Continuous integration pipeline implementation for Tech11 software

By
Aswin G Sugunan(13)
Jeffin Jacob(17)
Nitin Suresh(25)
Vishnu Bose(39)
Guided By
Mrs.Greeshma
Asst.Professor in CSE

COLLEGE OF ENGINEERING CHERTHALA

October 20, 2016

Overview

1. INTRODUCTION
2. PROPOSED SYSTEM
3. modules
4. modules
5. CONCLUSION

(Common) Scenario

- **Developers working on a project.**
  - They each implement a few class.
    - Code them.
    - Ensure well tested.
  - When they're done,They *integrate* them.
    - Every thing breaks.

Integration Hell

That awkward moment near the end of the project when everyone realizes that none of their classes interoperate correctly.

Integration Hell (cont..)

Integration hell is extremely risk for a project.

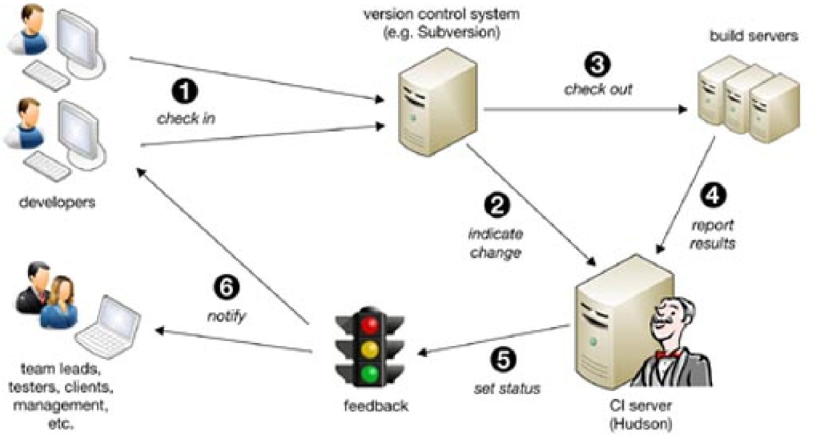
- **Difficult to determine how long it will take to resolve the integration process.**
  - May (vastly) exceed our budget.
  - May (vastly)exceed our schedule.

Continues Integration

- **Originated from eXtreme Programing (XP).**
- **Mitigates risk associated with integrating Software.**
- **Avoids integration hell.**
- **Integrate early and integrate often.**
    - *ie*, on every change.

Continues Integration Server

Automates the process of building,testing,reporting.



Benefits of CI server

- **Developer might forget to run the test.**
  - *Don't break the build.*
- **It may take too long to run the tests.**
- **We might need to test the code in various environments.**
  - Different architectures (32-bit,64-bit,ARM,PowerPC).
  - Different platforms (Windows,Linux,Mac,Solaris).

Benefits of CI server (cont..)

- **Reports provide useful insights to team.**
    - Can track metrics like line coverage.
        - Percentage of line executed by a program's test.
    - Can run all sorts of utilities on our code.
        - *CheckStyle, Findbugs, ..*
- **Can deploy automatically.**
    - Deploy a web project to a stagging server.
    - Deploy latest stable build of a desktop application to our website for download.

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator

Version Control

- This practice advocates the use of a revision control system for the projects source code.

- In this practice and in the revision control community, the convention is that the system should be buildable from a fresh checkout and not require additional dependencies.

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator

Artifact Manager

- An artifact repository is akin to what Subversion is to source code, i.e. it is a way of versioning code binary artifacts. In the Java world these artifacts could be jars, wars, ears, fully fledged applications, libraries or a collections of libraries that are packaged.
- Jenkins stores the artifacts as plain files without versioning while artifacts in an artifact repository can be version controlled.
- video

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator
- Test Automator

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator

Continuous Integration Handler

- It integrates with popular build tools (ant, maven, make) so that it can run the appropriate build scripts to compile, test and package within an environment that closely matches what will be the production environment
- It integrates with version control tools, including Subversion, so that different projects can be set up depending on projection location within the trunk.
- It can be configured to trigger builds automatically by time and/or changeset. (i.e., if a new changeset is detected in the Subversion repository for the project, a new build is triggered.)
- It reports on build status. If the build is broken, it can be configured to alert individuals by email.

