Mphasis
The Next Applied

Salesforce DX with DevOps for Faster Development

by Manohar Kotapati, Module Lead in Mphasis

Contents

1.
Abstract

DevOps culture for software development has gained rapid momentum in SFDC development industry in the last few years. It involves adopting agile software development methodologies such as Continuous Deployment, Continuous Integration (CI) and Continuous Delivery (CD). This helps resolve issues quicker, get instant feedback on new products and features, improve the quality of software and ultimately save cost and gain market share.

The paper describes how Salesforce DX platform is well suited to use the Continuous Integration and Continuous Delivery methodology, to help organizations become agile.

2.
Introduction

In the current software development scenario, it requires effort and money to develop the applications faster and just focus on the business requirements rather than the development process. Also, as we work in teams, it is important that we do not fiddle with any changes made by other team members, and that the changes should be reverted if there are issues in the deployed code.
As most of the projects follow agile methodology, the DevOps methodologies such as Continuous Development, Continuous Integration and Continuous Delivery can be used for quick delivery of application in the most efficient way.

Here, we shall see how the DevOps methodology can be used with Salesforce DX.

3.
Problem Statement

Business today faces many challenges, right from rapidly changing business requirements to need for scaling up in short time. To overcome the lengthy software development lifecycle, many businesses are using the agile methodology. Agile methodology is the practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project.

Since agile involves short sprints of development, it is important for the team to work efficiently, without fiddling with each other's code. Also, deployment using the traditional salesforce techniques such as Change Set is not recommended for large applications as they lack code repository and code revert mechanism.

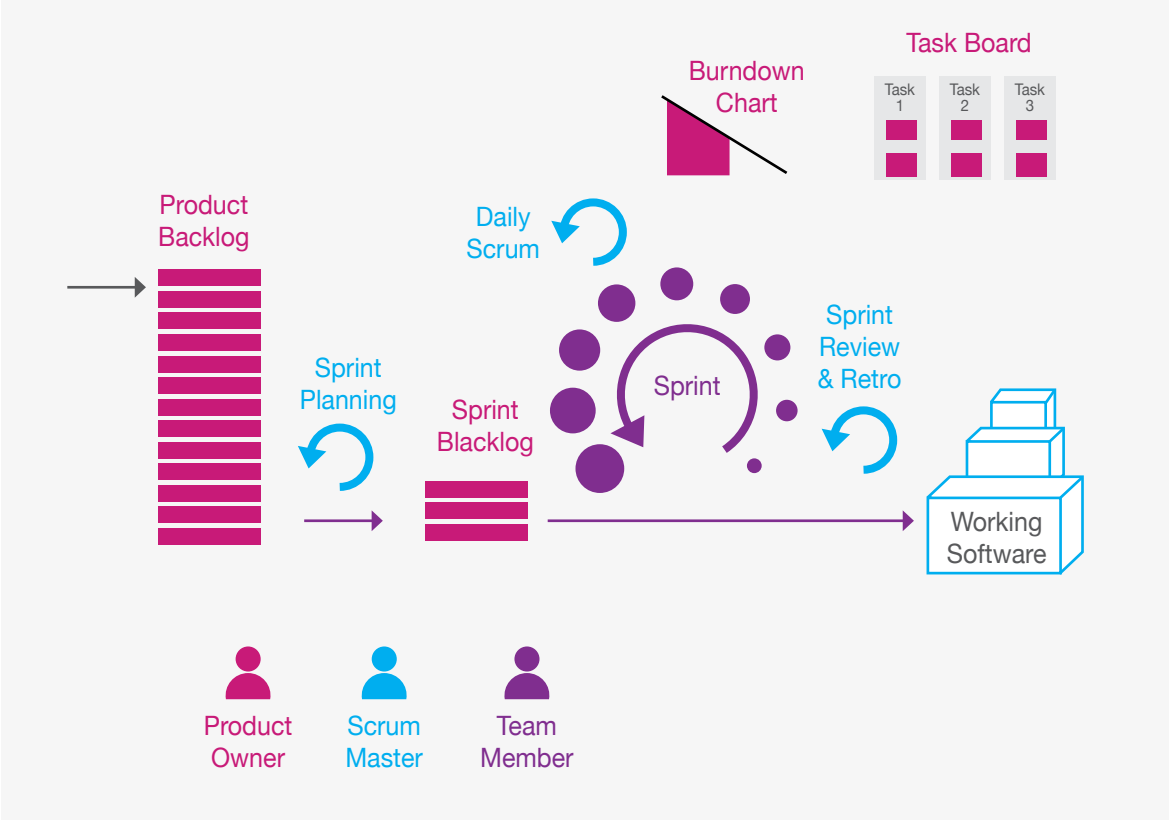

Fig. 1 Agile Diagram

This becomes even more important when the teams are spread across different geographies and locations. And that's when methodologies such as Salesforce DX (SFDX), Continuous Integration (CI) and Continuous Delivery (CD) come into the picture to help development teams achieve greater efficiency, enabling them to provide rapid software development changes while maintaining the system security and stability.

