# Importance of CI

## Lesson Plan

SKILLS

# Continuous Integration (CI):

Continuous integration (CI) is the practice of continuously integrating code changes into a shared repository, often several times a day. The goal of CI is to catch issues early in the development cycle before they can cause bigger problems later on. CI involves using automated tests to verify that code changes don't break existing functionality or introduce new bugs.

Setting up a robust CI pipeline—especially as part of DevOps and agile development—requires careful planning and configuration, including choosing the right tools, defining build and test workflows and configuring infrastructure. CI pipelines also require regular maintenance to accommodate changes in the code base, dependencies (such as APIs) and infrastructure.

Here’s why CI is so important:

## Earlier, more efficient error detection

CI processes enable teams to address errors early—sometimes within minutes of check-in. Early detection of bugs makes them easier and cheaper to fix.

## Improved team collaboration

Everyone on the team can change code, integrate code changes and identify code incompatibilities and integration errors, simplifying knowledge sharing and improving code and software quality through peer feedback.

## Accelerated software development

Because new code is integrated continuously, teams spend less time integrating and testing large batches of code. And the accelerated feedback loop CI tools offer helps developers iterate and deliver software updates and new products to end users faster.

## Reduced risk in the development process

Frequent code commits mean smaller, more incremental changes that are easier to understand, review and test. This reduces the risk of introducing significant issues into the code base during development.

## Improved Transparency and Accountability

CI provides visibility into the state of the codebase at all times. This transparency helps teams track progress, identify bottlenecks, and hold developers accountable for the quality of their contributions.

## Reduced Costs and Time

By catching and fixing bugs early, CI reduces the time and cost associated with debugging and rework. It also minimizes the risk of costly delays caused by integration issues at the end of a project.

## Faster Feedback

Automated tests run with each integration, providing developers with instant feedback on the impact of their changes. This quick feedback loop allows teams to address issues before they escalate, reducing the risk of introducing defects into the codebase.