Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 15 Issue 2 Version 1.0 Year 2015
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Changing Scenario of Testing Paradigms Using DevOps - A Comparative Study with Classical Models

By Surendra Naidu Mullaguru

JNTUK, India

Abstract- Many organizations come across themselves in a challenging conditions with the increase of web applications which depend on cloud. As cloud applications are supportive for rapid releases with respect to the queries requested by the user community. Most of the software organizations expects to release the product in time as required by the end user hence, it pressurize functional teams of the organization to do the needful. As perfection is rare to achieve , it is obvious to commit more and more defects and consequent annoyance to the team. This paper discusses about a new paradigm i.e DevOps which tries to address the identified problem by integrating Development and Operations teams. In this model, the development team would take care of operational requirements like deploying scripts, debugging and performance testing from the scratch and the operations team take care of well-informed support and feedback before, during, and after deployment.

GJCST-C Classification : D.2.5

Changing Scenario of Testing Paradigms Using DevOps – A Comparative Study with Classical Models

Surendra Naidu Mullaguru

Abstract- Many organizations come across themselves in a challenging conditions with the increase of web applications which depend on cloud. As cloud applications are supportive for rapid releases with respect to the queries requested by the user community. Most of the software organizations expects to release the product in time as required by the end user hence, it pressurize functional teams of the organization to do the needful. As perfection is rare to achieve , it is obvious to commit more and more defects and consequent annoyance to the team. This paper discusses about a new paradigm i.e DevOps which tries to address the identified problem by integrating Development and Operations teams. In this model, the development team would take care of operational requirements like deploying scripts, debugging and performance testing from the scratch and the operations team take care of well-informed support and feedback before, during, and after deployment.

I. Introduction

Most of the software development teams are moving towards a new trend of testing domain i.e DevOps. Organizations are given pressures to produce higher-quality code faster giving less time for Quality Assurance (QA). As it is emerging as prominent domain, the developers need to adjust themselves to gain more benefits. We cannot release the product quickly with timelines as persisting in compact and separated work teams. The DevOps promises to achieve this. With DevOps, team's participating skills are equally important as technical skills. This paper focus on how DevOps paradigm can change the testing set-up of conventional testing by integrating the development process and the QA with the operations functioning together to speed up development during problem resolution. The paper also focuses on the areas of application, its suitability wherein the conventional system fail to deliver testing inferences [7]. DevOps is a rising methodology from two popular methods of software development. i.e the agile system administration or agile operations and the later is the integration of development and operations teams in all the phases of the development lifecycle for creating and using a service. The definition of DevOps according to Jez Humble is "DevOps is a cross-disciplinary community of practice dedicated to the learn building, evolving and operating rapidly-changing resilient systems at scale". DevOps is an effort to enable operations engineers and development engineers together in total service lifecycle, from design through the development process to production phase and support. DevOps is also characterized by operations by team making use of many of the techniques as developers to carry out systems work in a routine pattern. Those techniques can range from using source control to testing to participate in an agile development process. In the industry the tag DevOps is referred as the relationship between development and operations. The reason for this is to get the advantage of both development team and operations team together for the growth of the business. The DevOps improves the organization potential for nonstop software release that can facilitate organizations to capture market opportunities. It also helps in evaluating customer feedback, cost, quality and risk. As DevOps analyzes the code systematically and assist in reducing the coding errors. According to the present market trend in the near future Devops is expected to emerge as a vital tool for rapid process develop. In the past development employees are considered as "makers" and the "Ops" employees are considered as "people who deal with the creation after its birth. In this way, DevOps can be implemented as a outcome of Agile Computing. Agile software development stipulates close association of customers, product management, developers, and QA to fill in the space and rapidly iterate towards an improved product. But the aim of DevOps is not only increasing the rate of change but also to deploy the features successfully into production without causing disorder and trouble to other services, while rapidly detecting and correcting incidents when they occur. DevOps is a simple extension to agile principles beyond the boundaries of "the code" to the entire delivered service. DevOps is especially complementary to the agile software development process, as it extends and completes the continuous integration and release process by ensuring the code production and provide value to the customer[6][1].