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator

Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- Test Automator

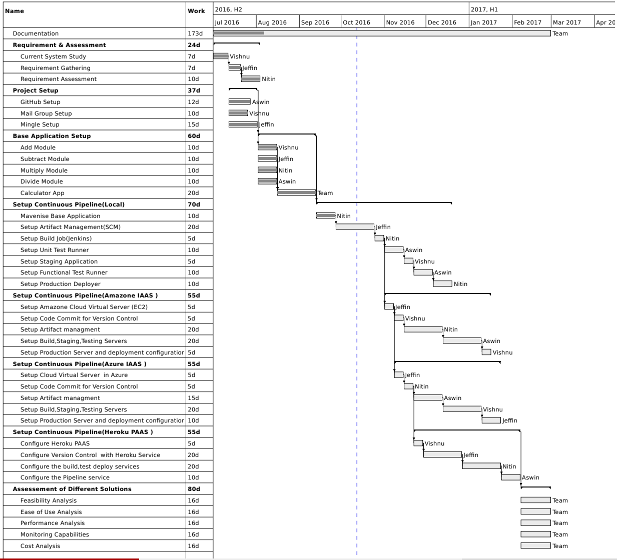
Test Automator

- In software testing, test automation is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes
- Test automation is critical for continuous delivery and continuous testing.
  - Graphical user interface testing.
  - API driven testing.

Operating Environment

- The system is expected to be operated in Linux as well as in windows with the support of respective JRE (Java Runtime Environment). This system based project is completely platform independent. The most important requirement is the internet connection.

Gantt Chart



Name	Work	
Documentation	173d	Team
Requirement & Assessment	**24d**	
Current System Study	7d	Vishnu
Requirement Gathering	7d	Jeffin
Requirement Assessment	10d	Nitin
Project Setup	**37d**	
GitHub Setup	12d	Aswin
Mail Group Setup	10d	Vishnu
Mingle Setup	15d	Jeffin
Base Application Setup	**60d**	
Add Module	10d	Vishnu
Subtract Module	10d	Jeffin
Multiply Module	10d	Nitin
Divide Module	10d	Aswin
Calculator App	20d	Team
Setup Continuous Pipeline(Local)	**70d**	
Mavenise Base Application	10d	Nitin
Setup Artifact Management(SCM)	20d	Jeffin
Setup Build Job(Jenkins)	5d	Nitin
Setup Unit Test Runner	10d	Aswin
Setup Staging Application	5d	Vishnu
Setup Functional Test Runner	10d	Aswin
Setup Production Deployer	10d	Nitin
Setup Continuous Pipeline(Amazone IAAS)	**55d**	
Setup Amazone Cloud Virtual Server (EC2)	5d	Jeffin
Setup Code Commit for Version Control	5d	Vishnu
Setup Artifact managment	20d	Nitin
Setup Build,Staging,Testing Servers	20d	Aswin
Setup Production Server and deployment configuratior	5d	Vishnu
Setup Continuous Pipeline(Azure IAAS)	**55d**	
Setup Cloud Virtual Server in Azure	5d	Jeffin
Setup Code Commit for Version Control	5d	Nitin
Setup Artifact managment	15d	Aswin
Setup Build,Staging,Testing Servers	20d	Vishnu
Setup Production Server and deployment configuratior	10d	Jeffin
Setup Continuous Pipeline(Heroku PAAS)	**55d**	
Configure Heroku PAAS	5d	Vishnu
Configure Version Control with Heroku Service	20d	Jeffin
Configure the build,test deploy services	20d	Nitin
Configure the Pipeline service	10d	Aswin
Assessement of Different Solutions	**80d**	
Feasibility Analysis	16d	Team
Ease of Use Analysis	16d	Team
Performance Analysis	16d	Team
Monitoring Capabilities	16d	Team
Cost Analysis	16d	Team

