

# Table of Contents

Continuous integration

*Disciplines & Methodologies*

*Tools & Technologies*

*Links*

3

8

8

9



# Continuous integration

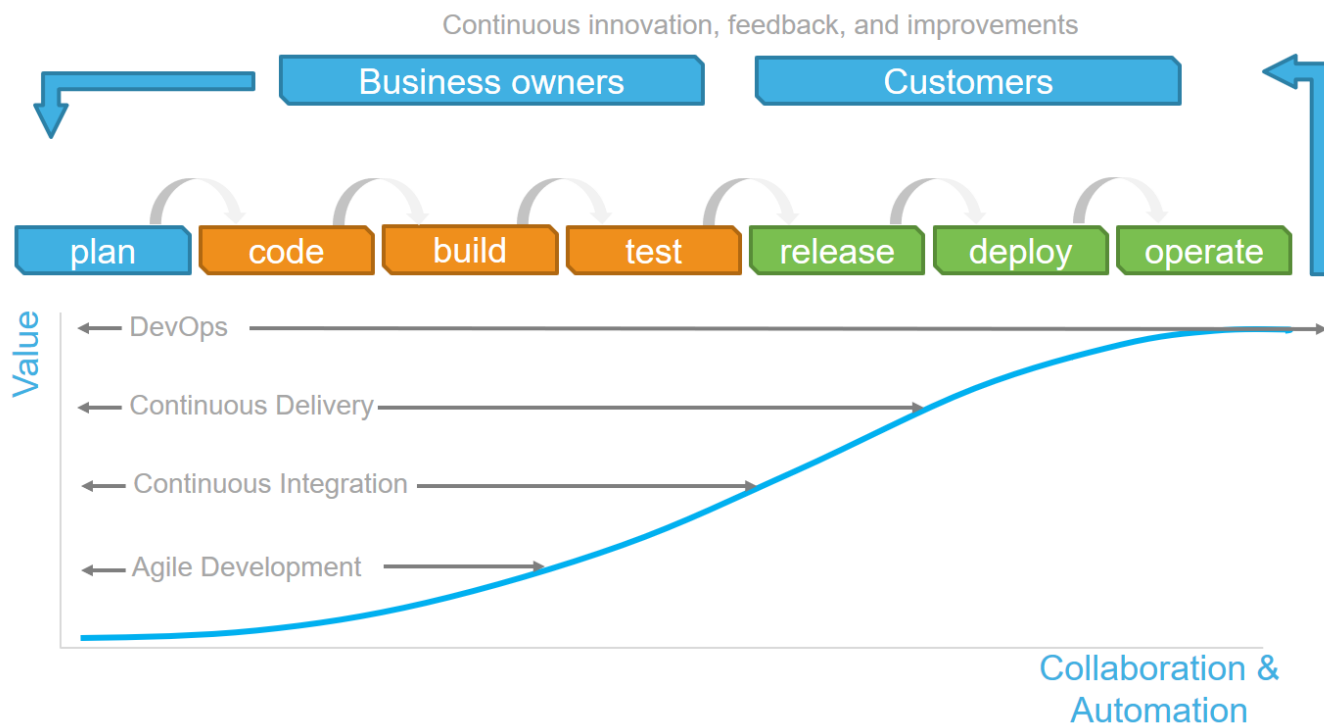
Continuous integration is a software development practice that aims to improve the quality and speed of software delivery. It involves integrating code changes from multiple developers into a shared repository frequently, usually several times a day. By doing so, it enables early detection and resolution of bugs, conflicts and errors, as well as faster feedback and validation of new features. Continuous integration also facilitates collaboration and communication among developers, testers and other stakeholders. It is often combined with other practices such as continuous delivery, continuous testing and continuous deployment to form a continuous software pipeline that delivers value to customers faster and more reliably.

- [CI - CD - CD](#)
- [DevOps Toolchain](#)

Continuous integration (CI) is a software development practice that involves integrating code changes from multiple developers into a shared repository frequently, usually several times a day.

