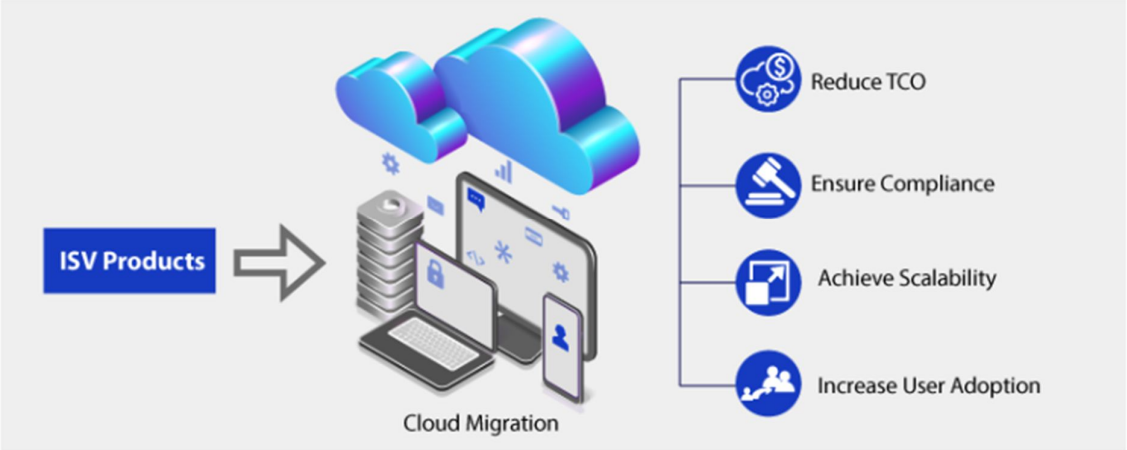
The Triple Play for ISV Product Optimization - DevOps, Observability, and SRE

Overview

The enterprise customers of ISVs are undergoing a growth spurt. They're running more SaaS, more cloud platforms, and more channels and digital ecosystems. Both NextGen ISVs and their enterprise clients are betting that cloud adoption and migration will extend their market reach to new global customers and revenue opportunities. To keep pace, ISVs are racing to migrate and/or cloud enable the last of their on-prem operations. But moving to the cloud is only one step (though an important one) on the path to delivering agile, cloud-ready products and services to demanding enterprise customers.

Development teams have to ensure that their skills, tools and methods are able to generate high-quality, scalable software that can interoperate – securely and seamlessly - with their customers distributed, diverse infrastructures. This means that not only is 99.999 cloud uptime critical to cloud-enabled ISV operations, but the reliability of the environments developers use to build, test and run their software is also paramount to achieving 'continuous delivery' and rapid adoption of high-quality ISV products.

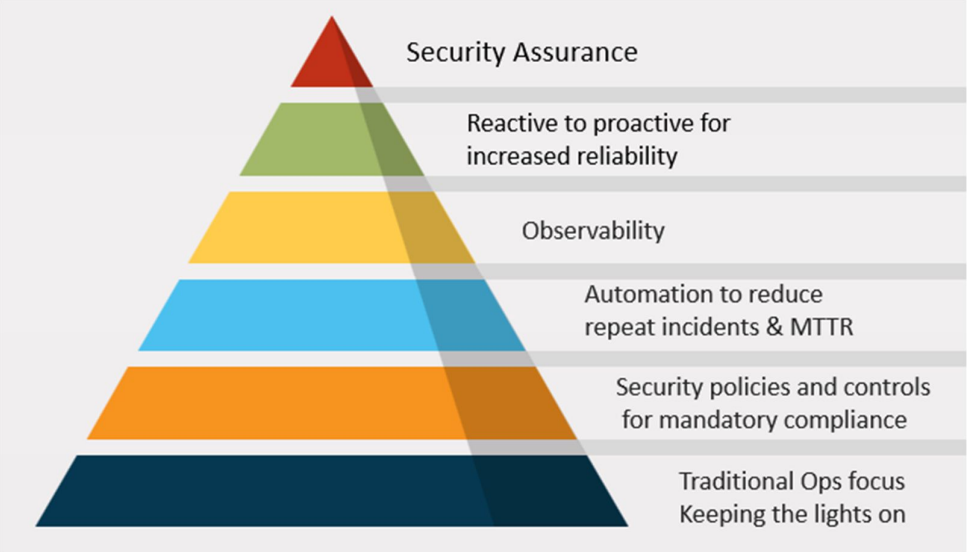


Obstacles & Challenges

The reality for development teams today is that the pressure to deploy code for production in minutes, not days, leaves no room for gaps between Dev and Ops. So, while they may understand the value of these development methodologies, the problems of increasing complexity, shrinking timelines and skill gaps persist, leaving best practices like DevSecOps and modern Site Reliability Engineering (SRE beyond the reach of many organizations.

