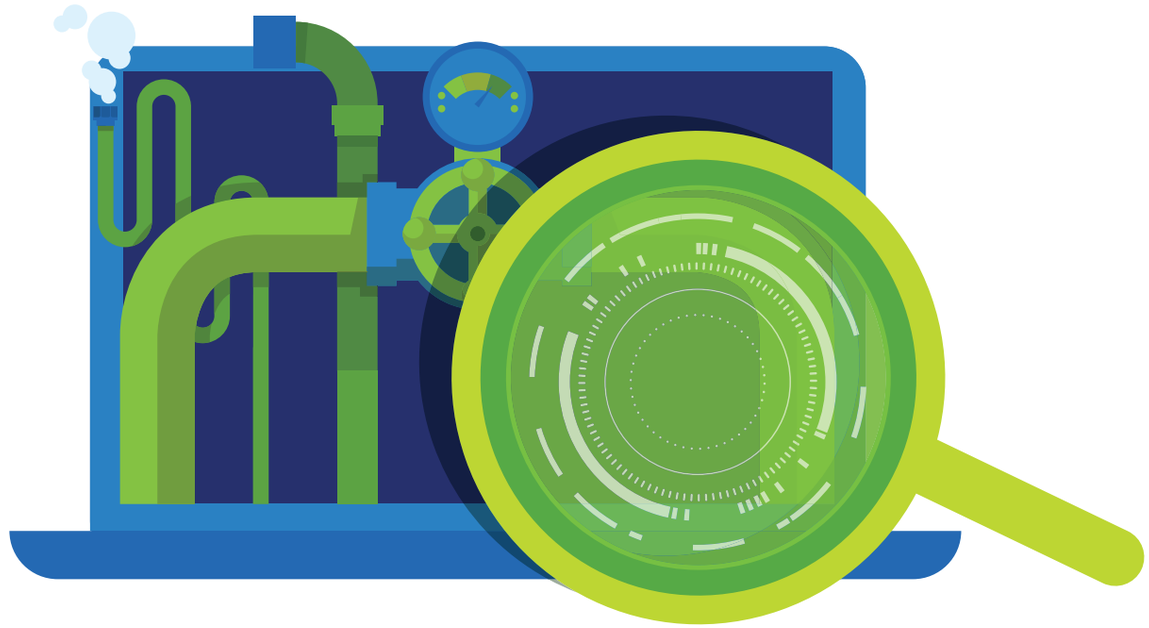
Blog: blog.dynatrace.com
Twitter: @grabnerandi

AG Andreas Grabner has 15+ years' experience as an architect and developer in the Java and .NET space and is an advocate for high-performing applications. He is a regular contributor to large performance communities and a frequent speaker at technology conferences. To hear Andi's thoughts, look for the purple talk bubble with the initials AG.



© 2017 Dynatrace

The promise of DevOps



AG I remember when I first talked about DevOps a couple of years ago. I actually got fingers pointed at me.

People would say, "Hey, because of people like you, DevOps is on every single piece of product collateral. Nobody knows what DevOps is all about anymore." I have to agree with them. DevOps is about doing the hard work of changing culture. The tools only play the supporting role.

I think the reason why people are so crazy about DevOps is that they know there is a major disruption happening. It is companies like Uber, Facebook, Alibaba, and Airbnb that totally change the way existing industries operate. They are very fast in providing new features and services to customers.

Now, there is even more disruption ahead of us with the Internet of Things (IoT) taking off. And the question is whether your organization is ready to respond to new use cases quickly. Many companies aren't. They cannot compete with these software-defined businesses. That's why the life span of corporations are getting much, much shorter.

A lot of us want to get to the new way of developing and deploying software. These leaders, the unicorns, are not doing two deployments per year, they are doing multiple deployments per day and this is delivering value faster to the consumer.

Unfortunately, a lot of organizations are still failing. They are not start-ups like the unicorns. Many are well-established companies. They are not building a new platform where they can use micro-services from the beginning. Established companies without DevOps principles put themselves on a treadmill when they try to automate processes without any of the other work that goes with it.

This eBook will not tell you the story of a unicorn. It will tell you the story of a real company just like yours, with real challenges building over years of operations. It will tell the story of how Prep Sportswear struggled, the steps they took to move the organization towards DevOps, and how performance management helped.

© 2017 Dynatrace

Section 1

The Prep Sportswear story

© 2017 Dynatrace

Super heroes, half-done projects, and the monolith

RD About 10 or so years ago, our CEO Chad Hartvigson started Prep Sportswear by personally pressing customized shirts for schools in his parents' basement.

