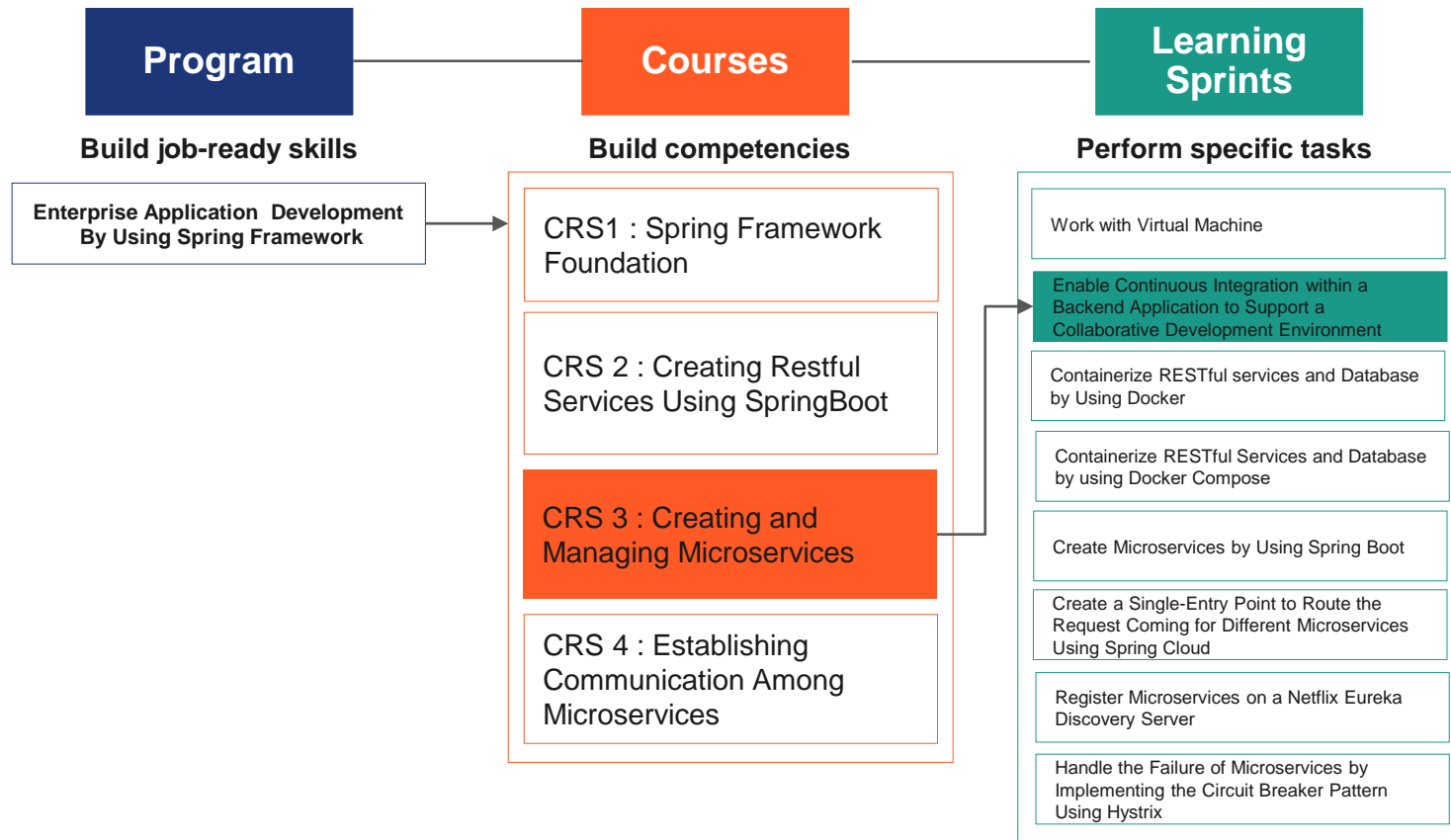


# Backend Program: Course 3: Structure

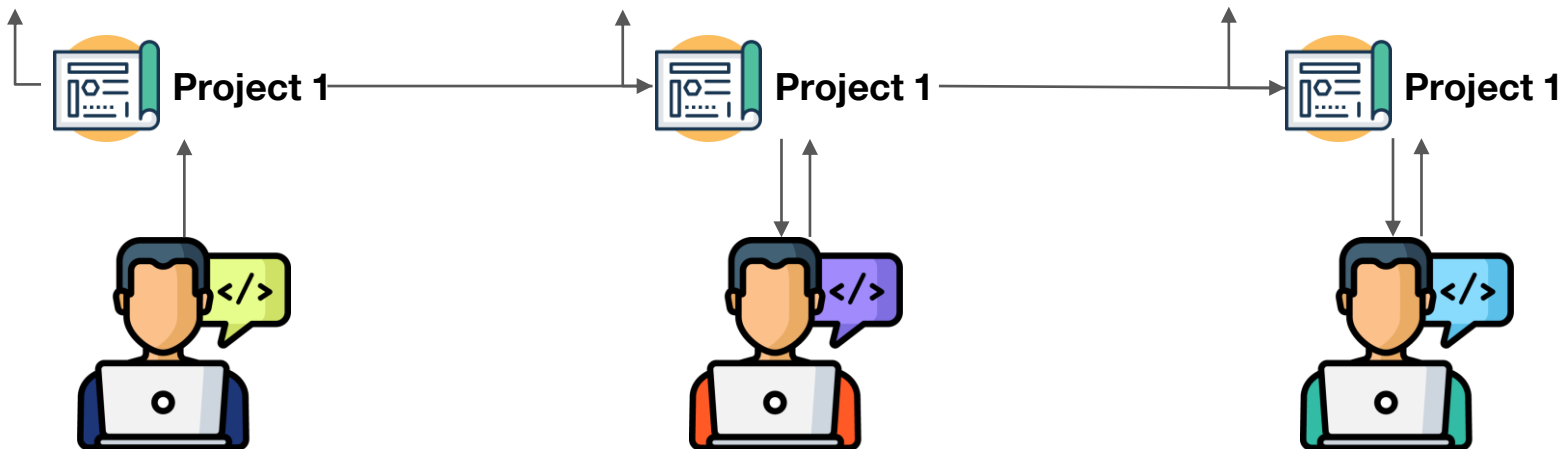


# Software Development Scenario

Jim works on Project 1, and saves his code on a shared Git repository.

Peter also works on Project 1 and saves his code on the same shared Git repository.

Sam is assigned the task of enhancing certain features of Project 1, after Peter and Jim have completed their work.



Is the code of Jim and Peter that Sam picks up from the Git repository bug free ?

Will Sam have to re-run all the test cases in his system ?

Can Sam ensure that the code cloned from the repository are tested and all the test cases have passed?

## Time to Think

- If the number of developers in the team increase, how will each developer ensure that testing is done for all the code?
