

# Software Engineering Processes

Course Code: XB\_0089

---

Fernanda Madeiral & Claudia Raibulet

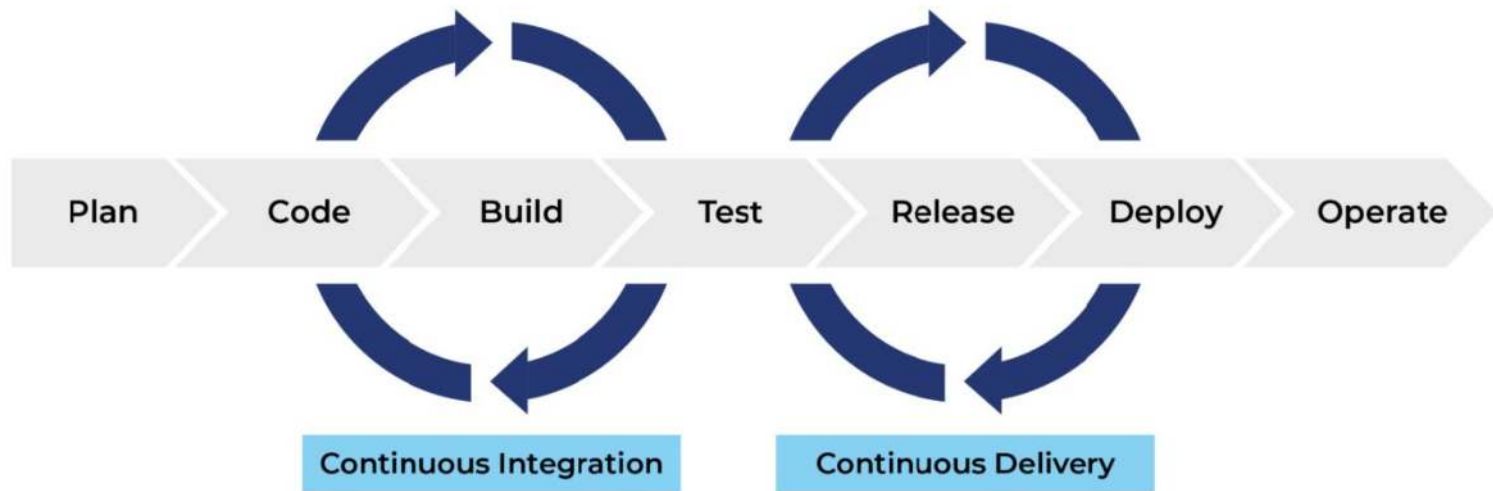
Lecture: CI/CD



Bachelor in Computer Science, 2023/2024

# CI/CD

---



# Continuous integration

---

Each time a developer performs a change to the project's core repository main branch, an executable version of the system is built and tested

# Continuous delivery and deployment



Continuous delivery: A simulation of the product's operating environment is created and the executable software version is tested

Continuous deployment: A new release of the system is made available to users every time a change is made to the project's core repository main branch

# Examples of CI/CD tools/services

---



Travis CI



**Jenkins**



# GitHub Actions

---

# GitHub Actions

---

Main goal:

- To **automate** software development tasks and processes

# GitHub Actions: workflow

---

A workflow is a configurable automated process that will run one or more jobs

Workflows are defined in a `.yaml` file checked in to your repository

Workflows are defined in the `“.github/workflows”` directory in the repository



# Workflow basics

---

A workflow must contain the following basic components:

- One or more **events** that will trigger the workflow
- One or more **jobs**, each of which will execute on a runner machine and run a series of one or more **steps**
- Each step can either **run** a script defined by the developer or a predefined action

# GitHub Actions: example



```
name: hello-world
on: push
jobs:
  my-job:
    runs-on: ubuntu-latest
    steps:
      - name: My step
        run: echo "Hello World!"
```