He quickly moved on to launch a website, but as customer demand increased, so did the website's code base. With the rapid development required keep up with new features, stability suffered.

Our company became very accustomed to heroes. There was no complete picture of how the website worked, and no process to find out the source of the problem. We had one or two of these "heroes" that fixed everything. But, guess what? That very same issue occurred again the next day, or the next week. It wasn't fixed — they had just applied a Band-Aid.

Another result of the fast development was a large buildup of half-done projects. The development team started many projects over the years that were not truly implemented. They were either never tested or the code pointed to some future remote service that was never implemented. As time went by, the monolith started to grow. Monoliths don't start as monoliths. They start as a single application on a single web server. People say "Let's just start with something, and we'll fix it later." It's that "fix it later" mentality that creates problems.

That's what happened at Prep Sportswear. Breaks were happening more frequently and they were completely unrelated to what was deployed. This happens a lot when you have a monolith because everything is twisted together in unknown ways.

The team was making two to three deployments a week at this time. And, how often were there issues? Two or three times a week.

The answer at the time? "Let's throw more hardware at the problem."

People say "Let's just start with something, and we'll fix it later." It's that "fix it later" mentality that creates problems.

© 2017 Dynatrace

A new beginning

RD

We had some hard decisions to make.

The first decision was that our CEO brought in the Director of IT — a veteran who knew how to build highly scalable and successful web services. And, then we added individuals to the team that understood what changes needed to be made.

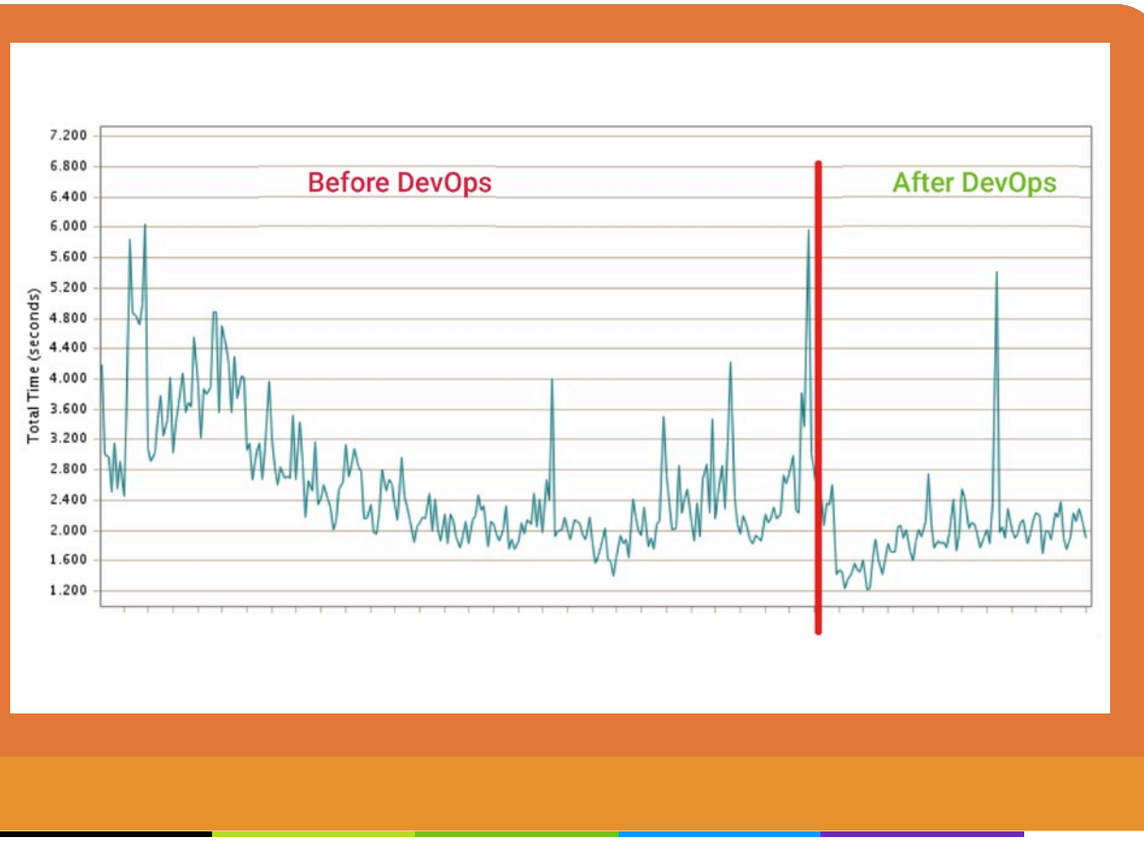
We soon adopted a DevOps mentality and moved into a two-week sprint process. We actually reduced deployments from two to three a week down to one every two weeks. While in DevOps you're supposed to increase deployment frequency, we had to decrease it in order to have better quality code.

