Shift-Left Performance Monitoring: Maximizing Agility in Software Delivery

ca technologies

Shift-Left Performance Monitoring:
Maximizing Agility in Software Delivery

Today, organizations of all sizes recognize the value of an agile software delivery process. Agility is the foundation for cost-efficiency, competitiveness, high team morale and customer satisfaction.

Yet most organizations face an agility paradox in that the more agile their software delivery workflow becomes, the higher the risk that instability or unpredictability will be introduced and undercut the investment they make in software production.

Traditionally, companies have tried to address this paradox by maintaining legacy software delivery processes (which are more stable but also less agile) alongside agile workflows. While this strategy helps mitigate the risk of critical problems or delays in software production, it also compromises the efficiency and competitiveness of the organization by constraining the ability to be as agile as possible.

Fortunately, there is a solution to this conundrum. It's called shift-left monitoring, and it allows your company to maximize agility in software delivery without assuming undue risks. This e-book explains how you can achieve agility and stability at the same time to streamline the path to digital transformation and innovation by shifting monitoring to the left of your software delivery workflow.

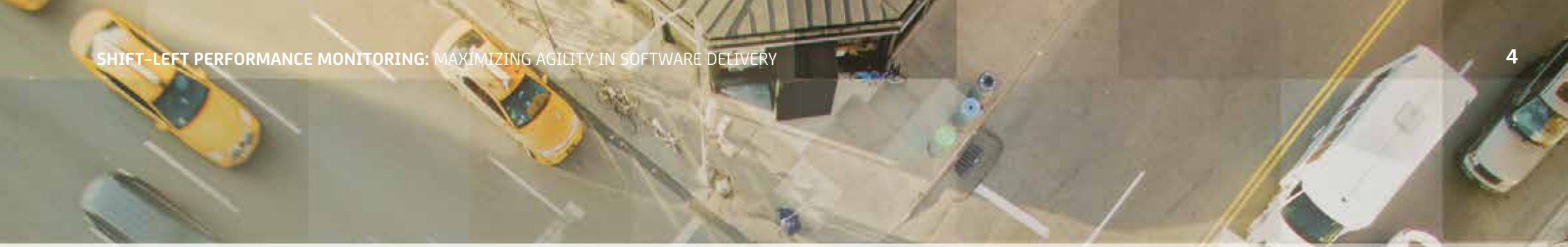
The State of Software Delivery Today: DevOps and Agility

To understand how shift-left testing can be applied to your software delivery process, you must first understand how most organizations currently approach software delivery.

Today, most organizations operate under intense influence from the DevOps movement, which has existed for almost a decade. DevOps emphasizes principles such as continuous delivery, which means software updates are designed, implemented, tested and pushed into production on a constant, rolling basis, with maximum collaboration between developers, admins and everyone else involved in software delivery.

Also central to DevOps is the concept of agility, where software delivery chains can scale up or down seamlessly. Developers and admins can switch easily between different types of programming frameworks. Personnel can change without affecting the quality or speed of software delivery, and the software delivery team can react rapidly to changes in user expectations or market demand. Being agile has become the sine qua non for organizations that wish to optimize expense by adopting the most efficient technologies. And for organizations that want to keep employees happy and productive by mitigating wasted time and unnecessary roadblocks, and stay ahead of the competition by providing end users with the software that best meets their needs.

As a result, today's most forward-thinking organizations strive to implement continuous delivery chains that are highly agile. They do so by deploying continuous integration servers like Jenkins, adopting workflow processes that allow different teams to work in parallel, implementing constant feedback loops so that developers enjoy high visibility into end user experiences, and so on.



The Agility Paradox

Traditionally, the ability and willingness of most organizations to adopt agile workflows has been limited by their need for predictability and stability.

By their nature, agile workflows make software production less predictable. The rapid pace of continuous delivery chains increases the risk of introducing errors or performance glitches into a software program and the potential failure to recognize them before they reach end users. Parallelized workflows, while more efficient, offer fewer opportunities for developers to pause and identify possible bugs in their code. And despite the emphasis on constant communication and collaboration across the organization, in practice, QA experts are not always able to break out of their silos and keep pace with the work of other teams in the software delivery chain.