4.
How Salesforce DX Solves the Problem

Currently, updates and releases are developed with deployment scoped to the Production Org. The positive side of this approach is the team becomes familiar with the lifecycle and technologies involved, but on the other hand, it can be tough to coordinate within the team and version control isn't seamless. In addition, this model doesn't support continuous integration. It is also difficult to automate testing, to roll-back releases/codes, or to create separate version modules. This is where Salesforce DX can help. It is observed that many organizations have multiple sandboxes created for new development or for application testing. This is now a thing of the past as with Salesforce DX, now we can create scratch orgs. This transition is necessary to provide more flexibility to developers. It also adds versioning at the center of the workflow so that the state of an org can be brought back to any point in the past. Also creating the scratch orgs is simple and quick. Along with Salesforce DX, we can implement CI and CD using Jenkins, etc., to automate the check-in and deployment process.

Following are some of the key principles involved in CI and CD:

a. Automation of build and deployment activities
b. Automated testing
c. A single source code repository
d. Build validation

The key software and tools include the CI Server, the source code repository and the automation testing tool. If we apply the same principles and tools to Salesforce DX development, it would mean deploying to either a Sandbox or a production ORG. We can push both configurations as well as customization changes to different ORGs using CI.

Some of the major challenges that are faced at the time of development are:

a. Release cycles are too long due to manual deployments across multiple orgs
b. Deployments are prone to errors as changes are done manually
c. Since multiple developers work on the code, there is a risk of code being overwritten
d. There is no one source of truth for code repository
e. Reverting the code is an issue in the absence of repository
f. Multiple sandboxes need to be created and maintained which becomes an overhead in long-term

Introduction to Solution

Salesforce DX can have GiT/BitBucket as the version control, but for our purpose, we will discuss SFDX with BitBucket. Version control software keeps track of every modification that is made to the source code, giving the developers an option to track changes and help identify who made the change and its purpose. Also, it can be integrated with other tools to implement Continuous Integration (CI) and Continuous Deployment (CD), reducing the turnout time for application in the process. Using a VCS (version control software) and other tools such as Jenkins, we can automate the testing and deployment, reducing the manual intervention.

