

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

CI-CD Automated Test Solution

Chaitanya Narayanam

M.Phil, MSC(IT)

Test Manager, TQA, Hitachi Consulting
Hyderabad – India

Abstract: This paper provides Test solution for those who are working in Agile Projects and Latest SDLC projects. This paper will provide a rid to all deployment issues in various environments. This is an automated test solution and very simple solution for integration of tools. This automation test solution will solve all issues and problems of manually code deploying project teams Automation test solution will help to deploy the code very ease and smoother transition to various environments Viz., Development, Testing, UAT and Production. CI- CD will be used to work in big and multiple teams are working at same time. The Code Merge will happen easily, daily builds will happen smoothly, code health checks will happen whenever code is getting merged in branch or trunk.

Keywords: Continuous integration, continuous Deliver, Code Merge, Daily builds, Code health check on Code check-in, Automated builds and Deployments, CI-CD with Test automation Solution.

I. INTRODUCTION AND HISTORY OF BUILD AND DEPLOYMENTS

SDLC projects are directive projects. Standard SDLC projects have phases for a release and for every phase there will be a build and deployment will be happened. Development phase will complete the Code developing. Then that code will be build and deployed in Test environments, after signoff the Testing then code will have new fixes for bugs which are raised in Test environment. Now this code will build again and deployed in Pre- production / Production environments.

Old SDLC	New SDLC
Directive	Adaptive
Task-oriented	Goal-directed
Specialized roles	Empowered teams
Resistant to change	Optimally responsive
Outsourced	Automated
Project optimization	Portfolio optimization

A decade back a new development framework has introduced called Agile. Many people are talking about agile today. Decreasing costs, increasing speed and quality, and the urge to empower employees to contribute their full work force to the growth and success of an organization. Scrum is a most popular in agile framework. Scrum Consists of Sprint and multiple teams will work in a sprint. Once Sprint is completed, Demo is over and Definition of done is met for user stories. Then every team will integrate their codes and make build for deploying in Stage/ Test environment. A configuration management tool will be used to integrate the codes. Here the actual problems arise. These problem fixing will take a lot of time and lead to slip the releases. Build issues and Deployment issues are very common for these teams.

Testing in Agile	Testing in DevOps
Test as early and as often as possible	Test continuously
Automate testing as much as possible	Automate almost everything
Continuous integration and testing is a step forward	Continuous integration and testing is mandatory
Potentially shippable code at the end of a sprint	Potentially shippable code following every integration

Manually performing code Integration, preparing a build and deploying code in environments need monitoring and high attention. In case Issues are occurred then it will take much time to fix all those issues. So, these projects teams are looking for an automated solution which will show all issues while code integration time itself. All these issues will be monitored easily with set of CI CD tools. CI CD automation solution will provide made ease of code integration, simple build making process and monitoring and do the deployments in various required environments. CICD automation solution is helpful to make daily builds in production as well.

II. CI-CD AUTOMATION TEST SOLUTION

CI-CD automation solution high level architecture is as per following figure I. Jenkins will play a vital role for gathering the code from repository viz Git.. Then the code will build with Building tools viz. Maven or Gradle. Once build is successful then code will be deployed in Test environment. An Automation Tests prepared by Selenium test will be triggered and performed smoke test on application and test results will be published. On successful tests, code will be deployed in Production. Again new code will be integrated, build will happen later code will be pushed to Test environment; tests will be conducted and so on.

