Pro ware ness | we on.

Continuous Delivery

Visual Studio Team Foundation Server 2013

Subjects

- What is Continuous Delivery?
- Continuous Delivery vs Continuous Deployment
- Building a release pipeline in Visual Studio Team Foundation Server 2013

Agile Manifesto

Agile Manifesto

Our highest priority is to satisfy the customer

through early and continuous delivery

of *high-quality and* valuable software

Cycle time

… **reduce** the time it takes from

deciding to make a change

to having it available to users

Feedback

Pro ware ness | we on.

Continuous Delivery

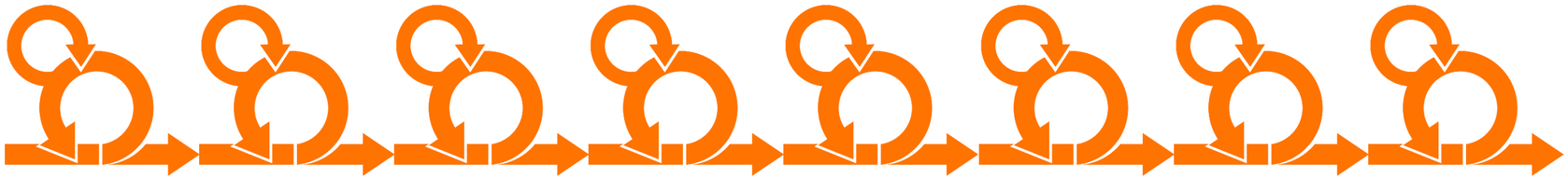
…a set of **practices** and **principles** aimed at,

building, testing, and releasing software

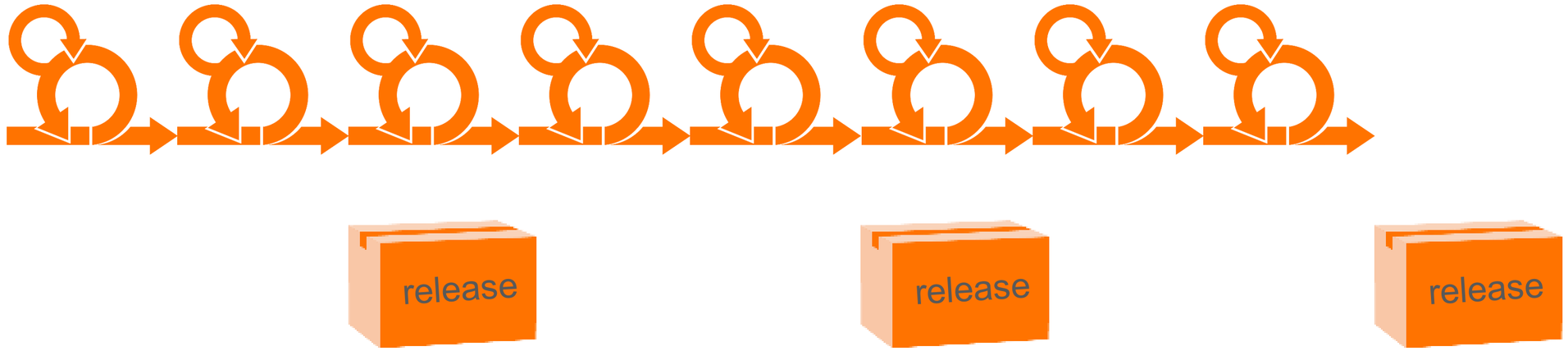
faster and more **frequently**.

Pro ware ness | we on.

The Last Mile



The Last Mile



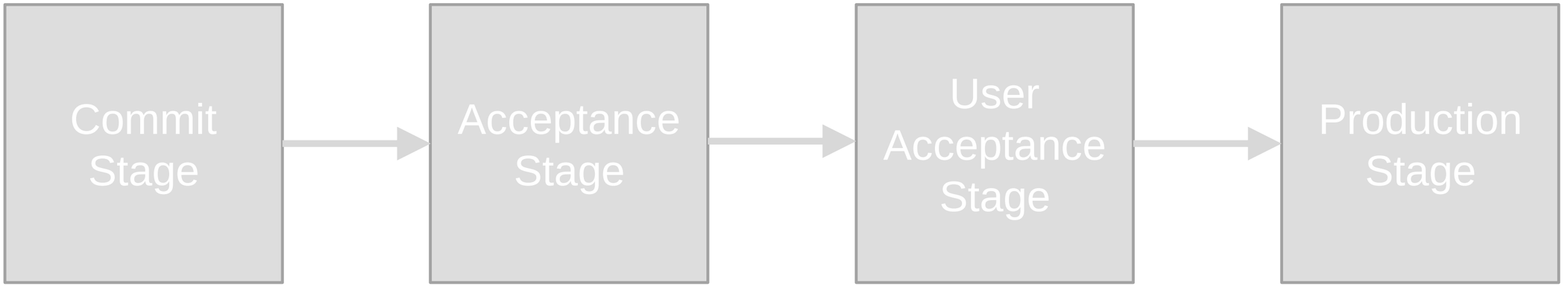
- Acceptance Tests
- Performance Tests
- User Acceptance Tests
- Manual Deployment on Environments