The testing tool must ensure to provide some basic provision to carry out testing facilities which are being considered by DevOps as a vital aid . Following are the some basic capabilities extend by Devops[4].

Author: Sharp Software Development India Pvt.Ltd, Test Lead, India
e-mail: surendranaidu0580@gmail.com

- Testing early pieces of code often.
- Integration of application development and test life cycles.
- Standardization of the testing methods for consistent results.
- Develop a broad test plan, as it creates the basis for the testing methodology.
- Extend dynamic approach and declare expected results.
- To manage business policies across the application.
- Usage of multiple levels and kinds of testing (systems testing, regression, module, integration, stress and load).
- Assessment and examining the tasks to lower costs and
- for plan for future.
- Easy interface for the clients to verify the extent of application development and feasibility.

II. Comparison of DevOps with Agile

The origin for the development of DevOps is emerging technologies. The DevOps is derived from the Agile System Administration movement and Enterprise Systems Management (ESM) movement. The characteristic of agile process is that it shows us how to create adaptive software development for quick feedback cycles through early customer participation. The primary characteristic of DevOps is that it contemplates on business results rather than technical details. DevOps acquires lot of lean principles like deploying in smaller batches aiming at enhancing the communication between developers and operations teams[3][2].

a) *Lean is the basis of Agile*

Lean signifies optimization from end-to-end process which produces value for your customer – from the original idea to generation of revenue. Lean philosophy focus on stream more than bottlenecks in the process and unproductive activities identification. Creating importance for your customers is all about judgment, validate product ideas, implementing these ideas and managing them to advantage the users. It is evident that many admirable schools of thought are there for structuring the processes such as Lean Startup and Running Lean to hash out product ideas. Scrum, and XP are used for creating features in an pliable manner. The DevOps enables the creation of a tradition of focusing on delivering worth for the customer instead of only concerning for technical issues[1][4].

b) *Lean Startup or Running Lean*

Eric Ries and Ash Maurya reviewed about a learning cycle during feature discovery. From authenticating ideas to testing probable solutions, they teach to resolve problems worth solving and keep your solutions more directed.

- *Scrum, XP and Kanban:* Typical agile processes show how to arrange adaptive software development for speedy feedback cycles through timely customer participation.
- *DevOps:* It focus on cultural aspects of business results relatively than technical details. DevOps integrate lot of lean principles, focusing on improvement in collaboration between developers and operations folks.

Lean principles are the foundation of entire product development course. In order to incorporate path of improvement, one must consider the whole as sub-optimizing element and just one part will hurt the business. The developer circles are emphasizing on DevOps and the entrepreneur circles are focusing at Lean Startup. The term, coined by Eric Ries, is a startup that achieves validated learning through the build-measure-learn loop. With a notion of learning what the customer actually need so that a useful product and a profitable business model would emerge[2][8].

It may not appear that the Lean Startup and DevOps are related. Small batches of tasks are the way for cross-functional teams, which work together on the smallest units of work possible while pushing that work to the customer as soon as possible. This way, the product gets to the customer swiftly, allowing the business model or product to be validated.

The common model presently is large batches. Instead of a small team working across the modules to push a product towards the customer rapidly. The team specializes, with a range of team members completing their segment of the product before passing it on to the next phase in the chain. If at any point a fault is found, the product has be sent back down to the chain. This leads to speedy feedback and rapid turnaround times, resulting in an improved product. DevOps is about accepting small batches, but the batch in this case is a thin part of the overall process, pushing a product from development to deployment to see it in use. DevOps is about allowing production behavior to quickly fed back to developers so that development can adjust swiftly to the situations in production. With the lean startup, the potential competence of an individual unit such as operations might emerge to weaken. The efficiency of the system as a whole speeds up. When developing a system it is simple to foresee how a system should run, but that is not enough. Often inevitable differences in hardware and traffic escort to differences in how the system actually performs with how it was predicted. DevOps is about tightening the same loop as the Lean Startup. The only difference is the narrower focus of DevOps. With the loop as rigid as possible, knowledge gained from the real system is fed back into the next cycle of the product so the system becomes better in small and tight increments. DevOps is a subset of the Learn Startup. Both are reactions to the today's fast changing world and require to produce a system that can optimize quickly, even if that means undercutting what appears to be the efficient in the entity unit.