- Is this going to be a manual process that each developer must do ?
- Do you think this process is time consuming?
- In this process, can defects get injected into the actual code?
- Do integration tests get missed out ?
- Can this process be human error free ?
- Can there be a pipeline that can run test cases and notify defects?



# Enable Continuous Integration Within a Backend Application to Support a Collaborative Development Environment



# Learning Objectives

- Describe Continuous Integration (CI)
- List the benefits and best practices of CI
- Set up the build process of CI
- Configure CI to run the jobs listed in CI file

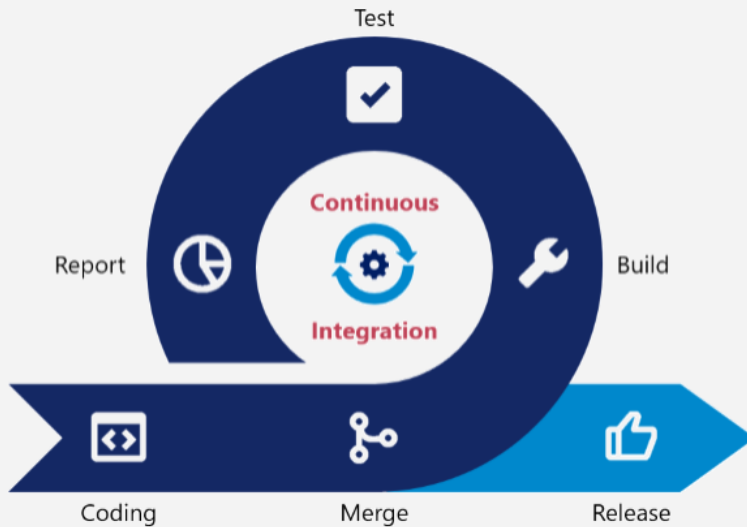


# Continuous Integration (CI)

# What is Continuous Integration ?

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day.

