CONTINUOUS INTEGRATION (CI)

Learning Objectives:

By the end of this discussion, students should be able to:

- Understand the concept of Continuous Integration (CI) in software engineering.
- Explain the benefits of CI in modern software development.
- Describe best practices for implementing CI.
- Identify common challenges and how to address them.

Definition:

- **Continuous Integration (CI)** is a software development practice in which developers frequently integrate their code into a shared repository, ideally multiple times a day. Each integration is automatically verified by running tests to detect errors as early as possible.
- enables faster feedback loops, better collaboration, and higher software quality.

KEY COMPONENTS

- Version Control System (VCS)
- Automated Build System
- Automated Testing
- CI Server
- Feedback Mechanism

Version Control System (VCS)

A Version Control System (VCS) manages and tracks changes to the source code. It allows multiple developers to collaborate, merge changes, and revert to previous versions when needed.

Version Control System (VCS)

Examples:

- Git – A widely used distributed VCS that tracks changes and allows developers to work independently before merging changes.
- GitHub, GitLab, Bitbucket – Platforms that host Git repositories and provide collaboration feature

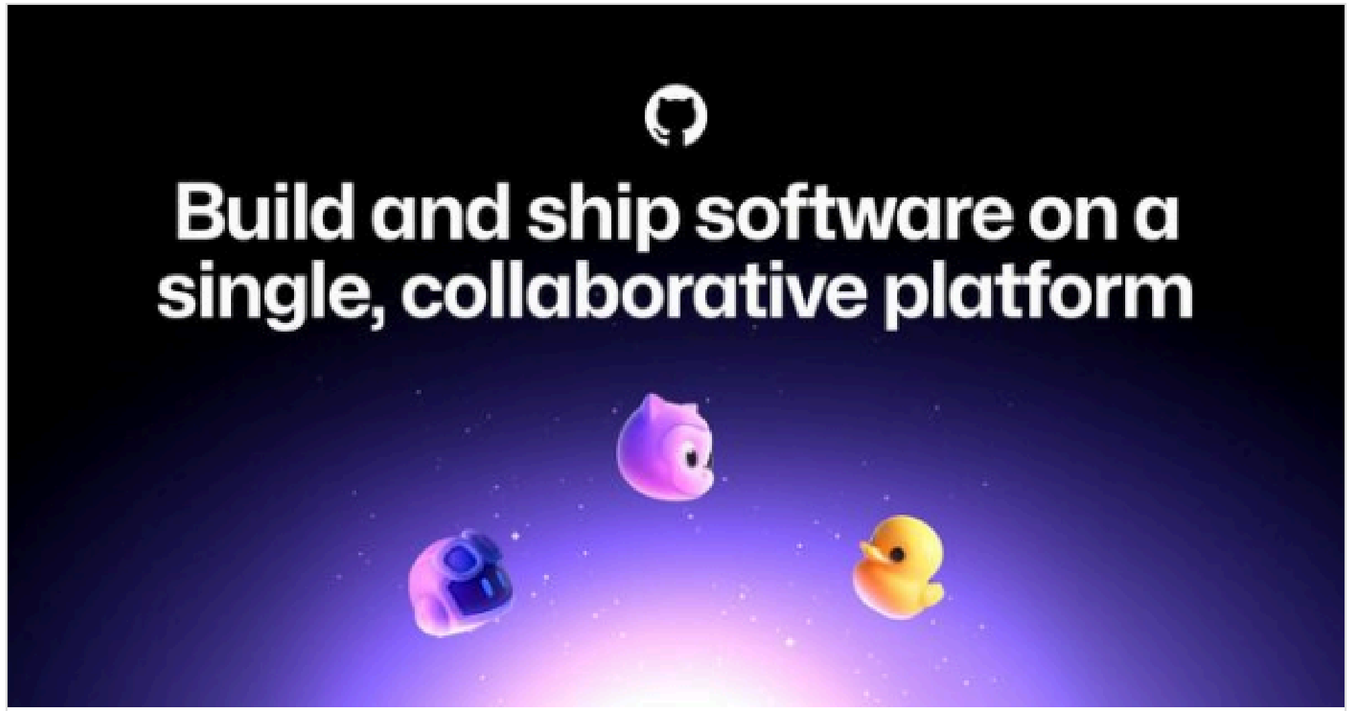
Version Control System (VCS)

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It has features like cheap local branching, convenient staging areas, and multiple workflows.

https://git-scm.com

Version Control System (VCS)



GitHub · Build and ship software on a single, collaborative platform

Join the world's most widely adopted, AI-powered developer platform where millions of developers, businesses, and the largest open source community build software that advances humanity.