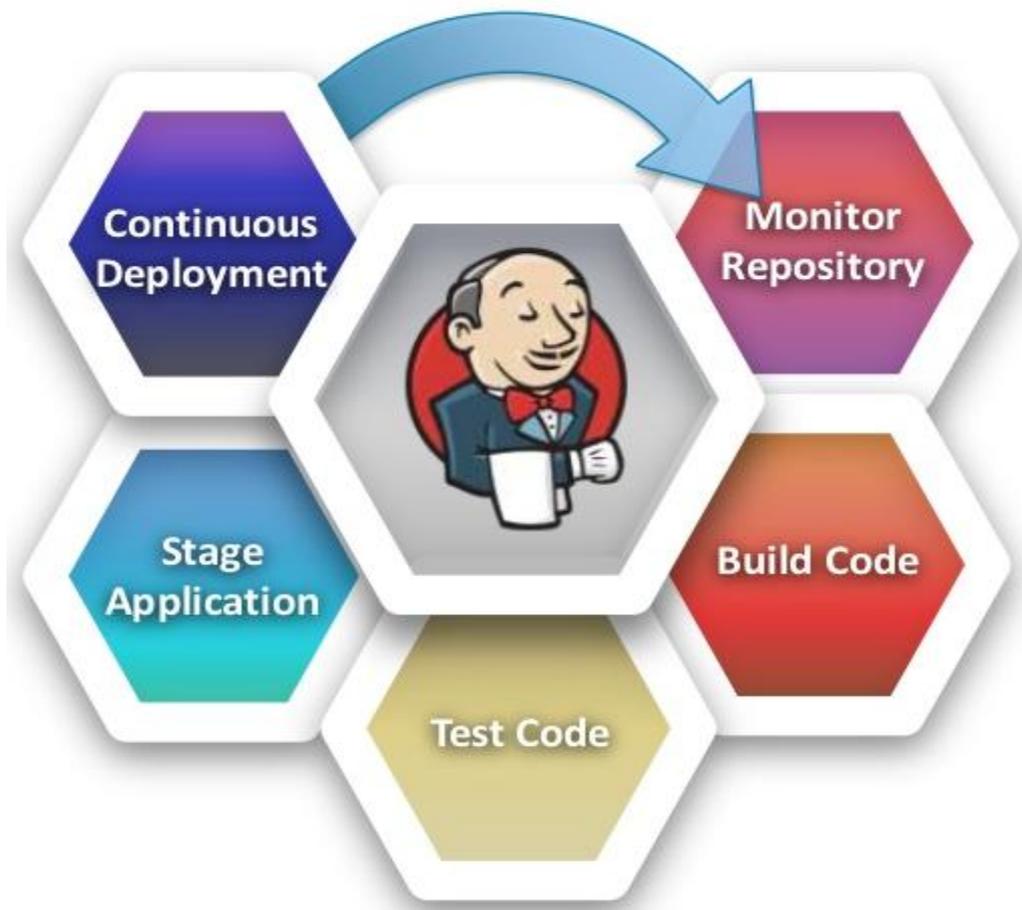


Figure I

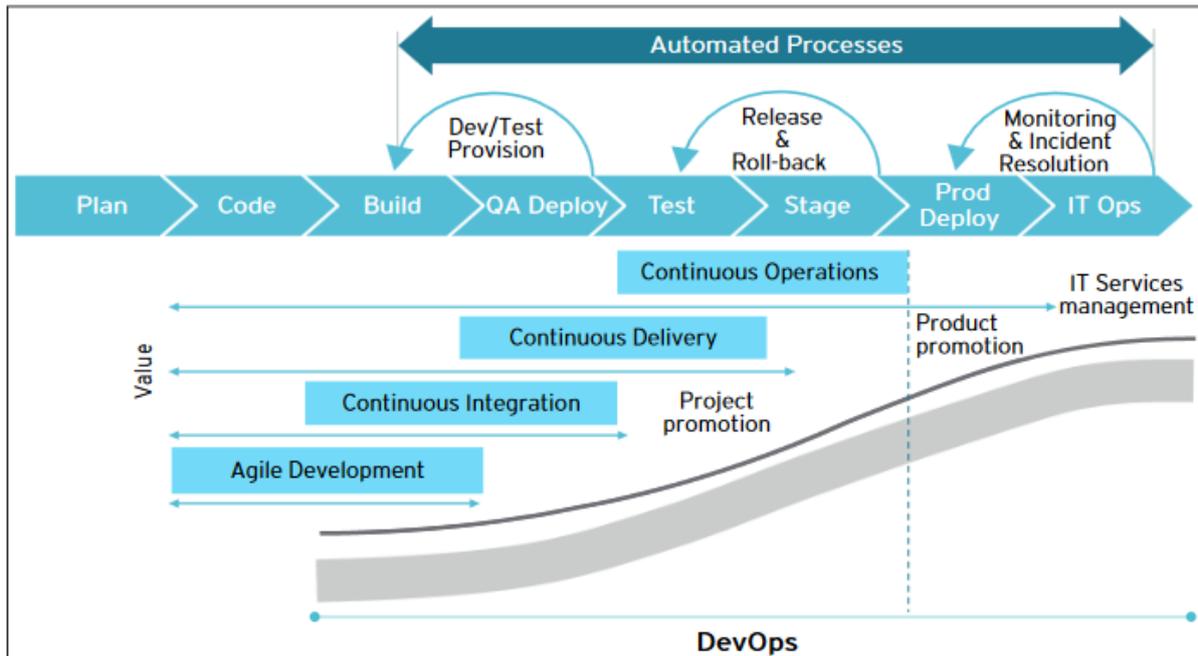


Figure II

Agile development will happen as shown in figure II. CI/CD automation solution will be the automation process in the above diagram. Each and every sprint team code will be integrated together from GIT by Jenkins. Build will happen with the help of Maven, once build is successful then Jenkins will push the code in QA deployment then Jenkins will trigger the selenium scripts to conduct test on application. Build health check by Selenium Script will happen in every environment like Test, Stage and Prod Deploy. Jenkins will monitor all the way of build health check. IT Ops team will check the Prod deploy test results and provide signoff for deployment of Production.

Every check point has the quicker feedback mechanism which is present at multiple layers of Automation Test suite. Typical tests include: **Health check Process:** As part of this process checks all required services are up after deployment. Typically, health checks run for just a few minutes. **Smoke tests Process:** As part of this process checks most of the high prioritised critical sub set of automated