# GitHub Actions: example

```
name: hello-world
```

```
on: push
```

Event

```
jobs:
```

```
  my-job:
```

Job

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: My step
```

```
        run: echo "Hello World!"
```

Step

# GitHub Actions: example

```
name: hello-world
on: push
jobs:
  my-job:
    runs-on: ubuntu-latest
    steps:
      - name: My step
        run: echo "Hello World!"
```

The screenshot displays the GitHub Actions interface for a repository named 'hello-world'. The top navigation bar includes links for Code, Issues, Pull requests, Actions (highlighted with a red box), Projects, Wiki, Security, and a menu icon. Below the navigation bar, the workflow run is titled 'Create hello-world.yml #1' with a green checkmark icon. To the right of the title are buttons for 'Re-run all jobs' and a three-dot menu. The left sidebar contains a 'Summary' section with links for 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The 'Jobs' section lists a single job, 'my-job', which is marked as successful with a green checkmark. The right panel shows the details for the 'my-job', indicating it 'succeeded' and took '6 minutes ago in 0s'. It includes a 'Search logs' input field and a list of steps: 'Set up job' (1s), 'My step' (0s), and 'Complete job' (0s). The 'My step' is expanded, showing a log with two lines: '1 Run echo "Hello World!"' and '4 Hello World!'.

# GitHub Actions: example

```
name: Java CI with Maven
```

```
on: push
```

```
jobs:
```

```
  my-ci:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: Check out code
```

```
        uses: actions/checkout@v4
```

```
      - name: Set up JDK 17
```

```
        uses: actions/setup-java@v3
```

```
        with:
```

```
          java-version: '17'
```

```
          distribution: 'temurin'
```

```
      - name: Compile with Maven
```

```
        run: mvn compile
```

Existing/predefined actions

This is a command line you would do in the terminal

# GitHub Actions: example



```
name: Java CI with Maven
on: push
jobs:
  my-ci:
    runs-on: ubuntu-latest

    steps:
      - name: Check out code
        uses: actions/checkout@v4

      - name: Set up JDK 17
        uses: actions/setup-java@v3
        with:
          java-version: '17'
          distribution: 'temurin'

      - name: Compile with Maven
        run: mvn compile
```

What does this workflow do?

Compile a Java project with  
Maven

# GitHub Actions: example

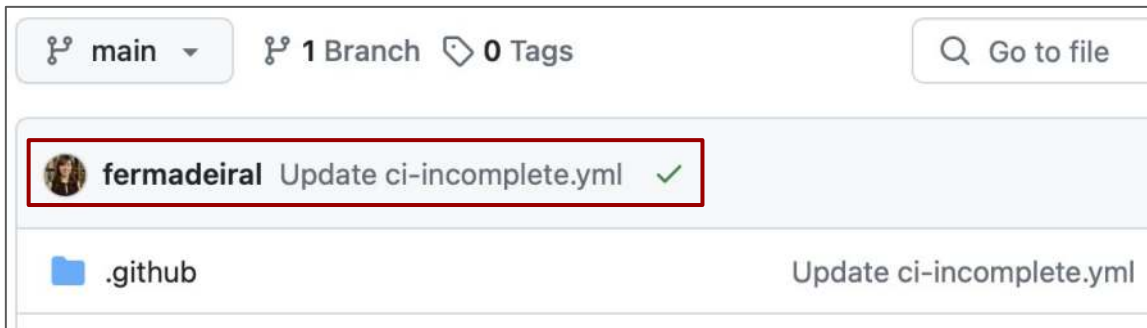
A commit has been pushed to  
the repository...

