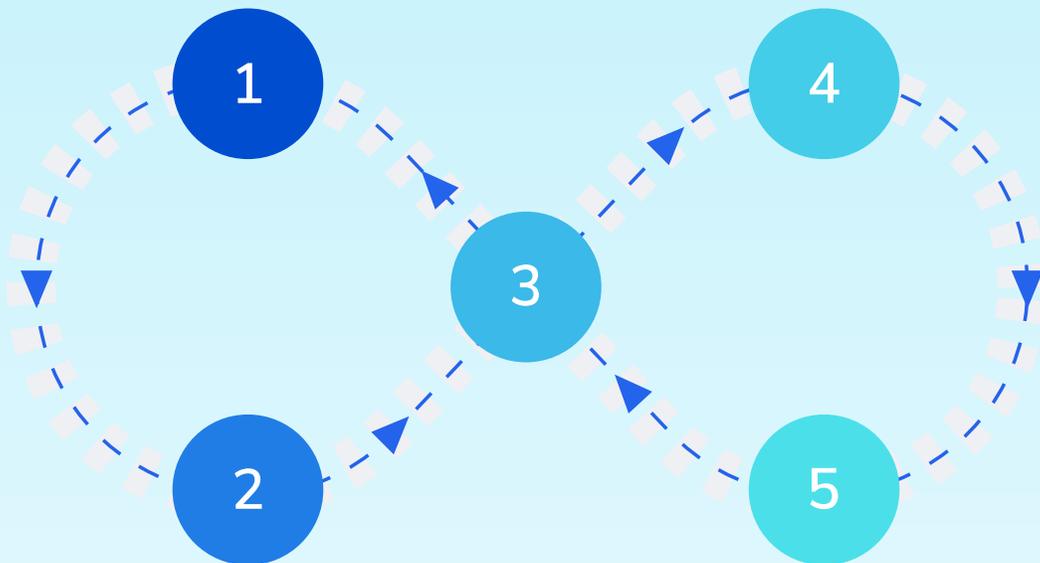


Copado's 5 Steps to DevOps Success

Follow in the Footsteps of
Today's Most Innovative Cloud Teams



#1 Low-Code DevOps & Testing Platform

Table of Contents

Copado's 5 Steps to DevOps Success

Introduction	1
Executive Summary	3
1 Step 1: Visibility	6
2 Step 2: Quality	10
3 Step 3: Speed	15
4 Step 4: Innovation	20
5 Step 5: Resilience	24
Now It's Your Turn	29



Introduction

Setting the Stage

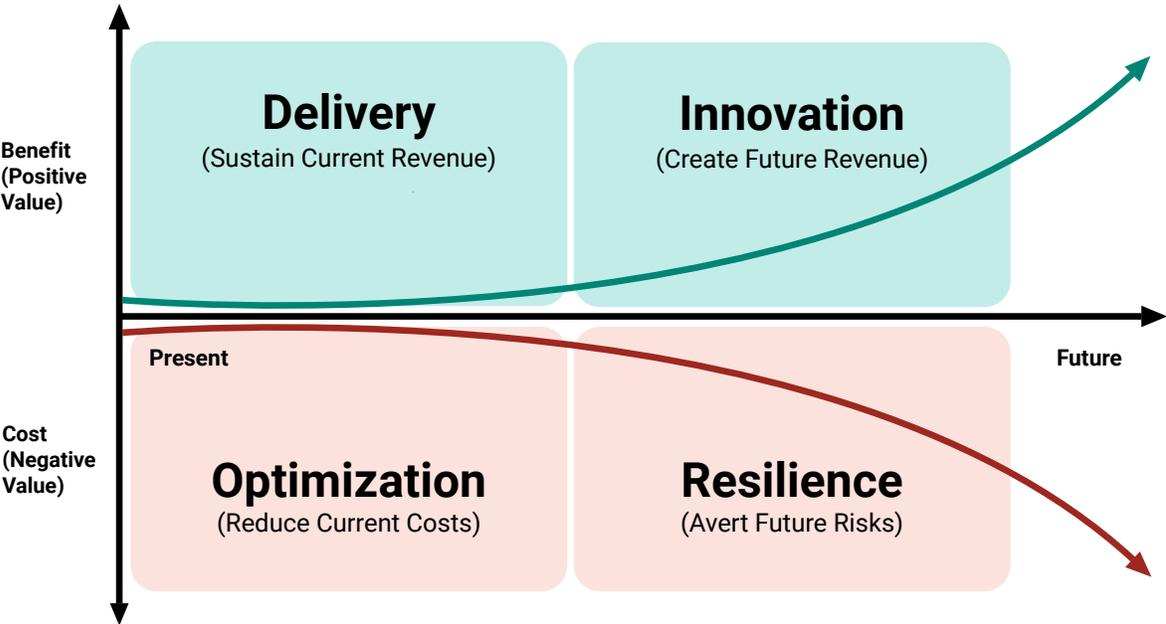
Today’s companies are under tremendous pressure to execute two intimidating tasks: Build the future of work for their employees and transform digital experiences for their customers. Teams need to accelerate top-line growth, improve margins and elevate business resilience – all at enterprise scale.

Why Is Low-Code App Development So Important for Digital Transformation?

Low-code cloud platforms are a critical way to leverage a larger community of innovators in the race to build a digital business. The avalanche of non-technical citizen developers reveals they face many of the same challenges as traditional coders: **Siloed technologies, processes and teams.**

A lack of connection and collaboration bottlenecks new innovation and business value – but low-code DevOps practices solve these challenges and give teams the digital edge they need to drive disruption. Low-code development is a powerful way to ramp up revenue and reduce costs: The two ingredients for business success.

Fundamental Business Concerns



Salesforce and Low-Code DevOps

Salesforce is a low-code development platform that enables admins to build applications through clicks – not code. But as teams have built more and more

customizations onto the Salesforce platform, the level of complexity has exploded. **And DevOps has emerged as the backbone that helps teams bring this complexity under control.**

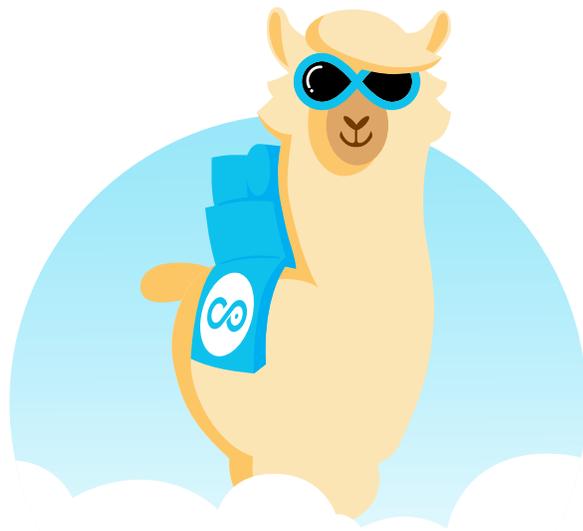
“By 2025, 70% of new applications developed by enterprises will use low-code or no-code technologies (up from less than 25% in 2020).”

Gartner

Deployments are often the trickiest challenge for Salesforce teams. It’s not just a matter of moving a few files – deployments require the sophisticated handling of thousands of XML files and careful management of dependencies. Companies like Copado equip teams with native low-code DevOps solutions that automate manual processes across the end-to-end pipeline and accelerate deployment.

Books like *Mastering Salesforce DevOps* explore how traditional techniques like modularization and continuous integration can be adapted for

the Salesforce platform. You can learn more about DevOps for Salesforce by reading Copado’s State of Salesforce DevOps Report.



Executive Summary

Copado's 5 Steps to DevOps Success

The success of your software development hinges on how your team works together. Unfortunately, many development projects miss the mark because stakeholders don't set the right benchmarks for success. They wind up locked into a DevOps methodology that does little to accelerate release cycles, improve quality, reduce risk or drive ROI.

Copado's Five Steps to DevOps Success are built to solve the everyday pain experienced by modern cloud teams. Each step offers a prescriptive – yet adaptable – playbook teams can follow to build a strong DevOps practice from the ground up and become a digital disruptor.

Visibility: You can't deliver business success without an end-to-end view of plans, changes, environments and performance. Situational awareness gives you the insight you need to improve.

Quality: A 360 view of your current state will likely reveal quality gaps. Shift-left testing bakes quality and compliance into the entire process – reducing change failure rates and time-to-recover from defects.

Speed: A firm foundation of visibility and quality lets teams move faster. Automating manual processes and eliminating downtime ramps up deployment frequency and time-to-value.

Innovation: You've mastered fast and friction-free deployments – and now it's time to optimize for impact and ROI. Fast feedback loops, adoption monitoring and effective prioritization unlock your potential for innovation.

Resilience: Digital disruptors are not only innovative – they adapt and pivot as business needs change. Resilient teams are laser-focused on continuous improvement, self-correction and experimentation.

Get the
fundamentals
down and
the level of
everything you
do will rise.

