DevOps Without Measurement Is a Fail

How to measure and track the 5 critical drivers of DevOps success

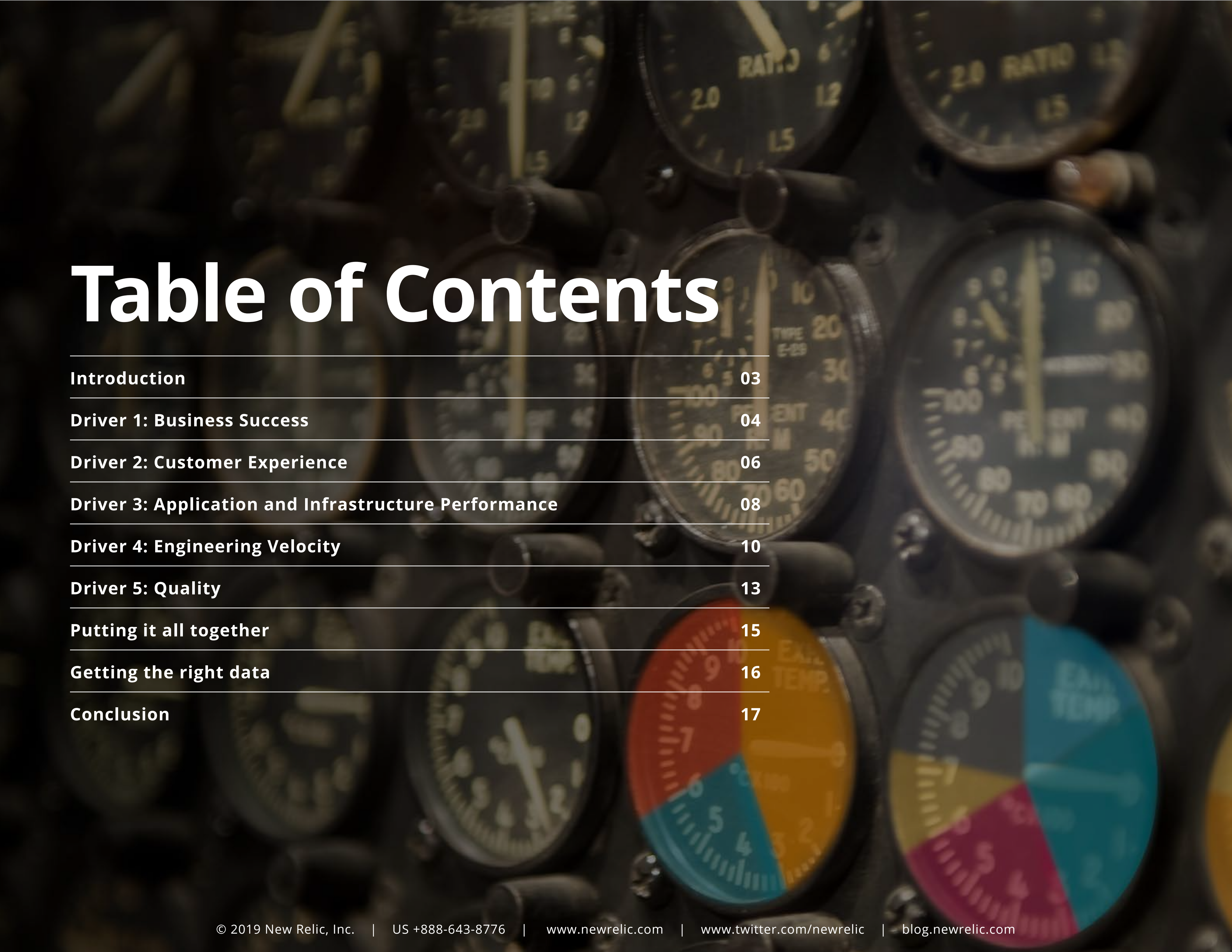


Table of Contents

Introduction

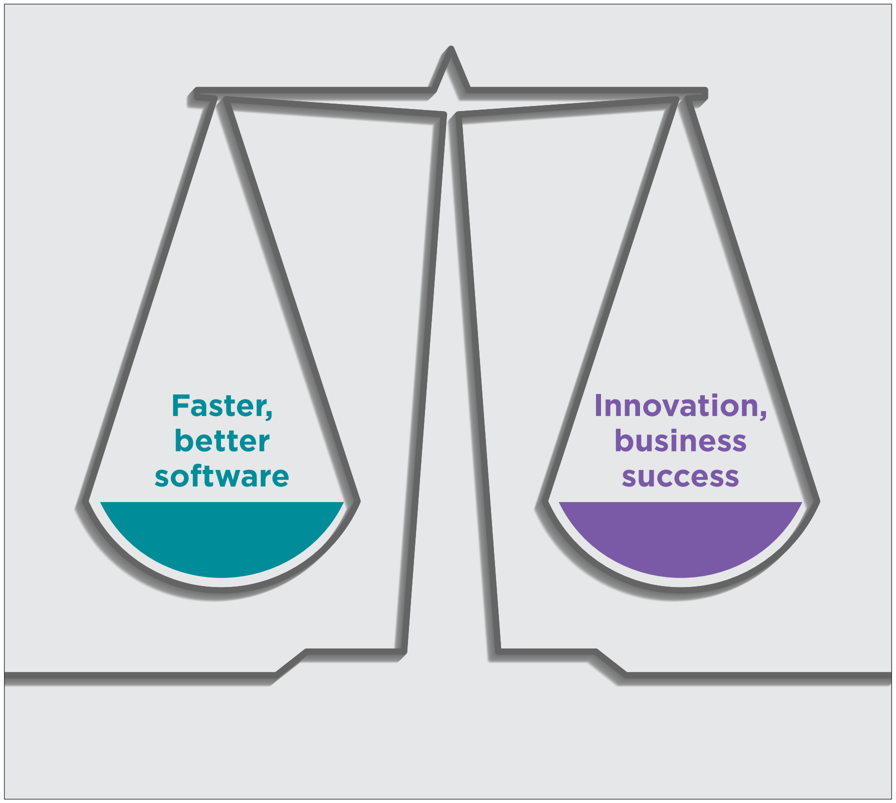
It's becoming an increasingly common scenario for many organizations: you're now months into your DevOps initiative but the results are still hard to see. While automation efforts have made it easier to provision resources, bottlenecks in the flow of work are still forming elsewhere and your engineers are struggling to reduce toil from unexpected outages and performance issues.

