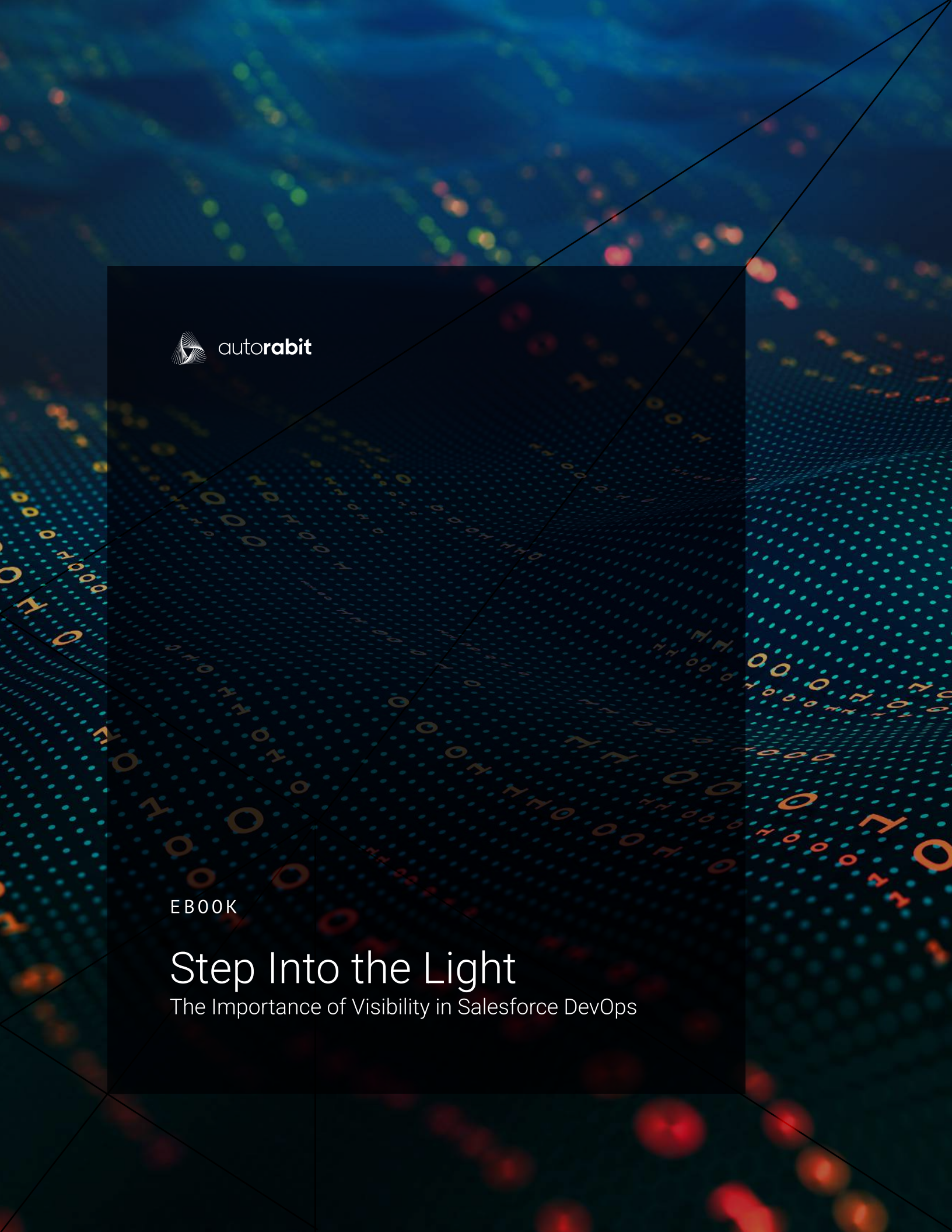



EBOOK

# Step Into the Light

## The Importance of Visibility in Salesforce DevOps

## INTRODUCTION

Confusion leads to data security risks, negative impacts on productivity, and frustrated team members. DevOps processes can get pretty granular and if you don't have systems and tools in place to clear up the complicated steps, errors and vulnerabilities are sure to follow.

Failing to preserve comprehensive visibility can lead to the following issues:

- Increased risk of error-filled applications and updates
- Difficult collaboration between team members
- Deployment failures.
- Security and compliance risks
- Inability to make data-driven decisions

Managing teams across various departments and numerous tools quickly becomes cumbersome. Overlapping responsibilities can lead to tasks becoming lost in the shuffle. The complicated nature of Salesforce DevOps makes it difficult to sort these issues out, but intentional choices and proper tool sourcing can clear things up. The first step to making improvements is to understand the various aspects that touch visibility in a DevOps pipeline.