Also problematic is the fact that some organizations confuse continuous integration with continuous delivery. They set up continuous integration servers but don't revamp their workflows in the way necessary to implement fully continuous delivery. Adopting DevOps tools without also adhering to DevOps cultural practices can lead to a fractured, risk-prone software delivery chain.

All the above is at the root of the agility paradox.

Traditionally, organizations have tended to address the agility paradox by retaining some non-agile practices while simultaneously migrating others to an agile workflow. They do so believing that by being only partially agile, they can leverage some of the benefits of agile software delivery while limiting the risks that come with an imperfect agile delivery chain.

This strategy may work, but it's far from ideal. Because it places strict limits on a company's ability to be fully agile, it undercuts cost-efficiency, productivity and competitiveness.

Solving the Agility Paradox With Shift-Left Monitoring

Shift-left monitoring is a better solution.

Shift-left monitoring refers to the practice of performing application performance and stability monitoring early in the software delivery process—in other words, to the left of a linear delivery chain that begins with design and coding and ends with releasing code into production.

The shift-left concept isn't new. It's already familiar to groups like QA teams, which have adopted shift-left testing to identify bugs earlier in the development process—when resolving them is exponentially less costly and time-consuming than fixing them in production. Equally important is shift-left monitoring, a practice that few organizations are currently embracing. With shift-left monitoring, businesses can benefit from earlier detection not just of software bugs but of application performance or stability problems. By finding performance problems early in the software delivery lifecycle, companies can address them much more efficiently—and greatly reduce the instability and unpredictability that have traditionally been the trade-off for agile workflows.

More specifically, shift-left monitoring offers the following key benefits.

Maximize stability and predictability.

Just as you wouldn't want to wait until software is in production to find out it has a bug that needs to be fixed, you want to be able to discover performance glitches before software reaches end users. By the time performance issues reach production environments, they've already negatively impacted your customers' experiences, and harmed your brand by extension.

In addition, fixing performance issues within production environments is much more difficult and costly than addressing them earlier in the development cycle. Resolving production-level performance problems requires applying patches to software that is already in use, and potentially disrupting users' work. It also often necessitates rolling back a host of code changes, because to get to the root of the coding or design error that causes a performance issue, you may have to modify a great deal of other code that depends on the code you'll change to fix the performance problem.

Keep IT teams happy and productive.

Developers and admins (who are likely among the highest-paid professionals at your organization) don't like wasting their time anymore than your organization likes wasting money on lost productivity. Developers and admins are happiest when the hard work they put into delivering software is rewarded with a smooth, efficient delivery pipeline.

Shift-left monitoring helps to ensure that pipelines remain smooth by allowing performance problems to be nipped in the bud. Rather than having to spend their time tediously fixing performance glitches that are not identified until production, developers and admins can focus on doing what they enjoy most and what delivers the best value to your organization—writing and maintaining better software.

Given that turnover for skilled IT personnel is remarkably high in today's labor market, the importance of keeping programmers and admins engaged and productive cannot be underestimated.

Build a fully continuous feedback loop.

To make the most of an agile software delivery workflow, organizations need to collect and interpret information about how software performs, then pass that information back to developers to use to improve the next generation of updates.

Traditionally, organizations have waited until software was in production among end users to collect feedback. While it's always important to monitor the end user experience, on its own, production-environment monitoring isn't enough. By performing preproduction monitoring as well, software delivery teams can gain information about application performance sooner so that they can act on it before the software reaches production. This is better for the development team—when performance problems are identified sooner, the team can fix them with less effort. It's also better for end users because many performance problems can be addressed before they ever make it to production.

Maintain environment parity.

To maximize application stability, maintaining environment parity across the delivery chain is key. Environment parity means that development, testing and production environments are as consistent as possible to ensure that software behaves in production as expected. DevOps tools like containers are one resource for helping to maintain environment parity.

Shift-left monitoring is another crucial practice. By running performance checks on an application during the development stage as well as the production stage, a software delivery team is in a better position to identify environment-related issues that cause performance or stability issues. In this way, performance data helps teams smooth over the environmental inconsistencies that (despite the best efforts at maintaining parity) are bound to exist within delivery chains.

Achieve continuous performance monitoring.

If your software delivery process is continuous, your performance monitoring should be, too. This isn't the case if you only monitor your applications once they're in production.