CI has many benefits for software quality, efficiency and collaboration, such as:

1. **Detecting and fixing errors early:** CI allows developers to catch bugs and errors before they become too costly or complex to fix. By running automated tests and code analysis tools on every code change, CI ensures that the code is always in a working state and meets the quality standards.
2. **Reducing integration conflicts:** CI reduces the risk of integration conflicts that can arise when developers work on different branches or features for a long time without merging their code. By integrating code frequently, CI ensures that the code base is always up-to-date and consistent across all developers.
3. **Improving feedback and communication:** CI provides fast and continuous feedback to developers about the status of their code changes, such as whether they pass or fail the tests, whether they comply with the coding standards, whether they introduce any security vulnerabilities or performance issues. This feedback helps developers to improve their code quality and learn from their mistakes. CI also facilitates communication and collaboration among developers by making their work more transparent and visible to others.
4. **Accelerating delivery and deployment:** CI enables faster delivery and deployment of software products by automating the build, test and release processes. By ensuring that the code is always ready for deployment, CI reduces the time and effort required to deliver software updates to customers or users.



CI is an essential component of both ALM and DevOps, as it enables faster feedback loops, higher quality standards, and more frequent releases. CI helps to align the goals and expectations of different teams and stakeholders involved in the software project. By integrating code changes regularly and automatically into a shared repository, CI ensures that everyone is working with the latest version of the codebase. This reduces conflicts, errors, and delays in delivering software products that meet customer needs.

See also [Continuous Delivery](#)

## What is continuous integration (CI)?

Continuous integration is a software development practice where developers frequently merge their code changes into a shared repository, which is then automatically built, tested, and integrated into the software.

## Why is continuous integration important?

Continuous integration is important because it helps to identify and address integration issues early in the development cycle, reducing the risk of defects and errors in the final software product.

## What are some benefits of continuous integration?

Benefits of continuous integration include faster development cycles, better collaboration among team members, improved software quality, and reduced risk of integration problems.

## What are some common tools used in continuous integration?

Some common tools used in continuous integration include Jenkins, Travis CI, CircleCI, GitLab CI/CD, and TeamCity.

## What is continuous delivery (CD)?

Continuous delivery is an extension of continuous integration, where software is automatically built, tested, and deployed to production or other environments.

## What is the difference between continuous integration and continuous delivery?

Continuous integration is focused on automating the build, test, and integration process, while continuous delivery is focused on automating the entire software delivery process, from build to deployment.

## What are some benefits of continuous delivery?

Benefits of continuous delivery include faster and more reliable software releases, reduced risk of deployment failures, and increased agility in responding to changing business needs.

## What are some common tools used in continuous delivery?

Some common tools used in continuous delivery include Docker, Kubernetes, Ansible, Puppet, and Chef.

## What is a CI/CD pipeline?

A CI/CD pipeline is a series of automated steps that take code changes from development through to deployment, including building, testing, and deployment.

## What are some best practices for implementing continuous integration and delivery?

Best practices for implementing continuous integration and delivery include ensuring a stable and maintainable codebase, using version control, automating the build and test process, monitoring performance and metrics, and fostering a culture of collaboration and continuous improvement.

Snippet from [Wikipedia](#): **Continuous integration**

In software engineering, **continuous integration (CI)** is the practice of merging all

developers' working copies to a shared mainline several times a day. Nowadays it is typically implemented in such a way that it triggers an automated build with testing. 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.

