IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Business and Delivery Challenges in Salesforce Development and Deployment without DevOps, Solutions to Adopt for DevOps Framework

Rishitha Kokku
Senior Software Engineer
Optum Services INC

Abstract This paper explores the problems that arise within Salesforce development and deployment without opting for standardized DevOps structure. Salesforce is one of the leading platforms for CRM, promising to provide out of the box features and flexibility of custom development to its customers. Although, organizations often encounter challenges when there are no DevOps practices employed. Inconsistent environments, mismatch in data, manual deployments, little to no control over releasing the code, quality gaps are some of the areas with complex difficulties. This will have a long-term business impact due to the compromised service provided to the clientele. This paper outlines such key challenges around code management, collaboration, testing, business processes and provides potential solutions that could improve project delivery and overall business efficiency.

Keywords DevOps for Salesforce, Salesforce Development, Salesforce Deployment, Continuous Integration and Delivery, Operational and Business inefficiencies, Salesforce Deployment best practices

1. Introduction

Many leading organizations today use Salesforce for its flexibility with customizations, integrations and implementation of complex business requirements. Salesforce offers a built-in development and deployment interface which enterprises tend to use to reduce costs but later discover the downside to it and realize the importance of practicing DevOps. Organizations that follow this traditional approach often face critical challenges that impact business operations [1], quality of the product or service, timely delivery and other technical areas. This research highlights such blockers and provides sustainable approaches to help businesses streamline their DevOps aiding Salesforce processes.

2. Business and Delivery challenges in Salesforce Development without DevOps

2.1. Inconsistent environments

Salesforce development practices involve usage of multiple development environments, a test and UAT environment and a Production environment [1]. Code, configuration and data are tightly coupled with each other to provide the expected output. In absence of DevOps, it is highly impossible to maintain the consistency with code, configuration and data across the environments. This results in significant differences in functional behavior of the application causing delay in releasing quality products to the customers.

2.2. Lack of collaboration between teams

To ensure successful delivery of a product or service, there needs to be continuous collaboration within the team [2,4]. Without DevOps such collaboration doesn't exist, and developers, testers and administrators work in silos

leading to communication gaps and timely responses. Following DevOps guidelines ensures that the code developers have developed is reviewed, maintained and deployed successfully across the environments.

2.3. Manual deployments

As Salesforce provides an in-built deployment interface developers choose to deploy their work manually from the dev instance to QA, UAT and even Production. This leads to a high number of inconsistencies and code overrides. One reason for this is when the changes are manually deployed there is often a chance of missing files [1], configuration changes and tracking of deployment steps as there is no track or history. This results in more dev time analyzing the differences between environments and increased number of deployments.

2.4. Delay in deployments, No automation

Without CICD pipelines and automated deployments, the deployments often take more time than usual especially when a high volume of files are included. Salesforce metadata is very complex [3] and has relationships within object hierarchy and multiple references. Without DevOps, tracking and maintaining such references and deploying them in order can be challenging. In large enterprises, circular dependencies are likely to be present and it is very tedious to deploy the changes to all the environments without running into deployment errors [1].

2.5. No Version Control History and Audits

When there is no version control there is no tracking of the files, the contributors and the changes made to these files. Manual tracking of the files can make it difficult to trace the changes [1]. This also impacts the audit trails when organizations are to comply with industry standards. If there is no history, there is no accountability when something is not working as expected.

2.6. Inconsistent Quality Assurance

Automated pipelines ensure that all test cases are covered when the code is deployed to the destination environment. Lack of this setup can contribute to insufficient testing, missed scenarios and human errors making a way for bugs into Production**.**

2.7. Lack of Continuous Integration

Since there is no CI and CD in place, the deployments that are manually taking place are repetitive in nature to all the environments including Production. Such deployments are highly unpredictable, override existing changes, are error and defect prone [1]. This also adds unnecessary deployment times due to its duplicate nature.

2.8. Delay in Marketing Time

Businesses have certain goals to launch enhancements and upgrades into the market to keep up with the competition [6]. Manual deployments that do not use any automation tools hinder in achieving the goals due to the time taking deployments resulting in missed opportunities and prevents them from being at the top of the game.

3. Solutions for Challenges in Salesforce Development and Deployment without DevOps

3.1. Introduce Version Control Systems

There are various version control systems in the market that each organization can choose from based upon their policies and budget. Some of such tools are Git, Bitbucket, Tortoise SVN that allow teams to track the changes, integrate with deployment tools, identify code conflicts, etc [1]. Developers can also maintain the entire history of the files making it easy for them to fall back on the changes that were made at a given point in time. Version control is one of the major pillars to achieve a successful DevOps implementation. With tools like Git Bitbucket, users can always roll back the changes to a previous version in case of application failures. Figure 1 below illustrates how multiple branches in version control can be pointed to different Salesforce environments using a deployment tool like Jenkins, aiding parallel development work and releases. The mapping of these branches to environments can be altered as per the business needs and release cycles.