The collaboration across siloed teams that DevOps promises is increasingly shown to deliver more code deploys, reduced mean time to resolution (MTTR), reduced lead time for changes, and ultimately business impact.[1] However, the transformation that's required to achieve those outcomes is certainly not easy. What's needed is a mindset shift that impacts the entire chain of work.

The question "how will you measure this?" is at the core of any good engineering culture. Whether it's a new feature deployed to production or a broader organizational initiative, measurement is the tie that binds your efforts to meaningful results for everyone to see. Conversely, work that doesn't link to measurable goals can experience a backlash when teams and individuals feel they are being exploited or undervalued (for example, being on call after business hours) versus using data to measurably prove success that builds a sense of being part of a shared journey.

You can't know for sure that you are achieving business success with your DevOps approach unless you measure the right things and manage your DevOps operation to continuously keep the key performance indicators in view. Think of it this way: DevOps without measurement—or without measuring the right things—is a fail.

This ebook will introduce you to five critical drivers of DevOps success and show you how and what to measure to achieve that success: **business success, customer experience, application and infrastructure performance, engineering velocity,** and **quality.**



[1] Puppet's 2017 State of DevOps report

DRIVER 1:

Business Success

In the Agile Manifesto, the first principle states “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.” Customer satisfaction is one example of a metric for software delivery that relates to business goals. But how can we define and measure the rest of that principle—that is, the delivery of valuable software? What does that mean for the business?

Driver 1: Business Success

We can define what valuable software means by relating it back to a set of measurable goals that are most important for the company. For instance, software can be valuable to the company when it helps grow the customer base, increase revenue, reduce customer service costs, or any number of other objectives.

Talk to your company's business stakeholders and executive suite to understand what the relevant business goals are, and which ones can be impacted by delivering innovative software, quickly.

Depending on your industry, company maturity, market share, and other variables, key performance indicators that your software could potentially influence include:

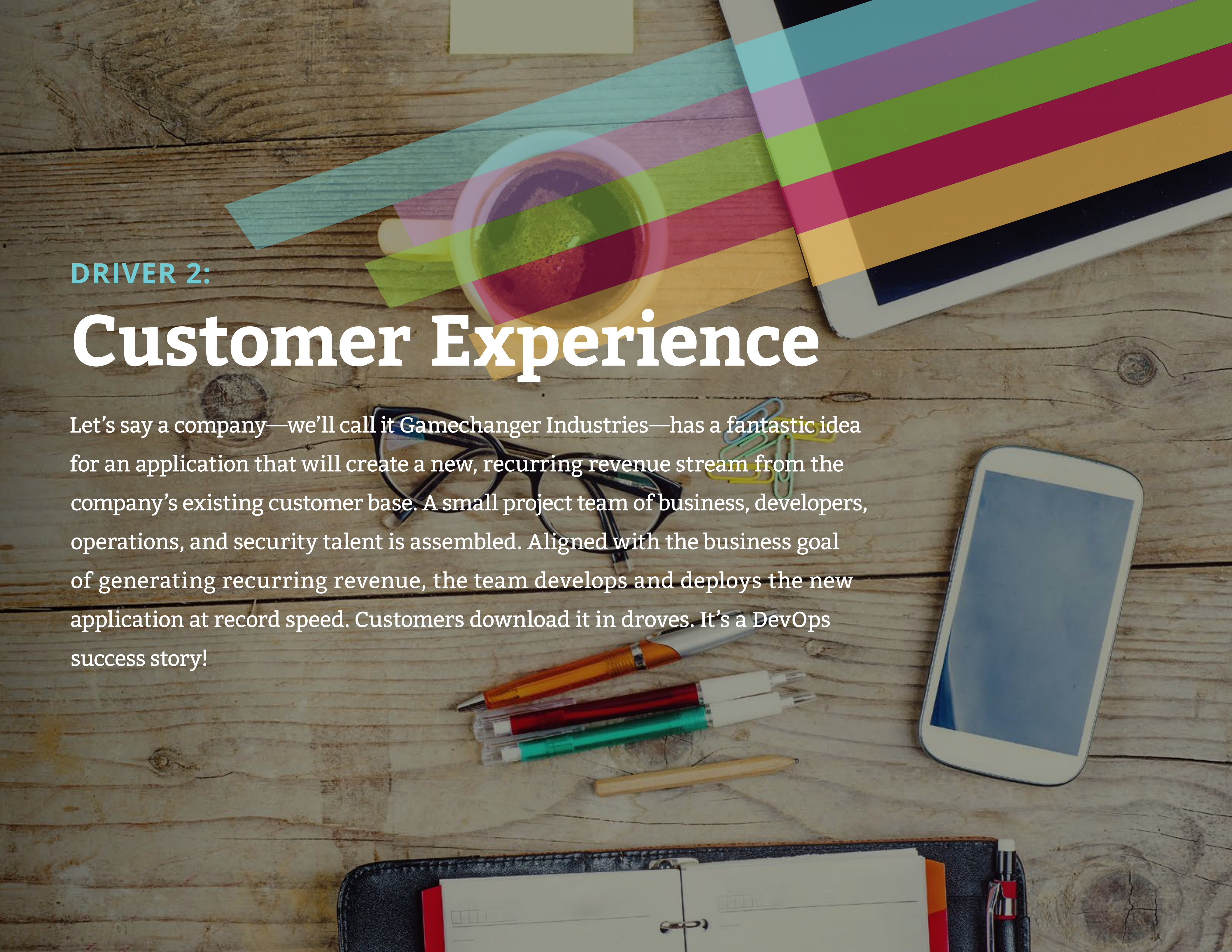
- Conversion rates
- Average revenue per user
- Customer acquisition costs
- Rate of customer churn
- Recurring or incremental revenue
- Renewals and/or subscriptions