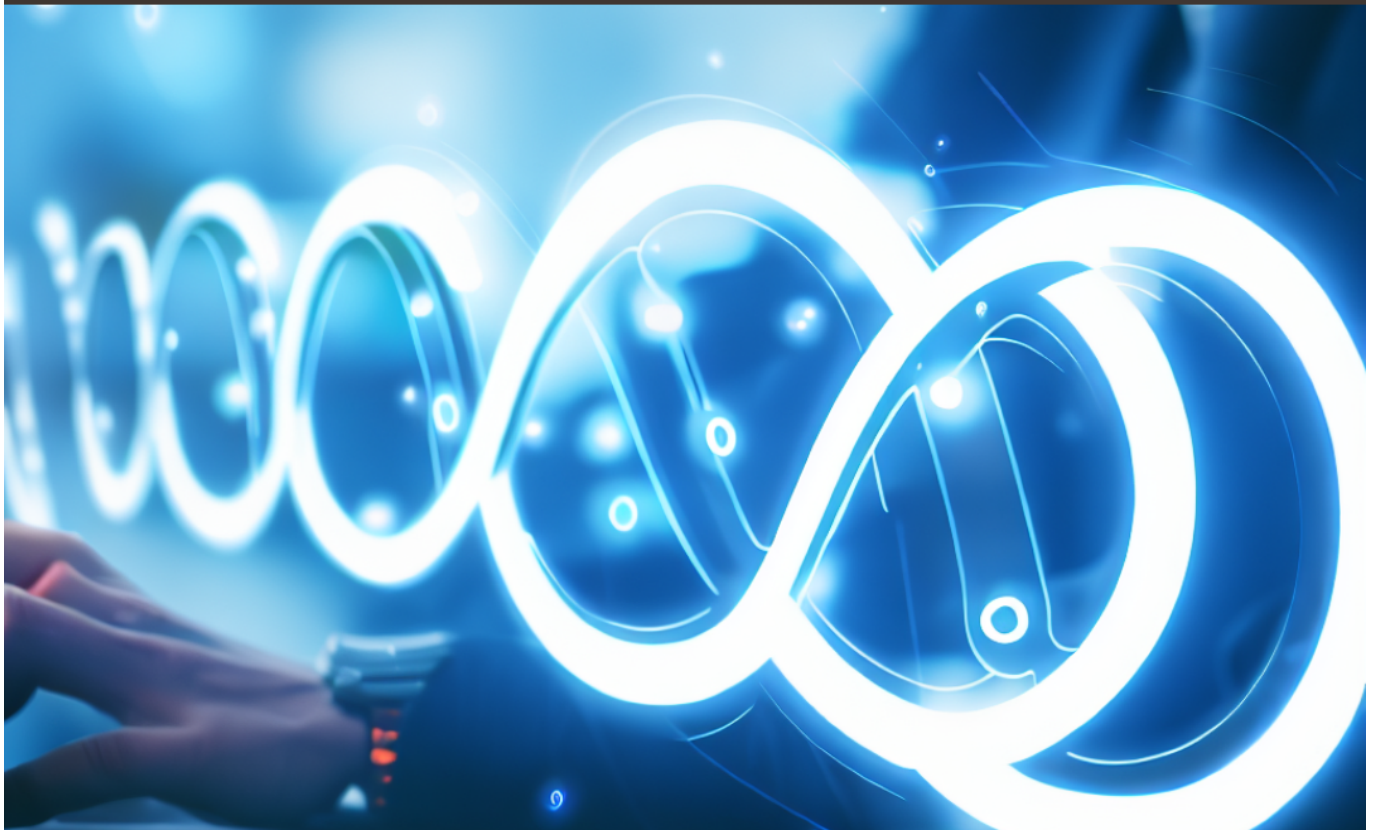
[Creative Commons Attribution-Share Alike 4.0](#)

# What is Continuous Integration

Continuous integration is the practice of regularly and automatically integrating code changes into a shared repository to detect and resolve integration issues early in the software development process

WHAT IS CONTINUOUS INTEGRATION?

**CONTINUOUS INTEGRATION IS A SOFTWARE DEVELOPMENT PRACTICE THAT AIMS TO IMPROVE THE QUALITY AND SPEED OF SOFTWARE DELIVERY**



IMPROVE QUALITY AND SPEED OF SOFTWARE DELIVERY

[Previous](#) [Next](#)

#### GitHub Topics

- <https://github.com/topics/continuous-integration>

Automatically build and test your code as you push it upstream, preventing bugs from being deployed to production. A complementary practice to CI is that before submitting work, each programmer must do a complete build and run (and pass) all unit tests. Integration tests are usually run automatically on a CI server when it detects a new commit.