c) Jenkins – A DevOps tool

Jenkins is a rapid Continuous Integration server. Continuous integration is a procedure in which all development work is integrated at as planned entity with a time constraint or event and the resulting work is routinely tested and built in automated environment as shown in figure 1. The idea is to development a scenario wherein errors are being identified at an early stage in the process. Jenkins is an open source tool to achieve continuous integration. The basic functionality of Jenkins is to perform a definite list of steps supported by a trigger. The trigger might be a change in version control system or a indication for time based trigger, e.g., a build every 15 minutes. To accomplish this the list of steps include:

- Perform a software build with Apache Maven or Gradle
- Run a shell script
- Archive the build results
- Start the integration tests

Basically Continuous Integration is the mechanism of running tests on a non-developer machine routinely each time someone pushes a fresh code into the source repository.

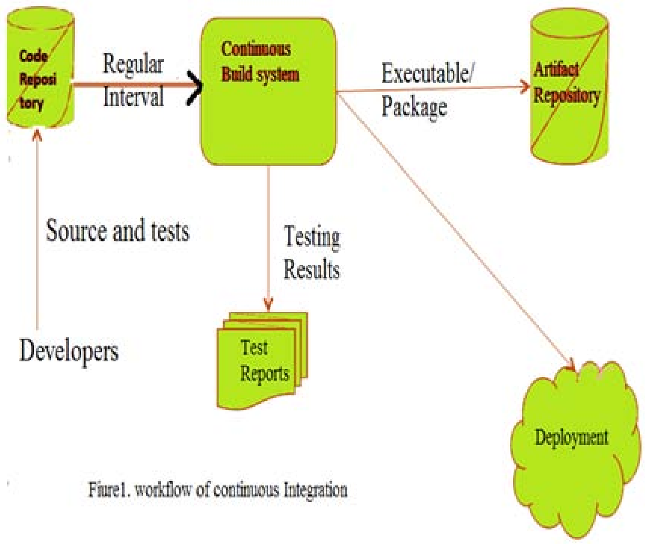
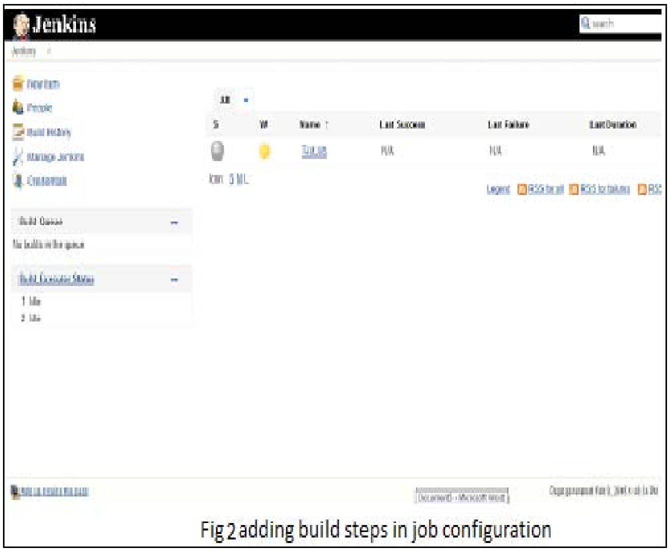