We just didn't have the kind of code base — we had a 50% deployment failure rate — to deploy more quickly. We were dealing with too many Band-Aids and poor quality coding.

Sometimes you have to take a step back and first focus on quality before you can move forward. We started to push back on marketing timelines so that we could actually fix the problems so they did not occur again.

From the charts (right), you can see that we had a substantial performance improvement when we started making these changes (from an average of 2.8 seconds to around 2.3 seconds), and we had more consistency. We didn't make any drastic code changes, we just started to deploy less often and with more confidence, and the users saw real differences.

We actually reduced deployments from two to three a week down to one every two weeks.



© 2017 Dynatrace

From log files to actual performance monitoring

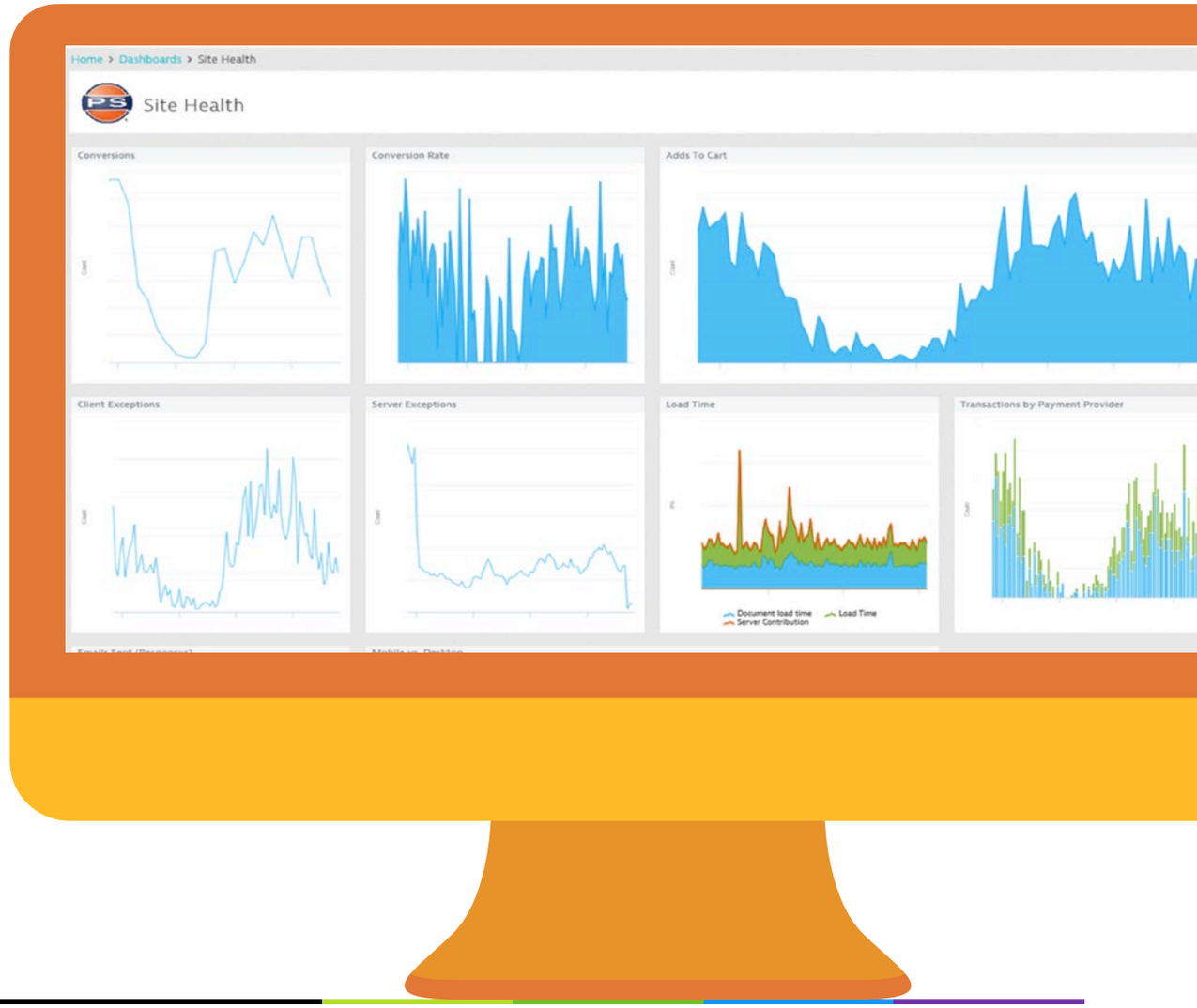
RD Once the new team was in place, we adopted Dynatrace for digital performance monitoring.