```
name: Java CI with Maven
on: push
jobs:
  my-ci:
    runs-on: ubuntu-latest

    steps:
      - name: Check out code
        uses: actions/checkout@v4

      - name: Set up JDK 17
        uses: actions/setup-java@v3
        with:
          java-version: '17'
          distribution: 'temurin'

      - name: Compile with Maven
        run: mvn compile
```



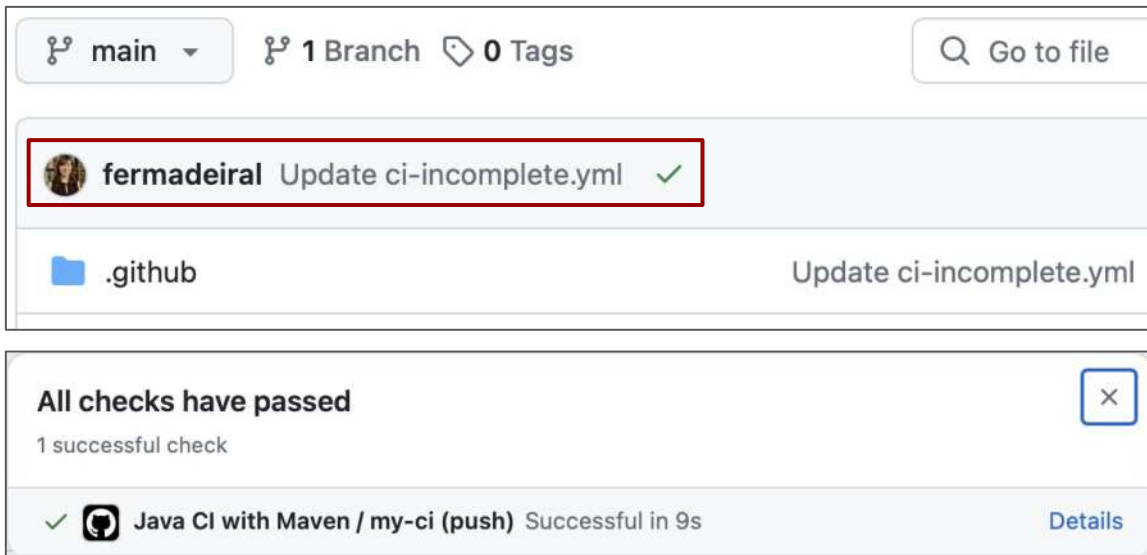
# GitHub Actions: example

```
name: Java CI with Maven
on: push
jobs:
  my-ci:
    runs-on: ubuntu-latest

    steps:
      - name: Check out code
        uses: actions/checkout@v4

      - name: Set up JDK 17
        uses: actions/setup-java@v3
        with:
          java-version: '17'
          distribution: 'temurin'

      - name: Compile with Maven
        run: mvn compile
```



The screenshot shows the GitHub Actions interface for a workflow run. At the top, it displays the current branch as 'main', 1 branch, and 0 tags. A search bar labeled 'Go to file' is on the right. Below this, a list of workflow runs is shown. The first run, by user 'fermadeiral', is titled 'Update ci-incomplete.yml' and has a green checkmark indicating it passed. This run is highlighted with a red rectangular box. Below the workflow runs, a folder icon labeled '.github' is shown, with the file 'Update ci-incomplete.yml' listed next to it. At the bottom, a summary box states 'All checks have passed' with a close button (X). Below this, it says '1 successful check'. A specific check is listed: 'Java CI with Maven / my-ci (push)' which was 'Successful in 9s'. A 'Details' link is provided for this check.

main 1 Branch 0 Tags

Go to file

fermadeiral Update ci-incomplete.yml ✓

.github Update ci-incomplete.yml

All checks have passed

1 successful check

✓ Java CI with Maven / my-ci (push) Successful in 9s Details



# GitHub Actions: example

```
name: Java CI with Maven
on: push
jobs:
  my-ci:
    runs-on: ubuntu-latest

    steps:
      - name: Check out code
        uses: actions/checkout@v4

      - name: Set up JDK 17
        uses: actions/setup-java@v3
        with:
          java-version: '17'
          distribution: 'temurin'