Figure 1 : Widow of continuous integration

This has remarkable advantage of knowing that all the tests work and attain fast feedback. The fast feedback is vital to know right after breaking the build and based on that one would repair or revert it for improvement. The developer must be alert when running the tests occasionally, as the problem that has emerged due to change in code may have surfaced since the last time. As it is hard to figure out which change has induced such problem. Built on continuous Integration is the platform of continuous deployment/ delivery wherein, after a successful test run the system instantly would release the latest version of the codebase. This makes the deployment a non-issue and helps in pace up development. Jenkins is a one of the DevOps application tool which supervises regularly executed jobs, such as building a software project. Among those things, presently Jenkins focuses on building/testing software projects continuously and supervise executions of externally-run jobs, such as cron jobs and procmail jobs, even though that are run on a remote machine[5].

The following section demonstrates the steps of a sample Java application testing using Jenkins tool. The implementation steps are as follows.
step1: Deploy Jenkins. war in web server
step2: Create a new job.
step3: Add build steps in job configuration.
step4: From the job home page click on "Build Now" to start the build process.
step5: Check the log for build status and then click1 on last run console output.

The following are the screenshots of a simple java application testing using jenkins tool. After deplyong the Jenkins. war in web server we create a new job. Then add build steps in job configuration. Once it is configured, we would get a resultant screen as figure 2.

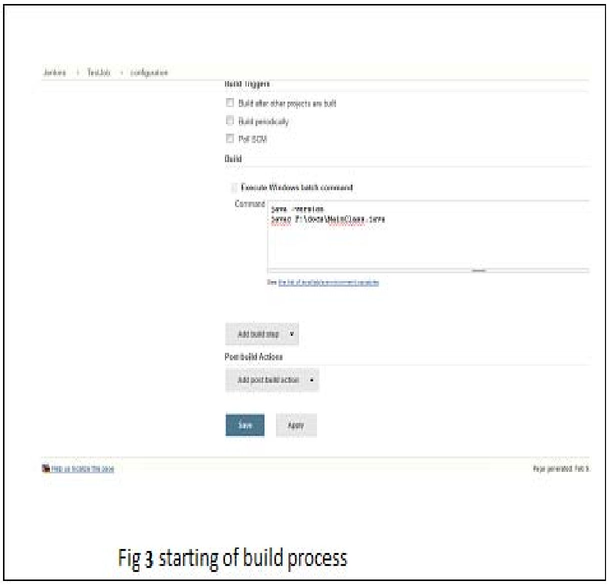


S	W	Name ↑	Last Success	Last Failure	Last Duration
		[illegible]	N/A	N/A	N/A

Fig 2 adding build steps in job configuration

Figure 2 : Adding build steps in job configuration

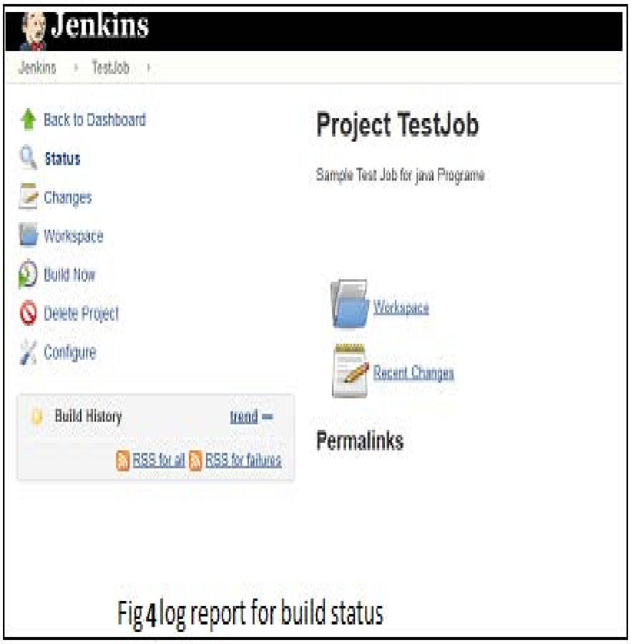
Once the build steps are added in the job configuration we started the build process which is depicted in the figure 3 and the corresponding log report is depicted in figure 4.