By adopting shift-left testing, you can gather and interpret performance information about your software at all stages of the delivery pipeline. In addition to presenting a more holistic, broader vision of your application's performance health, continuous performance monitoring allows you to identify variables within different stages of the delivery chain that can help you understand why performance problems occur. For example, poor performance that doesn't happen when running preproduction tests on simulated hardware (but is experienced in production) sends a signal that the performance issue is specific to real hardware environments, and that your team should look there first to resolve the problem.

Redefining Approaches to Performance Monitoring

From the perspective of performance monitoring, shift-left monitoring is a revolutionary change.

In most cases, monitoring has been the province of software delivery teams that are still working according to traditional schedules. They deliver software according to slow, staccato rhythms, then monitor its performance once it's in production. They place their faith in a gradual, deliberate design process to ensure reliable software performance, then confirm performance expectations once software is in production.

Meanwhile, agile software delivery teams have tended to implement performance monitoring tools haphazardly or not at all. They've relied on next-generation DevOps tools to maximize performance by making applications run faster. This is a mistake. Although technologies like containers and microservices can indeed improve performance significantly, plenty of performance issues can still occur using these technologies, and a monitoring solution that reaches across the delivery chain is the only way to identify them.

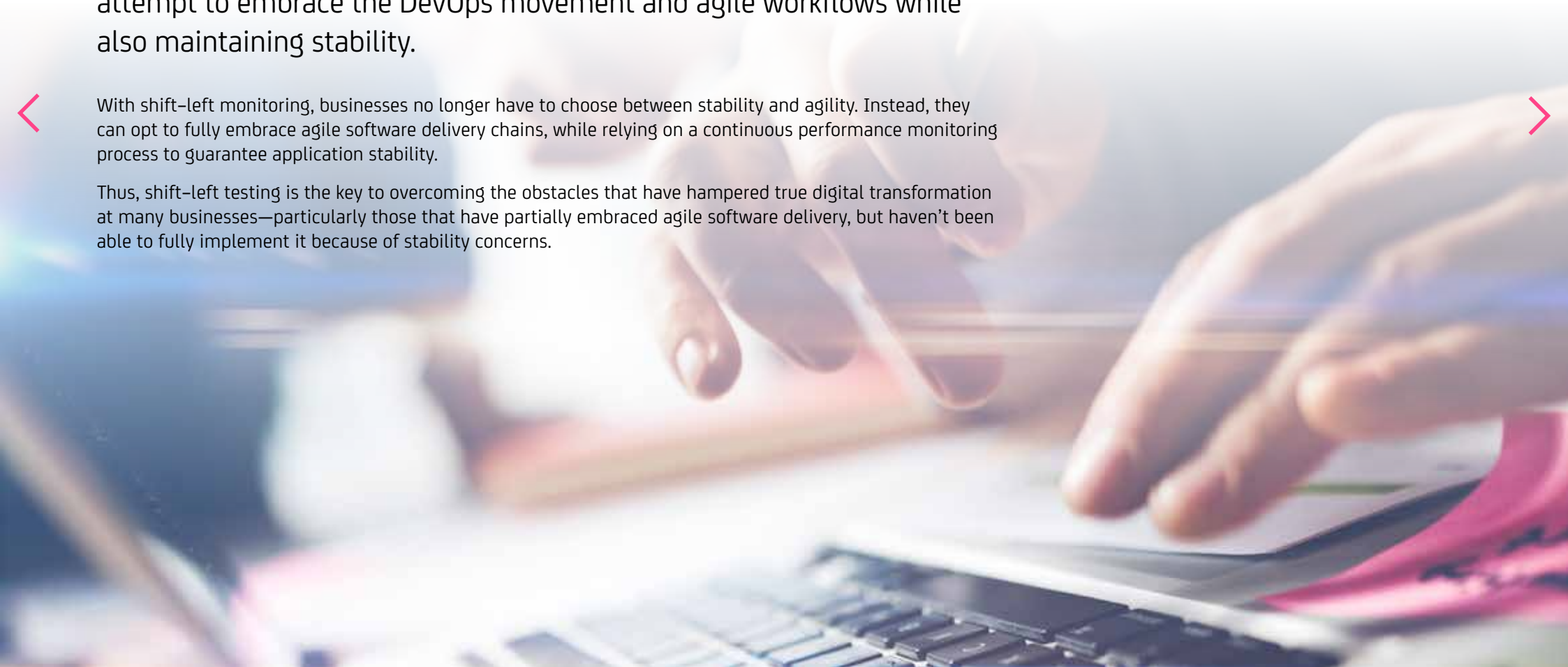
By rethinking performance monitoring and seeing it as something that can and should occur not only in production environments but at an earlier stage of the software delivery chain, organizations can ensure ROI of the time and energy that their software delivery team puts in. And they can make performance monitoring as continuous as the rest of their delivery process.