GitHub

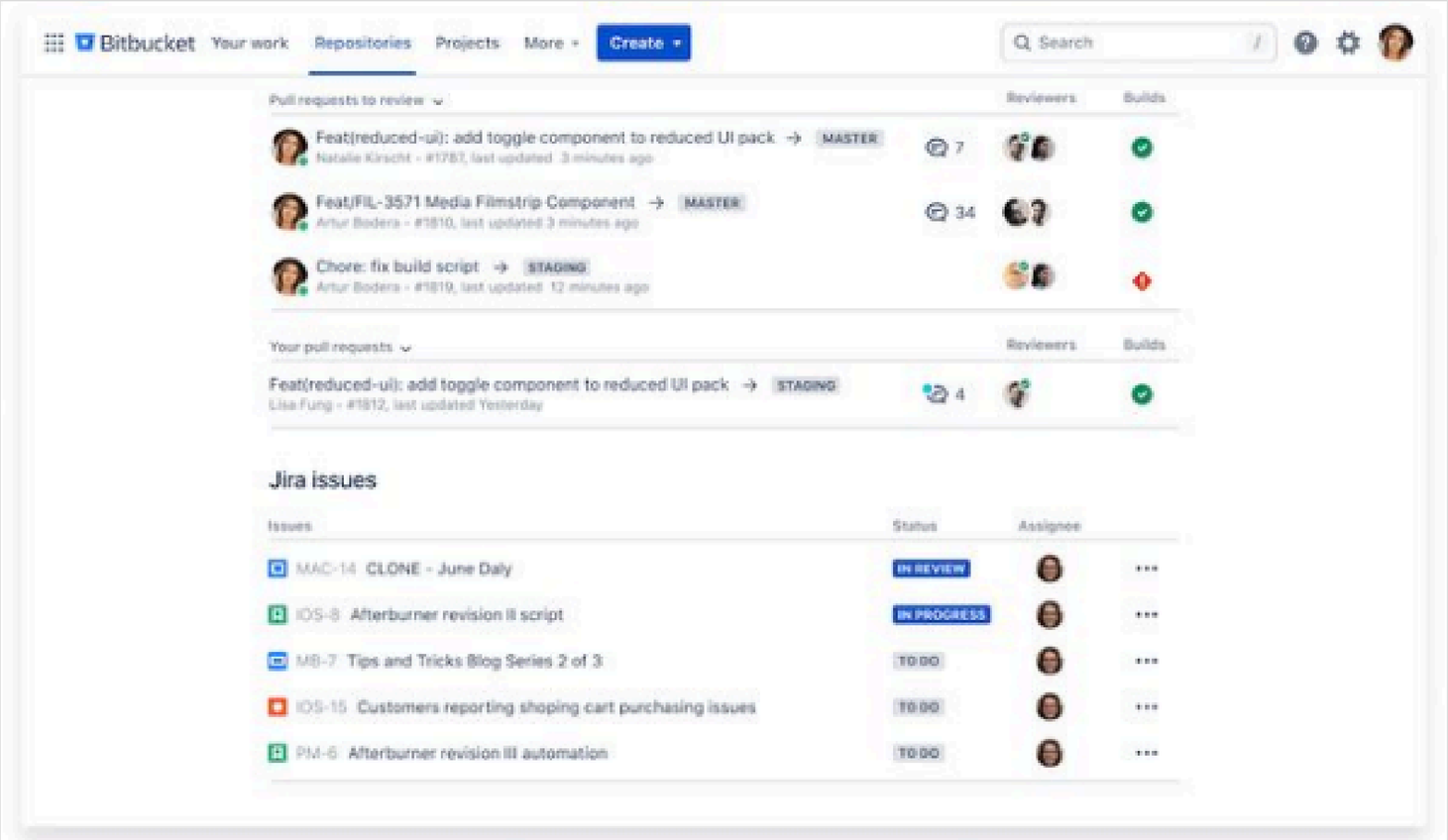
GitHub

https://github.com

Version Control System (VCS)

https://about.gitlab.com

Version Control System (VCS)



https://bitbucket.org/product

Git solution for teams using Jira

Bitbucket Cloud is a Git-based code and CI/CD tool optimized for teams using Jira.

Version Control System (VCS)

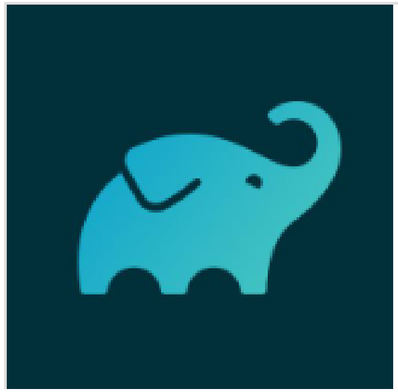
Example in Action:

- A development team uses Git with GitHub to store and manage their project.
- Developers create branches (e.g., feature-login for login functionality).
- They push their changes to GitHub and open a Pull Request (PR) for review.