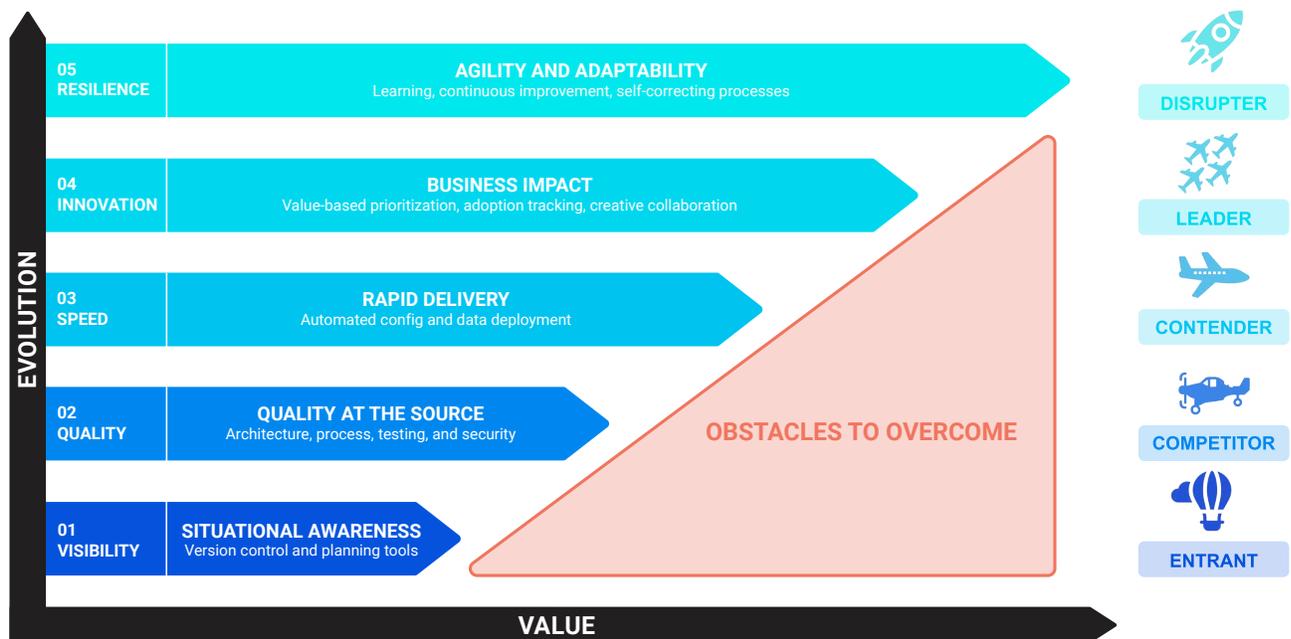
Michael Jordan



How High Can You Fly?

Gauge Your Organization's Risk Profile

Your organization's risk tolerance plays a big role in your innovation strategy. The five risk profiles in DevOps are **Entrant, Competitor, Contender, Leader and Disruptor**. Entrants encounter the least risk, but they arrive late to the party and miss out on the rewards of innovation. Disruptors bear the most risk, but they are in position to capture the most market share. Most companies reside somewhere in between. To begin unlocking DevOps ROI, you need to understand where you're at on the risk-to-innovation scale – and where you want to be. So you can ramp up your innovation without ramping up the risk.



Where Does it Hurt?

5 Root Causes of DevOps Pain

There's no such thing as a perfect software delivery cycle. Every team in the world has to deal with bottlenecks, bugs and broken functionality. When the flow of value is impeded, it impacts us as employees, managers and human beings. **So where are DevOps teams feeling pain?** These are the top culprits we've encountered:

- Confusion due to lack of visibility.
- Errors due to a lack of quality controls.
- Inefficiency due to a lack of speed.
- Poor ROI and a lack of innovation due to a lack of data analysis.
- "Unpleasant surprises" that stress out teams and are difficult to resolve.

How Do You Remove Waste from the System?

The source of DevOps pain is different for every company – and it can be tough to see the forest from the trees. In many cases, software delivery teams are missing clarity around planning, architecture, the history of changes and system interdependencies. Without visibility into the end-to-end DevOps process, teams waste time and money trying to gather everything they need to deploy software. By addressing these inefficiencies, teams unlock opportunities for innovation and avert future risk by adapting quickly to changing needs.

Keep in mind: You may have visibility, quality and speed in some areas but not in others.

You can pinpoint the source of your pain – without causing new problems – by implementing the Lean Production Thinking Process and answering three key questions:

- What needs to be changed?
- What should it be changed to?
- What actions will cause this change?

Going through the Lean Production Thinking Process for each of the 5 Steps will help you gauge what needs work and how to approach it.





Visibility

Step 1 to DevOps Success

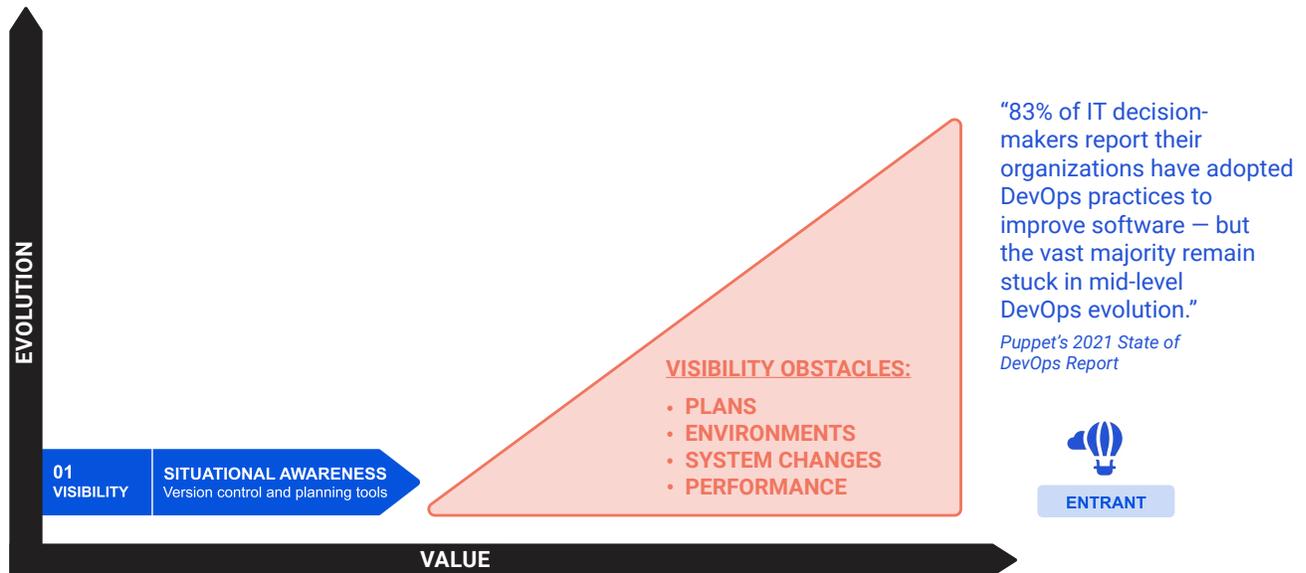
Can your teams see the big picture? Can they trace the end-to-end pipeline and track user story data in one place – from ideation to customer adoption? Today’s digital enterprises have a mountain of powerful insights at their fingertips, but they often struggle to connect the dots due to siloed tools and siloed teams. Blindspots emerge – and dreams of cloud transformation are derailed by poor decision making and wasted time.

Low-code development teams must find a way to collaborate across countries, continents and time zones. And the first step is **Visibility**. End-to-end version control and planning tools deliver traceability, auditability and compliance – giving teams the situational awareness they need to unblock bottlenecks and ramp up digital maturity without ramping up the risk.

Once you understand the bottlenecks your organization faces, you can start to solve them. Visibility gives you the information you need to turn your attention further down the pipeline. It lets you focus on increasing quality, improving your speed to deployment, tackling innovation and achieving resilient development processes.

Visibility is the foundation for effectiveness

Understand your current state, what's changing, and where you can go



Why is DevOps Visibility Important?

Modern software development is more complex than ever – a single enterprise project can have dozens of stakeholders and hundreds of APIs. **Without visibility, gleaning insights and troubleshooting issues is like looking for a needle in a haystack.** And that's why it's our First Step to DevOps Success.

Am I Missing Out On DevOps Visibility?

According to the Harvard Business Review, more than 8 out of 10 IT decision makers have adopted DevOps practices – but not all have reported success. When teams run into DevOps pain, a lack of visibility is often the culprit. Here are three signs you might be missing out on visibility:

1 Manual Methods	2 Muddled Goals	3 Merge Hell
Manually tracking changes is an easy habit to get into – but it fosters a culture of inconsistent tracking. It's easy to make a small change and forget to log it. Manual methods silo mission-critical data and make it tough for teams to stay in sync.	Development teams often get lost in the fog of war and forget about the big picture. Unclear goals and erratic feedback mechanisms make it a challenge to prioritize the right things. Strong feedback cycles drive a strong user experience.	Joining multiple DevOps branches can get complex in a hurry. Even teams with Continuous Integration are beset by merging bottlenecks. Full pipeline visibility helps teams simplify branching and manage even the most complex merges.