Overcoming Digital Transformation Roadblocks

In all the ways described above, shift-left monitoring helps organizations get past the roadblocks with which they've traditionally struggled as they attempt to embrace the DevOps movement and agile workflows while also maintaining stability.

With shift-left monitoring, businesses no longer have to choose between stability and agility. Instead, they can opt to fully embrace agile software delivery chains, while relying on a continuous performance monitoring process to guarantee application stability.

Thus, shift-left testing is the key to overcoming the obstacles that have hampered true digital transformation at many businesses—particularly those that have partially embraced agile software delivery, but haven't been able to fully implement it because of stability concerns.



Essential Strategies to Achieve Shift-Left Monitoring **and** Pull Business Value Right

A successful shift-left monitoring strategy involves much more than incorporating tools earlier within the software development cycle. Solutions must establish fast feedback loops, actively provide stakeholders with analytical insights to guide decisions, eliminate waste, remove bottlenecks, and remediate problems before software is released and customer experience impacted.

This can be achieved by adhering to the following guidelines.

Measure what matters: business outcomes and customer experience.

It's important to connect application delivery with business outcomes. Do this using data that links the impact of new software releases and IT performance to key business metrics, such as customer conversions, shopping cart fulfillment and time-to-value. This data is spread across multiple applications components, underpinned by hybrid cloud infrastructure and network services, so DevOps teams will need new ways to correlate information to expose key insights.

Analyzing information in aggregate enables DevOps teams to become far more data-informed. By understanding how IT performance and software quality correlates to business impact, teams can iterate quickly and pivot efforts toward those releases that deliver the most value. Many monitoring solutions provide part of the answer, but by correlating data across applications, infrastructure and networks, teams supporting each stage of the software delivery lifecycle can see the impact of their work in context of supporting shared business goals.

Strive to incorporate active monitoring.

Staff need to know about problems before customers are impacted. But most monitoring solutions are static in nature and only send alerts after a problem is detected. To address this shortcoming, organizations should consider more active monitoring methods that not only predict emerging problems but frame them in the context of business impact, and then automate DevOps workflows. This enables teams to focus collectively on business priorities and collaborate toward driving improvement across the development lifecycle.

There are other active monitoring methods that teams can use to learn about problems early. For example, a solution such as CA Application Synthetic Monitoring can continuously build accurate pictures of web app performance and customer experience before the software is released to customers. Similarly, in a mobile app A/B or split-testing scenario, CA App Experience Analytics can reveal information about functional usage and performance to guide design improvements.

Match development speed with quality and supportability.

Shift-left monitoring must help developers speed the development of great code while ensuring that what works its way downstream is supportable. To this end, solutions shouldn't force developers to use unfamiliar tools, but rather establish monitoring in the context of their own work practices. If, for example, a developer wants to compare performance across software builds, then that information should be revealed in the context of the continuous integration tools (such as Jenkins) that they use—and it should be available right from their own workstation.

This capability is available through an integration between CA Application Performance Management (CA APM) and Jenkins. By publishing key performance data from CA APM to Jenkins, developers can quickly and easily assess how their work impacts code quality. Performance metrics can be established as software build-pass conditions, enabling engineers to rectify problems early and understand which coding practices correlate with increased performance.

In this way, feedback becomes bidirectional. Jenkins build status information is published to CA APM. Any software build changes and the resulting impact on performance therefore become immediately visible and actionable.

Establish monitoring with continuous testing.

In a lean manufacturing plant, operators can pull an Andon Cord to stop production when they detect quality problems. Similarly, in our software factories, shift-left monitoring must incorporate methods that enable fast go/no-go deployment decisions based on performance goals. This ensures that low-quality code never makes it through to production and into the hands of customers. And with software quality now defined by functionality, usability and operability, finding ways to improve performance is just as important as new feature development.