Agreeing on the business goals for your software helps everyone on the DevOps team relate what they're doing back to a measurable set of indicators of success. This data-driven approach improves prioritization and decision-making—from which features to include in your software to resource allocation and the size of your DevOps team.

Going the extra mile for business success

Specialty retailer REI is committed to helping its members and customers get the most out of its large variety of outdoor recreation products and services. To ensure all of its apps and websites deliver on the promise of a great digital customer experience, the company's IT team follows a continuous deployment model, pushing code multiple times throughout the day.

As a result, developers require end-to-end visibility into the production environment to see any changes to business metrics, performance, and error accounts. According to Todd Wilson, REI's director of platform engineering, "We've been evolving site performance from a technical thing to an actual business driver with an impact on conversion and user experience. We've started creating performance dashboards that can be leveraged by the business as they think about performance as a business investment."



DRIVER 2:

Customer Experience

Let’s say a company—we’ll call it Gamechanger Industries—has a fantastic idea for an application that will create a new, recurring revenue stream from the company’s existing customer base. A small project team of business, developers, operations, and security talent is assembled. Aligned with the business goal of generating recurring revenue, the team develops and deploys the new application at record speed. Customers download it in droves. It’s a DevOps success story!

Driver 2: Customer Experience

But wait ... it turns out that after a few tries, customers stop using the app. In fact, many delete it from their devices. What happened?

One culprit could be a customer experience that didn't meet expectations. Was the app too confusing for customers? Was it too slow? Were there too many steps involved in critical interactions? Did the team dissolve after release and move on to another project? If the customer is too frustrated to complete the transaction that generates additional revenue for the company, the DevOps approach will have failed.

If this were your company, you wouldn't know if customer experience was the problem unless you were measuring and tracking the right KPIs within your software. These customer experience KPIs could include:

- Perceived response times of key transactions
- Frequency of key transactions
- Number of visits per user/per week
- User growth rates
- Funnel/conversion rates
- Amount of time spent in app
- A/B test results
- Customer satisfaction

The moral of the story is that aligning innovation and software development with business goals isn't enough. You need to also link those aspects of success to key areas of the customer experience and measure how well you are doing.

Technology performs 'beyond par' for superb guest experience

Global sports entertainment leader Topgolf entertains more than 13 million guests annually at more than 41 venues across the world. When the company added 20 new venues within a short period of time, its DevOps teams at headquarters required real-time visibility into the software running at each new location.

Performance monitoring helps Topgolf's technology group find and fix problems before they can affect guests or venue staff. Additionally, real-time and historical trend analytics guide the team in how to proactively improve the guest experience. Says Kenny Sullivan, director of applications at Topgolf, "We have to make sure that not only is the software up and running in each venue, but that it's not slow and impeding gameplay. Software performance means the difference between a really pleasant experience for our guests or a negative one."

DRIVER 3:

Application and Infrastructure Performance

Back to our Gamechanger Industries example. The company has an innovative idea for an application that will generate new, recurring revenue from the customer base and help the company meet its business goals. This time, Gamechanger also sets goals for the customer experience and optimizes the application to meet those goals. The efforts pay off, and after developing and launching a beta, the Gamechanger app is a hit, so much so that they struggle to keep up with demand.

Driver 3: Application and Infrastructure Performance

Despite proactive measures to scale the infrastructure, customer support is suddenly getting calls from angry customers complaining about slow performance and errors. With a distributed architecture built in the cloud, it's not exactly easy for the incident response team to pinpoint the exact location of the issue. Is it the new code? Is it the cloud service scaling improperly? Is it a problem from just one host? Or is it something else completely?