7 Elements of DevOps Visibility

Time and energy are precious in today's innovation economy. Teams can't afford to fly blind and hope for the best. Visibility is the backbone of DevOps because it uncovers the insights you need to get work done – in the most efficient way possible. Let's take a look at seven elements that power true end-to-end visibility.

1 Plans	2 Environments	3 System Changes
Visibility sheds light on use cases and project roadmaps. Say goodbye to redundancies, missed benchmarks, solution overlap and scope creep.	A clear environmental view helps teams identify and prioritize work in progress vs. work ready for deployment – solving the headaches of managing multi-cloud development.	Each change must be tracked to guarantee consistent versions. Visibility also eliminates overwrites, metadata conflicts and the introduction (or recurrence) of bugs.
4 Architecture	5 Value Streams	6 Dependencies
Architecture is hidden inside intricate relationships stitched across metadata – making visibility a challenge. Clear governance is the key to unlocking clear views of architecture under the hood.	Value streams are all about flow. Visibility helps teams track the flow of work and diagnose benefits and risks throughout the pipeline.	Visibility connects dependencies across the software supply chain (teams, systems, packages, codebases) and visualizes the potential impact of changes.
7 Performance		
Visibility into team performance is a great way to identify bottlenecks, work distribution and opportunities for improvement. Tracking the four key DevOps metrics (lead time, deployment frequency, change fail rate, time to restore) reveals how well the team works together.		



Visibility Best Practices

Ready to roll up your sleeves and inject visibility into your development cycle? Tackle these visibility best practices to drive DevOps ROI.

Adopt Version Control. Version control automatically tracks code changes. Updates are recorded in a centralized log for developers and stakeholders to reference and roll back changes.

Connect User Stories and Metadata. Syncing changes in version control to user stories gives you a sweeping story that covers the who, what, where, when and why.

Organize the Architecture. Noisy metadata can obscure architecture visibility. Shine a light on your architecture by organizing it into folders and packages.

Sync the Cloud Stack. Digital enterprises rely on Multi-Cloud applications and development processes. Cloud-native DevOps provides visibility into Multi-Cloud tech stacks and ties them together.

Map Value Streams. Visualize workflows to locate bottlenecks, eliminate waste and focus on your strongest value drivers. Value stream management unlocks a complete picture of tools, applications and software solutions in one place.

Leverage Four Key DevOps Metrics. Gauge your performance based on lead time, deployment frequency, change fail rate and time to restore. Find out whether your performance is elite, high, medium or low.





Quality

Step 2 to DevOps Success

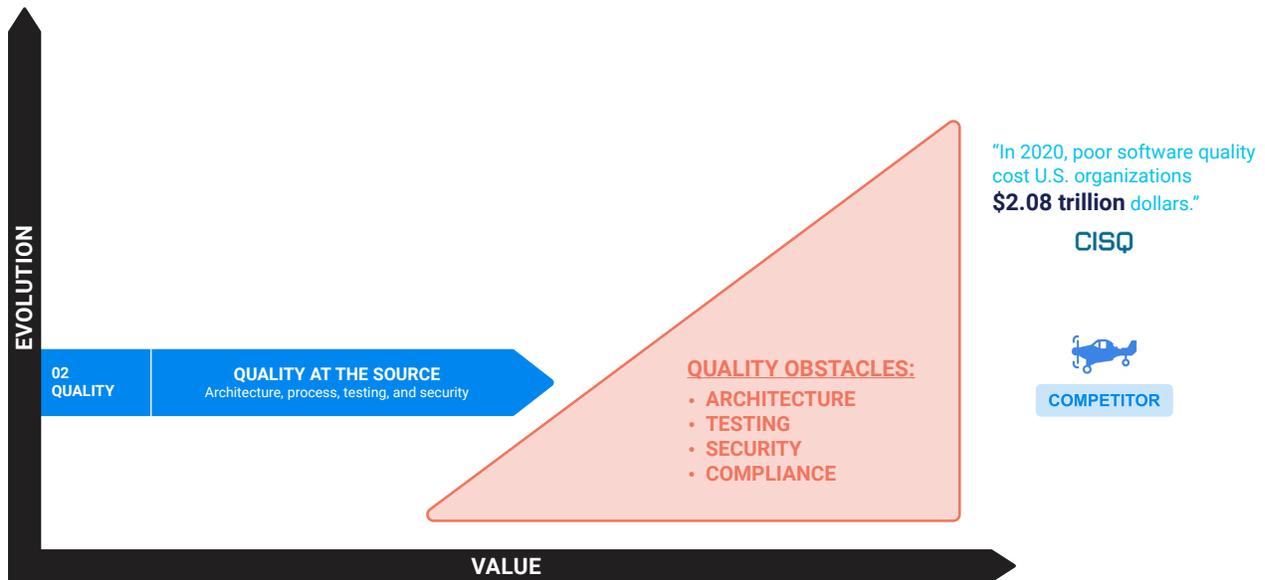
Can you measure the quality of your pipeline? Can you control it? Many of today's development teams still rely on QA professionals to manually test code after it has been merged and promoted to staging. This takes forever and leaves a lot of room for human error and lack of documentation. As a result, bugs creep into the code and releases are bottlenecked at the finish line.

The global pace of innovation demands end-to-end quality built directly into your DevOps pipeline. Modern teams must shift both left and right to bake testing into the entire development cycle – from deployment to live in-production UX tests. Most important of all: **your testing should be automated.**

Comprehensive testing covers UI functionality, API data and security systems across multiple platforms. Going the automated testing route allows you to pursue continuous delivery – from enhancing test coverage to eliminating production failures to expediting the release of high-quality software.

Reliability is the limiting factor on growth

Optimize for quality so you can move faster ... safely



Why Do You Need a Test Automation Strategy?

Today's digital enterprises must adapt to upgrades and innovations at lightning speed in order to survive. Rapid transformation is a recipe for growing pains – but nearly any headache can be remedied by strategic implementation and organizational readiness. The roadmap to a swift and successful deployment is fueled by a software test automation strategy.

Testing Challenges

Data discrepancies are the Achilles' heel of automation implementation projects. The technology you use is only as good as the data you feed it. Test automation solutions can predict release quality, evaluate the quality of test metrics and repurpose test cases. To avoid rewrites, your data source storage and maintenance must be up to standard.

Another common testing challenge: Understanding which systems to prioritize and test first to protect the business. Creating a simple and highly visible flow of information is one of the most powerful steps teams can take to improve test effectiveness.

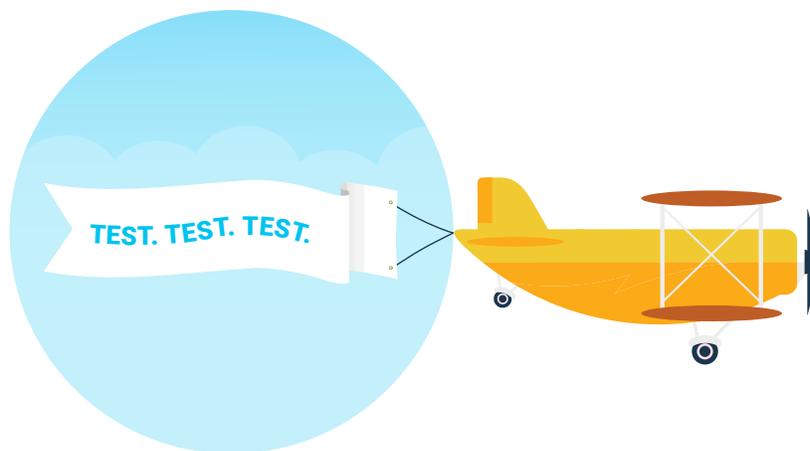
Meeting End User Expectations

Technology continues to raise the bar on user expectations. Modern users are looking for three things:

Speed. Instant access to new features. Lightning-fast load times. Error-free updates and a product that runs in production without a hitch. Your company's competitive edge depends on how much speed you can muster.

Quality. Poor quality experiences lead to broken customer trust. Teams must ensure high-quality data, product design and software functionality.

Experience. A poor user experience can spell doom for your product and your brand image. Listen to your end users throughout the testing process and apply their feedback. End-to-end testing can help companies take a step back and observe the lifecycle of a system from start to finish.



4 Models of Software Testing

The realm of software testing features four schools of thought.

Classic Model

In the classic model, a centralized QA team owns testing. Development projects are outsourced to specialized system integrators under high-value, long-term contracts who scope them in advance – “Waterfall” style. In some cases, the testing phase happens months (or years) after the development phase. QA teams are often under pressure to cut corners and complete work quickly. QA managers map out testing strategies, spend resources and oversee teams of testers.

Agile Model

This model requires testers to work closely with developers and business analysts and implement new sets of requirements introduced in every sprint. Agile requires exploratory testers to rapidly test software releases and weed out defects in new features. This approach is superior to the Classic Model when it comes to flow speed and test effectiveness.

Shift Left

Shift-left testing amplifies quality by addressing it at the source and baking it into the planning and development process. Teams that adopt a shift-left mindset practice test-driven development and leverage a small number of business users to guide test design during the early stages of a project.

