



EBOOK

The Definitive Handbook to Quality Management in Salesforce DevOps

INTRODUCTION

There are a lot of ways to assess the success of a Salesforce DevOps project. How long did it take? How much money was spent during the production cycle? What is the expected return on the investment? But there is one aspect of a DevOps project that has the greatest impact on an array of factors: **quality**.

A high-quality DevOps product will be one that functions as intended, doesn't contain any bugs or errors, and supports the larger data security strategy.

But even though quality is a major contributor to success, business considerations like ROI and speed are often prioritized. This can lead to a loss of productivity, security, and ultimately, money.

An application or update done right the first time reduces the amount of redundant work needed to get a product to market. Managing quality levels across your platform can seem cumbersome and difficult, but keeping in mind some important factors will give your team the tools they need to consistently achieve high-quality releases.

We'll explore these seven crucial considerations for quality management in Salesforce DevOps:

1. Principles of Quality Assurance in Salesforce DevOps	004
2. The Importance of Test Automation	006
3. Continuous Integration + Continuous Deployment/Delivery (CI/CD)	008
4. Best Practices for Collaborative Workflows	010
5. Scalability of Approach	012
6. Processes Monitoring	014
7. Data Security + Compliance	016

01

Principles of Quality Assurance in Salesforce DevOps →

PRINCIPLES OF QUALITY ASSURANCE IN SALESFORCE DEVOPS

Quality assurance refers to the practice of ensuring the reliability, functionality, and performance of Salesforce applications throughout the DevOps lifecycle. Successful quality assurance strategies create high-quality products, efficient processes, higher productivity, and more satisfied customers.

Here are six key principles to guide your decision toward a more reliable DevOps approach:



Continuous Feedback: Begin testing as early as possible to receive prompt feedback on code changes, ensuring issues are found and fixed immediately.



Comprehensive Test Coverage: Aim for comprehensive test coverage across all aspects of Salesforce applications, including functionality, performance, security, and scalability.



Adherence to Best Practices: Follow industry best practices, such as maintaining a clean codebase, effectively utilizing version control systems, adhering to coding standards, and implementing robust deployment strategies.



Collaborative Approach: Foster collaboration between development, QA, and operations teams through clear communication channels to ensure a shared understanding of quality goals and strategies.



Security and Compliance: Integrate security checks and compliance measures into the QA process. Ensure that security practices are embedded within every stage of the development cycle to safeguard sensitive data and comply with regulatory requirements.



Continuous Monitoring: Implement monitoring systems post deployment to gather real-time data on application performance. Use this data to inform future decisions.

Adhering to these principles empowers DevOps teams to build high-quality, reliable, and scalable applications and updates while maintaining flexibility throughout the application lifecycle.

02

The Importance of Test Automation

THE IMPORTANCE OF TEST AUTOMATION

A contemporary approach to quality management and DevOps isn't complete without the utilization of test automation. Manual processes are time consuming and error-prone. Automating these crucial tasks drastically increases your team's capacity for productivity while also boosting quality.

Automation expedites the testing stage by executing test cases rapidly and consistently. This speed allows for faster feedback on code changes, enabling quicker releases and iterations. Functional, regression, integration, and performance testing can all be automated to ensure that critical aspects of the Salesforce application are thoroughly validated.

Tasking your team members with a manual approach to testing will monopolize their time and lead to poor results. Even the most talented developers will occasionally make mistakes. And if these mistakes aren't caught, the quality of the eventual product will degrade. Automated tests are precise and repeatable, eliminating the risks associate with manual testing. The resulting consistency contributes to higher accuracy in identifying defects and ensures reliable results.

While the initial setup and scripting for test automation require investment, automated tests save time and resources in the long run. They enable parallel testing and lower the cost of bug fixing by catching them early in the DevOps pipeline, when they are less costly to rectify.

Automated test scripts are scalable and reusable. Once created, they can be utilized for different environments and iterations, saving time and effort in testing across various configurations.

Regulatory compliance relies on strong processes and detailed reporting. Once an automated testing tool identifies a defect, it will provide a detailed log and information about the issue. This comprehensive reporting helps developers quickly pinpoint the source of an error so it can be addressed. And when a regulatory audit occurs, the reports provide proof of proper data handling.