      - name: Compile with Maven
        run: mvn compile
```

The screenshot displays the GitHub Actions interface for a repository named 'Java CI with Maven'. The 'Actions' tab is selected, showing a workflow run titled 'Update ci-incomplete.yml #2' with a green checkmark indicating success. Below the title, there are links for 'Summary', 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The 'Jobs' section shows a single job named 'my-ci' with a green checkmark. A detailed view of the 'my-ci' job is shown on the right, listing the steps: 'Set up job', 'Check out code', 'Set up JDK 17', 'Compile with Maven', 'Post Set up JDK 17', 'Post Check out code', and 'Complete job', all marked with green checkmarks. The job status is 'succeeded 2 minutes ago in 9s'.

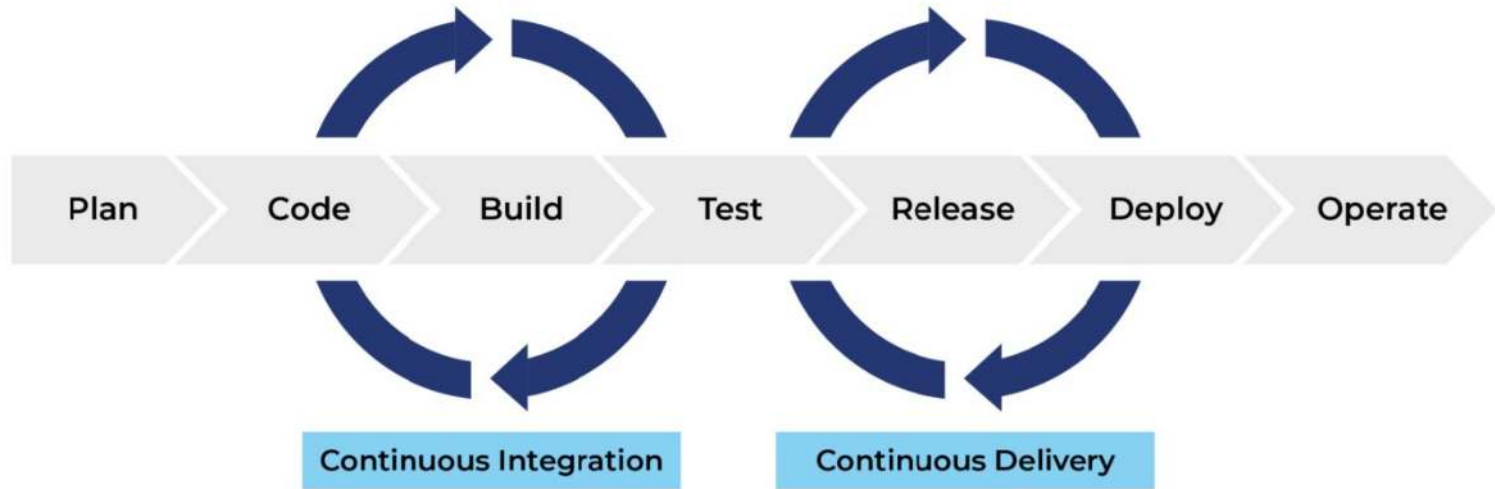
# Build tools: examples

---

- For Java:
  - Maven
  - Gradle
  - Ant
- For Python:
  - PyBuilder
  - Poetry

# CI/CD

---



Is it only about testing and delivering?

# Software quality checks

---

- Executability and functionality are covered by the basic checks (build and tests).
- What more could we check and how?

# Static analysis tools: examples

---

- For Java:  
Checkstyle  
Spotbugs  
PMD  
SonarQube
- For Python:  
Pylint  
Pyflakes  
pycodestyle

TODO



# Reading

---

## Exam material:

- Ian Sommerville, “Engineering Software Products”, 2020: Chapter 10 - “DevOps and Code Management”, Sections 10.2 and 10.3

## Recommended material:

- GitHub Actions documentation:  
<https://docs.github.com/en/actions>

# Takeaways?

---