Shift Right

A shift-right approach tackles testing in production – critical for complex systems whose behavior can't be predicted in advance. Traditional test strategy assumes production systems will behave in predictable ways. But web-scale systems are often too expensive and complicated to replicate for testing. They involve feedback loops, delays and an enormous range of possible data and user behaviors that make their behavior impossible to predict.

Shift-right testing provides continuous feedback and real-time observability. Practices like feature flags allow capabilities to be turned on and off in production – which lets teams test and experiment even after software has been released into the wild.



Crafting a Future-Proof Test Automation Strategy

What's your definition of success? Are you looking to optimize the experiences of your internal and external customers? Are you measuring the risk of different changes? How do you balance upfront costs (creating automated tests) with the long-term savings of automation? The first step: distinguishing creative work from repetitive tasks. **Repetitive work can be automated – creative work cannot.** For each testing scenario, you need to estimate the frequency of changes, the likelihood of broken functionality and how long it would take to manually test vs. automate.

Low-code test automation empowers both technical and non-technical users to quickly write tests. A solution that enables more tests to be automated more quickly dramatically reduces the overall risk and cost to the business. QA professionals are famous for their patience and persistence – but concern for quality is their real skill. Easy-to-use test automation tools make it easy for anyone to exercise creativity and protect functionality.

The software testing lifecycle includes **requirement analysis, test planning, test case development, environment setup, test execution and test closure.** Whether tests are manual or automated, these stages are the same. Automation simply allows teams to move faster and execute greater test coverage.

Build Your Testing Toolkit

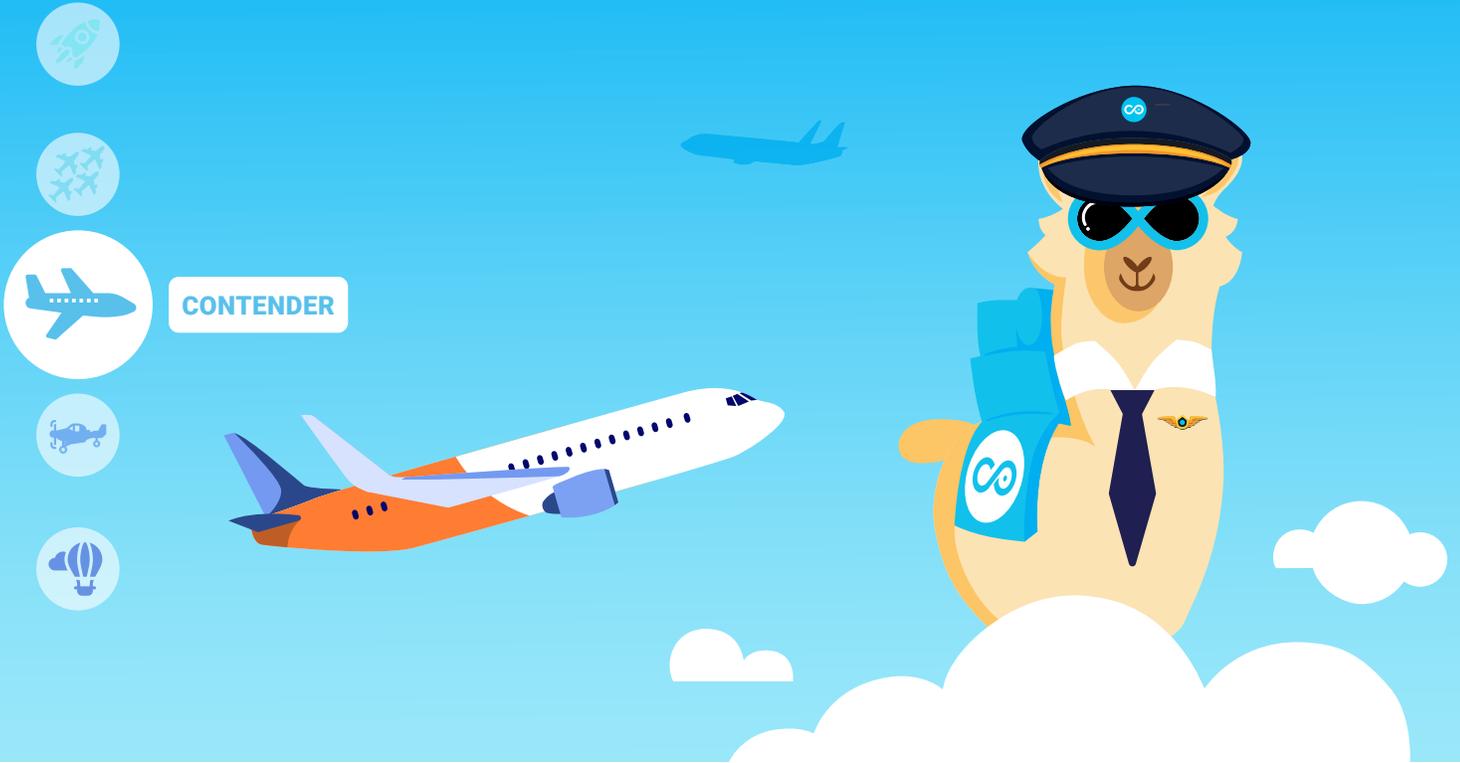
Any new software testing tool needs to play nice with your existing tech stack. Follow our checklist of recommended features for your future software test automation strategy.

Low-code or no-code functionality. Low-code and no-code automation lowers the barrier for anyone to create and execute tests without years of coding experience.

Intelligent automation. Your testing solution should automate more than routine tasks. Intelligent automation can report potential defects, take automatic screenshots of errors and predict the quality of upcoming releases.

End-to-end integration. Modern businesses leverage a combination of business process management tools. Make sure the automation solution you choose is built to support each of the applications your business relies on.





Speed

Step 3 to DevOps Success

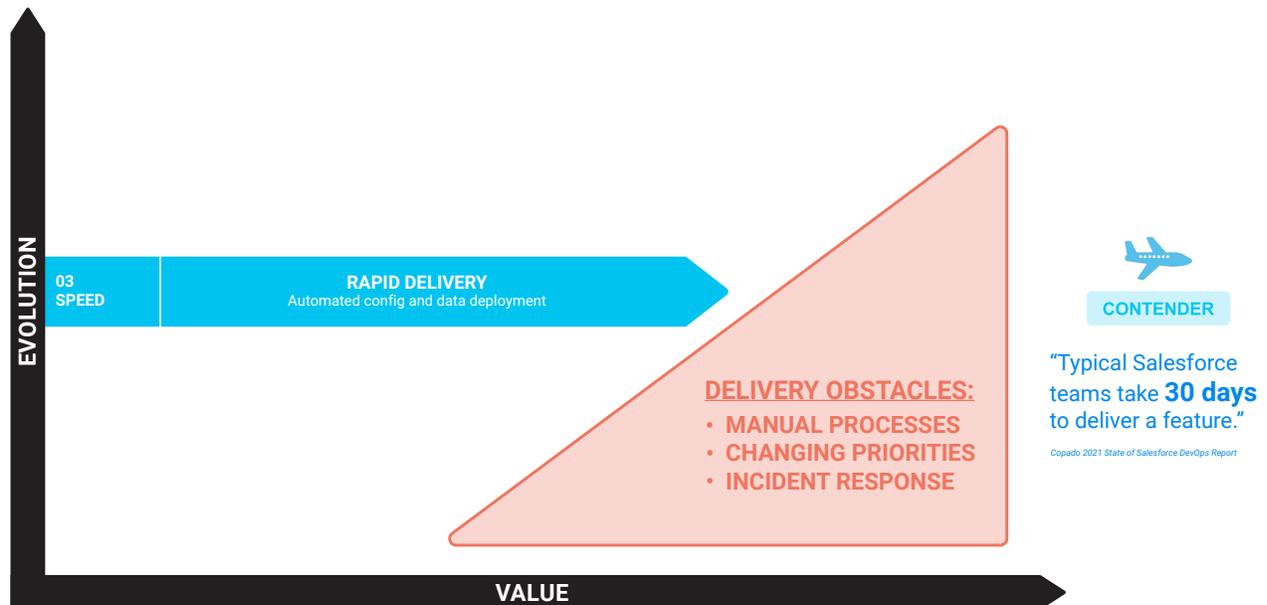
"The pandemic fast-tracked digital transformation by more than five years according to 96% of leaders." – Twilio

If teams can't bring both quality and speed to the table, they'll be bogged down by an endless number of manual processes and shifting priorities. To stay competitive in this race for innovation, companies must automate deployments to drive development velocity and stop wasting valuable time. An end-to-end CI/CD process lets you proactively address conflicts, automate manual tasks and templatize pipelines.

Building software is slow, painstaking work – but development teams are under pressure to move as fast as possible. Automating key parts of the development process removes the tedious tasks that bog down developers and frees up teams to release software at lightning-speed.

Speed lets you capture opportunities faster

Speed opens the door to innovation



3 Common Challenges that Slow Down DevOps

Bottlenecks: Breakneck development means a massive uptick in deployments. But you can only deploy as fast as your slowest dependency – and this creates pipeline bottlenecks.