It is one of our biggest IT costs. Before we started these changes, there was virtually no monitoring whatsoever. The only thing we had was log files. There was really no other way for the new team to know how the system was operating.

Now we have performance dashboards to show us all our KPIs — conversion rates, traffic, exceptions, errors, load time, etc. One of my favorite dashboards is the one that shows business impact and technical root cause: conversions are going down because load time is up and load time is up because the CPU is maxed out due to a problem with the latest deployment. And so on. We also use these charts immediately after deployments to see whether it had a positive or negative impact on conversion, and then we drill down to find the root cause.

We have another dashboard focused on performance metrics for DevOps, and another more marketing-friendly chart. The marketing chart is designed so that anyone from the company can easily look at it and get insight into the end user experience.

Before we started these changes, there was virtually no monitoring whatsoever... Now we have performance dashboards to show us all our KPIs.



© 2017 Dynatrace

Back to Table of Contents

Section 2

Five ways to use performance monitoring to make DevOps a success

© 2017 Dynatrace

From development to the experience of your end users

AG Richard and his team are right. Performance monitoring is a critical part of any DevOps journey.

You must have insight across the delivery pipeline from development to testing through to production and the experience of your end users. Over years of experience helping hundreds of companies improve performance, Dynatrace has identified five ways performance monitoring should be used to make DevOps a success.

#1: Don’t check in bad code

It is time for developers to “level-up” in the DevOps journey. You can execute your tests as you always do, but be sure to do so with performance monitoring, and verify that the code works as intended before it leaves IDE.

#2: Stop bad builds through metrics

At Dynatrace, we are a big fan of continuous integration and delivery. Your performance monitoring approach should cover every single unit, integration, and REST API test. If the key architectural metrics change from one build to the next, then you need the capability to automatically stop the build. That's very critical.

#3 Monitor your services and end users after deployment

You always want to be monitoring — especially, in production. If the usage goes down, why does it go down? Is it because response time is going down? If so, then you need to know whether the recent deployment changed the behavior of the database.

And then you need to drill down to find out whether the cause was due to a bypassed database cache, a configuration problem, too many service calls, etc.

#4 and #5: Understand your end users and optimize their behavior

From the business perspective, performance monitoring should help you answer the questions: Do my campaigns work? Who are my users?

What browsers and devices do they use? Where are they coming from? How are they navigating through the system? Are they using all the features? Does performance have an impact on behavior?

