

# Continuous integration pipeline implementation for Tech11 software

By

Aswin G Sugunan(12143605)

Jeffin Jacob(12143611)

Nitin Suresh(12143616)

Vishnu Bose(12143626)

Guided By

Mrs.Greeshma

Asst.Professor in CSE

COLLEGE OF ENGINEERING CHERTHALA

# Overview

- 1 INTRODUCTION
- 2 PROPOSED SYSTEM
- 3 MODULES
- 4 IMPLEMENTATION
- 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

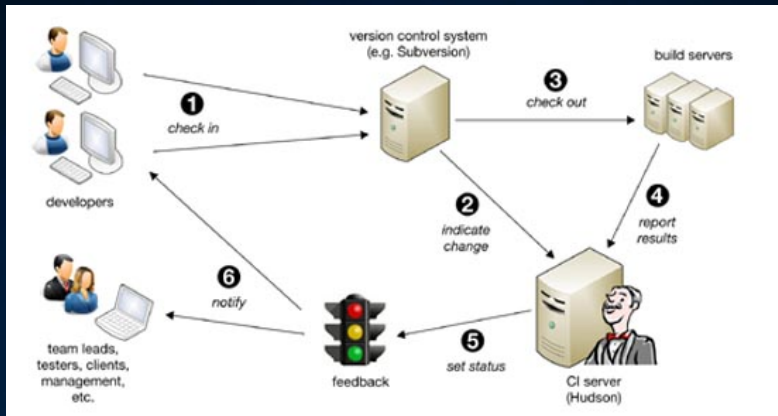
- **Originated from eXtreme Programing (XP).**
- **Mitigates risk associated with integrating Software.**
- **Avoids integration hell.**
- **Integrate early and integrate often.**
  - *ie, on every change.*

# Evolution of software delivery

- Waterfall
- Agile
- Continuous Delivery
  - Another subset of agile which in which the team keeps its software ready for release at all times during development. It is different from traditional agile in that it does not involve stopping and making a special effort to create a releasable build.

## Continues Integration

**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 staging server.
  - Deploy latest stable build of a desktop application to our

# 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.

# Version Control

- Here we use Git-Hub for version controlling.
- GitHub is a web-based Git or version control repository and Internet hosting service. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features.



# Modules

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

# Modules

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



# Artifact Manager

- Artifact repositories are great at managing multilevel dependencies, much better than the old text file with a list that developers update and maintain.
- Critical for reducing errors and ensuring the right pieces make it with each build/deployment/release, especially in large-scale business applications.
- video

# Modules

- Version Control
- Artifact Manager
- Continuous Integration Handler
- 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.



# Data Flow Diagram

- level 0



Figure : level 0 dfd

# Data Flow Diagram

- level 1

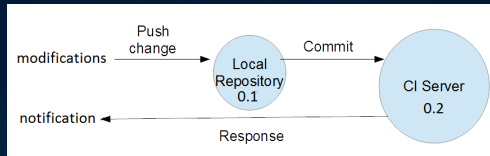
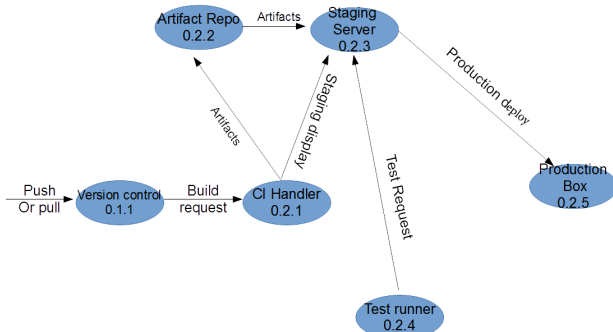


