All software projects should follow a continuous delivery and continuous integration process.

Current software market is affluent. Customers expectation is ''sky's the limit'' and often on-demand, so company's software delivery strategy have to be agile. In recent years it is observed that to cope competition with rivals, software market organizations need to pay adequate attention to develop and deliver quality software to customer door step quickly. As per consecutive meeting with the engineering team and market operation team it is evident that though engineering team are sync with market operation team feedback and collaborate well, also they are building software as per market need, but still customers not getting those feature any faster. It is clear that we "Don't mistake activity with achievement." To accelerate the development and delivery of software features timely and without compromising quality all software projects should follow a continuous delivery and integration process.

According to experts opinion continuous delivery and continuous integration is the key steps in software delivery methodology which is more customer centric in terms of quality and acceleration. As per company traditional software delivery method as showing in Figure-1 , uses a linear approach, where each stage of the software development process has dependency on before stage. This means a stage must be completed before the next one begins. Here, once each person completed their work then other persons start work which is time consuming and if any change need during implementation phase then it need to revisit the preceding phase again to check back and work.

This stages usually consists the following:

1. Requirements gathering and documentation
2. System design
3. Code and unit testing
4. System testing

5. User acceptance testing
6. Bug fixes
7. Product delivery

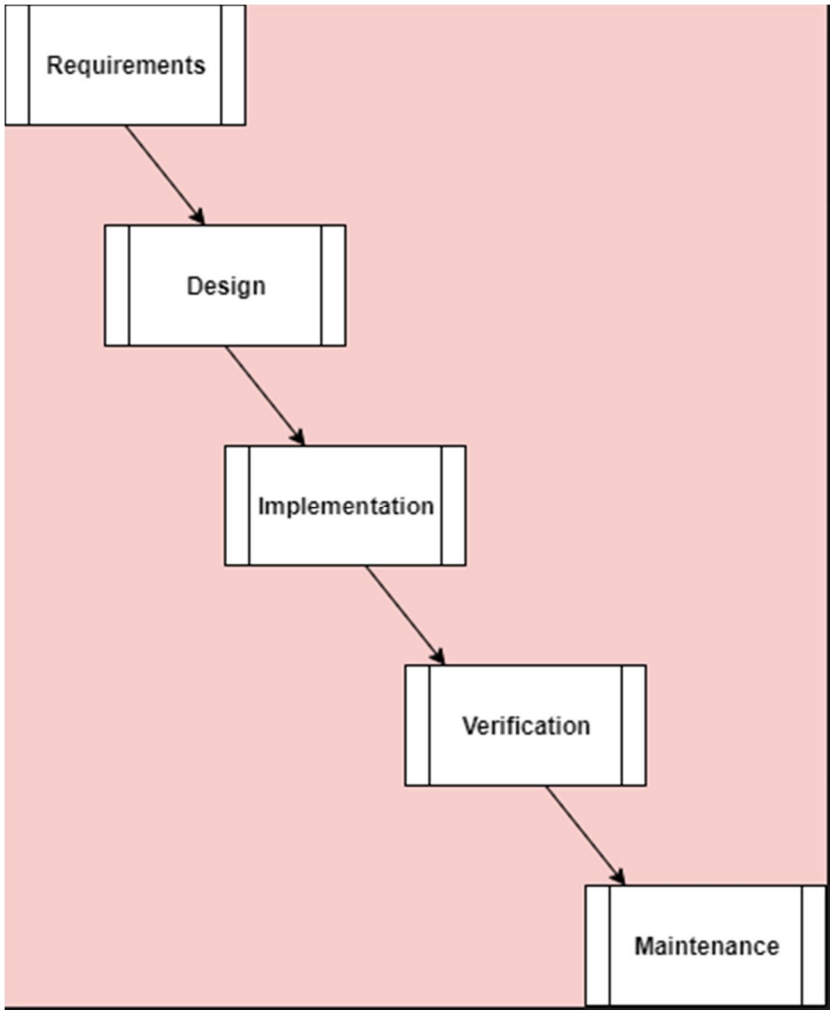

Figure 1: Company current traditional software delivery method

In this way both in engineering team both development and operation apartment become overloaded and frustrated as well as it consume time and cost as well. This method also not suitable for large software projects where consistent change is required as per market feedback.

Now let's explain why Continuous Integration and Continuous delivery should adopt.