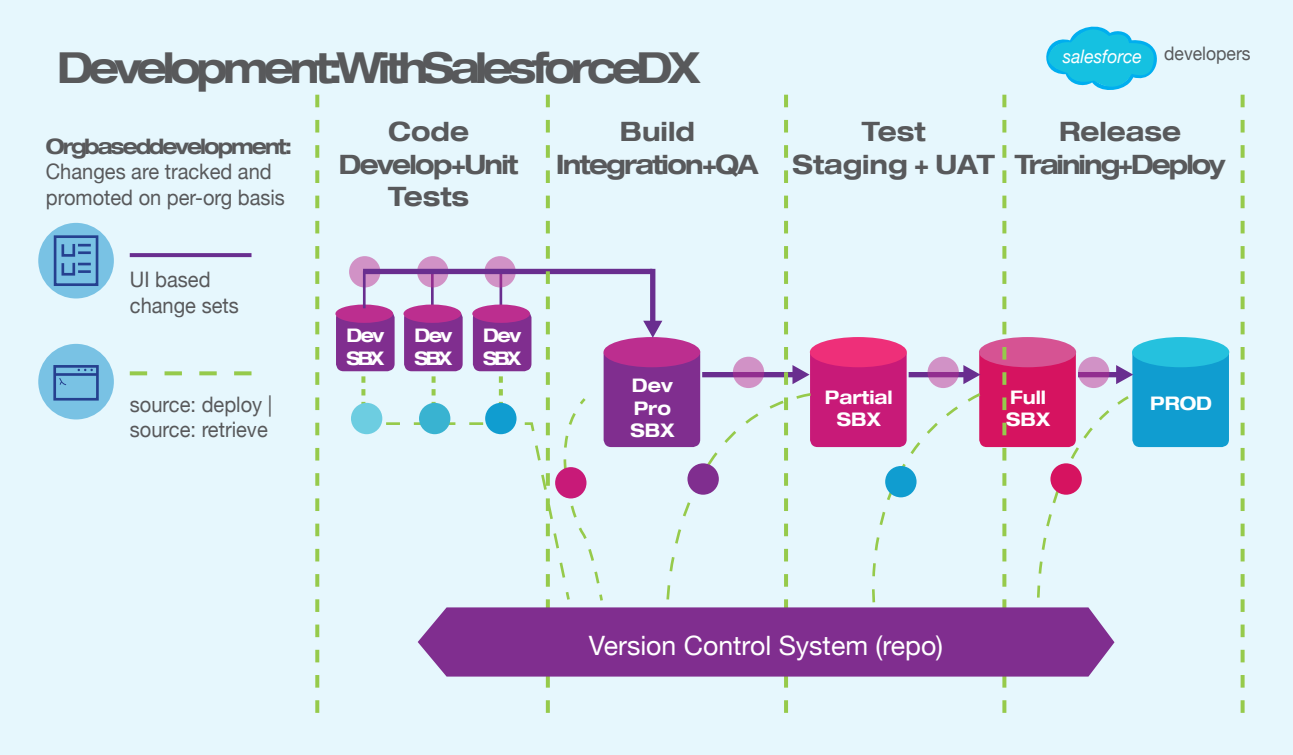

Fig. 2 Development with Salesforce DX

As shown in above figure, with Salesforce DX, multiple developers can work in their respective scratch orgs and commit the code to the repository at regular intervals. Here, repository acts as single source of truth and contains all the code changes. The code from the repository is then deployed to the other sandboxes for testing and UAT, where repository could be specific to the project or can be used for the series. The deployment to production happens from the repository in the final step and hence if there are any issues in the code, it's easy to revert to the old working code, without breaking the application.

Success Story

The client is leading an advertisement domain company involved in online and offline advertisement and Search Engine Optimization (SEO) business. Some of the challenges faced by them are:

1 Need to have efficient way for bi-weekly live deployment
2. Development teams spread across multiple locations, geographies and time-zones
3. No repository or version control
4. Need for a tracking mechanism to identify the source of any customization and configuration code change
5. Creating and maintaining sandbox is time-consuming and an overhead
6. Manual deployment to sandboxes should be automated to reduce the development cost
7. Need to avoid last minute deployment issues such as test class failure, code coverage issue and many more

Detailed Solution

Following are the tools needed to achieve the defined goal:

a. BitBucket repository: The repository to access the code on the internet
b. JIRA: Project tracking tool
c. BitBucket Pipelines (previously known as Atlassian Bamboo): A cloud and docker-based continuous build and deployment tool for BitBucket repository
d. Force.com Migration Tool: ANT-based build and migration tool for Salesforce
e. Eclipse IDE: IDE for code development

First, there is a need to setup the Salesforce code repository with multiple branches on BitBucket and then use that to setup the build and deployment scripts. These scripts are written to ensure the code is retrieved and deployed to any sandbox with minimum configurable parameters.
We can create a new scratch sandbox and code from that can be merged in the master branch and deployed to testing sandbox for testing. Once the testing is completed, the code can be merged to the UAT branch in BitBucket. Additionally, development teams can also configure JIRA to track the requirements/issues.

When deciding on the CI tools, there are several options that can be considered such as Jenkins, Hudson, etc. Some of the reasons for using BitBucket pipelines are:

a. It's completely cloud-based and does not require any server at our end
b. The configuration and setup is simple
c. They are from the same company and hence tightly coupled with BitBucket

Once the tools are decided, the pipelines can be setup, which would trigger the build and subsequently deployment script can automatically deploy and run test cases when any codes are checked into repository. This will help in early identification of possible code issues and last minute code coverage issue too. Internally, the BitBucket pipeline uses Docker Container to execute build and deployment steps. The pipelines are configured using a single configuration file named (bitbucker-pipelines.yml), which contains the steps to be executed when any code is checked into the repository. It is possible to have separate execution steps for separate branches. It also supports setting-up manual pipelines that can be triggered when needed. Once the build and deployment is completed, the automated status e-mail is sent to the specified members.

