



# CI / CD Pipeline

Let's define a CI/CD Pipeline



# CI/CD

In this context CI/CD only covers: **Continuous Integration / Continuous Delivery**

*Not Continuous Deployment* - as this requires a very strong ability to get Continuous Delivery right first, and the ability to rollback or roll forward, whatever the case might be.

**Continuous Integration:** is a *practice* which the code is continually integrated and each commit is verified by an automated build, allowing problems detection early.

**Continuous Delivery:** is a *practice* of delivering every change to a **production-like** environment and assurances are provided through rigorous automated testing



# Define the Principle of CI/CD first

What is a principle?

With Google's help

## Dictionary

principle



## principle

*/ˈprɪnsɪp(ə)l/* 

*noun*

1. a fundamental truth or proposition that serves as the foundation for a system of belief or behaviour or for a chain of reasoning.  
"the basic principles of justice"  
*synonyms:* [truth](#), [proposition](#), [concept](#), [idea](#), [theory](#), [postulate](#); [More](#)
2. a general scientific theorem or law that has numerous special applications across a wide field.



# Principle of CI/CD

Following Agile thinking, a CI/CD pipeline is made up of:

0. People

1. Practices

2. Tools

3. Processes



# Principle of CI/CD

The previous list shape the following CI/CD Principle

## **People and Practices *over* Tools and Processes**

This doesn't mean that Tools and Processes are not or less important.

This means that People and Practices will put the foundation, thinking and selection of these Tools and Processes.

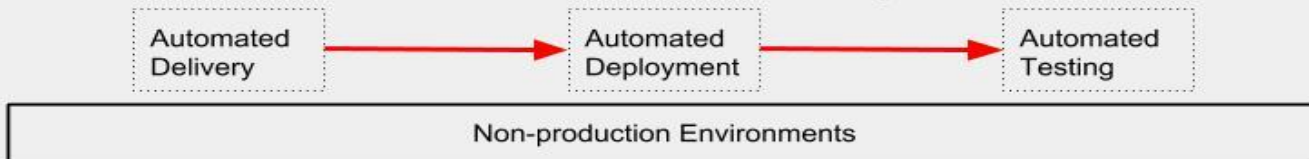


# CI/CD

## Continuous Integration



## Continuous Delivery





## CI/CD

On top of the Practices, the team would want to have a team agreement. For example:

- to not have snowflake pipelines
- to have and define test coverage baseline
- to strive for a x-functional team
- when fixing a bug - write unit test first!
- to create and maintain living documents



# CI/CD

While teams now have:

0. The principles of Continuous Integration and Continuous Delivery

1. Team Agreements

Teams will still need to:

- Define
- Build and
- Maintain

their own CI/CD pipeline as each team will have different requirements and set up.

For example, a JavaScript team will have different set up to a Java or Java and JavaScript team.





# CI/CD

Organisational governance, and architecture, are still providing direction and guidance.

