#Introduction

In software engineering, Continuous Integration (CI) is the practice of merging all developer working copies to a shared mainline several times a day.Continuous Integration (CI) is a development practice that requires developers to integrate code. Grady Booch first proposed the term CI in his 1991 method, although he did not advocate integrating several times a day. Extreme programming (XP) adopted the concept of CI and did advocate integrating more than once per day – perhaps as many as tens of times per day.

#Advantages

- Integration bugs are detected early and are easy to track down due to small change sets. This saves both time and money over the lifespan of a project.

- Avoids last-minute chaos at release dates, when everyone tries to check in their slightly incompatible versions

- When unit tests fail or a bug emerges, if developers need to revert the codebase to a bug-free state without debugging, only a small number of changes are lost (because integration happens frequently)

- Constant availability of a "current" build for testing, demo, or release purposes

With continuous automated testing benefits can include:

- Enforces discipline of frequent automated testing

- Immediate feedback on system-wide impact of local changes

- Software metrics generated from automated testing and CI (such as metrics for code coverage, code complexity, and feature completeness) focus developers on developing functional, quality code, and help develop momentum in a team

#Tools

- TeamCity

- Travis

- Jenkins

- GoCD

- Gitlab

##GitLab CI

### Advantages:

- It is an integral part of the open-source Rails project GitLab and a free hosted service.

- It can run on Windows, Linux, OSX, FreeBSD, and Docker.

- It can run multiple jobs concurrently.

### Disadvantages:

- GitLab is open core and anyone can contribute changes directly to the codebase.

##Jenkins

### Advantages:

- Jenkins is an open-source CI tool written in Java.

- Flexible.

- Jenkins plugin list is very comprehensive.

- Jenkins prides itself on distributing builds and test loads on multiple machines.

- Easy installation.

### Disadvantages:

- UI Interface is very intuitive.

- Configuration is a bit confusing.

- Errors messages are not very clear about what went wrong.

##Tarvis

### Advantages:

- Comes with free cloud-based hosting that requires no maintenance or administration

- Capable of running tests on Linux and Mac OS X simultaneously

- Supports the following languages: Android, C, C#, C++, Clojure, Crystal, D, Dart, Erlang, Elixir, F#, Go, Groovy, Haskell, Haxe, Java, JavaScript (with Node.js), Julia, Objective-C, Perl, Perl6, PHP, Python, R, Ruby, Rust, Scala, Smalltalk and Visual Basic

- Lightweight and easy to set up

- Free for open source projects

- No dedicated server needed

- Build matrix feature

### Disadvantages:

- Enterprise plans come with a cost

- Limited options for customization

##TeamCity

###Advantages

- TeamCity provides a great integration with git, especially Bitbucket.

- When a new code release (build) fails TeamCity has a great tool for investigation and troubleshooting.

- TeamCity provides a user-friendly interface. While some technical knowledge is required to use TeamCity, the design helps simply things.

##Disadvantages

- Upgrading TeamCity is a long and manual process.

- Java skills are needed to fully utilize TeamCity, although they are not necessary for basic or medium-level use.

##GoCD

###Advantages

- Easy Setup for deployment pipeline

- Environment Variables for each step

- Supports both Windows and Linux agent

- Highly customizable

##Disadvantages

- UI can be improved

- Location for settings can be re-arranged

- API for setting up pipeline