# Let’s look into 7 critical considerations relating to maintaining visibility in Salesforce DevOps:

01

# The Relay Race

## THE RELAY RACE

Moving a project from planning to production involves the work of a series of teams. Think of the application or update as the baton being passed from one stage to the next. This relay race means that once a team finishes their aspect of the job, they are out of the loop on how it's progressing. Likewise, teams further down the pipeline aren't updated to the particular circumstances surround a project before it drops in their laps.

This segmentation leads a breakdown in communication and is a prime scenario for unnoticed errors. Team members spend unnecessary time getting themselves up to speed on a new product and if an error is detected and it needs to be sent back, developers can become overwhelmed with ad-hoc tasks.

### The Impact of Increased Visibility

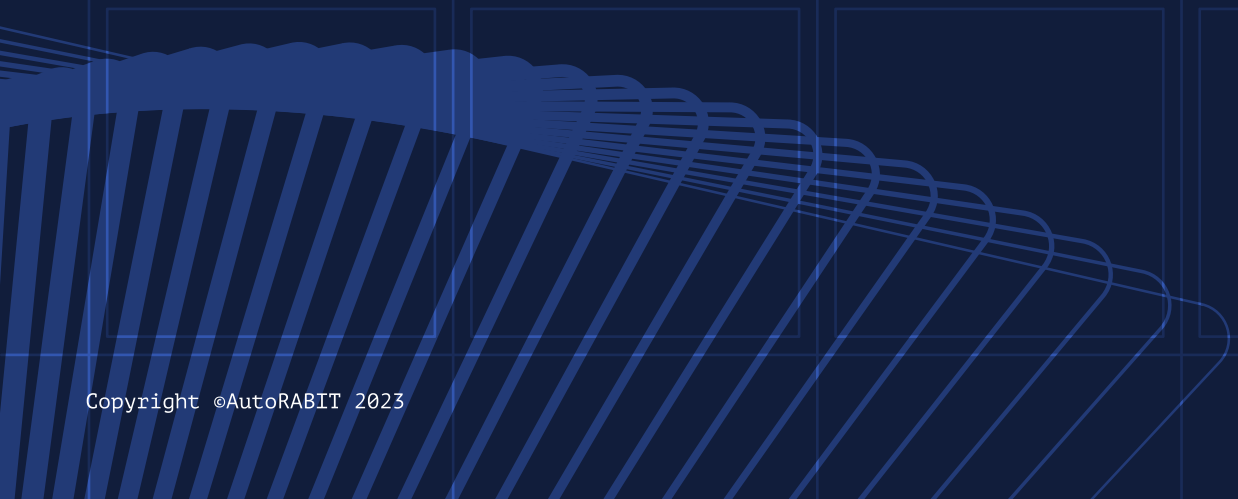
The stages of the DevOps lifecycle can be clearly defined—for example, planning, development, deployment, and production. But just because these stages can be described apart from each other doesn't mean they need to be separate in their approach.

Facilitating communication and visibility between these teams reduces confusion and ensures code flows smoothly from one stage to the next.

This can be accomplished with guidance from administrators for team members to communicate freely across departments. Utilizing a visualization tool that brings all of these considerations into a single interface will also make it much easier for team members to keep track of all the moving parts within an application or update.

02

# Wasted Time



## WASTED TIME

Amplifying productivity is a goal for every business. Reducing the time it takes to produce a quality product increases ROI along with the time it takes to produce a DevOps project. But when team members get bogged down with unnecessary questions and redundant tasks, they find themselves talking more about work than actually working.

Unclear expectations of team members and an inability to maintain updated views on current projects for administrators makes it more difficult to complete projects in a timely manner. The resulting confusion leaves team members waiting for answers when they could be completing their work. Eliminating this wasted time increases release velocity while also improving job satisfaction for team members.

### The Impact of Increased Visibility

Let your workers work. Managing every stage of a DevOps pipeline can seem daunting, but automated tools are available to help admins streamline their oversight. Ensuring teams further down the pipeline have the information they need to hop right into their tasks reduces wasted time, but that can only be accomplished if administrators maintain a clear view of what's happening.

How do errors impact other stages of the development pipeline? Are there any manual processes that can be automated? Is the flow of code clearly defined?

Heightened visibility over the entire application lifecycle makes it possible for Salesforce administrators to quickly answer questions like these—and this information helps them reduce wasted time.

03

# Recognizing Errors

## RECOGNIZING ERRORS

You can't find an error if you aren't looking for them. This is why multiple layers of testing are a Salesforce DevOps best practice. But sometimes, we can become so focused on pushing a project through production that we miss errors which become bugs in a live environment.

**This creates a series of issues:**

1. Negative impacts on functionality
2. Potential data security vulnerabilities
3. Loss of consumer trust
4. Strain on support team
5. Regulatory issues

These bugs become drastically more expensive to fix compared to those found in the code writing stage. An inability to spot these issues is an ongoing problem that can cost an organization a lot of time and money.