Continuous Integration (CI)

As per traditional software delivery method Integration take place at end of the project after several stages finish their work. Sometimes it takes lot of time to perform the integration successfully. But, in Continuous integration practice can achieve the integration phase earlier in the so that building, testing and integrating code happens on a more regular basis as showing in Figure 2.

Here different stakeholder Developers write their code and integrate their code into common repository. Then it can be build the combined software from the scrap they each wrote and test that it works the way they expect.

Generally Developers generally use CI server to do the building and the integration for them. CI server need developers code and integrate those code together and run the build. This is code that tests itself to ensure that it is working as expected, and these tests are often called unit tests. When unit test completed successfully then they will see a green build. In this way Developers can verified that their changes are successfully integrated together, and the code is working as expected by the tests. Besides, share developers code at version control system will help them to maintain version of the code properly, here instead of waiting of someone other test and verified the code they can do the test by themselves by the help of automation. Though the integrated code is successfully working together, but it not yet ready for production because it has not been tested and verified as working in production-like environments. This phase is discussed in the Continuous Delivery section below.

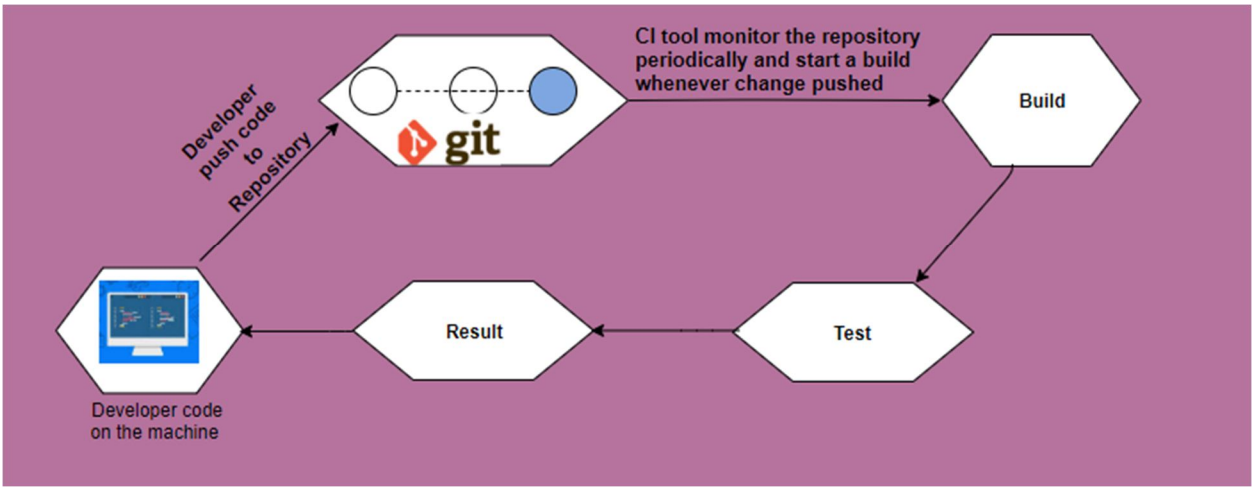

Figure 2: Continuous Integration Steps

Continuous Delivery (CD)

In Continuous Delivery means each time developers develop and push the code to repository, integrate and build the code, then CD ensure that the code *also* automatically test this code on environments which are similar to production environment. In this stage as environment, containerization technology is widely used. CD is aimed to ensure that an application is always at production-ready state after successfully passing automated tests and quality checks. This practice offers several benefits like reduced deployment risk, lower costs and getting user feedback faster. Here we can test the code at different environment which will give operation team confidence that the code will work on the production environment when the code is deployed there .Besides, Teams can detect any code that is not fit for delivery and then reject the code and provide feedback as early as possible. Crucially, the code is only promoted to (tested on) the next environment in the deployment pipeline *if* it passes the tests of the previous environment. Figure-3 showing steps of Continuous delivery steps