Code Conflicts: Collaboration without visibility sparks confusion. Without a strong tracking system, transparency in the development cycle becomes impossible. Teams struggle to merge branches, track errors and resolve issues. Shifting priorities and industry standards can also introduce conflict and slow down the process.

Repetitive tasks: Tedious and time-consuming tasks still require a high level of precision. For instance, many testing processes are extremely repetitive but still prone to human error. If a tester is in a rush and overlooks a defect, it may not be caught until later in the release cycle – when it’s exponentially more expensive to fix.



The Solution: CI/CD

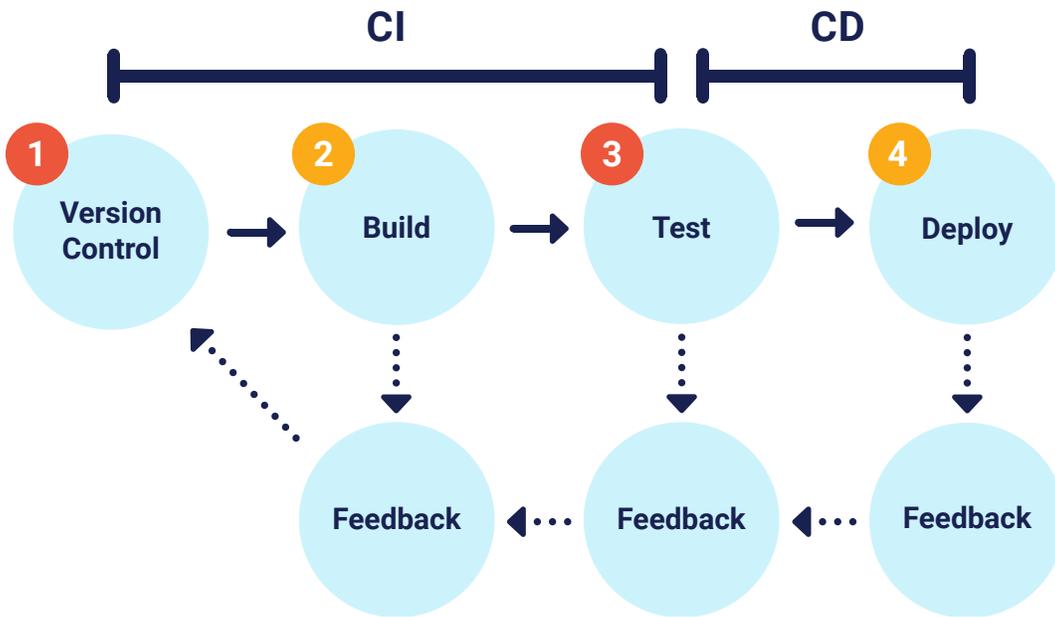
It's easy to avoid bottlenecks and code conflicts if your release cycle is slow and steady. However, today's need for digital transformation gives companies no choice: **Go fast or get left behind.** So how do teams accelerate their development cycle without sabotaging quality? They embrace a **continuous** mindset.

Continuous Integration: Companies that practice CI integrate code and merge small batches of changes early and often from multiple developers. The goal is to resolve conflicts in small batches rather than waiting for one large update and letting issues accumulate. Smaller commits help teams discover issues faster and patch them up before they cause disruption. Automated testing provides a layer of protection – and efficiency – by checking and rechecking integrations thousands of times.

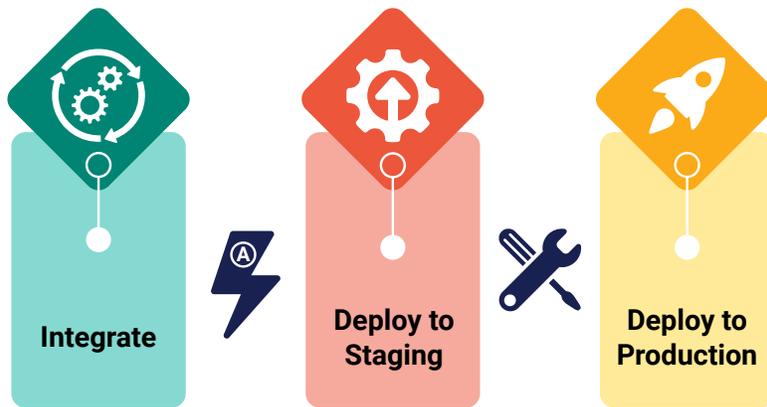
Continuous Delivery: Companies that practice Continuous Delivery harness the power of integrated teams and technologies to deploy on demand. They streamline testing, staging and deployment to automatically package new software and roll it out as soon as it's ready for the customer. Reaching a state of mature CD tends to take much longer than CI and requires the perfect fusion of technology, process and people.

Continuous Deployment: Companies that practice Continuous Deployment build on Continuous Delivery by automating the final stage of deployment. Every change that goes through the pipeline is automatically deployed directly to customers. This can be useful for hot fixes – but is risky with more complicated changes. Many organizations opt for Continuous Delivery over Continuous Deployment or limit Continuous Deployment to specific use cases.

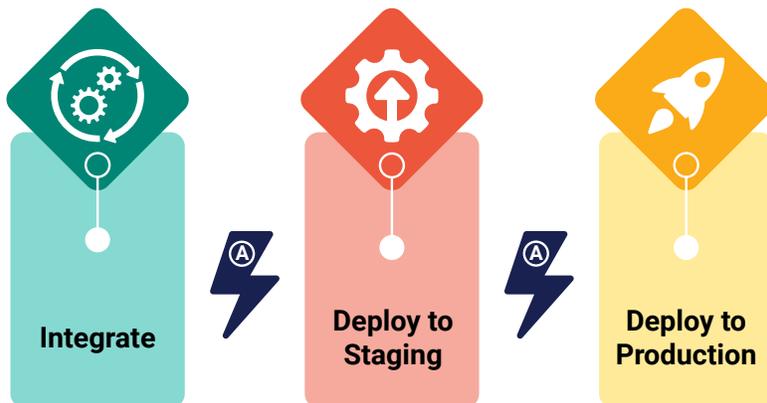




Continuous Delivery



Continuous Deployment



Why You Need Both CI and CD for DevOps Success

A CI failure may waste a few hours of a developer's time and delay delivery of a change. But a CD problem can prevent people from doing their jobs, corrupt production data and expose security vulnerabilities. The mission of CI is to run a fast set of tests for each change to confirm nothing broke. But work delivered to production must navigate a battery of comprehensive tests to make sure it meets every functional and structural requirement. Teams that practice both CI and CD have the ability to:

- Optimize version control branching for speed and clarity.
- Auto-resolve merge conflicts and auto-apply manual resolutions.
- Fully or partially automate deployments.
- Deploy small changes to production every day.
- Quickly create and refresh environments – and keep them in sync.

Automation: The Engine that Fuels CI/CD Pipelines

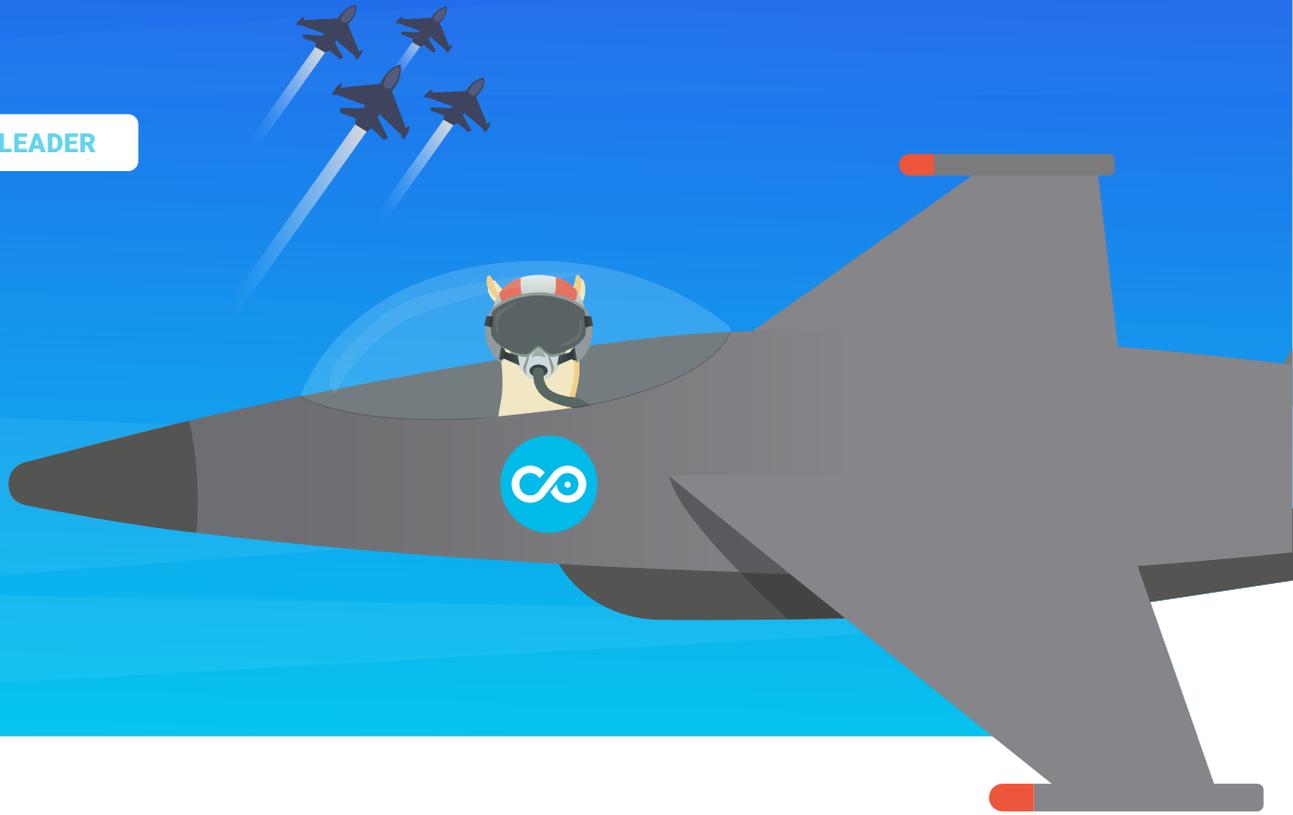
Automation is central to the success of the CI/CD pipeline. It allows Developers to commit a change and see it in a deployable state within minutes. When developers introduce new commits, the system automatically generates a build and runs it through a gauntlet of automated testing. If it passes, the build moves on to the application servers and eventually gets released into the wild. Every step of the way, admins are kept in the loop and notified if the build fails a test. **And it's all made possible by the power of automation.**