© 2017 Dynatrace

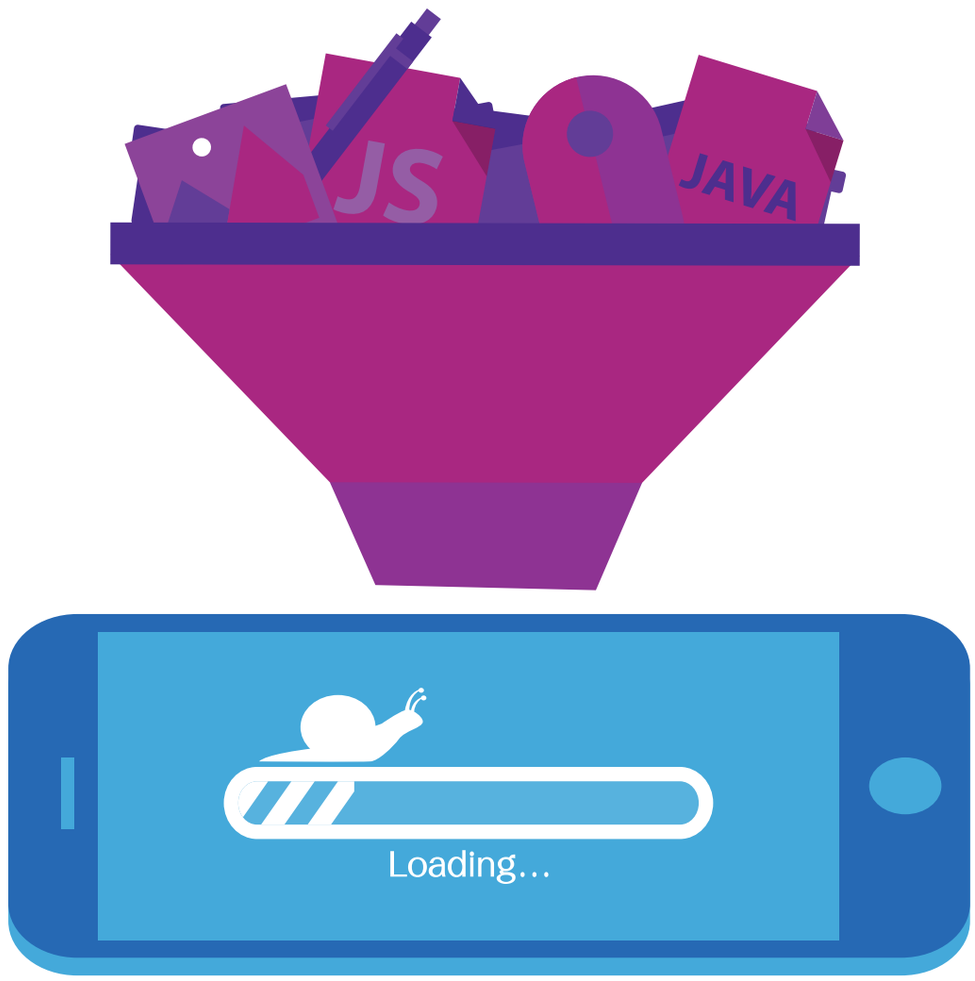
Back to Table of Contents

Section 3

The way forward: Microservices

© 2017 Dynatrace

Breaking up the monolith



RD The way forward for us is microservices. They are helping us move away from the monolith by allowing us to deploy very small segments of code with confidence.

We needed to start breaking apart one of the biggest components, and that's our image server which creates on-demand personalized product images. Our monolith literally contained the entire business — administration, image processing, the e-commerce site, customer service, and design — in one gigantic piece of code.

We started analyzing what happens when a product image request hit the site. Turned out that it had to go through a lot of code that wasn't even doing image processing. With this new insight, the engineering team could extract just the image processing code into its own microservice so that requests got directed to the new service right away instead of being processed by unnecessary code.

When we extracted the code we started with running the old and new version side-by-side. We allowed half of the image requests to hit the old implementation and the other half went to the new implementation. We immediately saw the positive impact in multiple ways. The much smaller service runs much faster, with fewer resources, and a smaller hardware footprint.

Our monolith literally contained the entire business — administration, image processing, the e-commerce site, customer service, and design — in one gigantic piece of code.

© 2017 Dynatrace

Your Next Steps

Without the change to DevOps, Prep Sportswear was faced with unhappy users and an unsustainable amount of technical debt. Prep Sportswear took DevOps principles and adopted them for their specific needs for their specific environment, exactly what they should have done. We wish them the best of luck as they continue to improve on their DevOps journey.

We know you are facing the same changes as Prep Sportswear — in markets, requirements, and user expectations. Your users want a responsive digital experience and they want it now. The answer is continuous delivery supported by a DevOps culture and a performance monitoring safety net. We hope the Prep Sportswear story gives you some ideas for your own DevOps journey. Go forward to automate as much as possible, and keep focusing on faster and better!

Here are our some additional resources that will help you on your journey.

Click here for your
DYNATRACE FREE TRIAL

Webinar replays

> 6 Ways DevOps helped Prep Sportswear move from Monolith to MicroServices

Blogs and Podcasts

> Dynatrace APM on DevOps
> PurePerformance Podcast Series

Recommended reading

> *Continuous Delivery* by Jez Humble and David Farley

© 2017 Dynatrace

Learn more at dynatrace.com

Dynatrace has redefined how you monitor today's digital ecosystems. AI-powered, full stack and completely automated, it's the only solution that provides answers, not just data, based on deep insight into every user, every transaction, across every application. More than 8,000 customers use Dynatrace to optimize customer experiences, innovate faster and modernize IT operations with absolute confidence.

3.22.17 727_EBK_PrepSportswear_TipsForOps_jg

dynatrace