[Callout: 50**%** of enterprises say organizational silos are a challenge to delivering value to the market faster, 49% cite legacy technology as a challenge, and 46% say resistance to change is a major issue (Google).]

The end result may be that precious IT resources are consumed by manual development tasks and old school "keeping the lights on" IT rather than gaining the competitive advantages that modern DevOps, automation and security best practices deliver.

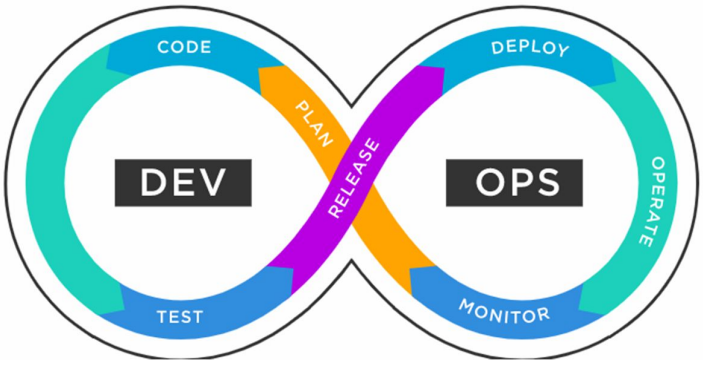


The Triple Play Solution

What is proving to be a competitive differentiator in the race to innovate software products is an environment where **development**, **operations** and **security** teams bring their specialized knowledge, tools and solutions into a constantly evolving, collaborative framework, where each shares responsibility for the entire product lifecycle. This point of view is based on multiple Xoriant engagements where we have demonstrated that by implementing DevOps (included DevSecOps), Site Reliability Engineering (SRE) and automated (and intelligent) monitoring and observability, ISVs and enterprises can actually reduce release cycles, strengthen process and product security, and leapfrog further complexity using the power of automation and, potentially, artificial intelligence (AI). Think of these practices as a strategy-driven Triple Play for optimization of every aspect of your SDLC. And, like a traditional triple play, the ultimate score depends on covering all the bases.

Triple Play #1: Building a Mature DevOps Environment

In a mature DevOps environment, the functional silos and knowledge gaps that created endless rework and last-minute testing are obsolesced by the integration of "Dev" and "Ops" flows and functions into every phase of the product lifecycle. When design, build, QA, test, UAT, deploy, monitor and feedback from SMEs, SREs and customers are in a constant loop, teams are able to achieve continuous improvement, rapid fixes, and streamlined development cycles.



By establishing a collaborative DevOps environment, your team – and your products - will benefit from the ability to deliver new products, features and updates faster with the confidence that they will meet constantly changing customer needs and preferences.

And, as your DevOps practice matures, individual team members will gain cross-functional knowledge and expertise with popular trends, such as no code/low code development, as well as much-needed time to ideate and innovate.

Hierarchy of Competence

Benefits

- Faster delivery and adoption of higher-quality products
- Increased scalability and availability
- More resilient operating environments
- Better resource utilization
- Pervasive automation with AI/ML
- Greater visibility into apps, systems
- More time and ability to innovate
- Increased customer satisfaction and revenue

Adding to the urgency, DevOps is rapidly moving beyond CI/CD automation in order to build a secure foundation for integration of emerging DevOps trends such as GitOps, AIOps and next-gen Site Reliability Engineering (SRE) (see Play #3). However, building that secure foundation requires incorporating DevSecOps across the SDLC.