Automation tackles repetitive, time-consuming tasks like testing and merge conflict resolution. Collaboration goes through the roof when developers can automatically integrate code. To alleviate risk, teams can automate quality gates and build compliance controls directly into their pipelines. Automation lets you move faster – with less error – while freeing up your developers to work on more creative tasks.





LEADER



Innovation

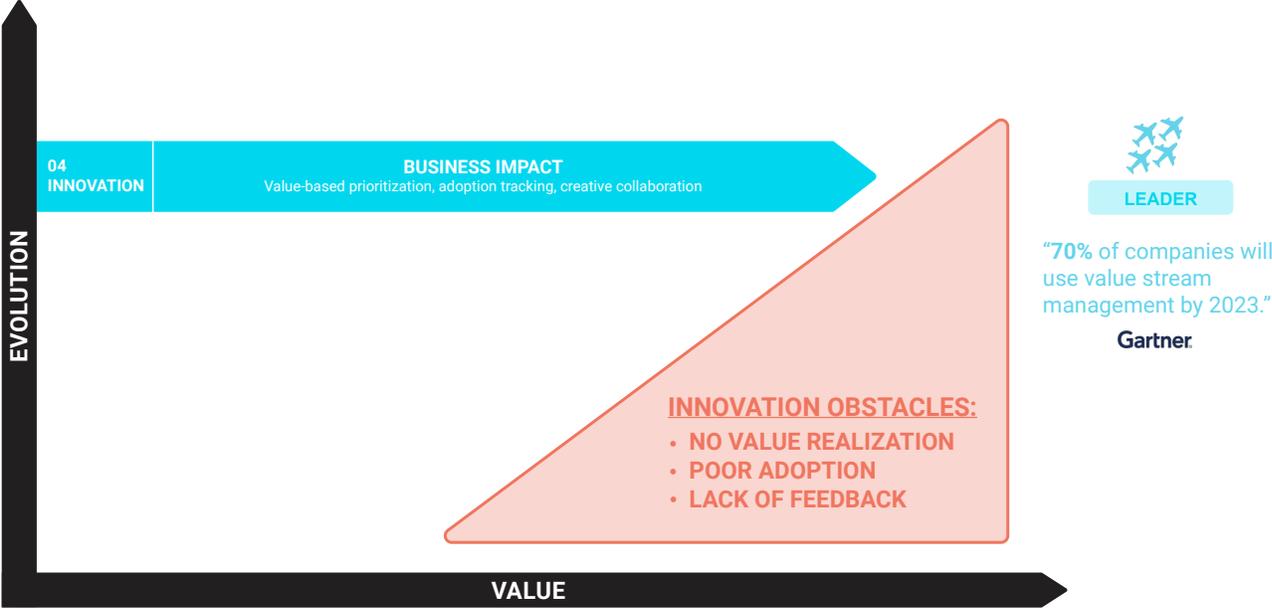
Step 4 to DevOps Success

Organizations farther along in their digital maturity don't just innovate more — **they innovate differently**. According to Deloitte, only one in 10 organizations view their innovation as a strength. But for digitally mature companies, a whopping 81% are happy with their level of innovation.

At this point in your DevOps journey, you've laid the correct technical foundation for visibility, quality, and speed. Now the floodgates have been opened for innovation and improvement. Innovative firms free up the time and resources they need to ask the right questions — questions that tie directly to business value.

Building a digital business requires five key strengths

5 steps to Salesforce DevOps success



Build a Seamless Flow of Information

Seamless communication and fast-feedback loops are the lifeblood of DevOps success. CI/CD processes rely on cross-functional collaboration and shared data – they cannot exist within a traditional, siloed development environment. By creating an ecosystem that fosters collaboration, you empower your teams to pursue innovation.

How Does Low-Code Break Traditional Barriers to Development?

What happens when you combine low-code platforms with DevOps strategy and CI/CD best practices? **You create the conditions for a culture of innovation.** Tools don't create culture – but they can lay the groundwork for new processes and foster new ways of thinking. By reinforcing your DevOps practice with iterative processes and an innovation-first culture, you can achieve better outcomes and maximize ROI.

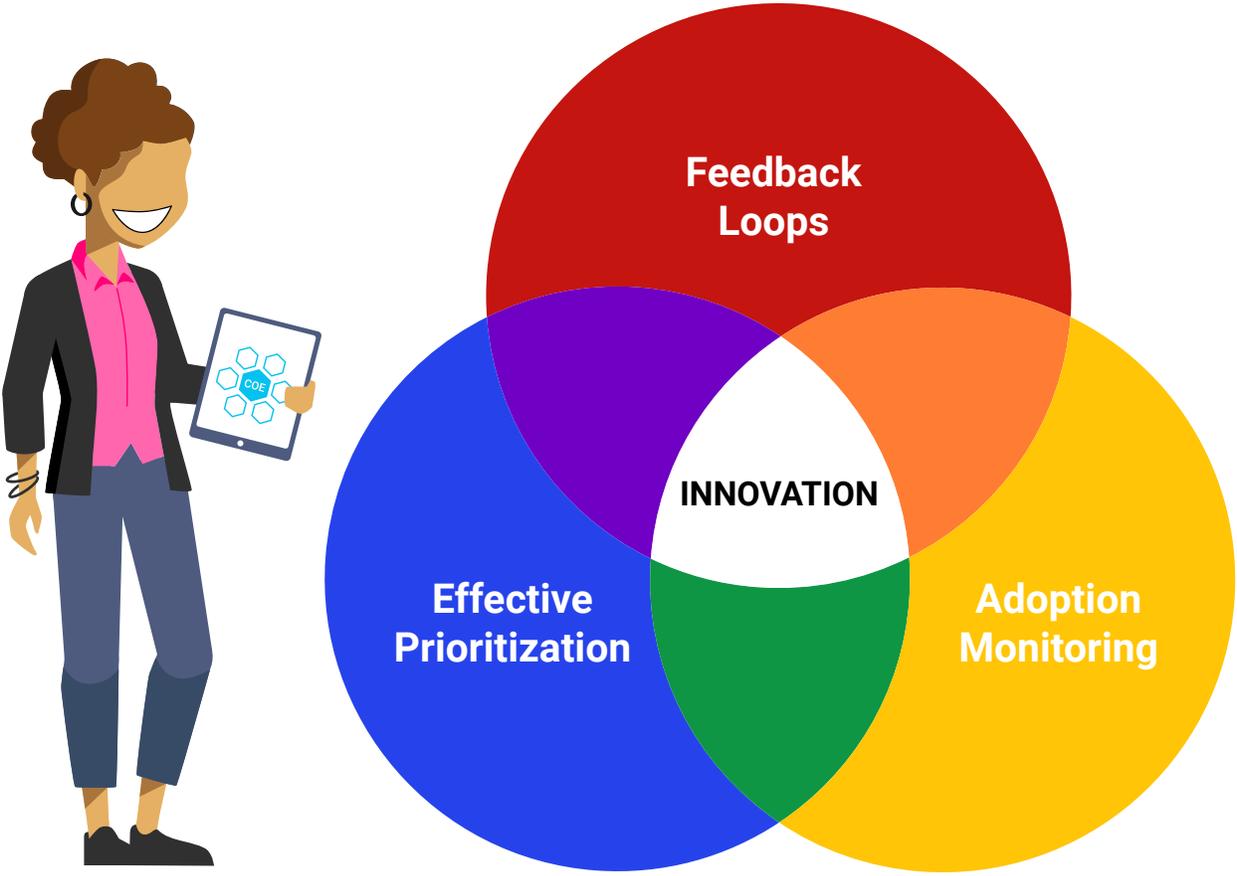


3 Ingredients of Innovation

Feedback Loops, Adoption Monitoring, Effective Prioritization

Innovation bottlenecks can stifle any dream – but teams can identify and resolve issues with speed and ease thanks to fast-feedback loops, the adoption of value stream management and insight-driven prioritization.