03

Continuous Integration + Continuous Deployment/ Delivery (CI/CD) →

CONTINUOUS INTEGRATION + CONTINUOUS DEPLOYMENT/DELIVERY (CI/CD)

Properly managing quality requires streamlined processes in the DevOps pipeline. Integrating CI/CD tools is a great way to optimize processes, enhance collaboration, and ensure consistent and reliable delivery of high-quality applications.

It does this by utilizing various automated testing capabilities. CI/CD pipelines automate the integration of code changes into a shared repository, triggering automated tests. This ensures new code integrates smoothly with the existing codebase, allowing for early detection of any errors or conflicts that could compromise quality.

This advanced detection allows for faster feedback on code changes. Automated testing and deployment processes provide immediate feedback to developers, enabling them to address issues promptly, improving the quality of the codebase overall.

Discrepancies emerge when deployment environments are inconsistent. CD ensures consistent deployment environments across various stages to avoid unnecessary errors and ensure that the eventual product functions as it's intended.

Large releases make a great press release, but they are prone to errors that ultimately degrade the quality of the product. Incremental deployments enable more flexibility and simplified reviews. CI/CD tools enable teams to push small, manageable changes into production. This reduces the risk of introducing bugs and makes it easier to identify the cause of issues if they arise.

Quality assurance processes feed into each other. Each additional link in the security chain strengthens those around it. CI/CD pipelines seamlessly integrate with automated testing frameworks, static code analysis tools, and quality gates, ensuring that only high-quality code passes through the pipeline and gets deployed into a live environment.

04

Best Practices for Collaborative Workflows →

BEST PRACTICES FOR COLLABORATIVE WORKFLOWS

Optimized collaboration among the various teams in a DevOps pipeline is crucial to supporting quality management. Create cross-functional teams that share knowledge and responsibilities to ensure a holistic approach to quality management, including QA engineers, operations teams, and other stakeholders involved in the development process.

To achieve this, clear and open communication channels must be established. Utilize collaboration tools like Slack and Microsoft Teams to facilitate real-time communication, discussions, and updates.

Everyone must understand what is expected of them. Roles and responsibilities should be clearly defined to avoid confusion and instill confidence that team members understand their contribution to quality management practices. Daily stand-up meetings, sprint planning, and retrospectives promote an environment that values open communication.

Overwrites are a major issue for multi-developer teams. Utilizing a version control system while leveraging a static code analysis tool allows team members to collaborate on code changes, review each other's work, and verify code quality before integration.

Continuous feedback loops should be integrated into collaborative workflows. Establish processes where team members provide feedback on each other's work. This facilitates continuous improvement while giving them the information they need to promptly address any issues that come up.

Encourage the documentation of processes, best practices, and lessons learned. Foster a culture of knowledge sharing through wikis, documentation platforms, or internal knowledge bases to facilitate learning and alignment among team members.

A truly collaborative environment makes for a better workplace while also giving your team the information it needs to produce high-quality applications and updates.

05

Scalability of Approach



SCALABILITY OF APPROACH

Growth is a good thing for a business. However, the expansion of your team, your products, and your breadth of resources creates a challenge: How do you ensure your quality management strategy addresses all the new complications created by expansion?

Scalable quality management practices align with the organization's growth trajectory. They accommodate increased user bases, expanded functionalities, and evolving business requirements without compromising on quality standards.

As Salesforce applications grow in complexity with additional features and functionalities, a scalable approach allows for testing across various configurations and scenarios. This ensures that the quality is maintained regardless of its technical sophistication.

Performance and load testing are critical during this process. Ensuring the application performs optimally—even during peak usage—maintains quality standards and user satisfaction.

The infrastructure on which your system is built will also need to be addressed. Resource allocation must be flexible to support the expanded needs from your testing tools. With scalable infrastructure, teams can simulate diverse environments and perform tests in parallel, improving testing efficiency without compromising quality.

An expanding Salesforce DevOps approach incorporates numerous environments. Consistency must be maintained across these environments to reliably detect and address bugs and errors early in the application life cycle.

The process of scaling your quality management strategy generates valuable testing data and metrics that can be used in future decision-making processes. These insights enable teams to make data-driven choices for optimizing future quality management practices.