DevOps shifts left to incorporate DevSecOps

While the exact naming and parameters are evolving, the need for continuously integrated, automated security across the entire SDLC is becoming more urgent as cyberattacks against ISV customers escalate. When your ISV teams incorporate security practices (aka DevSecOps) during the planning stages of your DevOps implementation and/or cloud migrations, it can provide an effective, automated security layer and repeatable processes to ensure consistent delivery of optimized, digital-age products.

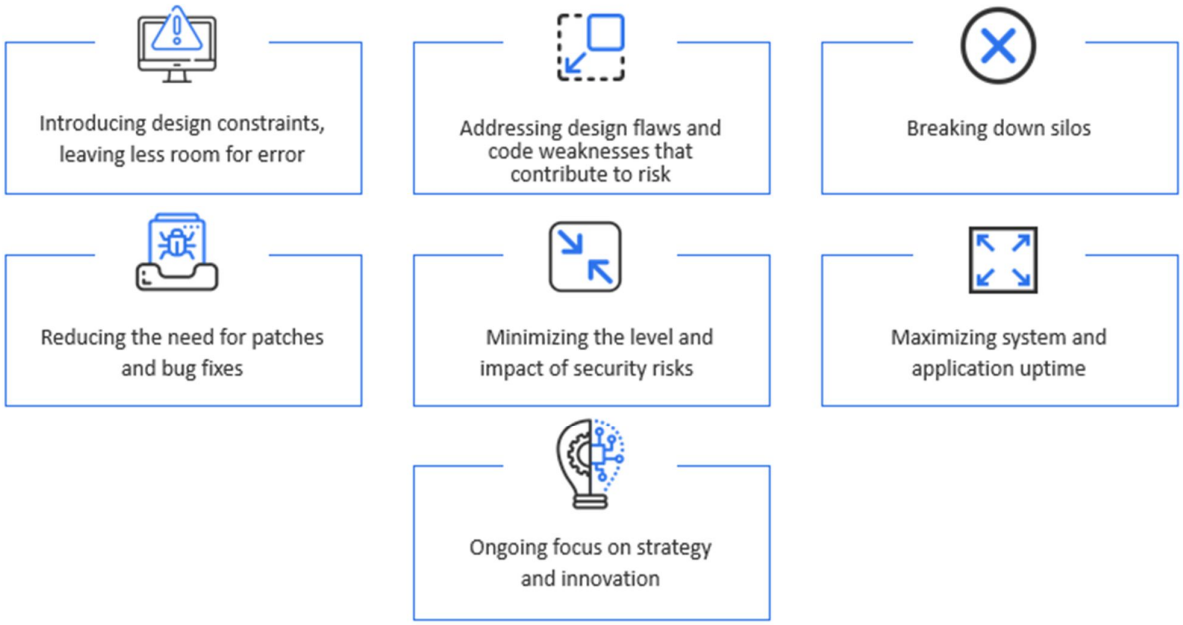
Far from being yet another add-on to DevOps, DevSecOps is an entire culture and tooling change that allows developers to receive automated outputs of SDLC security status throughout the development process. And, as more organizations adopt serverless, microservice architectures, Docker, Kubernetes, and other modern cloud technologies, automated SDLC security will probably become part of DevOps by default.*

Overcoming objections

According to a recent study, only 59% of respondents say they're building more security automation into their pipeline and only 22% of organizations have developed a formal DevSecOps strategy for integrating security into SDLC processes. However, an overwhelming percentage of the DevSecOps users reported accelerated incident detection (95%) and response (96%) efforts.1 So what's holding up adoption?

There are still questions impeding adoption. For example: Can implementing DevSecOps extend time-to-market cycles? Potentially, but this usually only happens if teams don't sufficiently automate the security code review process. As long as that is covered, a modern Security-as-Code powered product lifecycle can prove faster than the older DevOps model. Also: Can DevSecOps add steps to your DevOps? Yes, but more importantly, it will automate code verification and eliminate last-minute patches and rework, speeding time to market with higher-

quality, modern products designed to perform reliably in customer infrastructures and digital ecosystems by applying these best practices:



Benefits

- Security vulnerabilities are identified and remediated during development, avoiding customer dissatisfaction, revenue and reputational damage
- Increased return on investment (ROI) on existing security infrastructure.
- Automated, security means fewer mistakes or admin failures leading to cyber-attacks and downtime.
- Automation frees cybersecurity architects from configuring security consoles so teams can handle other pressing issues.
- Better communication and collaboration between teams.
- Greater flexibility in managing sudden changes during the development lifecycle.
- More opportunities for quality assurance testing and automated builds.