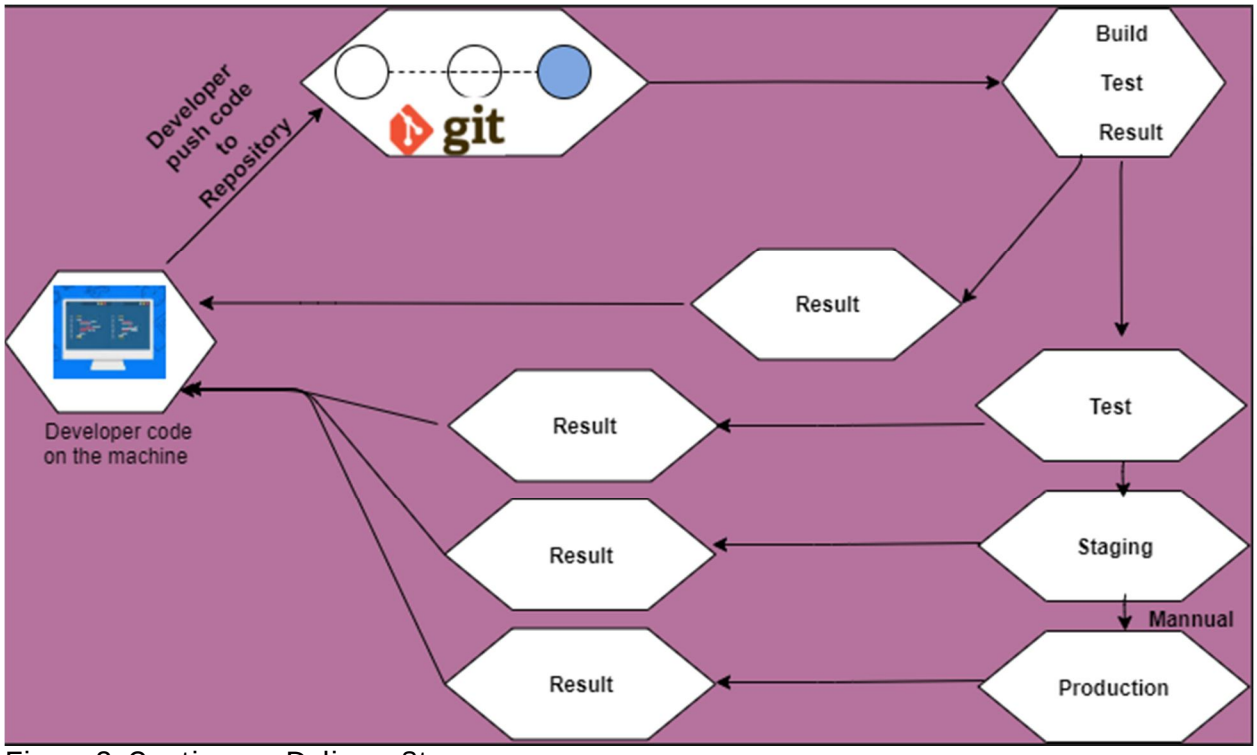

Figure 3: Continuous Delivery Steps

Tools Use in CI/CD Practice:

Most automated delivery pipelines include at least tools in these categories:

Source-code management: Tools include Git.

Build: Tools include Ant, Make, Maven, and Gradle.

Continuous integration (CI) server: Tools include Jenkins and Travis-CI.

Configuration management: Tools include Ansible, SaltStack, Chef, and Puppet.

Deployment and provisioning: Tools include IBM® UrbanCode® Deploy, Bamboo, and Chef.

Testing frameworks: Tools include xUnit, Behave, and Selenium. Testing frameworks tend to be programming-language specific.

Sample example how CI/CD Practice work to test and delivery code:

Figure 4 shows a simple diagram regarding automated webpage testing using Jenkins & Docker.

In steps key parts are:

- Maintain a GIT code repository.

- Scheduling CI Server Jenkins Build where Jenkins can monitor the repository and start a build whenever any changes have been committed by choosing Poll SCM option and entering syntax of cron.

- Once build start it will create delivery model docker container and apply web application according to pushed code.

- As testing; Script will check web application running status and check for a particular text “Hello World” in the webpage, if test success then it will send an email to user and perform automatic Deployment to production by using docker container and if test failed it will send a email notification to user. The test will be consider as failed is test script find any other extra word instead of “Hello World” as an example “<<Hello World>>”.