- Each check-in is then verified by an automated build, allowing teams to detect problems early.
- By integrating regularly, you can detect errors quickly, and locate them more easily.
- Continuous Integration is a way to increase the code quality without putting an extra burden on the developers.
- Tests and checks of your code are handled on a server and automatically reported back to you.



<https://www.automation-consultants.com/continuous-integration-2>

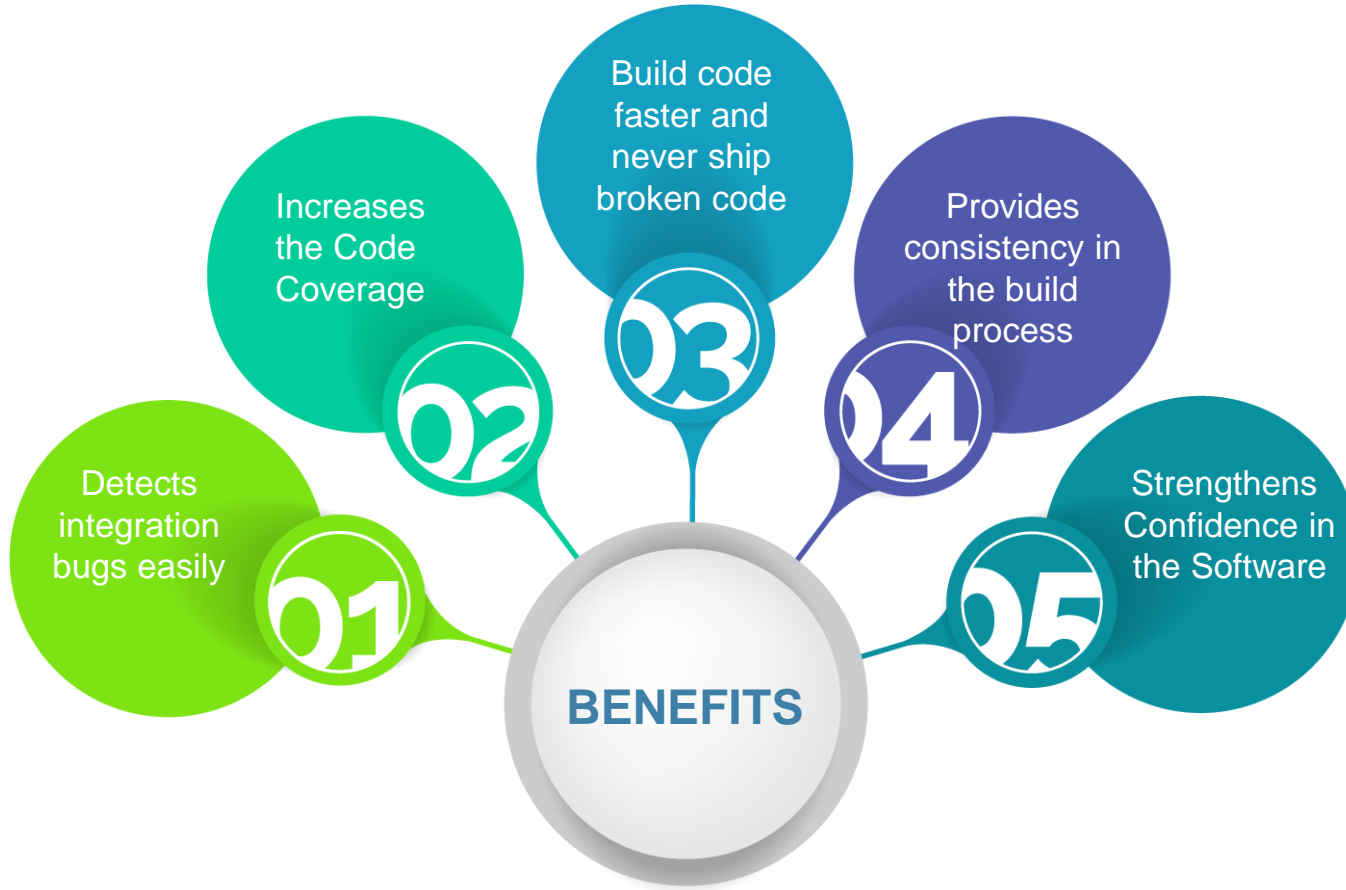
# Use of Continuous Integration

- Runs your tests in the real world
- Increases your code coverage
- Deploys your code for production
- Triggers build and compilation process