Triple Play #2: Observability & Monitoring

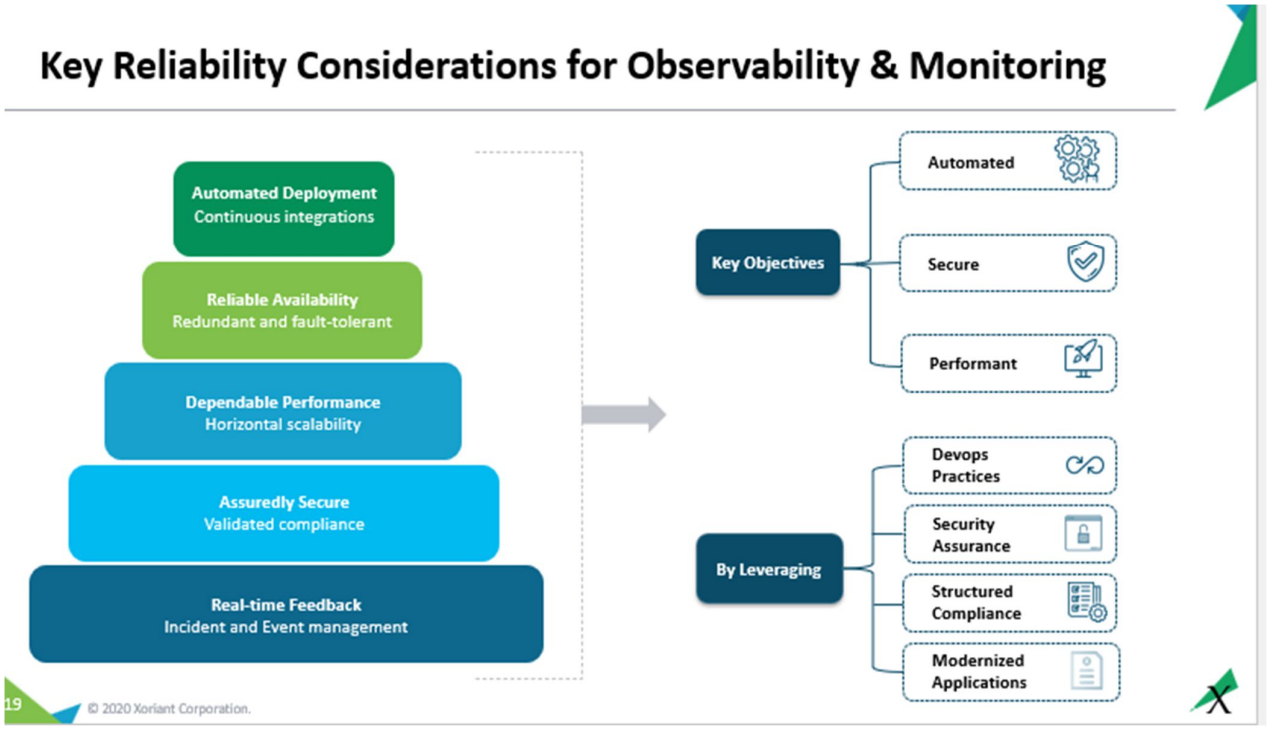
According to Gartner[2], "Enterprises are frustrated with the limitations of existing monitoring tools and, despite decades of investments, continue to rely on customers to notice an outage." [2]

Just as security lagged when DevOps introduced new processes and layers of complexity to product development, so monitoring tools have failed to keep up with modern digital and cloud-native production systems and environments. For example, compared to legacy technologies like virtual machines and monolithic architectures, containerized microservices increase the level of cognitive complexity due to the interdependencies between components. Issues like losing all modifications after container restart, and incompatible versioning between separately released components are common.*

Are observability and monitoring the same thing?
To be clear, observability and monitoring are not the same. Rather observability is composed of three components: monitoring, logging, and tracing. Monitoring provides details when a defined service level or quality criterion defined by the app developer has not been met. The logs contain the error reports of each individual software component. Tracing allows teams to identify the path a call has taken between services. This information should be accessible in a central dashboard.