Modern application architectures offer tremendous agility and scale, but also introduce complexity. The customer's experience is often influenced by many factors spanning a network of server-side application services, dependencies from third-party services, and a set of dynamically allocated compute resources, which may be located in your own data center or that of a public cloud provider. The confidence to make changes in production that DevOps inspires requires fast feedback on which piece of the application stack is causing customer experience issues. And as cloud costs eat an increasingly larger portion of your IT budget, it's important to deliver the best customer experience as efficiently as possible.

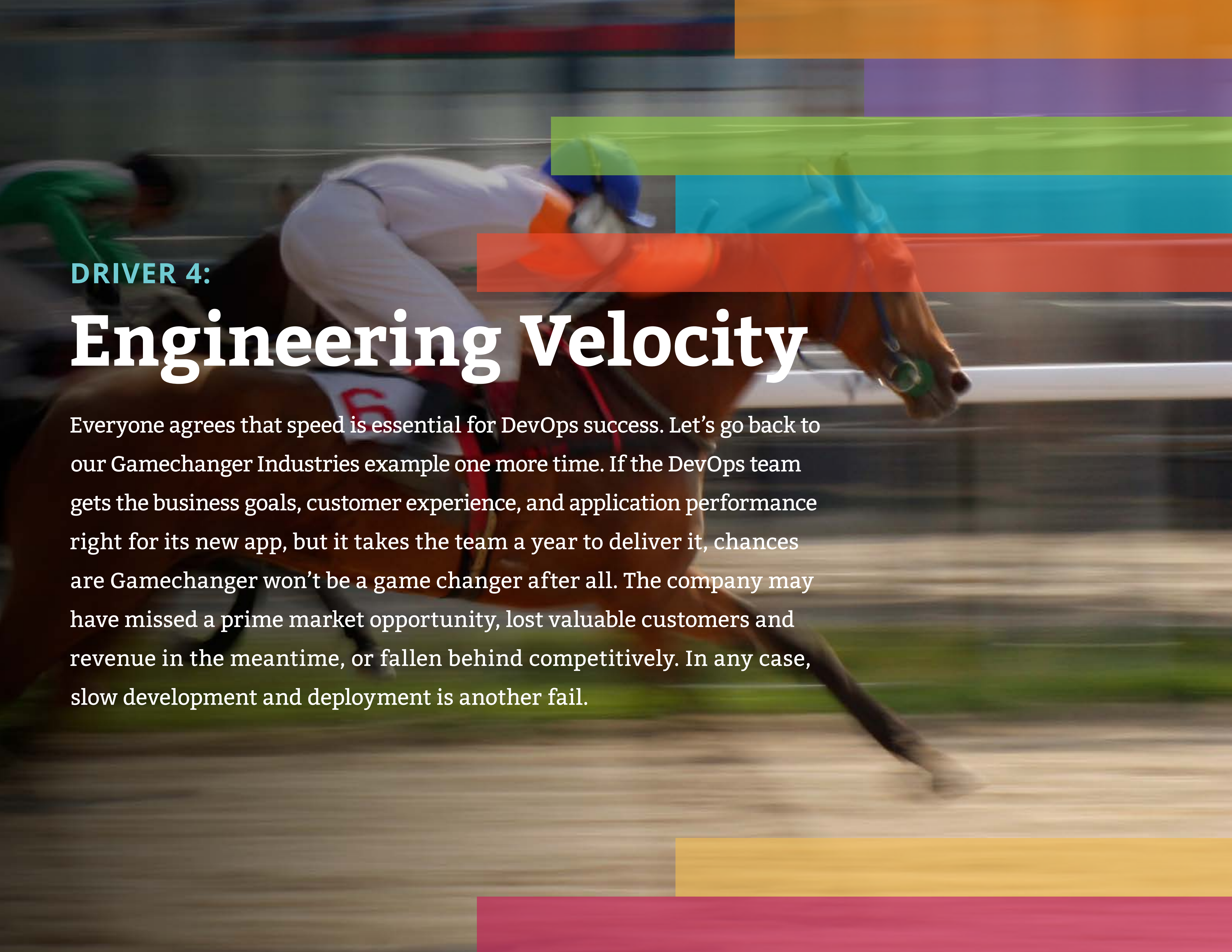
Catching performance problems before they manifest requires tracking and monitoring performance KPIs across the application and infrastructure. Hopefully, your team is already focused on some or all of these KPIs:

- App response time
- Database response time
- Third-party service performance
- Resource utilization
- Cloud cost

Auditing its app for performance issues

The world's largest tax-preparation firm, H&R Block has been operating in Canada for over 50 years where it currently has more than 1,200 offices. The ability to monitor the performance of its DIY online tax-filing software is crucial for the business, which revolves around a single, all-important deadline and the months leading up to it.

To handle the huge spikes in user traffic that are so critical to its business, H&R Block requires comprehensive, real-time insights to deal with performance issues on-the-fly and get fast feedback on how performance is affecting customers. "The DevOps team can drill down to the line of code causing the error and get references back to the data model triggering the error. As a result, we've been able to radically improve the quality of the product," says Dave Falkenberg, director of product development and compliance for H&R Block Canada.



