# Software Engineering

# CI – Continuous Integration

**Essentially, CI is an integrated test service that works alongside the source repository service and operates automatically and continuously for the purpose of determining the quality of code, in terms of correctness, robustness, simplicity, readability, or other particular standards, in a development project.**

**Its use was mainly a supervisor to**

* **Ensure some coding standards,**
* **Reduce evitable bugs, recurring bugs at an early stage,**
* **Determine the compatibility of code,**
* **Enable hands-free test and deployment**

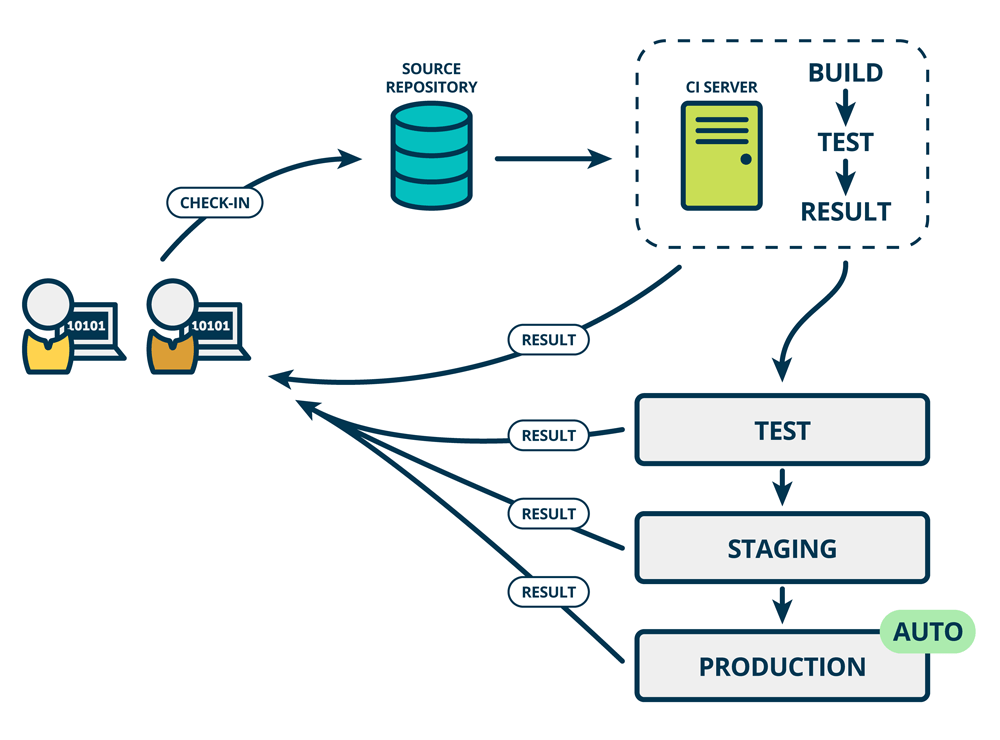
**CI is often integrated with Version Control System. CI will automatically run generations and tests every time a team member has submitted changes.**

**CI encourages developers to merge changes into the version control repository after each small task is completed to share code and take advantage of all unit tests available.**

**Submission of code will trigger the automated build system to retrieve the latest code from version control system and then generates, tests and validates the complete latest master branch.**

**CI has become a common practice since it’s extremely helpful when it comes to code integration: developers often work alone and need to integrate with the team’s codebase. CI has been a very suitable tool that provides a holistic method to aid the integration of tasks and codebases, reducing duplication of efforts and shortening the testing period.**

**One of the CI’s most palpable advantage is that the CI has ensured a ready-to-deploy status of the main branch. When a task is allocated, developers would clone a new branch of code and make changes in its own code branch. After completing changes, developers could submit a pull request to be merged into the main branch. The CI comes when the pull request is created and starts the tests during this time. It’s only after building and testing by the CI could the branch actually be merged, which in turn guarantees that the main branch after any merge is ready to build and passing all previous tests.**



# Jenkins

Jenkins is a self-contained, open source automation server which can be used to automate tasks related to building, testing, and delivering or deploying software.