Jenkins › TestJob › configuration
Build Triggers
□ Build after other projects are built
□ Build periodically
□ Poll SCM
Build
Execute Windows batch command
Command
See the list of available environment variables
Add build step ▾
Post-build Actions
Add post-build action ▾
Save
Apply
Help us localize this page
Page generated: Feb 5

```
java -version
javac F:\docs\MainClass.java
```

Fig 3 starting of build process

Figure 3 : Starting of build process.

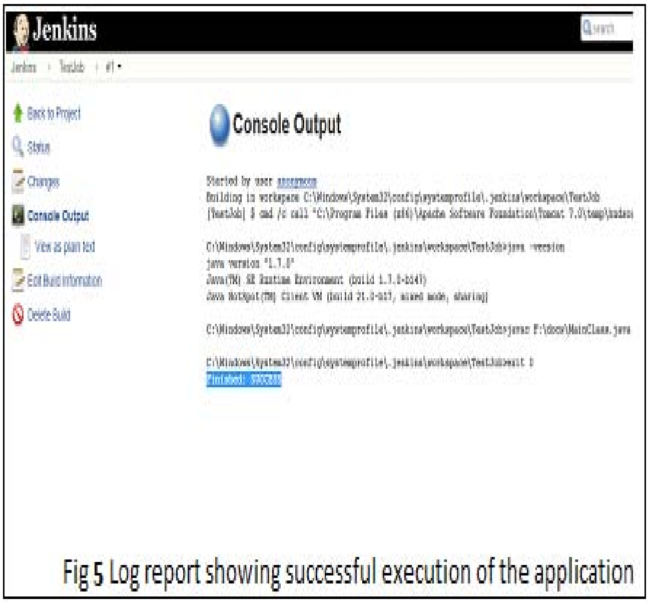


Project TestJob

Sample Test Job for java Programe

Fig 4 log report for build status

Figure 4 : log report for build status.

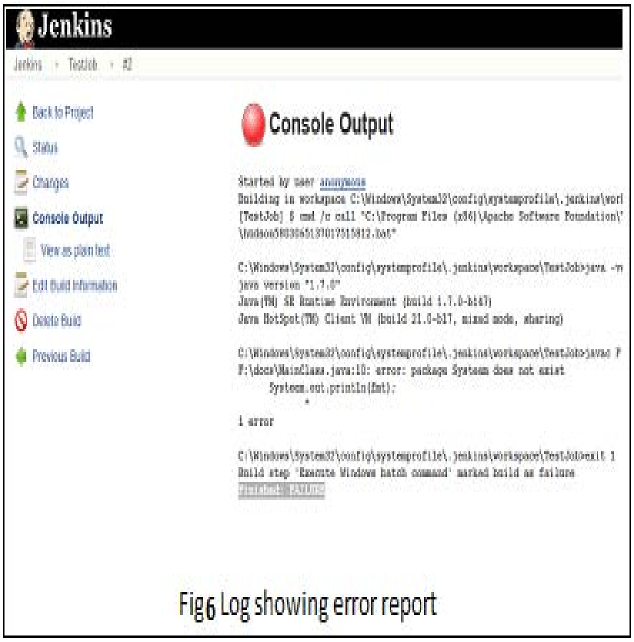


```
Started by user anonymous
Building in workspace C:\Windows\System32\config\systemprofile\.jenkins\workspace\TestJob
[TestJob] $ cmd /c call "C:\Program Files (x86)\Apache Software Foundation\Tomcat 7.0\temp\hudso

C:\Windows\System32\config\systemprofile\.jenkins\workspace\TestJob>java -version
java version "1.7.0"
Java(TM) SE Runtime Environment (build 1.7.0-b147)
Java HotSpot(TM) Client VM (build 21.0-b17, mixed mode, sharing)

C:\Windows\System32\config\systemprofile\.jenkins\workspace\TestJob>javac F:\docs\MainClass.java

C:\Windows\System32\config\systemprofile\.jenkins\workspace\TestJob>exit 0
Finished: SUCCESS
```

Fig 5 Log report showing successful execution of the application

Figure 5 : Log report showing successful execution of the application.