DRIVER 4:

Engineering Velocity

Everyone agrees that speed is essential for DevOps success. Let's go back to our Gamechanger Industries example one more time. If the DevOps team gets the business goals, customer experience, and application performance right for its new app, but it takes the team a year to deliver it, chances are Gamechanger won't be a game changer after all. The company may have missed a prime market opportunity, lost valuable customers and revenue in the meantime, or fallen behind competitively. In any case, slow development and deployment is another fail.

Driver 4: Engineering Velocity

DevOps teams need to focus on the speed of development, delivery, and response to issues that occur in production. Metrics to consider for helping you track progress and success in this area include:

- Lead time for changes
- Frequency of code releases
- MTTR

When it comes to speed, many organizations rely on cloud computing to streamline and accelerate the development and deployment of software. Through Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS), DevOps teams can gain access to the services they need to bring new products to market quickly, without the effort to provision and manage servers, networks, or storage.

As with our other drivers of DevOps success, we can't focus on speed in exclusion of our other goals—particularly software quality—as it relates to the customer experience and application performance. In fact, development and deployment speed can become a negative if the deployment quality is poor, resulting in more time and effort needed for fixing quality issues that appear in production.

The need for speed

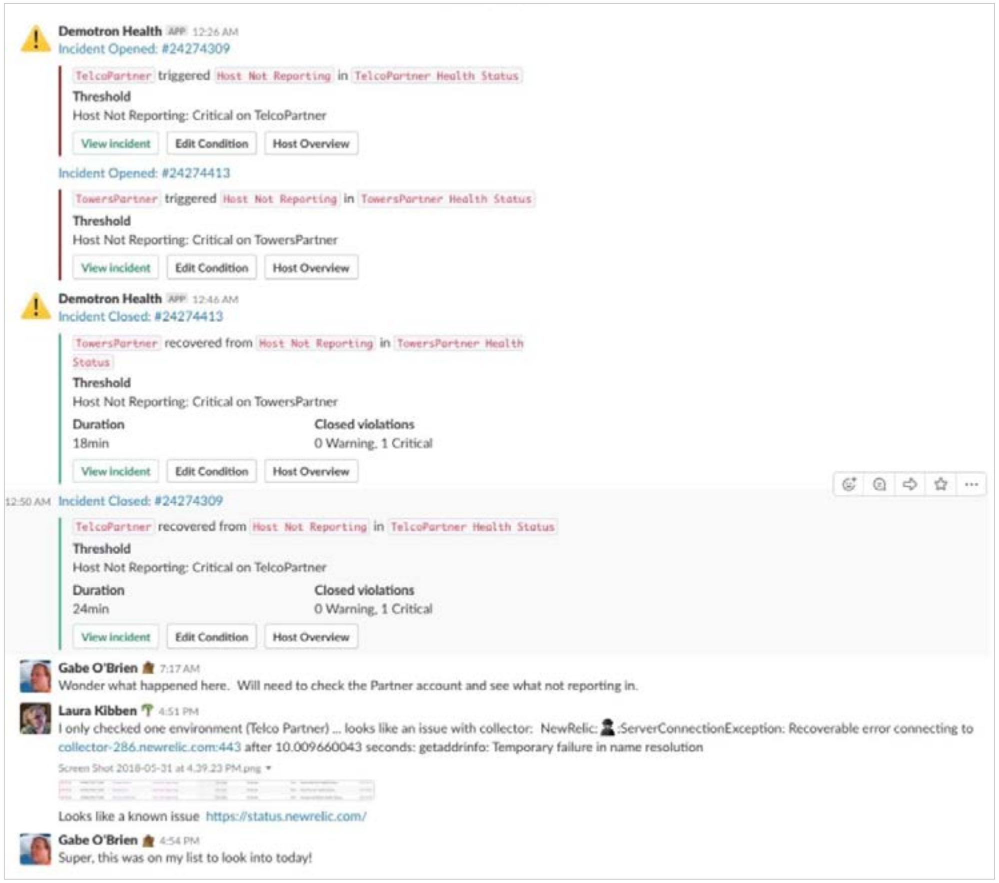
Trek Bicycle Corporation is a bicycle and cycling product manufacturer committed to providing a wealth of superior cycling-related information and experiences through its online presence. Meeting ever increasing customer expectations, however, requires a strong system of continuous feedback and agility.

Fast feedback on engineering efforts is critical to achieving faster development cycles. Says Jason Endres, web technologies manager at Trek, "On a couple of occasions when we've done major platform upgrades, we've had performance issues. Without New Relic, we would likely have spent a week or two longer debugging them. Instead we cut down on our development cycles by pinpointing precisely where we need to look. And that, in turn, frees up our DevOps team to work on more strategic projects."

ChatOps as an enabler for collaboration

It's almost inevitable that your system will fail. When that happens, it's critical that incident response is activated quickly to resolve the issue. ChatOps tools like Slack have become an important ingredient for DevOps teams to maximize the efficiency of communication during an incident. A group chat room is auditable and allows all the stakeholders involved in an incident response to stay informed about what's happening. In addition to chat discussions, plugins and integrations can provide real-time information from various development and operation tools. Centralized, time-stamped communication logs, real-time interactions with DevOps tools, and assistance from "chatbots" has defined a new way for DevOps teams to collaborate—this is the way of ChatOps.