[https://en.wikipedia.org/wiki/Comparison\\_of\\_continuous\\_integration\\_software](https://en.wikipedia.org/wiki/Comparison_of_continuous_integration_software)

## Disciplines & Methodologies

- [Build automation](#)
- [Container Management](#)
- [Continuous configuration automation](#)
- [Continuous Delivery](#)
- [Database Automation](#)
- [DevOps toolchain](#)
- [GitOps](#)
- [Repository Management](#)
- [Software development process](#)
- [Software documentation](#)
- [Static program analysis](#)

## Tools & Technologies

- [Ansible](#)
- [Apache Archiva](#)
- [Apache Continuum](#)
- [AppDynamics](#)
- [Automic Continuous Delivery Director](#)
- [AWS DevOps](#)
- [Azure DevOps](#)
- [Bamboo](#)
- [Bash](#)
- [BigPanda](#)
- [Buddy](#)
- [Buildbot](#)
- [Buildkite](#)
- [CA Automic](#)
- [Chef](#)
- [Clarive](#)
- [CloudRepo](#)
- [Cloudsmith](#)
- [AWS CodeBuild](#)
- [CruiseControl](#)
- [Electric Cloud](#)
- [FinalBuilder](#)
- [GitLab](#)
- [Google DevOps Solutions](#)
- [Gradle](#)
- [HP ALM](#)
- [Inedo ProGet](#)
- [Jenkins](#)
- [Jfrog Artifactory](#)
- [LaunchDarkly](#)
- [Maven](#)
- [npm](#)
- [NuGet](#)




- [Octopus Deploy](#)
- [Open Build Service](#)
- [Otter](#)
- [PTC Integrity Lifecycle Manager](#)
- [Puppet](#)
- [QuickBuild](#)
- [Rally Software](#)
- [Rational solution for CLM](#)
- [Rational Team Concert](#)
- [Salt](#)
- [Sonatype Nexus](#)
- [Spinnaker](#)
- [Team Foundation Server](#)
- [TeamForge](#)
- [Terraform](#)
- [Travis CI](#)
- [UrbanCode](#)
- [UrbanCode Build](#)
- [Visual Studio](#)
- [VSALM](#)
- [Visual Studio Team Services](#)
- [XebiaLabs](#)
- [Yarn](#)

## Links

- [Comparison of open-source configuration management software](#)
- [Change control](#)
- [CI/CD](#)
- [Git](#)
- [Intelligent Process Automation \(IPA\)](#)
- [Representational State Transfer \(REST\)](#)
- [Software versioning](#)
- [TFVC](#)
- [Universal package manager](#)
- [Version control](#)

[development](#)

**## ToDo ##**

-  - [Support Us...](#) →
- [CircleCI](#)
- [Codeship](#)
- [Concourse CI](#)
- [Continua CI](#)
- [Drone.io](#)
- [GitLab CI](#)

- Gump
  - Harness.io
  - Hudson
  - Hydra
  - Jenkins X
  - Nevercode
  - ProductionMap
  - Semaphore
  - Shippable
  - Snap CI
  - Solano CI
  - TeamCity
- 
- Continuous Integration Process
  - Continuous Integration Tools
  - Continuous Integration Best Practices
  - Continuous Integration Benefits
  - Continuous Integration Challenges
  - Continuous Integration Automation
  - Continuous Integration Testing
  - Continuous Integration Configuration Management
  - Continuous Integration Deployment
  - Continuous Integration Environment Management
  - Continuous Integration Metrics
  - Continuous Integration Monitoring
  - Continuous Integration Server
  - Continuous Integration Pipeline
  - Continuous Integration Workflow
  - Continuous Integration Code Quality
  - Continuous Integration Code Coverage
  - Continuous Integration Code Review
  - Continuous Integration Collaboration
  - Continuous Integration Scalability
  - Continuous Integration Flexibility

[skill](#), [ci](#)

From:

<https://www.almbok.com/> - **ALMBoK.com**

Permanent link:

<https://www.almbok.com/ci/ci>

Last update: **2023/05/01 18:16**