Teams must be able to collaborate quickly, pinpoint adoption lags and zero in on the most important action items and user stories. Mastering these three areas creates an agile, innovative environment that can easily adjust to even the most volatile markets. And be sure to audit innovation practices and perform post-mortems on disruptive incidents to better determine where to make improvements and adjustments.



Visualize Your DevOps Pipeline with Value Stream Mapping

You can't measure innovation if you can't tie it to business outcomes. Teams that fly blind often end up wasting their resources on projects that never add true value to the business. Even worse – misplaced innovation can lead to a loss of competitive advantage and market share.

Assemble a Value Stream Map to investigate the “Why” behind your DevOps results. Identify the metrics you need to accurately gauge the health and performance of your delivery process. Mapping out the flow of business value from ideation to completion tells you whether or not your activities are directly driving value for your customers.

Promote Measurable Innovation With Value Stream Management

For innovation to yield results, it must be continuously measured. Constant feedback loops, course corrections and actionable insights combine to create measurable patterns of innovation.

Today, enterprises leverage hyper-automation and quality intelligence to make their systems more measurable. They rely on algorithms and AI to track the four key DevOps metrics: **lead time, deployment frequency, change fail rate and time to restore.**

Automation: The Secret Sauce of Innovation

Architecting the DevOps pipeline with automation that supports CI/CD helps teams execute two tasks: **Complex data merges and conflict flagging.** Automation can also help you achieve a shorter feedback cycle to make sure the features you're developing are on track and on target. More automated feedback gives teams the confidence to make the right strategic decisions.

Finally, automation can help you prioritize work by value rather than a “first in, first out” approach. All customer requests are not created equal – and prioritization can improve customer satisfaction.





DISRUPTER



Resilience

Step 5 to DevOps Success

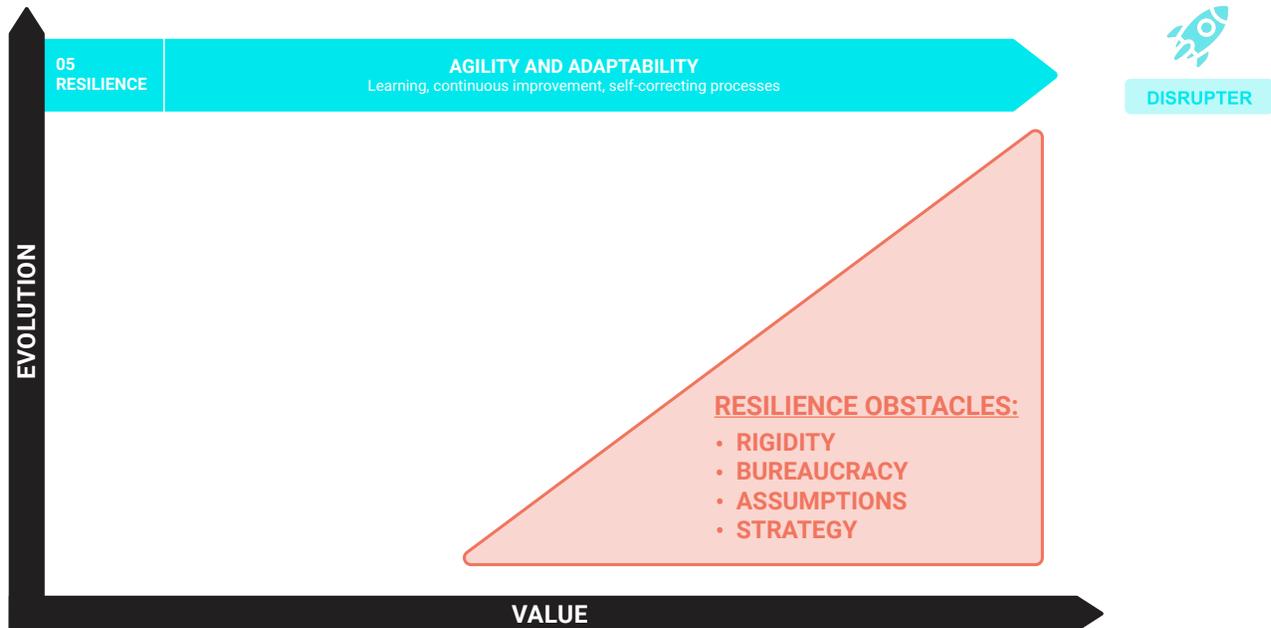
The first four steps to DevOps Success (Visibility, Quality, Speed and Innovation) establish a foundation for teams to deliver valuable work at enterprise scale. And the fifth step – Resilience – provides the final piece to the puzzle. It unlocks the ability to adapt to changing conditions in real-time.

Change is necessary for survival in today's hyperspeed world – and experimentation is the driver of change. Resilient DevOps teams leverage an agile culture that adapts and self-optimizes in real-time as their business needs evolve. They relentlessly search for better ways to resolve organizational challenges. Resilient organizations are reliable, consistent and compliant – all while reaping the benefits of faster release cycles and end-to-end digital transformation.



Building a digital business requires five key strengths

5 steps to Salesforce DevOps success



The Twin Hallmarks of Resilience

Building Predictable and Adaptive DevOps Teams

Predictable DevOps provides peace of mind. Without it, organizations can't guarantee the quality, security or compliance of their product. To build a resilient DevOps lifecycle, teams must be able to consistently respond to failure states and adapt to internal and external pressures. Resilience improves business continuity, reduces disruption and paves the way for digital experimentation and transformation.

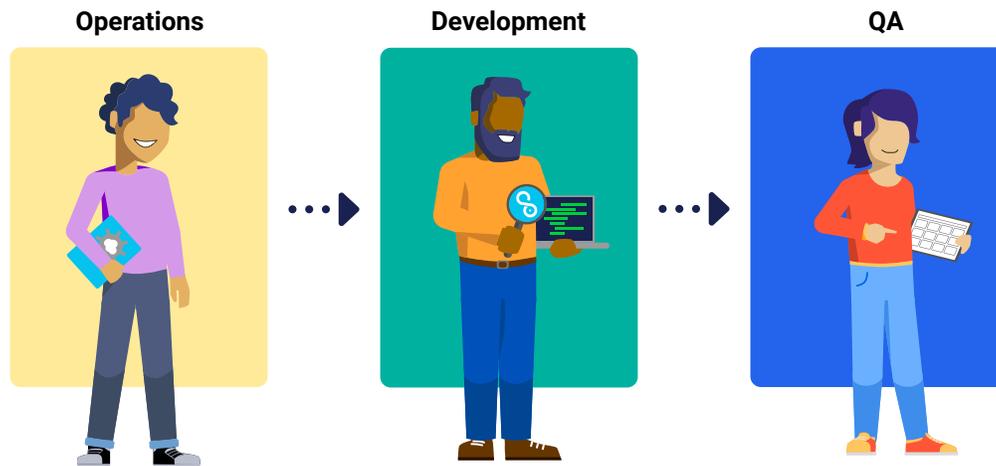
Micro-Resilience and Macro-Resilience

Change is constant – from business reorganization to personnel shake ups to external changes due to new market opportunities, trends and competitors. And the larger a company is, the more changes they have to track. On a micro scale, resilience means being able to adjust to all the tiny challenges that arise each day. On a macro scale, resilient teams adjust to seismic shifts like new competitors, new market opportunities, COVID and supply chain headaches.

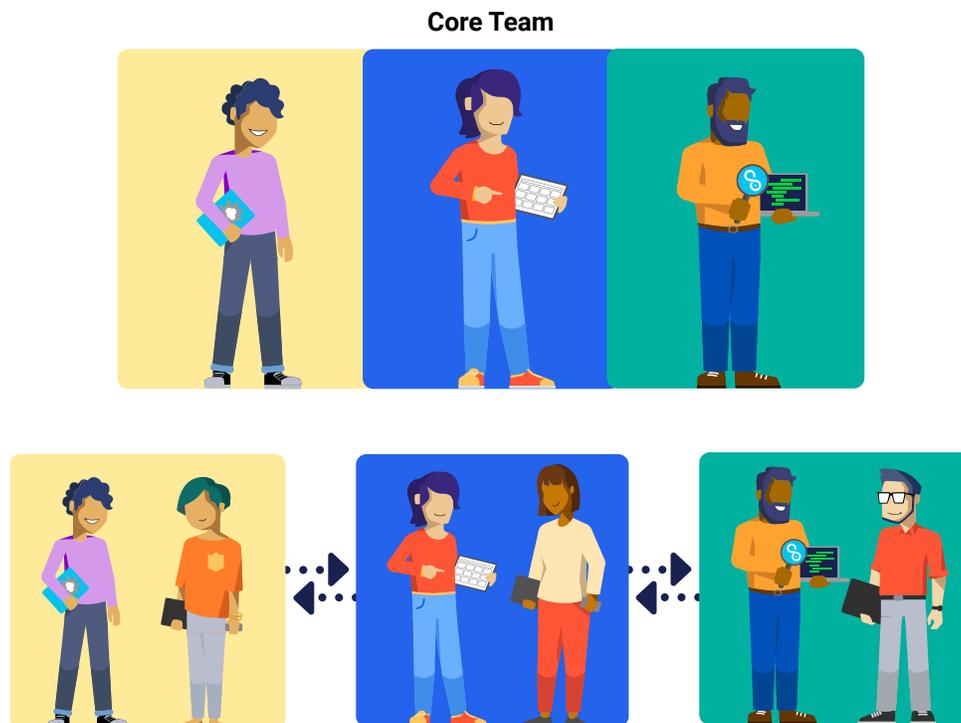


Driving Resilience via Cross-Functional Collaboration

No more silos between operations, QA and development – 70% of digitally maturing organizations have adopted cross-functional teams (Deloitte). **DevOps disrupts traditional models of team management and hierarchy.** Cross-functional teams mix and match roles so they can meet project needs with greater agility.