ChatOps provides a central place to record information from both individuals and tools in real time. This recorded history of events provides a lot of transparency as to how things are working and how problems get solved when they arise. In the spirit of speed and agility, the real-time nature of ChatOps as a tool for conversation-driven development and ops is a great fit for DevOps teams and those responsible for their success.



Demotron Health APP 12:26 AM
Incident Opened: #24274309

TelcoPartner triggered Host Not Reporting in TelcoPartner Health Status
Threshold
Host Not Reporting: Critical on TelcoPartner

Incident Opened: #24274413

TowersPartner triggered Host Not Reporting in TowersPartner Health Status
Threshold
Host Not Reporting: Critical on TowersPartner

Demotron Health APP 12:46 AM
Incident Closed: #24274413

TowersPartner recovered from Host Not Reporting in TowersPartner Health Status
Threshold
Host Not Reporting: Critical on TowersPartner
Duration 18min
Closed violations 0 Warning, 1 Critical

12:50 AM Incident Closed: #24274309

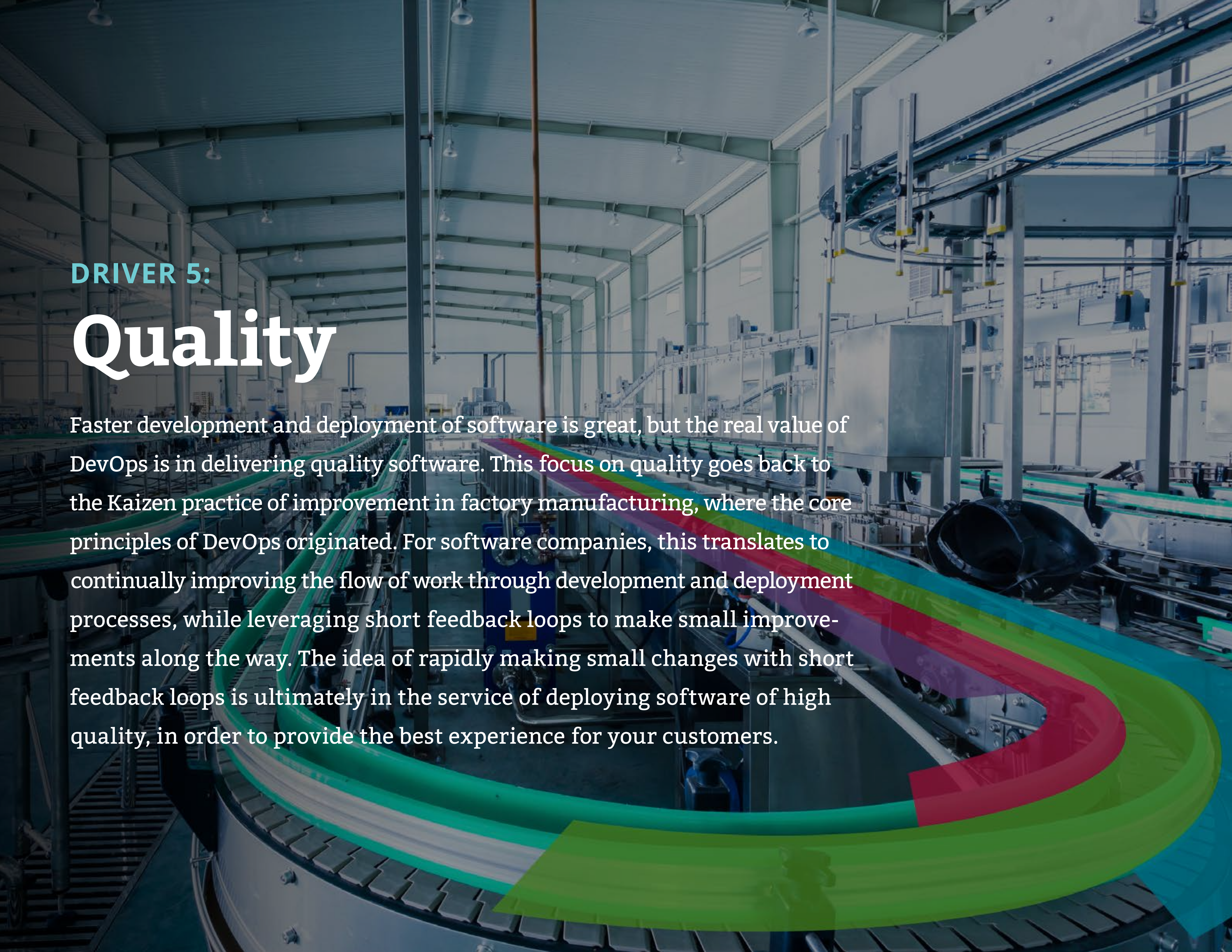
TelcoPartner recovered from Host Not Reporting in TelcoPartner Health Status
Threshold
Host Not Reporting: Critical on TelcoPartner
Duration 24min
Closed violations 0 Warning, 1 Critical

Gabe O'Brien 7:17 AM
Wonder what happened here. Will need to check the Partner account and see what not reporting in.