One excellent way to achieve this is to run massively scalable performance load testing against apps using a leading solution such as CA BlazeMeter®, and integrate with an application performance management solution such CA APM. By incorporating the results of load tests as key performance metrics, application support analysts can make better capacity and service-level decisions. Issues can also be detected and resolved earlier and faster because CA APM can trace problems surfaced by front-end performance load testing to the actual root cause—whether a slow transaction, a problematic database call, a memory leak or something else.

Detect signals within the noise.

Fast feedback distinguishes high-performing DevOps teams from the rest of the crowd. For developers and QA to make the right decisions, that feedback has to be clear, timely and accurate. Paradoxically, the simplicity of modern microservice and API-centric architectures and deployment patterns introduces more monitoring complexity—not least in managing an explosion in dependencies, alerts and alarms. All this noise can adversely impact decision making.

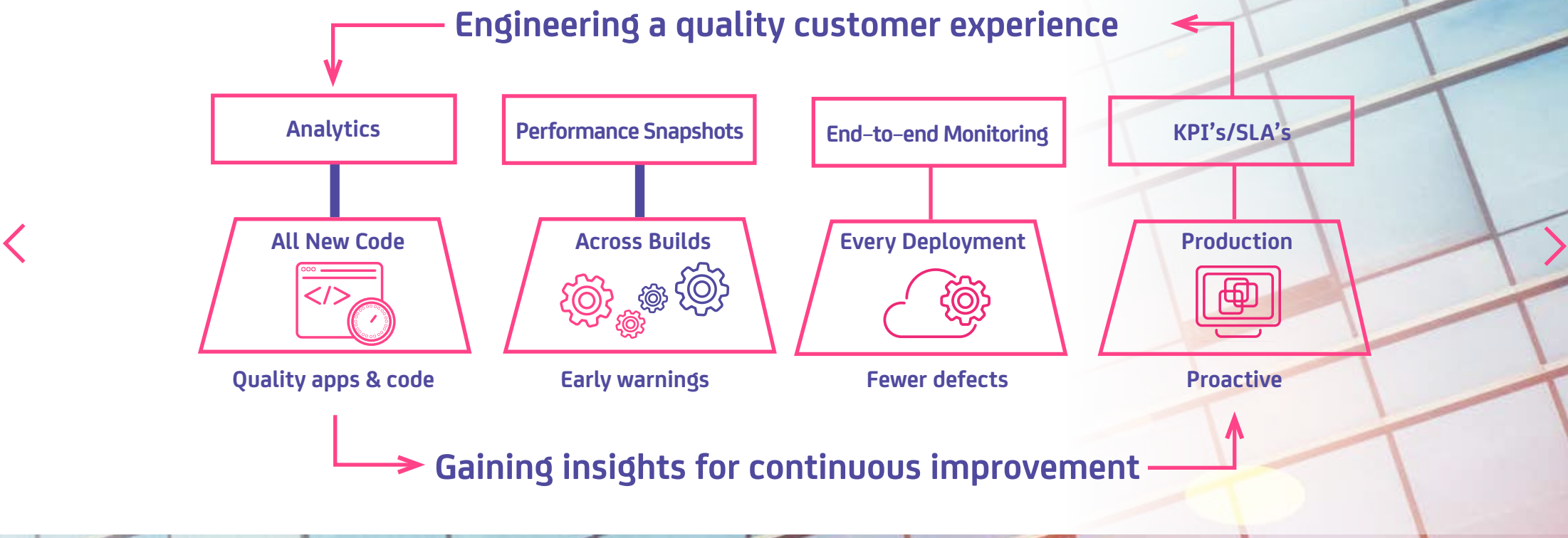
It's therefore essential that shift-left monitoring incorporate analytical methods to predict application performance requirements, detect emerging anomalous behaviors and distill and cultivate volumes of alerts down into key indicators of performance. One such method, known as differential analysis, has been incorporated into CA APM to distinguish false-positive nuisance alerts and anomalous trends worthy of action. When applied to the development and monitoring of microservices, DevOps teams can feel confident that the stream of alerts they receive are actionable problems.

Modern systems require modern monitoring.

With distributed systems, all of the complexity shifts from individual apps to the relationships between them. Critical issues such as API performance and app latency, along with back-end problems, need to be identified before production. But with container systems such as Docker supporting rapid application delivery, DevOps practitioners can no longer spend time configuring monitoring solutions. Things change too quickly.