### The Impact of Increased Visibility

It's critical for DevOps teams for fix errors before the project moves farther down the pipeline. And the earlier you understand an error, the quicker you can fix it. Visibility into your project can be taken from a macro or micro level. Static code analysis is an essential aspect of maintaining visibility into the health of your code from the moment it's written.

This automated DevOps tool provides developers immediate alerts when a coding error is detected. This enables them to fix the issue without needing to go back later and redo any contingencies that were built off the error.

04

# Improving Feedback

## IMPROVING FEEDBACK

A cycle of feedback is important for creating strong DevOps projects. Without it, developers don't have the information they need to improve upon current approaches to writing code, and administrators can't refine DevOps processes between projects.

Any lag time between noticing errors and communicating the problem increases development time. And when your DevOps pipeline is segmented between each team and stage, this type of information doesn't flow to where it needs to go.

Ideally, feedback continually flows back to each stakeholder as code moves from development to production. And these flows feed into each other, from the entire release, all the way down to the individual keystroke.

### The Impact of Increased Visibility

Clear definitions of how this feedback is meant to flow between teams is essential. Work with team leaders to determine which type of situations need to be raised to surrounding teams. Reducing the delay in the transfer of information will have a ripple effect of productivity across the DevOps project.

But of course, your team must first be aware of these issues in order to communicate them. A data governance strategy will establish a hierarchy of data relationships and how it should be protected. The team in charge of maintaining this should be in constant contact, ensuring any issues are flagged and addressed.

05

# Continuous Learning

## CONTINUOUS LEARNING

Every business hopes to grow and expand their value over time. The only way this can happen is if the team members are able to grow and learn alongside the organization. And this is only possible if they can analyze their approach, learn from mistakes, and continuously refine their skills.

A lack of oversight robs team members of the opportunity to grow in their roles. How can someone learn from their mistakes if they don't know they did something wrong?

The ability to measure success and failures across DevOps projects provides actionable context for both the organization and individual team members. This context can be analyzed for how to streamline processes and avoid future mistakes.

### The Impact of Increased Visibility

Automated reports are available from a series of DevSecOps tools. Code health, deployment success rates, time to market—these types of metrics help teams find areas that are in need of improvement.

Seemingly simple mistakes have the capacity to spread into other aspects of a release pipeline and exacerbate the ramifications. It's important to first locate and then rectify these mistakes—especially if they continue to happen.

Analyzing dashboards and reports made available from automated release management tools helps team members better understand how their actions fit within the pipeline. This data is crucial to helping them refine processes over time, growing in their role while simultaneously benefitting the organization.

01

# Security Concerns

## SECURITY CONCERNS

A lack of visibility across departments—and within the coding process itself—will inevitably lead to data security concerns. Coding errors that aren't caught, unauthorized access, conflicting updates—all of these scenarios compromise the integrity of an organization's Salesforce platform. This can lead to hacks, accidental deletions, and exposure of sensitive information.

An inability to maintain a contemporary view of an IT system's stability can result in unnoticed security breaches. This allows cybercriminals to roam freely within the environment until the organization finally realizes the intrusion has occurred.

For example, the 2017 Equifax breach started in March and wasn't noticed until May. And during this time, the sensitive information of approximately 143 million people was exposed.

### The Impact of Increased Visibility

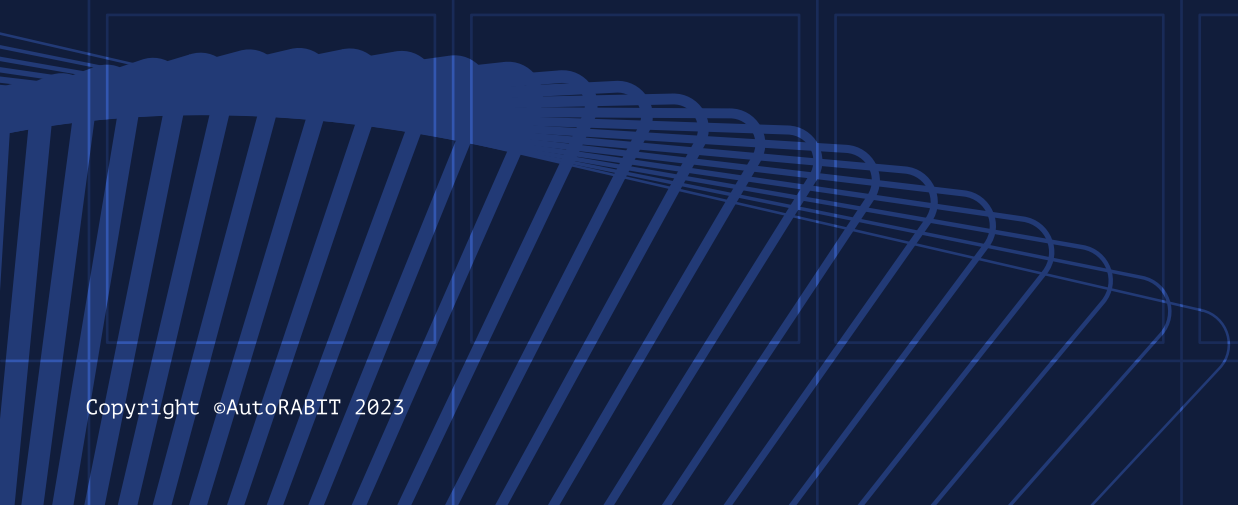
The same dashboards and reports that point to potential improvements in your release pipeline can also be used to identify a data breach. Continuous reviews of access logs, exported data, and a basic awareness of anything out of the ordinary by team members will help find a problem before it has a chance to get out of hand.