Automated Build System

An automated build system compiles the source code and ensures that all dependencies are properly included. It is responsible for checking whether the latest changes break the application.

Automated Build System



Gradle Build Tool

Accelerate developer productivity. Gradle helps teams build, automate and deliver better software, faster.

Gradle

https://gradle.org/
(Java-based projects)

Gradle Build Tool accelerates developer productivity

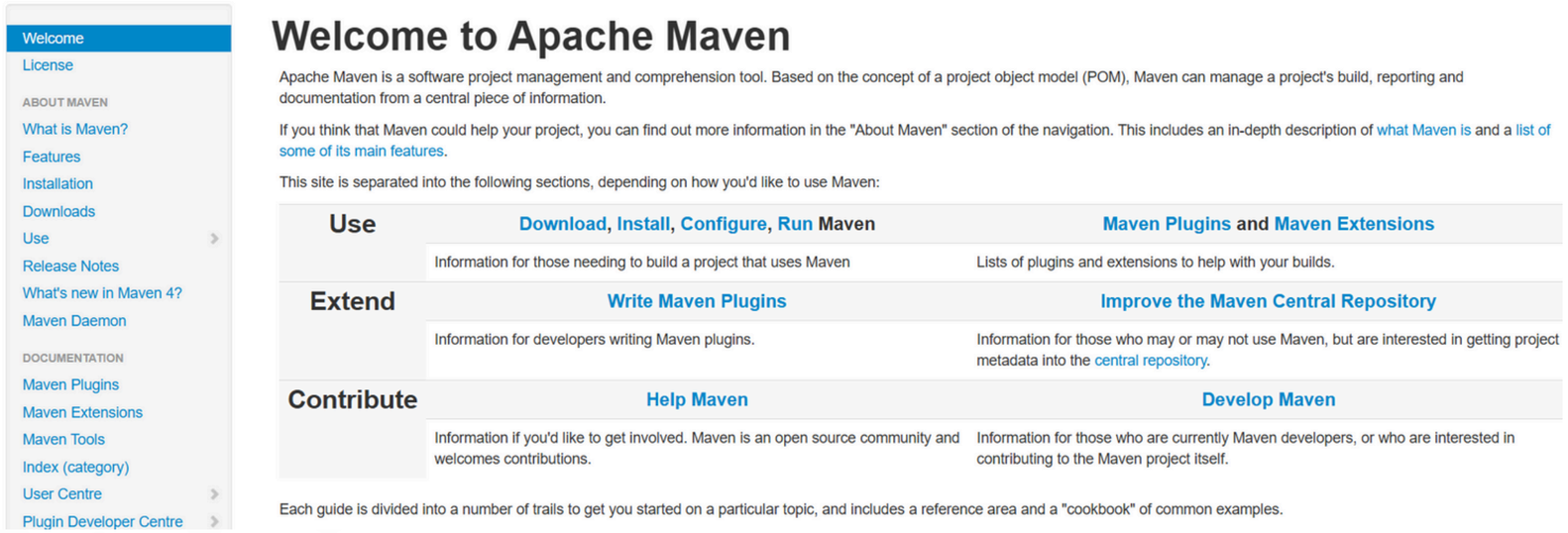
Gradle is the open source build system of choice for Java, Android, and Kotlin developers. From mobile apps to microservices, from small startups to big enterprises, it helps teams deliver better software, faster.

Automated Build System

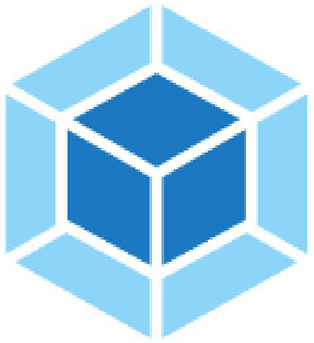
Apache Maven Project
http://maven.apache.org/

Maven™

https://maven.apache.org
(Java, Spring Boot applications)



Automated Build System



webpack

webpack is a module bundler. Its main purpose is to bundle JavaScript files for usage in a browser, yet it is also capable of transforming, bundling, or...

webpack

https://webpack.js.org/
(JavaScript applications)

Automated Build System

Example in Action:

- A developer commits a new feature.
- The CI pipeline triggers a build using Gradle.
- If there's a missing dependency or syntax error, the system fails the build and notifies the developer.