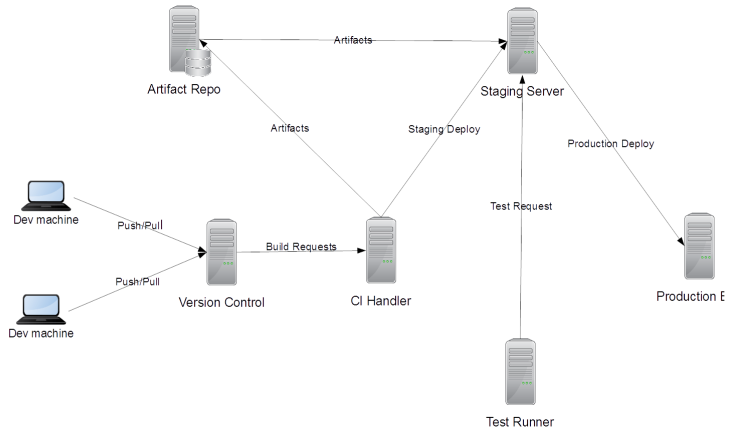
Figure : level 1 dfd

# Data Flow Diagram

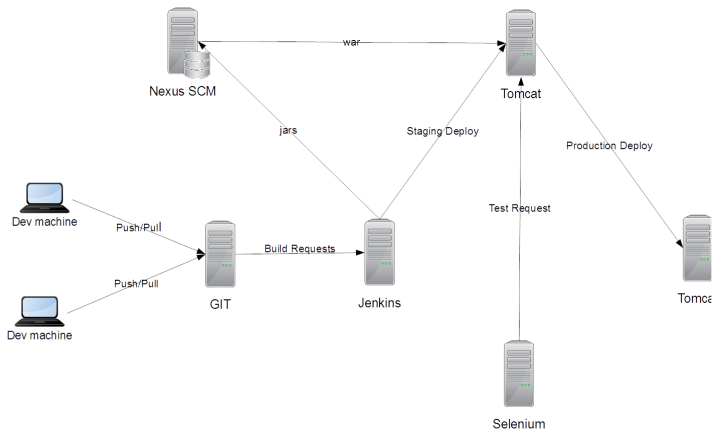
- level 2



# Flow Diagram



# CI Pipeline Diagram



# Sequence Diagram

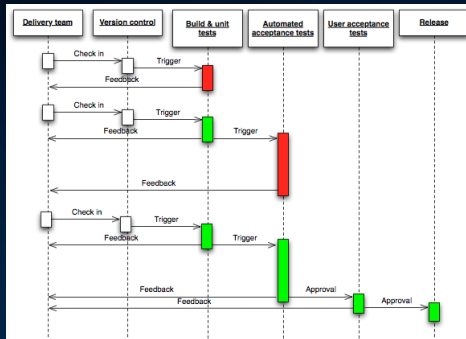


Figure : Sequence Diagram

# Implementation

- For demonstration to the Tech 11 Software Team, We decided to implement a Calculator application (using Java, javaScript, NodeJs, Angular...)
- We divide the work
  - Addition Module - Vishnu Bose
  - Subtraction Module - Nitin Suresh
  - Multiplication Module - Aswin G Sugu
  - Division Module - Jeffin Jacob
  - View - All

# Implementation

## For Version Control

- Created a Repository named LogicBaker in Git-Hub and cloned to individual systems.
- Create module given to each members in java
- After Coding the individual modules the team members commit to Git-Hub, irrespective on the platform which each one has build.



# Implementation

## **Artifact Repository**

- An artifact repository is akin to what Subversion is to source code, i.e. it is a way of versifying 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.

# Implementation

## **Artifact Repository**

- The Difficulty in integrating the individual modules due to the syntax mis-matching is overcome by Mavenising the Individual modules.

# Implementation

## Artifact Repository

- Maven was originally started as an attempt to simplify the build processes in the Jakarta Turbine project. There were several projects each with their own Ant build files that were all slightly different and JARs were checked into CVS. We wanted a standard way to build the projects, a clear definition of what the project consisted of, an easy way to publish project information and a way to share JARs across several projects.

# Implementation

## Artifact Repository

- The Same Calculator Test application code is mavenised by each team members and again pushed to the local repository.
- The whole can integrated and build in each team members local system much easier.
- With listed dependencies docs.

# Implementation

## **Continous Integration Handler**

- To review intermediately the generated code, Understanding what or How it looks like when Deployed?
- Thats the idea behind Continuous Integration, We provide a Automation platform when all team members and the project manager can VIEW the application and discuss.

# Implementation

## Continous Integration Handler

- For Automation we use Jenkins.
- Jenkins is an open source automation server written in Java. The project was forked from Hudson after a dispute with Oracle. ... It is a server-based system running in a servlet container such as Apache Tomcat.

# Implementation

## **Test Automator**

- The test Automation plugging is added to Jenkins depending upon the Nature of the development process dealing with.
- Which involves programming framework, language etc.
- Since we use Java , we use find-bugs and Check style in Jenkins..
- Also added Jacoco , For line coverage analysis

# 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.*