Deployment Pipeline

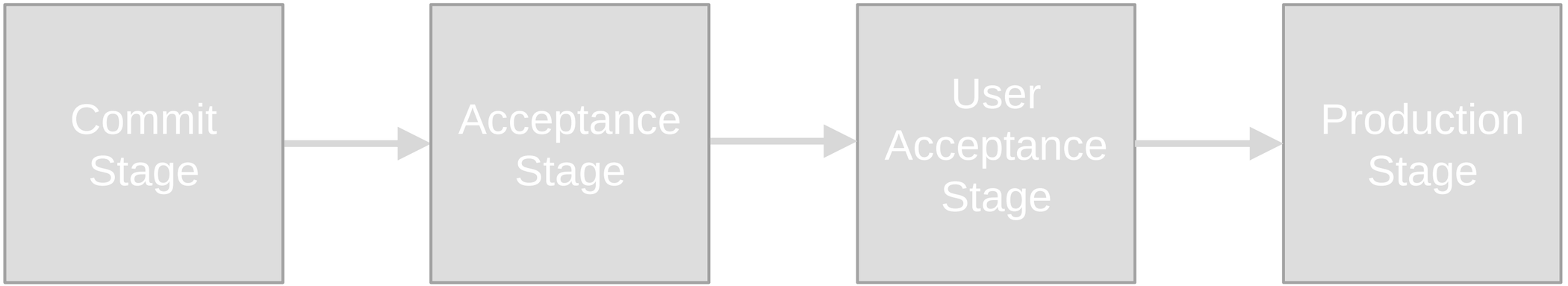
Deployment Pipeline

- Stages
- Steps
- Instances

Deployment Pipeline



Deployment Pipeline



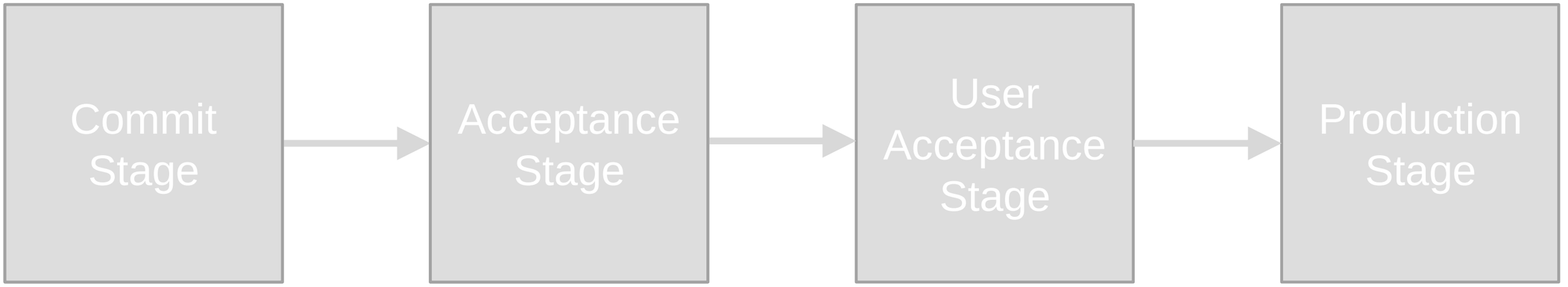
- Build
- Unit Test
- Code Analysis

- Acceptance Tests
- Performance Tests

- UAT
- Exploratory Testing

- Deploy to production
- Smoke-Test

Deployment Pipeline



- Build
- Unit Test

- Deploy to TST
- Acceptance Tests

- Deploy to ACC
- Exploratory Testing

- Deploy to PRD

Principles

Create a **repeatable**, **reliable**

process for **releasing** software

Principles

Automate almost everything

Principles

Keep everything in **Version Control**

Principles

If it **hurts**,

do it more **frequently**,

and bring the pain **forward**

Principles

Build **Quality** in

Principles

Done means **Released**

Principles

Everybody is **responsible**

for the **delivery** process

Principles

Continuous **Improvement**

Practices

Only **build** your binaries **once**

Practices

Deploy the **same way**

to **every environment**

Practices

Smoke-Test your deployments

Practices

Deploy into a **copy** of **production**

Practices

Each **change**

should **propagate** through

the pipeline **instantly**

Practices

If **any part** of the pipeline **fails**,

stop the line

Pro ware ness | we on.

MELVIN NIJHOLT

Agile ALM Consutant

m.nijholt@prowareness.nl