- Parallel build supports for test and build processes on multiple machines
- Decreases the code review time





# Benefits of Continuous Integration



# Benefits of Continuous Integration (contd.)

- Say goodbye to **long and tense integrations**
- Increase visibility enabling greater communication
- **Catch issues early** and nip them in the bud
- **Spend less time debugging** and more time adding features
- Build a solid foundation
- Stop waiting to find out if your code's going to work
- Reduce integration problems allowing you to deliver software more rapidly

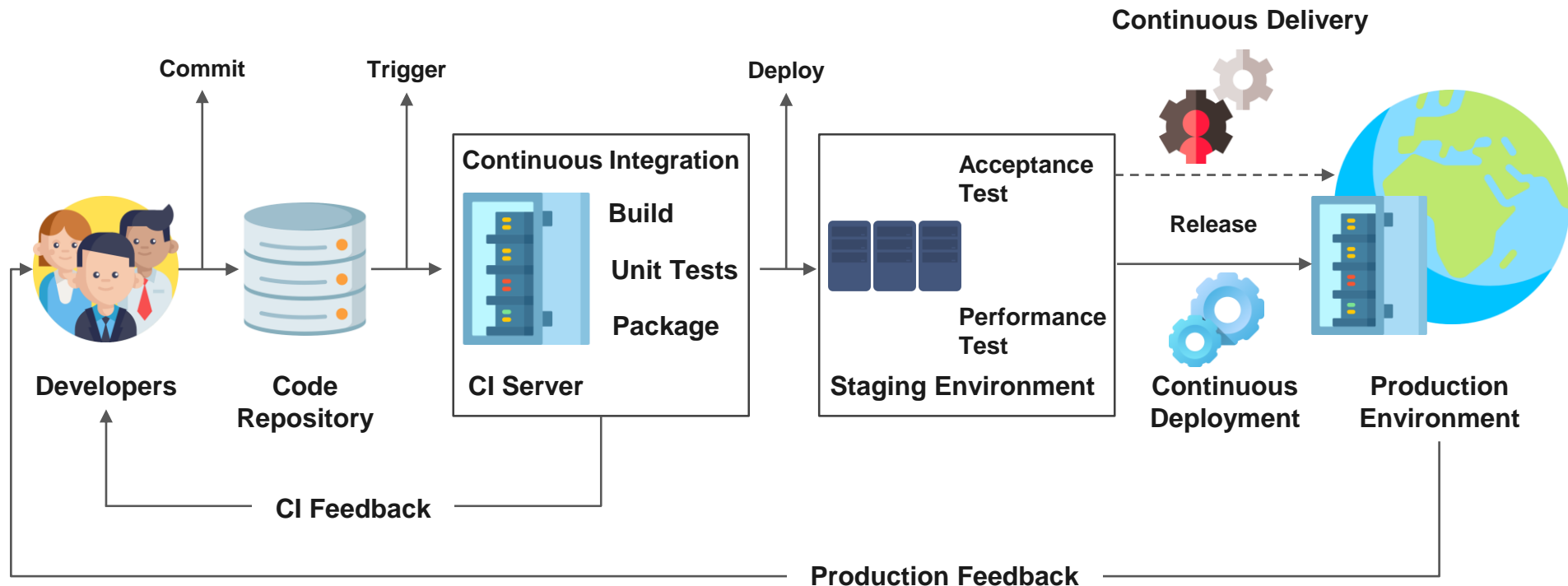
# Continuous Integration - Best Practices