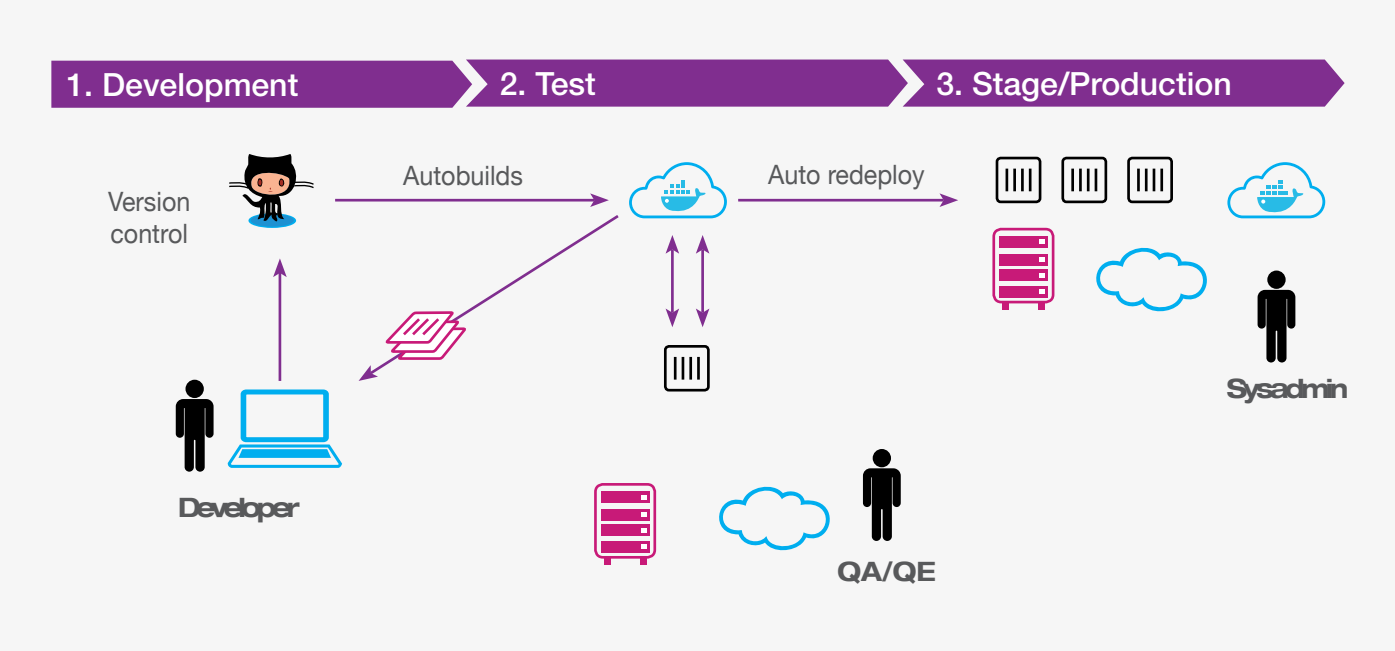

Fig. 3 BitBucket pipelines sample path

Business Benefits of the Solution

Following are the benefits that can be achieved by using the Salesforce DX along with Continuous Integration:

a. Simple and quick interaction with increased visibility enabling greater interaction

b. Issues identified early in development lifecycle and fixed before they turn into major problems

c. Having the code repository helps manage different versions and provides an option to deploy them if needed

d. Since debugging is simpler and faster, developers can utilize their time in developing new features

e. Create a repository of good quality code which follows the best practice and is reliable

f. Provides mechanism for automating the test class execution, which helps stakeholders verify if the functionality developed is working as expected

g. Deliver quality software more rapidly

Conclusion

In today's scenario, the applications are expected to go-live in a short span of time for which people are required to work in teams and collaborate with each other across geography and use modern development techniques.

Using Salesforce conventionally has no single repository and version control. These shortcomings are addressed with the Salesforce DX, which helps development teams to work together and easily integrate with other DevOps tools. Hence, issues such as code overwrite are rectified with improvement in automated testing and deployment. Using the version control system such as GiT, BitBucket, etc., maintain the code repository and help us track the changes.

The teams can move fully into agile-based development with shorter sprints and hence push business functionality more frequently to users. This can be enhanced in future to have automated testing done after every integration build as well as having a refined multi branch development rollouts.

Resources

https://developer.salesforce.com/docs/atlas.en-us.sfdx_dev.meta/sfdx_dev/sfdx_dev_ci_jenkins.htm

https://trailhead.salesforce.com/en/content/learn/trails/sfdx_get_started

https://developer.salesforce.com/platform/dx

Author

Manohar Kotapati

With more than 7 years of experience in Salesforce, Manohar is currently working as a Module Lead in Mphasis. He has worked on several salesforce projects around configuration and customization. He is an expert in Sales Cloud, Service Cloud and Marketing Cloud (Pardot). Manohar is certified in Salesforce Certified Administrator, Salesforce Platform Developer 1 (PD1) and Salesforce Certified Sales Cloud Consultant.

About Mphasis

Mphasis (BSE: 526299; NSE: MPHASIS) applies next-generation technology to help enterprises transform businesses globally. Customer centricity is foundational to Mphasis and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized ($C = X2C^{2}_{TM}= 1$) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization are key to building strong relationships with marquee clients. To know more, please visit www.mphasis.com

For more information, contact: marketinginfo@mphasis.com

USA
460 Park Avenue South
Suite #1101
New York, NY 10016, USA
Tel.: +1 212 686 6655

UK
88 Wood Street London
EC2V 7RS, UK
Tel.: +44 20 8528 1000

INDIA
Bagmane World Technology Center
Marathahalli Ring Road
Doddanakundhi Village
Mahadevapura
Bangalore 560 048, India
Tel.: +91 80 3352 5000

www.mphasis.com

VAS 07/05/19 US LETTER SIZE BASIL 5319