Modern systems such as CA APM address this challenge by incorporating an agentless approach to detect the creation of new containers and immediately monitor the applications contained within them—all without requiring additional configuration. This way, DevOps teams can scale monitoring in the same way that they scale dynamic, cloud-based applications. With deep monitoring of containerized applications, teams can better understand how microservice functionality and communication impacts overall performance, pinpoint load issues and optimize container deployments for a flawless customer experience.body

Shift Monitoring Left



Conclusion

Shift-left monitoring allows you to enjoy the best of both worlds—achieve a fully agile workflow and stable, high-performing applications at the same time. In turn, shift-left monitoring will keep your workflows lean and efficient, maximize the morale and productivity of your software delivery team, keep your end users happy and maximize your competitiveness.

Pairing stability with agility is the key to effective software delivery today. Shift-left monitoring is the essential ingredient that helps you get there.

Solution information

CA Technologies provides leading solutions that organizations rely on to establish the essential DevOps workflows that accelerate the benefits of a shift-left approach to performance monitoring. Engineered to address the requirements of cross-functional teams, CA solutions seamlessly integrate with existing development, testing, release processes and toolchains to ensure a high-quality experience and consistently deliver superior software performance. Solutions include:

- CA App Experience Analytics is a comprehensive solution to visualize and analyze user experience and app health, capture and investigate issues and inform app teams with real-world customer usage data to better understand digital experience. Using these insights to optimize the customer journey helps businesses attract and retain new customers, increase revenue, achieve faster resolution times and deliver innovations quickly by boosting development productivity. Armed with visibility throughout the app lifecycle, DevOps teams can collaborate to continually improve quality, design, performance and the user experience.
- CA Application Performance Management helps organizations deliver a loyalty-building user experience by speeding and simplifying the triage of performance problems and identifying opportunities for improvement. With CA APM, developers and operations teams can increase the quality and supportability of applications while maintaining the pace of continuous delivery. Role-relevant notifications enable better communication so teams can resolve problems faster, with analytics and workflows providing essential feedback on the software performance requirements needed to conduct a profitable digital business at scale.

Additional Resources
from CA Technologies

To further assist DevOps teams in advancing their shift-left goals, CA Technologies has contributed a range of practitioner-specific content and case studies. Ranging from real-world customer experiences to best-practice techniques, this material outlines many important shift-left considerations together with ways to measure progress and demonstrate ROI.

- DevOps for Digital Leaders. Chapter 7 of "Agile Operations: Powering the Modern Software Factory" describes critical monitoring strategies associated with shift-left practices, including thinking beyond resilience with design for failure, how to make support a top design issue and active monitoring approaches.
- Shift-Left Monitoring Strategies. This practitioner-level white paper and video describe the value of integrating CA APM with Jenkins' Continuous Integration. By exposing rich information in the context of Jenkins build workflows, CA APM gives DevOps teams clear visibility into the impact of code changes on performance.
- How National Australia Bank (NAB) used CA APM during performance testing to improve transaction performance by 85%. This case study describes NAB's DevOps journey toward establishing user experience as a key business differentiator, and how CA APM helped align Dev and Ops behind this goal. See how NAB significantly reduced transaction times, and could trace problems across mainframe and new core banking platforms.
- DevOps Perspectives. This collection of DevOps articles from leading practitioners and thought leaders across the world discusses many aspects of IT operations and shift-left performance monitoring, including culture, process development and tooling.
- DevOps and Agile Operations: Insights from the Experts. "The Phoenix Project" author Gene Kim joins other industry experts in an exclusive analysis of what the three approaches of DevOps mean for operations teams. Learn how IT orgs change to adopt lean thinking, how monitoring can shift left and how ops teams can embrace continual experimentation.
- Using APM to increase AWS Cloud application resilience. This white paper discusses DevOps approaches and advanced APM tooling techniques to accelerate design-for-failure architectures and increase the resilience of cloud applications and services.
- This extensive collection of blogs, articles, surveys and analysis outline the benefits of DevOps methods, including shift-left performance monitoring.

To learn more about the technologies and processes needed to advance DevOps initiatives, visit **ca.com/devops.**

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments. Learn more at **ca.com.**

CS200-271755_0517