tests ensures that key high prioritised sub set of system features are operational and no blocking defects occur. Smoke tests are typically executed in less than 15 minutes, and as the process matures, the response time should be further optimized. **Intelligent regression Process:** As part of this process checks Optimized set of prioritised set of regression tests for that release will be selected and executed. **Full regression Process:** this process will run all required Tests for complete regression testing suite for current release. The goal is to minimize feedback time by running a parallel execution of automated tests through multiple threads or machines.

The build deployment process has tests Viz. Health Check, Smoke Tests, Intelligent Tests, and Full Regression Tests. . out of these, in case any fail, the deployment process is halted, and everyone involved in delivering software – developers, testers and operations staff – is notified immediately, triggering corrective action. Shorter feedback loops enable the team to fail fast and recover quickly.

III. DEVOPS ANALYSIS

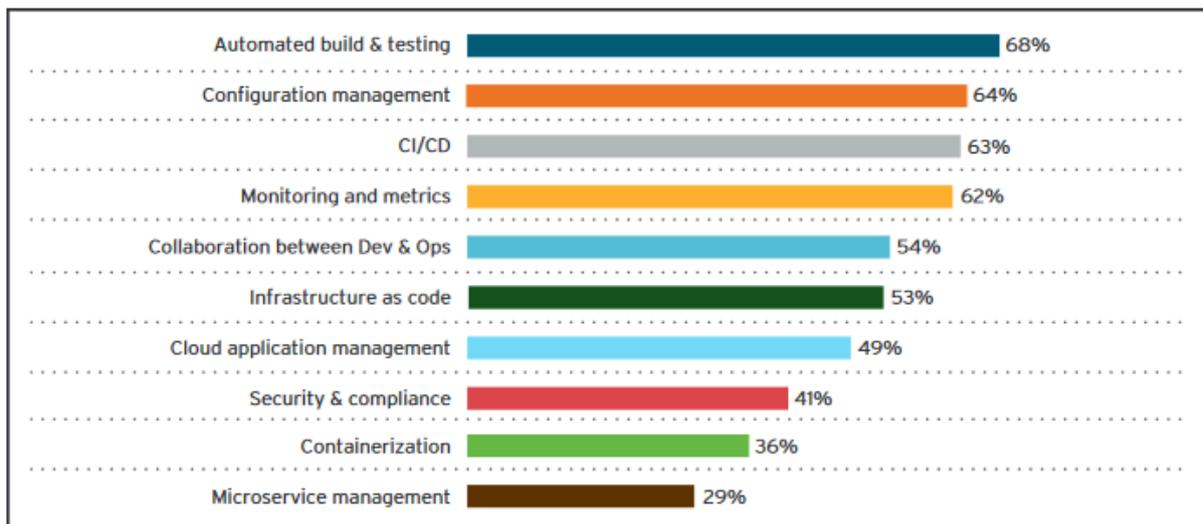
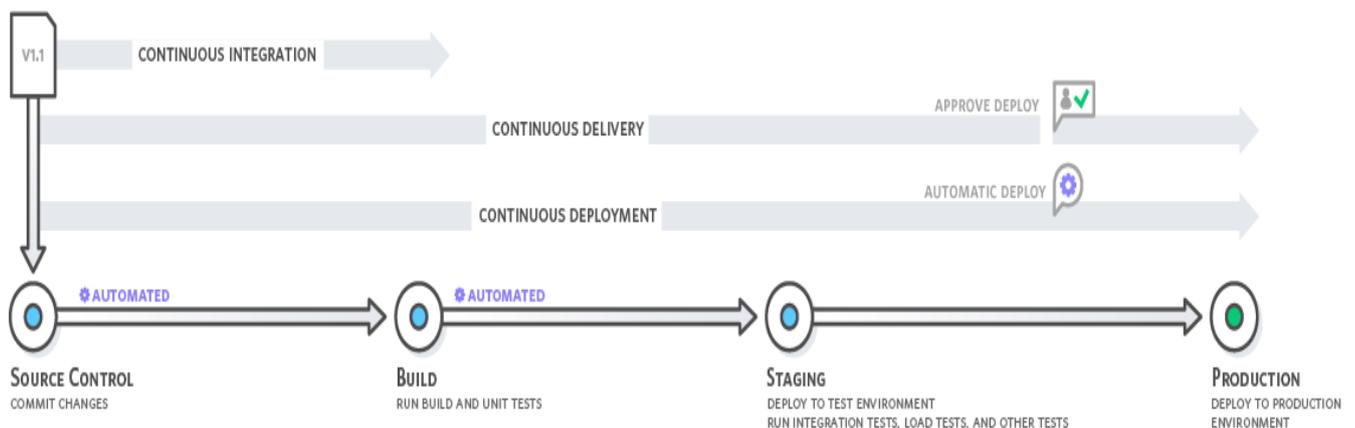