- Maintains a single source repository
- Automates the build
- Allows self-testing of the build
- Builds are faster with the most recent changes
- Makes it easy for anyone to get the latest executable version
- Allows viewing of the latest build

**“Continuous Integration doesn’t get rid of bugs,  
but it does make them dramatically easier to find  
and remove.”**

— Martin Fowler, Chief Scientist, ThoughtWorks

# Process of Continuous Integration



# Process of Continuous Integration (contd.)

- Developers first check out code into their private workspaces and then commit the changes into the repository.
- The CI server monitors the repository and checks out the changes when they occur.
- The CI server builds the system and runs the unit and integration tests.
- After that the CI server releases the deployable artifacts for testing.
- The CI server assigns a build label to the version of the code it builds
- The CI server informs the team of the successful build
- If the build or tests fails, the CI server alerts the team.
- The team fixes the issue at the earliest opportunity.
- The team fixes the issue at the earliest opportunity and continues to integrate and test throughout the project

# Setting Up of Continuous Integration

- Create a Spring Application
- Create a repository in the Gitlab
- Setting up the CI with SCM (Gitlab)
- Configure CI
- Review the CI build

# Configure Continuous Integration

- GitLab offers a continuous integration service. To configure CI, we need to add a `.gitlab-ci.yml` file to the root directory of our repository, next to `POM.xml`.
- This file defines a set of jobs in the project
- This file is used by GitLab Runner to manage your project's jobs.
- In GitLab CI, Runners run the code defined in `.gitlab-ci.yml` file. They are isolated (virtual) machines that pick-up jobs through the coordinator API of the GitLab CI.
- A **Runner** can be specific to a certain project or serve any project .
- A **Runner** that serves all projects is called a **shared Runner**.





## Configure CI (contd.)

```
image: 'maven:latest'
stages:
- test
- build
maven-test:
  stage: test
  script: "mvn test"
maven-build:
  stage: build
  script: 'mvn package'

artifacts:
  paths:
    - target/*.jar
```

- **image: 'maven:latest'** - This is related to the docker which we will cover in future videos.
- **stages:** This element defines the lifecycle of our build. We associate each job with our stage. Next stage is initiated when the previous stage is completed. In **stages** we can have various set jobs but here we are using two jobs the first is **test** and second is **build**.
- **maven-test:** refers to job **stage**, by default it is test and refers to shell **script** to be executed for this job by the runner. We are using image of Maven.
- **artifacts:** To persist the executable jar and share it across jobs.

# Ways to Review the CI Build

Status	Job	Pipeline	Stage	Name
 failed	#32948  master  aca8e201	#7702 by 	test	maven-test
 passed	#549  master  780cb4e3	#319 by 	build	maven-build
 passed	#548  master  780cb4e3	#319 by 	test	maven-test
 passed	#547  master  363159bb	#318 by 	build	maven-build
 passed	#546  master  363159bb	#318 by 	test	maven-test
 skipped	#545  master  ff3c6236	#317 by 	build	maven-build

# Quick Check

What steps are in Continuous Integration?

- a. Compilation
  - b. Unit Tests
  - c. Code quality Checks
  - d. Build
- 
- 1. a and b
  - 2. a,b,c and d
  - 3. c and d
  - 4. d



# Quick Check : Solution

What are the steps of Continuous Integration?

- a. Compilation
  - b. Unit Tests
  - c. Code quality Checks
  - d. Build
- 
- 1. a and b
  - 2. **a,b,c and d**
  - 3. c and d
  - 4. d



# Continuous Integration of Customer Application

In the previous session we created a Customer Boot Application, let us do a continuous integration of our application using the GitLab ci tool at all stages of the build and add jobs.

**DEMO**



# Key Takeaways

- Continuous Integration (CI)
- CI Benefits and Best Practices
- Understand CI Build Process
- Configure CI



Thank you!