Figure 5 shows the successful execution of the application. In case if the java application is modified then automatically jenkins will trigger error report which is shown in the form of log report as depicted in the figure 6.



```
Started by user anonymous
Building in workspace C:\Windows\System32\config\systemprofile\.jenkins\wor
[TestJob] $ cmd /c call "C:\Program Files (x86)\Apache Software Foundation\
\hudson5803065137017515812.bat"

C:\Windows\System32\config\systemprofile\.jenkins\workspace\TestJob>java -v
java version "1.7.0"
Java(TM) SE Runtime Environment (build 1.7.0-b147)
Java HotSpot(TM) Client VM (build 21.0-b17, mixed mode, sharing)

C:\Windows\System32\config\systemprofile\.jenkins\workspace\TestJob>javac F
F:\docs\MainClass.java:10: error: package Systeem does not exist
        Systeem.out.println(Dmt);
              ^
1 error

C:\Windows\System32\config\systemprofile\.jenkins\workspace\TestJob>exit 1
Build step 'Execute Windows batch command' marked build as failure
Finished: FAILURE
```

Fig6 Log showing error report

Figure 6 : Log showing error report

III. Conclusion

Automated test data generation is an central area of research for sinking cost of software development. DevOps is definitely a new approach in testing strategies which increases the organization throughput. Based on the test information requirements the test data generation methods, diverse program analyzers are utilized To face the today's cloud based environment DevOps is the right choice for better results and to speed up customer query processing. In this paper we addressed the issues related to agile method and how the DevOps resolved the problems of agile.

References Références Referencias

1. Top Technologies For The Infrastructure & Operations BT Agenda, February 4, 2015,Laura Koetzle, Glenn O'Donnell, Eveline Oehrlich with Christopher Voce, Michael Caputo, Megan Doerr.
2. DevOps Market - By Technology (Sonar Qube, Continuous Integration (CI), Lean startup, Agile software development, Automation, Cloud Computing) - Business Models, Technology Roadmap, Advancements, Forecasts & Analysis (2014 - 2019)
3. White paper on What Smart Businesses Know About DevOps, TechInsights Report | September 2013.
4. X. Yuan and A. M. Memon. Iterative execution-feedback model-directed GUI testing. Information & Software Technology, 52(5):559–575, 2010.
5. Tsong Yueh Chen, Fei-Ching Kuo, Zhi Quan Zhou, "Automated software test data generation" Proceedings of the Fifth International Conference on Quality Software (QSIC05) 2005 IEEE.
6. Xiao Qu, Myra B. Cohen, Katherine M. woolf, "Combinatorial Interaction Regression Testing: A study of Test Case Generation and Prioritization" , 2007
7. Rullian Zhao and Qing Li., "Automatic test generation for dynamic data structure" , Technical report, Beijing University of Chemical Technology, 2006
8. Harrold, "Testing: A Roadmap", ACM Trans. on Software Engineering 2000

Table I : Comparison of Agile and DevOps.

Methodolgy	Principles	Methods	Practices	Tools
Agile	Follows strategic approaches	XP, Scrum	Standups, planning poker, backlogs, CI, all the specific artifacts a developer uses to perform their work.	JIRA Agile aka Greenhopper
DevOps	System thinking, Amplifying feedback loops (Gene Kim), "Infrastructure as code "are commonly cited DevOps principles.	Same as agile but Scrum with operations Kanban with perations and style change control	Automated Testing, Integrated Configuration Management, Integrated Deployment Planning	jenkins, travis, team city puppet, chef, ansible, cfengine, AWS, OpenStack, vagrant, docker