# Git – version control

Git is one of the most popular version control system today, used world-wide and accepted as a common practice by many open-source projects, also adopted by GitHub, on which millions of projects hosted are all using GitHub as their version control system.

Git is initiated as a version control system to be implemented by the Linux developer community, with Linus Torvalds as one of the founders of the project. At that stage, the git was designed as a tool for global contributors to cooperate and to ease the increasingly complex submission and reviewing process.

This comes down to 4 major features :

1. Easy-to-use branching and merging
2. Comparison between histories and different branches
3. Distributed repository (without main server)
4. Code review (especially peer review)

Early in planning of this, git was used in such a gigantic Linux project which have given it a widespread reputation and continuous improvements.

Its contributors:

1. Junio C Hamano – maintainer of the ‘ git ‘ project

Studied with Professor from UCLA,

I’ve changed a part of this open stuff, that made my life better,

Makes somebody life better, become better place.

Get involved,

Between projects, upside down. Browsing the mailing list, interested, what’s going on. We are switching to something that doesn’t exist yet, so we are building one.

Without previous development experience with source control systems !!!

Heard the name ‘bitkeeper’ , yet none used. Free Beer / No SourceCode

Proprietary software (although free), yet still proprietary, 2004-2005

Business, free stuff.

I want to contribute back.

Tiny little code (200,000 lines), hundreds lines of code.

Start making,

Work is nowhere related to source code control.

**Time was sponged.**

Early in the morning or late after work.

**Cubicle at work place, endowed the eventuality of hacking, cultivated his creative way of using time.**

**Junio Hamano**

Major contributor of Git since 2005, still active maintainer.  
**MTS Google**

### Feature Outline:

#### Branching and merging

Branching is tailored for feature-oriented development, for the fact that most of the developers’ work are organized in this way, that different people works on different features. Git has enabled a unique way for different groups of people to independently develop on a core basis ’main branch’, and each doing a different add-on to the core, which in the end merged and integrated to the core.

Branching and merging has also enabled many types of workflows, either testing and developing branch apart from deployment ‘main branch’ or a double verification on merges with different types of tests. With the distributed feature of git, everyone could change their code branch with in seconds and coordinate their code with others without a tiring human-involved conde integration.

#### Small and fast

Git stores the branching and merging histories inside the project file folder, yet have consumed a minimal amount of space on disk.

The extra amount of space that was used to store the whole project history is actually acceptable compared to other version control system.

Yet this does not affect the blazing rapidity of cloning, merging and other branching processes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** |  | **Git\*** | **Git** | **SVN** |
| Clone | Clone and shallow clone(\*) in Git vs checkout in SVN | 21.0 | 107.5 | 14.0 |
| Size (M) | Size of total client side data and files after clone/checkout (in M) |  | 181.0 | 132.0 |

Source : git-scm.com

#### Distributed

The git server itself has a distributed way of storing code bases, unlike its predecessors, git does not require a host-server that controls the versions and everyone have to transfer the data to.

Every one in a git-based project could commit changes or make merges, clones locally and share any branches with others later. This also means that one could entirely copy the repository with all its due history from anyone in the project, needless to access the central server, this have liberated many types of cooperation, coped with the single-point failure and in the process reduced the cost of a central server.

### Secure

Git implements a host of checksum and encryption process for users to have a better experience when it comes to authorization, reviewing, and network problems.

Every file and commit are checksummed and retrieved by its checksum when checked out. It guarantees a secure and robust downloading and uploading procedure. With inherent SSL support, git can be very useful to tackle fake hosts and identity fraud.

# Building and Generating Tools

C / C++

## Linux

Building System

GNU C

GNU Make

### Pmake

## Windows

### MSBuild

Used by the Visual Studio IDE,

### NMake

Microsoft Program Maintenance Utility (NMAKE.EXE) is a toolset delivered with Visual Studio, using a Descriptor File to automate the build process.

### Ninja

## macOS

Xcode workaround

Clang / LLVM

## Cross Platform

Directories / Functionalities

Library , Executable

# Java Building Toolset

## Maven

## Gradle

## Ant