Laura Kibben 4:51 PM
I only checked one environment (Telco Partner) ... looks like an issue with collector: NewRelic: :ServerConnectionException: Recoverable error connecting to collector-286.newrelic.com:443 after 10.009660043 seconds: getaddrinfo: Temporary failure in name resolution
Screen Shot 2018-05-31 at 4.39.23 PM.png
Looks like a known issue https://status.newrelic.com/

Gabe O'Brien 4:54 PM
Super, this was on my list to look into today!

Example of ChatOps using Slack and New Relic Alerts



DRIVER 5:

Quality

Faster development and deployment of software is great, but the real value of DevOps is in delivering quality software. This focus on quality goes back to the Kaizen practice of improvement in factory manufacturing, where the core principles of DevOps originated. For software companies, this translates to continually improving the flow of work through development and deployment processes, while leveraging short feedback loops to make small improvements along the way. The idea of rapidly making small changes with short feedback loops is ultimately in the service of deploying software of high quality, in order to provide the best experience for your customers.

Driver 5: Quality

If you think about it, the underlying premise of DevOps is virtually an oxymoron: faster and better software delivery. Haven't these been mutually exclusive goals? Shorter development cycles and closer collaboration under a DevOps approach mean you catch problems before they go to production, and so spend less time fixing them under duress.

The timeless battle of supporting software in production is often about balancing time spent on reactive firefighting versus proactive efforts such as building and testing. Software quality is directly correlated to customer-impacting incidents, so it pays to invest the time and energy in addressing quality issues earlier in the cycles. Besides, you don't want customers to be the ones discovering your issues! Approaches such as TDD (Test Driven Development), CI (Continuous Integration), and CD (Continuous Deployment) can all help to prioritize work that is focused on quality improvements.

Here are metrics that you can track to help quantify the quality of your software's overall continuous improvement efforts:

- Uptime
- Deployment success rate
- Error rates (both client and server-side)
- Mobile crashes
- Incident severity

Since many quality issues don't appear until the software hits production, the ability to compare all metrics pre- and post-deploy is extremely helpful in understanding the impact of a specific release. This approach also allows teams to respond quickly and roll back a release or provide a quick resolution to any incidents that have occurred.

Distilling their process to perfection

Irish Distillers is Ireland's leading supplier of spirits and wines as well as the producer of the world's most well-known and successful Irish whiskeys. With its Jameson brand leading the way as the fastest-growing Irish whiskey, Irish Distillers exports its brands into more than 130 markets. To ensure the quality of its digital presence matches the quality of its products, Irish Distillers requires insight into the performance of its consumer-facing marketing sites—particularly as deployment velocity increases—both in terms of end-user experience and backend code performance.

As head of IT marketing and sales solutions for Irish Distillers, Pete McVicar encourages his teams to embrace the DevOps model of constant, agile development and also ownership for the quality of new features. McVicar notes, "We do the two-week sprints in which the weekly email reports from New Relic play a critical role—allowing us to see not only how well new features are performing but how they're affecting the rest of the site."

Putting it all together

Let's put the five drivers of DevOps success—business success, customer experience, application and infrastructure performance, engineering velocity, and quality—together for one overall picture of the areas that you need to monitor and track. As we've shown in our Gamechanger Industries examples, it takes a focus across all these areas to deliver on the promise of DevOps and drive measurable business outcomes.

Depending on how much progress you've already made in achieving your company's goals, you'll need to decide which KPIs are most important to track at this time. If you're not tracking any or only some, it's time to get started. You can do that by establishing a baseline and monitoring the metrics to make sure they are increasing or decreasing accordingly.

One thing we haven't discussed is how to balance resources and time across the DevOps success drivers. The allocation will rarely be static and will need to change as the business environment and overall strategy evolves. For instance, companies early in a product development lifecycle will spend a majority of time on innovation to develop new capabilities. A product in growth mode needs to consider performance and stability as the product matures and becomes more popular. Mature products, in particular, need to prioritize both innovation and quality—innovation to stay competitive and evolve the product, and quality improvements focused on reducing "technical debt."

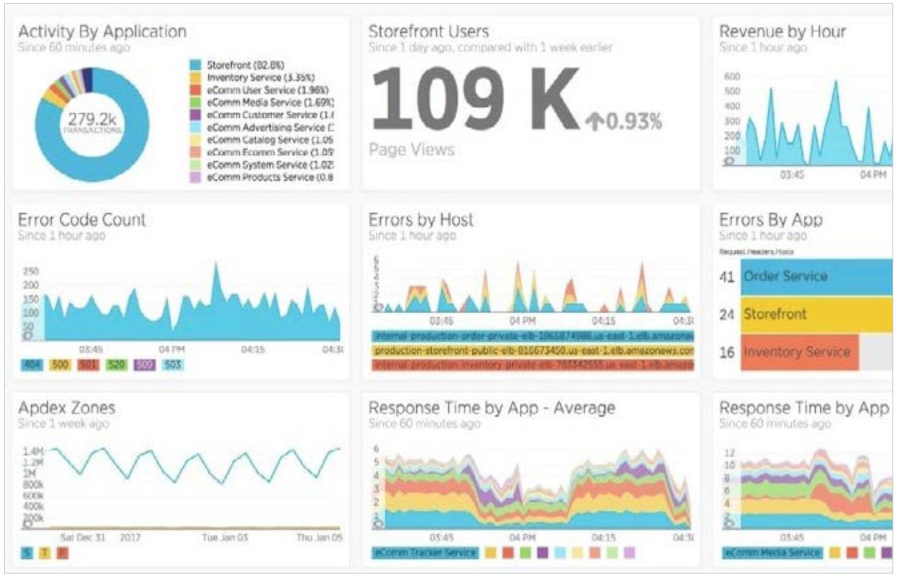
Technical debt (a term coined by New Relic's own Ward Cunningham, developer of the first wiki) is the extra development effort required to rewrite and improve code that teams incur as a result of doing what is necessary to release code quickly. According to DevOps expert Gene Kim, if you don't pay your 20% debt-reduction tax, you'll end up spending 100% to fix the accumulated issues.