06

Processes Monitoring →

PROCESSES MONITORING

The continuous observation, measurement, and analysis of various aspects of Salesforce applications and the underlying infrastructure provide crucial insights for quality management. This includes real-time insights into the performance, reliability, and user experience of your applications.

Various tools exist to help your DevOps team keep track of the success of its DevOps efforts.

- Deployment monitoring tools provide detailed logs and status tracking during deployments. Users can monitor deployment statuses, track errors, and view deployment history to ensure successful releases.
- Automated testing allows users to create and execute test suites, view test results, and generate reports to track testing coverage and identify issues.
- Comparing and tracking changes in Salesforce metadata between different environments aids in monitoring metadata changes, ensuring consistency across environments and preventing unintentional modifications.
- Integrating version control systems allows teams to monitor changes in code and configuration, review changesets, and ensure version control best practices are followed.

Continuous monitoring enables the rapid detection of anomalies, errors, or failures in the application. This allows teams to promptly investigate issues, identify root causes, and take corrective actions, minimizing downtime and enhancing reliability.

Data on user interactions, user journeys, and user satisfaction metrics can also be gathered by monitoring tools. Understanding user behavior and experiences helps in identifying usability issues, ensuring that the application meets user expectations and maintains high quality.

Monitoring processes are integral to maintaining and enhancing the quality of Salesforce applications in DevOps by providing actionable insights, ensuring performance, security, compliance, and user satisfaction, all of which contribute to the continuous improvement of the application's quality.

07

Data Security + Compliance



DATA SECURITY + COMPLIANCE

Regulated industries need to always keep compliance with data security regulations in mind. Every interaction with your Salesforce data either supports or undermines your compliance strategy. In many ways, the goals of quality management tightly align with compliance needs.

Quality management aims to ensure that Salesforce applications handle sensitive data securely. Adhering to data security practices, such as encryption, access controls, and secure coding standards, ensures the protection of sensitive information. Maintaining data integrity and confidentiality contributes to the overall quality of the application.

Compliance needs should be factored into quality management practices to ensure that the application meets specific compliance standards, such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), SOC-2, or industry-specific regulations. Noncompliance can result in severe repercussions.

The monitoring tools mentioned above can also be used to evaluate data security measures and verify compliance controls within the Salesforce application. This includes regular audits with logging and tracking changes to ensure that security measures and compliance standards are consistently met.

Role-based access control principles ensure that individuals within the development and operations teams have appropriate access levels and permissions. This practice reduces the risk of unauthorized access or misuse of sensitive data, contributing to the overall security and quality of the application. A policy scanner can be used to automatically verify adherence to internal standards such as this.

Data security and compliance are integral components of quality management in Salesforce DevOps. Ensuring robust data security measures and compliance with regulations contribute significantly to the overall quality, reliability, trustworthiness, and success of your Salesforce DevOps approach.

CONCLUSION

Quality management is an ongoing process that requires a commitment to excellence, collaboration, and a willingness to adapt and improve over time. Like DevOps itself, it's important to be open to feedback. Communicating with your team members is the best way to ensure everyone is on the same page.

The quality of your applications and updates directly impacts your ability to maintain data security and compliance standards. Buggy applications, misfires, and outages are unacceptable, both from a business and a compliance point of view.

If there's one secret to successful quality management, it's visibility. Unseen errors have the potential to create incredibly costly, devastating results. But you can address this concern up front by requiring adherence to best practices, getting everything right the first time, and doing it consistently.

Automated DevOps tools are a crucial part of maintaining a successful quality management strategy. Combining these with intentional practices and comprehensive oversight ensures your Salesforce DevOps efforts will produce high quality products that are secure and reliable.

ABOUT AUTORABIT


AutoRABIT is a DevSecOps suite for SaaS platforms that 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 control, advanced data loading, orgs and sandbox management, test automation, and reporting. Its services complement and extend Salesforce DX.

AutoRABIT Vault is a backup and recovery solution that streamlines Salesforce data, simplifies data backup challenges, offers disaster recovery, and provides end-point data protection in the cloud.

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

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

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

CERTIFIED
+ COMPLIANT



CALIFORNIA
CONSUMER PRIVACY ACT