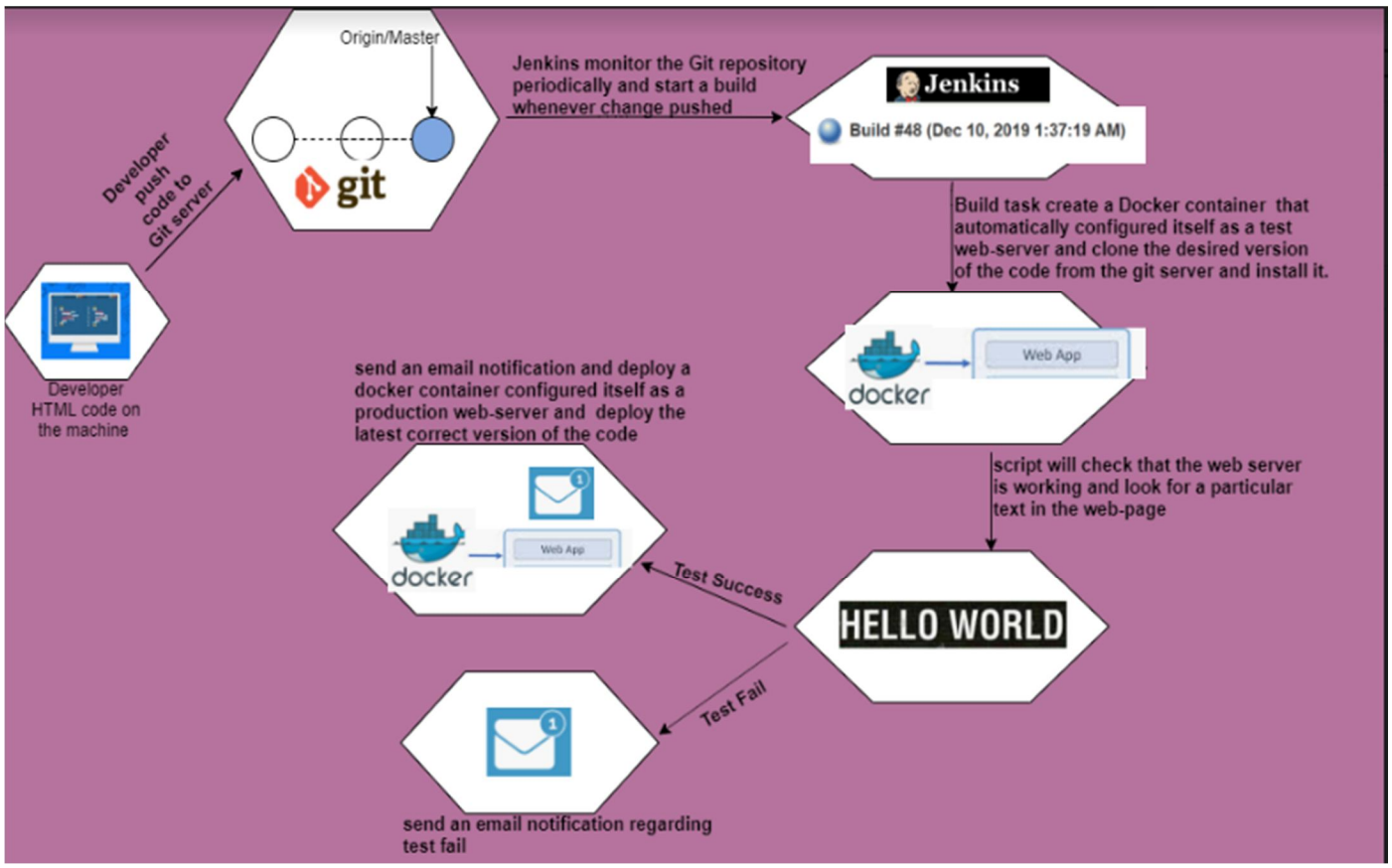

Figure 4: Automated web-page testing diagram using CI/CD Tools

Now, why don't we take a look at worlds big tech companies? How often they release software into production? Big tech companies — Facebook, Amazon, Netflix, Google — are releasing software updates thousands of times a day. Take Amazon, for example. In 2013, the company was doing a production deployment every 11.6 seconds. By 2015, this had jumped to 50 million deployments annually, or one every second, so there is no telling what cadence could be reached in 2019. How these big companies are achieving this big task? Applying continuous practice can help the company move forward to achieve more complex task in future.

Finally, Continuous Integration & Continuous Delivery process is very empowering for those of us in the business this approach. This method should be considered whenever possible, as it provides more benefits, especially for startups. It enables a complete functional software application to be released faster. Besides, if Developers code tests pass on all the environments, then we can know that the code is likely to be work as intended. Once the tests pass in all environments, we get to decide whether your end users get it or not. It is also more cost-effective, as making changes is less costly than with the traditional approach. Moreover, because the customer is highly involved and changes are constantly made to the application, a higher quality of work is achieved.