In order to figure out the balance and allocation of the success drivers, you need to think about and understand the business strategy, the competitive landscape, and the state of the product in the lifecycle. The drivers are essentially dials that can be set at different levels to match the constraints, and should change to support the business priorities.

Getting the right data

Now here's the most important part of measuring success: the measurement of data. Achieving DevOps success requires access to the right data, essentially all the KPIs you identified that you need to track and monitor as well as related insight into engineering velocity, quality, application and infrastructure performance, customer experience, and business success.

Becoming a data-driven DevOps organization not only helps you track and monitor the progress and success of your team's efforts, but it also fosters a culture of collaboration and empathy. Data lets you remove emotion and finger-pointing from the decision-making process while establishing a common language across skills, experience, and roles.


Activity By Application
Since 60 minutes ago
279.2k
TRANSACTIONS
Storefront (82.8%)
Inventory Service (3.35%)
eComm User Service (1.96%)
eComm Media Service (1.69%)
eComm Customer Service (1.
eComm Advertising Service (
eComm Catalog Service (1.05
eComm Ecomm Service (1.05
eComm System Service (1.02
eComm Products Service (0.8
Storefront Users
Since 1 day ago, compared with 1 week earlier
109 K
↑0.93%
Page Views
Revenue by Hour
Since 1 hour ago
03:45
04 PM
Error Code Count
Since 1 hour ago
404
500
501
520
502
503
Errors by Host
Since 1 hour ago
internal-production-order-private-elb-1065874988.us-east-1.elb.amazona
production-storefront-public-elb-816673450.us-east-1.elb.amazonaws.cor
internal-production-inventory-private-elb-783342555.us-east-1.elb.amazo
Errors By App
Since 1 hour ago
41 Order Service
24 Storefront
16 Inventory Service
Apdex Zones
Since 1 week ago
Sat Dec 31
2017
Tue Jan 03
Thu Jan 05
S
T
F
Response Time by App - Average
Since 60 minutes ago
eComm Tracker Service
Response Time by App
Since 60 minutes ago
eComm Media Service

Sample New Relic dashboard

Getting the right data isn't difficult, but you'll need instrumentation capabilities that capture data automatically and let you analyze, view, and share the data quickly and easily. The solution you choose should provide detailed, real-time, and historical data on the five drivers of DevOps success. At a minimum, you'll need capabilities that include:

- **Proactive application-centric performance monitoring:** As modern technologies like cloud, containers, and serverless functions abstract away the infrastructure, understanding the performance of the application is increasingly important. The application is closest to your customer and closest to business value. Look for a tool that helps you deeply understand performance across your applications and infrastructure in the context of what matters most for your customers.
- **Customer experience management:** Any tool you choose should also provide data that lets you measure customer experience across all your digital channels, 24/7 (even when you don't have users on the system). The solution you select should help your team manage the frontend processing tier, where more and more code is starting to reside.
- **Robust analytics:** To create a data-driven culture across developers, operations, and business stakeholders, you need flexible tools including configurable dashboards and the ability to perform deeper technical analytics that quantify business success.
- **Deployment management:** You'll also need a reporting capability that shows recent deployments and the before/after impact on application performance and end users, including any errors that occurred. This will help you track and measure your performance-related KPIs.

When evaluating a solution, keep in mind that tracking and monitoring data through a single platform simplifies and streamlines the management of your DevOps efforts. A single platform lets you share a unified view of your software across the team and across the business for better decisions. Having developers and operations using the same tool is very effective in driving a shared understanding of how software is deployed and operates. Integrated alerts, dashboarding, and analytics let everyone work smarter and respond faster. A software analytics platform such as New Relic offers all of the capabilities mentioned above and more, integrated into a holistic solution.

Building diversity with data

Data can help you create a cohesive, collaborative, and high-performance DevOps team. It can also help you create a team with greater diversity, varied in personality, culture, skills, race, gender, and more. Data becomes the common language for team members to communicate in a way that avoids finger-pointing and nurtures a meritocracy. That's important because team diversity helps you retain more employees by creating a rewarding, successful work environment that is supportive and inclusive.

Conclusion

Achieving success across the five drivers discussed here would be impossible using a traditional waterfall approach to software development and deployment.

Data-driven DevOps makes success in all five areas a reality. When you track and measure the right metrics, you gain the insight you need to accomplish amazing results, whether it's disrupting markets with an innovative new app or taking customer experience to a new level.

As you embark or continue on your DevOps journey, your team will be instrumental to helping your company become a better software business. And that's job number one in today's digital economy, where every business is a software business.

Do DevOps right with New Relic

Visit newrelic.com/devops to learn how.

New Relic®