Hierarchical Team Model



Resilient DevOps Teams

Attributes and Advantages of Resilient DevOps Teams

Dreaming of Decentralization

DevOps teams require true decentralization to function. If your data or your process has to pass through a single role or department, bottlenecks rear their ugly heads and block improvements.

Agility = Stability

DevOps is designed to drive greater stability through better agility. When you can roll back changes in a nanosecond, you can limit the blast radius of disruptive commits. And when you can pivot to changes, rise to challenges and shore up gaps, you unlock faster releases and smarter iteration.

Have you ever wondered how a cheetah is able to dash 80 miles per hour without losing its balance — or its prey? Cheetahs have an uncanny ability to pivot without slowing down. Likewise, agile organizations and resilient teams don't need to worry about a single misstep. Instead, they can focus all their energy on acceleration and growth.

Data-Driven Testing

The most innovative companies in the world use automation to test thousands of potential failure states and generate more resilient environments. Data-driven testing finds faults before and after a commit is released. And automated testing makes it possible to thoroughly test even the smallest of commits and discover defects in a flash.

Automated Failure Resolution

“Fool me once, shame on you. Fool me twice, shame on me.” DevOps is all about the continuous cycle of improvement — and automated failure resolution makes it easy to learn from hiccups so you can recognize and heal them in the future.

DevOps teams must be in total sync to collect incident data and create automated, self-healing solutions. As data flows across the CI/CD process, your team becomes equipped and empowered to proactively tackle failure states.



Future-Proof Compliance

Resiliency doesn't just improve stability – it also heightens compliance. CI/CD provides greater levels of control over your platform and process so you can keep up with ever-changing regulations without sacrificing agility.

Resilient teams integrate and automate compliance scans and quality gates to create a system that is both fully compliant and easy to audit. The ultimate goal: To prevent data leaks, application crashes and other errors. Validating functionality and performance in any new or updated features is a great way to catch these potential defects before real clients run into them.

Back Up and Roll Back

A safety net of backups and rollback gives resilient DevOps teams the peace of mind they need to innovate. Rollback solutions deliver support for fast, iterative commits and aggressive changes without system-wide disruption.

Leverage Deep Insights and Trends for Data-Driven Decisions

Managing the end-to-end development cycle can be as disorienting as looking into a kaleidoscope. Things are often fragmented and tricky to analyze. Every team is potentially developing in a different cloud environment, which makes it a challenge to integrate tools and measure performance. This adds up to hours spent on reporting and analysis – and less time for strategic value-added work. **Value stream management** is a powerful way to standardize, simplify and speed up the flow of data to drive more informed decision making.

Experimentation Is Everything

The highest performing organizations systematically and continuously experiment at every level. They test hundreds of hypotheses and validate or invalidate them based on the outcomes. Resilient teams are constantly phasing out tests and creating new ones to perfect the QA process. They leverage self-healing AI to future-proof test scripts and intentionally break things in production to probe vulnerabilities and gauge how systems handle the added stress.



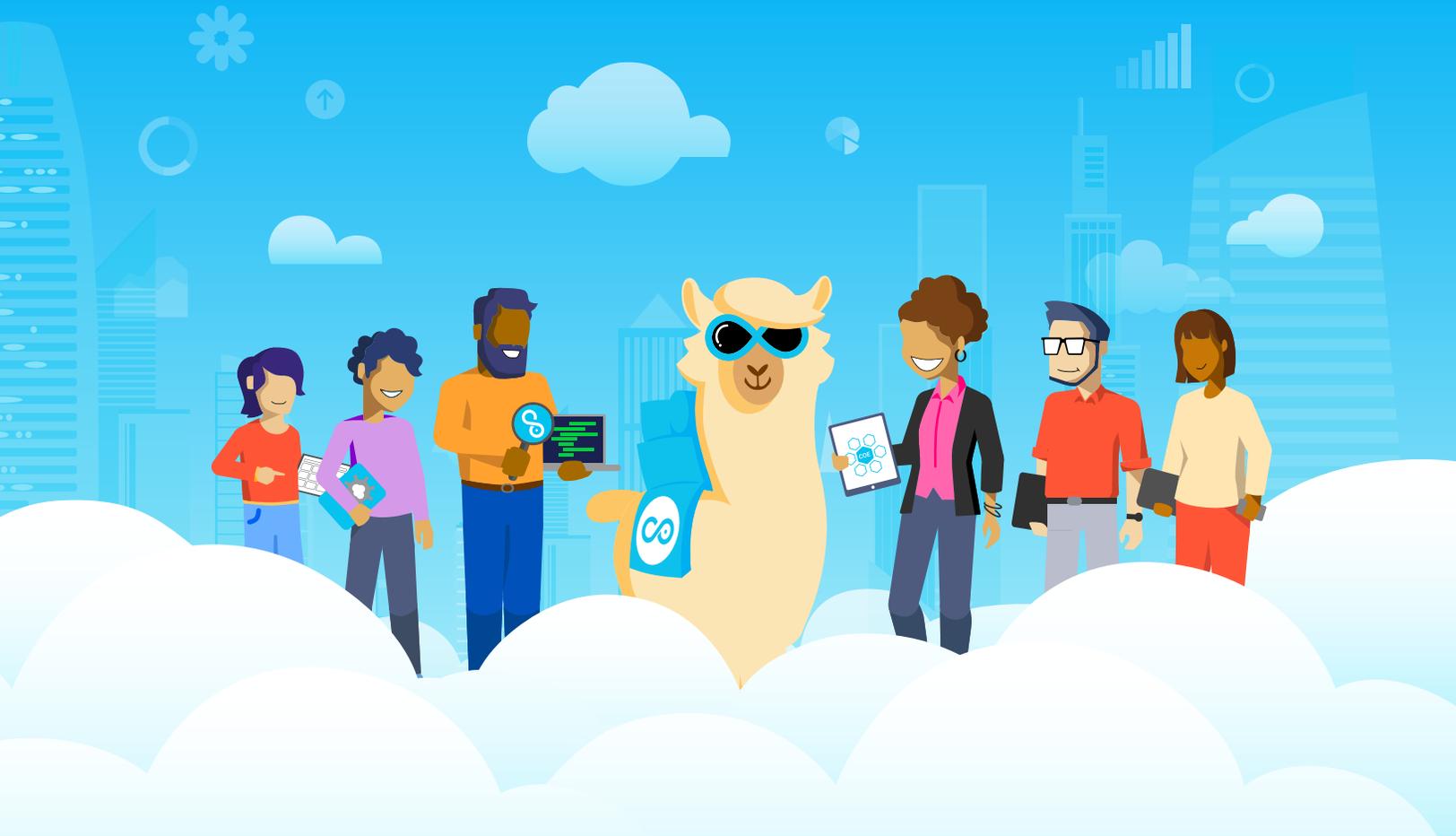
Now It's Your Turn

The Path to Digital Disruption Awaits

We've introduced you to the Five Steps to DevOps Success: Visibility, Quality, Speed, Innovation and Resilience. Together, these key ingredients give development teams everything they need to bring their digital dreams to life. At Copado, everything we do is centered around this step-by-step DevOps framework. Across the globe, our teams enable these capabilities for teams and accelerate their DevOps maturity.

Now it's your turn. How can you apply these principles to your team and your work?





About Copado

Copado is the #1 DevOps Platform for Salesforce and the Cloud. We harness the power of native CI/CD and Robotic Testing to drive digital transformation for 1,000+ of the most innovative brands on the planet – from Coca-Cola to eBay to Volkswagen.

Our low-code platform unites non-technical admins and pro-code developers on the same system and empowers enterprises to scale end-to-end software delivery across multi-cloud environments. Backed by Insight Partners, Salesforce Ventures and SoftBank Vision Fund, Copado has been proven to help digital enterprises unlock 20X faster releases, 94% fewer production bugs and 46% higher Salesforce ROI.