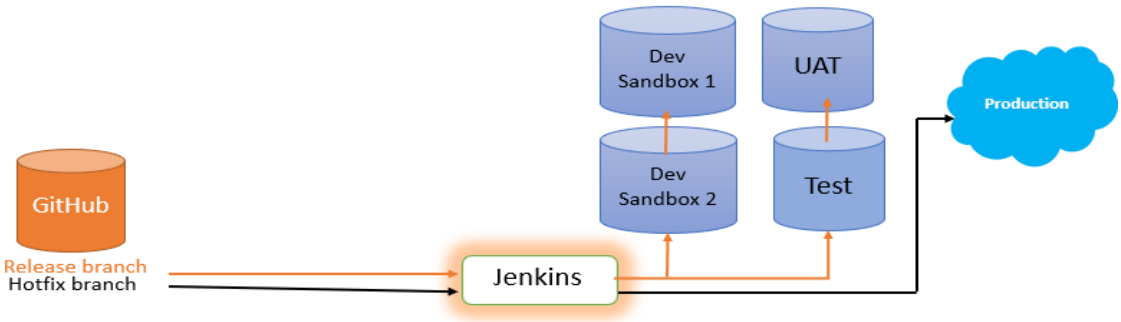

Figure 1. Version control integration with Salesforce environments

3.2. Implementing CICD and Automation of Testing

Teams can implement automated testing and CICD to catch issues in the early phase of deployments that can save time and effort. Tools like GitHub and Jenkins along with SFDX [3] can be integrated with the Salesforce environments to ensure fast and successful deployments, consistency across environments including production, quality assurance, reduced human efforts and errors and eliminate the duplicate work done by developers with manual deployments.

3.3. Environment management

As shown in figure 1, there can be multiple Salesforce environments that are in active use for various purposes. Keeping these environments in a synchronous state is essential for a smooth delivery and adhering to the timeline of the releases. These sandboxes should often be production like and to ensure that, refreshing these environments in regular intervals and deploying the source code across all the environments is crucial to identify any potential risks. This also helps in releasing the code from Development environment to Production in a seamless manner promising delivery on time.

3.4. Introducing Agile Methodology

One of the first steps to a successful DevOps model is to introduce Agile practices to develop the Salesforce application. The Agile model ensures there are regular release cycles, plans the team's capacity in each sprint, ensures that development and testing is done within the deadlines [2,4]. This helps the businesses to release in smaller intervals which is important to address any end user issues and to keep up with the market. Agile practices also ensure an easy management of code and deployments because of its incremental nature and the planning involved to release the features into Production benefiting the users.

3.5 Explore Deployment Tools

Organizations can always choose from various deployment tools that are available in the market that suit their requirements [1]. Implementation of full DevOps in a traditional manner using Jenkins is one way to achieve it as shown in Figure 1. Salesforce CLI, Salesforce DX and Visual Studio Code [3] can help in automating deployments to an extent. Teams can start with using CLI commands to get a hold of how automated deployments work and later proceed to building the CICD pipelines. Salesforce Developer Experience also known as SFDX offers an easy management of packaging and deploying the code into the destination environments. There are other tools such as Copado, AutoRABIT, Gearset that are pre-loaded with most of the configuration required for a successful DevOps implementation. Analyzing these tools and picking the one that is right for you improves the efficiency of the team. It is possible to gain a complete hold of CICD with the use of any tool mentioned above including the Static Code Analysis and Test class reports which are essential from a security standpoint.

3.6. Encourage Collaboration within the team

In addition to Agile and DevOps practices, collaboration and communication between Salesforce developers, testers and operations team ensures that all the changes are successfully deployed, tested and signed off by the business team. Break the habit of working in silos and encourage the team to communicate more efficiently to maintain transparency and help to achieve smooth deployments [2]. Open discussion often helps in identifying potential issues in an early stage, reducing the number of bugs released to production which is directly related to the quality provided and business performance [7].

3.7. Regular Deployment Intervals

When Agile setup is combined with DevOps framework, deployments are often in smaller packages and regular intervals due to the defined sprint cycles. This will ensure that the risk of downtime of the application is reduced [1], and thorough testing of the iteration can be done in the sandboxes before releasing to Production. Such deployments provide detailed release documentation making it easy for the users and business to understand what has been enhanced or improved with each release. Frequent upgrades to the application also lets the customers believe that a satisfied service is guaranteed [6].

4. Conclusion

Applications building on the Salesforce platform that aren't following DevOps practices experience significant challenges that directly affect the performance of the business, unsatisfied user experience, exceeding the delivery timelines, increased operational costs. The above-mentioned solutions like version control, automated testing, collaboration between the teams, implementation of CICD, following Agile practices can help organizations overcome inefficiencies associated with manual deployments. These methodologies have proven to be successful for Salesforce projects helping in faster and reliable deployments, thus improving the business outcomes.

References

1. Dive, P., & Gornalli, N. (2018). *DevOps for Salesforce: Build end-to-end Salesforce DevOps solutions to automate processes and effectively manage releases*. Packt Publishing Ltd. ISBN: 9781788477961.
2. Davis, A. (2019). *Mastering Salesforce DevOps: A comprehensive guide to building and managing a Salesforce DevOps pipeline*. Apress.
3. Salesforce.com, *Salesforce Metadata API Developer Guide*. [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.236.0.api_meta.meta/api_meta/meta_intro.htm Accessed: May 8, 2022.
4. Bankar, A. S., & Dhepe, P. P. (2018). *Trending agriculture by vision of DevOps & Salesforce*. Scholars Press. ISBN: 978-620-2-11563-0.
5. Salesforce.com, Deploying Apex. [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.238.0.apexcode.meta/apexcode/apex_deploying.htm. Accessed: June 8, 2022
6. Kim, G., Behr, K., Spafford, G. (2014). *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win* (3rd ed.). IT Revolution Press. IT Revolution Press. ISBN: 9781942788317.
7. Krief, M. (2018). *Learning DevOps: A beginner's guide to implementing DevOps in your organization*. Packt Publishing. ISBN**:** 9781788832698