The critical role of observability
Monitoring is helpful when we understand how systems fail, but as applications become more complex and distributed. so do their failure modes, making it extremely difficult to predict how and where they will fail. By making a system observable, teams can understand the internal state of the system. Using a combination of observability and more powerful monitoring tools, they can determine what is not working correctly and why.

The following considerations can help to proactively reduce the number and impact of errors and incidents.



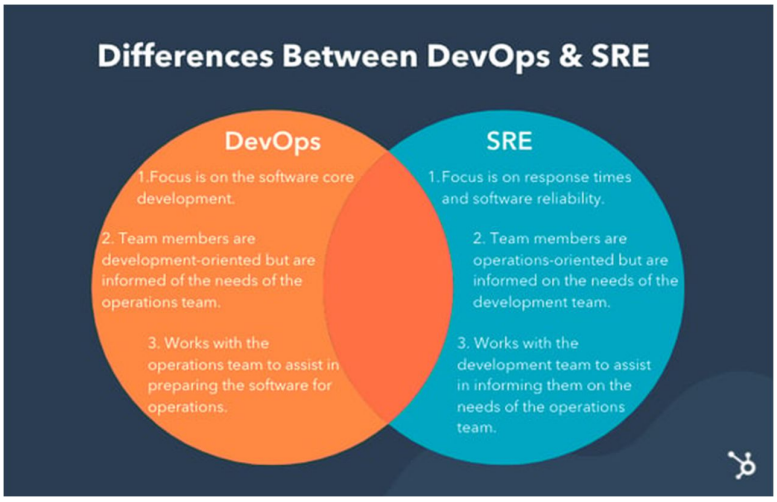
Finally, observability does not exist in a vacuum; it is both a result of, and integral part of, the DevOps, SRE, and cloud-native movements. The continuing evolution of these technical paradigms – and completely new innovations – will have a much better chance of successful adoption when an organization fully commits to the collaborative principles of DevOps, SRE and the shared responsibility of monitoring and observability.

Benefits

- Higher levels of application reliability and resiliency
- Increased efficiency and reduced costs through automation
- Developer time savings increases innovation opportunities
- Driving a culture of continuous improvement
- Improved customer satisfaction and retention

Triple Play #3: Implementing Modern SRE

SRE and DevOps are often referred to as two sides of the same coin, but they are complementary *not* interchangeable. The premise is the same – better collaboration between teams, automation, and several other factors. However, DevOps focuses more on the delivery, while SRE focuses on building system reliability and:



- Ensuring transactions are error free within cut-off time
- Automating issue detection and ensuring defects are fixed promptly
- Improving collaboration between different teams by reducing silos
- Reducing failure rates and downtime
- Automating repetitive tasks

Since Ops are software-defined, SRE tooling and techniques monitor toil (labor) and reliability to ensure consistent service delivery. While DevOps gathers metrics through a feedback loop, SRE employs SLIs, SLOs, and SLAs to measure and enforce system/application availability and reliability, and to automate and enhance ITOps functions such as:

- Release Management and Inventory mapping
- Disaster response
- Capacity planning and scalability
- Monitoring and observability
- Optimization and proactive fixes

It's important to note that any data, whether business indicators, system health metrics, or the time required to perform critical operational activities, is useful to help the team implement improvements that can ripple up and down the SDLC. So competent data management is also a critical component – and an increasingly difficult challenge – of the SRE function.

How Xoriant Expertise Can Help

From DevOps and DevSecOps to our SRE Services, Xoriant uses the best-practice approach where the reliability of the environment is key to product success. For example, every pipeline built for release/deployment embeds observability for the application and the infrastructure, as well as monitoring and logging/log-analytics capabilities.

By integrating security-as-code deep into the development process, DevSecOps ensures development teams carry out the task of programming with a security-first mindset. Done right, DevSecOps can help improve the quality, security, and functionality of enterprise products while keeping up with the accelerated pace of delivery, innovation, and evolving security regulations.

When combined with the rigorous oversight of Xoriant SRE, customers are assured consistent deployments, reliably available services, dependable performance and secured environments, with visibility on metrics and logs from day 0.