If cybercrime exists in the shadows, visibility and continuous oversight are the spotlights that find it. However, the available metrics aren't going to be useful if it's present in an overly complicated UI.

AutoRABIT's intuitive interface for CodeScan, ARM, and FlowCenter ensures even a casual user will be able to navigate the powerful functionality.

02

# Visualizing Flow



## VISUALIZING FLOW

As we've seen, the majority of Salesforce DevOps issues stem from an inability to provide easily understandable oversight. The release pipeline involves the work of a series of teams. Any breakdown in communication between these teams increases the chances of costly mistakes that can result in a problematic update or application.

DevOps pipelines can get messy. A team of administrators are often necessary to oversee a single project. This can lead to lines getting crossed and important information being lost in the shuffle. It might sound counterintuitive, but shrinking the team tasked with oversight can actually help increase visibility. All they need is the right tool.

### The Impact of Increased Visibility

AutoRABIT Flow Center helps teams visualize the release pipeline in a way that hasn't been possible until now. Moving a piece of code through the pipeline has been simplified by creating a visualization of the various stages, highlighting any failures or errors.

Stages of the release pipeline are defined. From there, users can drag and drop the various steps to build them out. All manual and automated activities can be defined in every stage.

This visualization simplifies the process of overseeing a DevOps project while increasing overall visibility of the mechanics involved along the way. The result is a constant stream of continuous feedback and updated states of successes and failures.

## CONCLUSION

Visibility is an intangible aspect of a Salesforce DevOps approach, but it's incredibly important. You can't fix a problem if you don't know it exists, and you can't learn from your mistakes if you don't recognize them.

Failing to maintain total visibility leaves organizations open to data security vulnerabilities, costly downtime, and massive amounts of redundant work.

Automated DevSecOps tools reduce the risks of manual error while streamlining processes. AutoRABIT's complete DevSecOps platform offers everything a team needs to maintain oversight from both macro and micro levels.

AutoRABIT Flow Center takes this consideration another step forward by simplifying the ways we monitor and manage the progress of an application or update as it moves through the DevOps lifecycle. Continuous improvement, data security support, increased productivity, and a reduction of errors are all possible when visibility is maintained with the help of intentional, automated tools.

## ABOUT AUTORABIT

AutoRABIT is a DevSecOps suite for SaaS platforms, which automates and accelerates the entire application development and release process. This enables continuous integration and delivery by providing fast, simple, and secure end-to-end automation across all Salesforce implementations. AutoRABIT tools help enterprises achieve higher release velocity and faster time to market.

AutoRABIT features static code analysis, automated metadata deployment, version controlling, advanced data loading, orgs and sandbox management, test automation, and reporting. Its services complement and extend Salesforce DX.

AutoRABIT Vault, our backup and recovery solution, streamlines Salesforce data, simplifies data backup challenges, offers disaster recovery, and provides endpoint data protection in the cloud.

CodeScan enables Salesforce developers and administrators with full visibility into code health from the first line written through final deployment into production, along with automated checks of Salesforce policies.

Flow Center helps Salesforce delivery teams to achieve their release velocity and DevSecOps maturity goals with flow automation, DevOps metrics, and visualization tools that can't be found anywhere else.

Record Migrator enables automatic bundling of all feature dependencies for Salesforce-managed packages. The deployment of templates ensures fast, efficient, and seamless releases.

Visit us at www.autorabit.com to learn more.

CERTIFIED
+ COMPLIANT

SOC 2 TYPE II CERTIFIED

CALIFORNIA CONSUMER PRIVACY ACT

GDPR

ISO 27001 INFORMATION SECURITY CERTIFIED

HIPAA COMPLIANT