Data Flow Diagram

- level 0

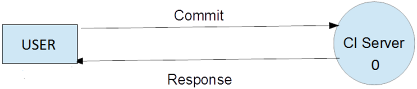


Figure : level 0 dfd

Data Flow Diagram

- level 1

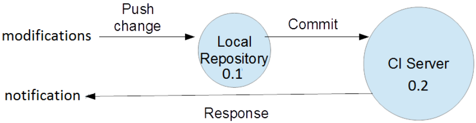


Figure : level 1 dfd

Data Flow Diagram

- level 2

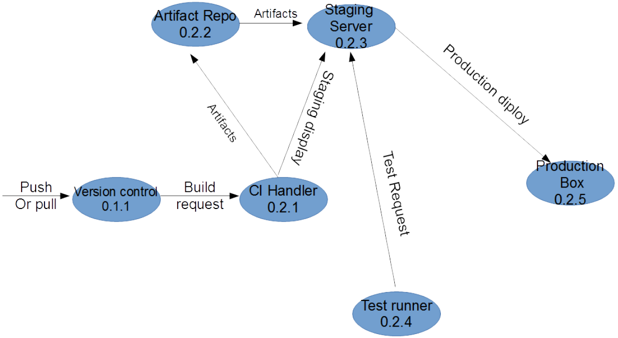
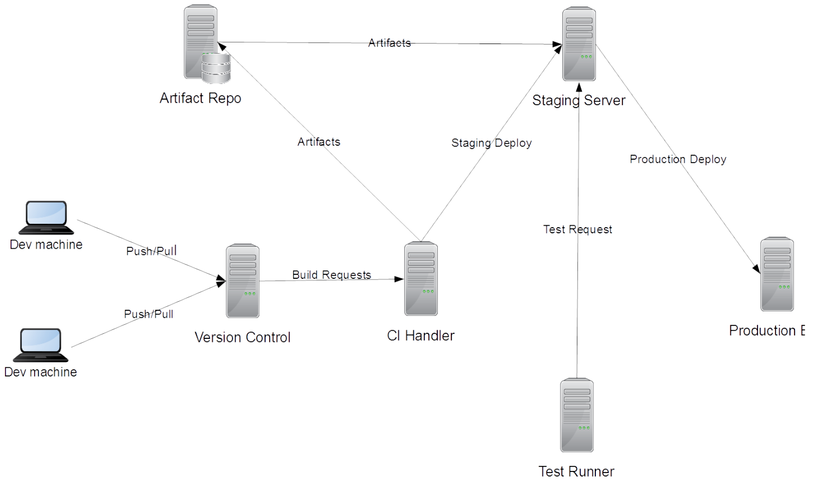
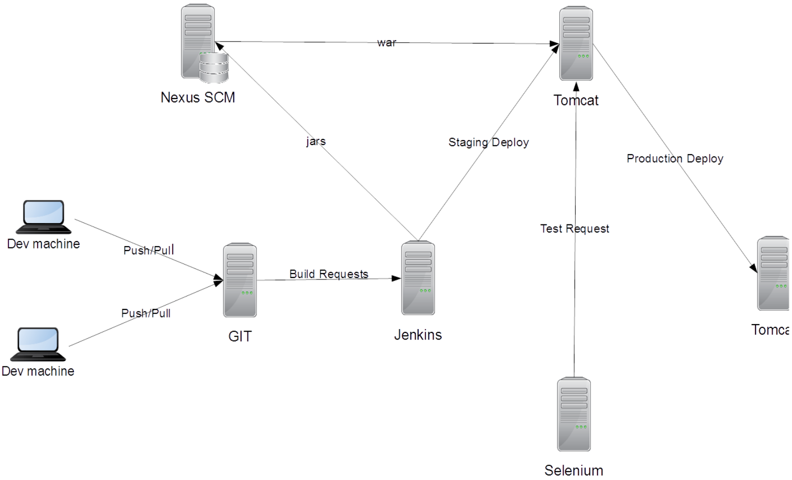


Figure : level 2 dfd

Flow Diagram



CI Pipeline Diagram



Sequence Diagram

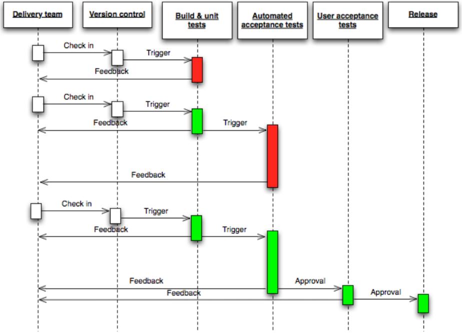


Figure : Sequence Diagram

CONCLUSION

- **Continually integrate and test to reduce risk.**
- **Detect problems early.**
- **Always have a deployable build.**
- **Generate metrics to guide project management.**
- **Continuous Integration is:**
  - *A good practice in any software development method.*
  - *Vital for agile development.*

thank you!