Figure III

The above diagram figure III explains about the DEVOPS at glance. According to a recent industry report published by online educational resources purveyor DevOps.com over 70% of IT organizations have embraced DevOps in some form. The rush to DevOps is on, and for some, the journey has just begun. Figure III is a result of 300 participants, including DevOps specialists, IT engineers and executives across a wide range of organizations.

Continuous integration refers to the build and unit testing stages of the software release process. Every revision that is committed triggers an automated build and test. With continuous delivery, code changes are automatically built, tested, and prepared for a release to production. Continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.



Continuous delivery automates the entire software release process. Every revision that is committed triggers an automated flow that builds, tests, and then stages the update. The final decision to deploy to a live production environment is triggered by the developer. With continuous deployment, revisions are deployed to a production environment automatically without explicit approval from a developer, making the entire software release process automated. Automation testing will happen after every CI and CD steps to validate whether it happened as expected or not.

References

- <https://en.wikipedia.org/wiki/DevOps>.
- <http://www.gartner.com/it-glossary/devops/>.
- <http://devops.com/the-real-value-of-devopsdownload/>
- 4 Kurt Bittner with Phil Murphy, John R. Rymer, Jeffrey S. Hammond, Tom Grant, Ph.D., Steven Kesler, "Continuous Delivery Is Reshaping the Future Of ALM," Forrester Research, July 2013, <https://www.forrester.com/Continuous+Delivery+Is+Reshaping+The+Future+Of+ALM/fulltext/-/Eres99321>.

5. Selenium is an open source suite of tools (<http://www.seleniumhq.org/>) to automate Web browsers across many platforms. Selenium runs in many browsers and operating systems and can be controlled by many programming languages and testing frameworks.
6. Gene Kim, Keving Behr, George Spafford, The Phoenix Project: A Novel about IT, DevOps and Helping Your Business Win, IT Revolution Press, January 2013, <http://itrevolution.com/books/phoenix-projectdevops-book/>.

A UTHOR(S) P R O F I L E



Chaitanya Narayanam, received the M.Phil. Degree in Computers from Bharathi Dasan University and M.Sc. degree in Information Technology from Alagappa University. Currently, he is working as a Manager in TQA department at Hitachi consulting. He is having overall 14 years of experience in IT.