Automated Testing

Automated testing runs various types of tests to identify errors early, before deployment. This includes:

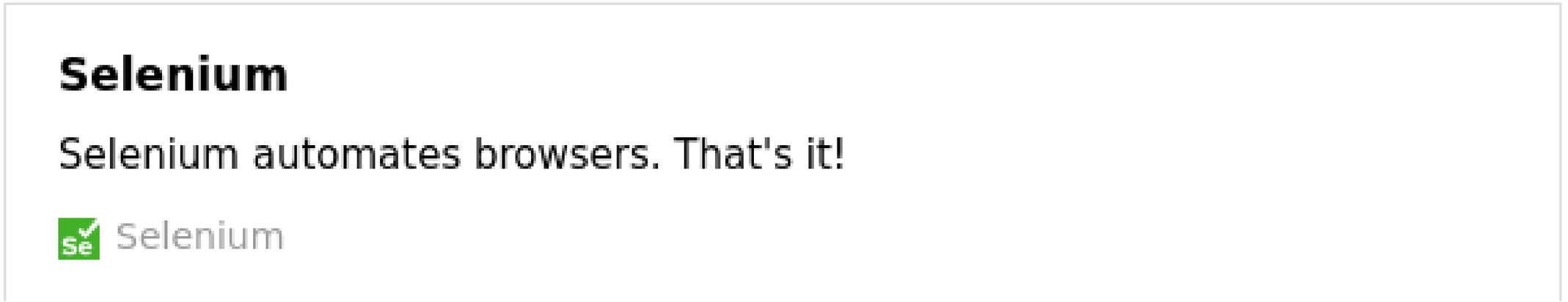
- Unit Tests – Check individual components.
- Integration Tests – Validate interactions between modules.
- Functional Tests – Ensure the application works as expected.

Automated Testing

https://junit.org/junit5/

Unit testing framework.

Automated Testing



https://www.selenium.dev/

Automated browser testing.

Automated Testing

pytest: helps you write better programsThe pytest framework makes it easy to write small, readable tests, and can scale to support complex functional testing for applications and libraries. pytest requires: Python 3.8+ or PyPy3.

Framework for writing unit and integration tests.

Automated Testing

Example in Action:

- A developer updates a function in a banking application.
- Automated tests run to verify that the function does not break other features.
- If a test fails, the developer is notified, preventing bugs from reaching production.

CI Server

A CI server automates the entire CI process by:

- Detecting changes in the repository.
- Running builds and tests automatically.
- Providing feedback to developers.

CI Server

Examples:

- Jenkins – Open-source automation server.
- GitHub Actions – Built-in CI/CD for GitHub.
- Travis CI – Cloud-based CI/CD tool.
- CircleCI – Fast, scalable CI/CD tool.

CI Server

Example in Action:

- A developer pushes code to GitHub.
- GitHub Actions detects the change and triggers a build and test.
- The CI server provides immediate feedback if something breaks.

Feedback Mechanism

A feedback mechanism provides real-time feedback to developers if a build fails or tests break. It helps in:

- Identifying issues quickly.
- Reducing debugging time.
- Keeping the project stable.

Feedback Mechanism

Examples:

- Email notifications – Jenkins sends an email if a build fails.
- Slack integration – GitHub Actions posts updates in a team's Slack channel.
- Dashboards – CircleCI provides real-time build status in a web dashboard.

Feedback Mechanism

Example in Action:

- A developer commits code.
- Jenkins runs the build and detects an issue.
- Jenkins sends an email alert to the developer with logs explaining the failure.

Summary Table of CI Components

CI Component	Purpose	Examples	Example in Action
Version Control System (VCS)	Tracks and manages code changes	Git, GitHub, GitLab, Bitbucket	Developers push changes to GitHub and open a pull request for review.
Automated Build System	Ensures code compiles successfully	Gradle, Maven, Webpack	CI pipeline triggers a build after a developer pushes code.
Automated Testing	Runs unit, integration, and functional tests	JUnit, Selenium, PyTest	A new feature is added, and automated tests verify that it doesn't break other features.
CI Server	Automates builds and tests	Jenkins, GitHub Actions, Travis CI, CircleCI	GitHub Actions detects code changes and runs automated tests.
Feedback Mechanism	Provides real-time feedback to developers	Email alerts, Slack notifications, Dashboards	Jenkins emails developers when a build fails, showing the error logs.

- CI is an essential practice in modern software development.
- It allows teams to integrate code frequently, detect errors early, and deliver stable software faster.
- By using version control, automated builds, testing, CI servers, and feedback mechanisms, teams can maintain high-quality code and improve development efficiency.