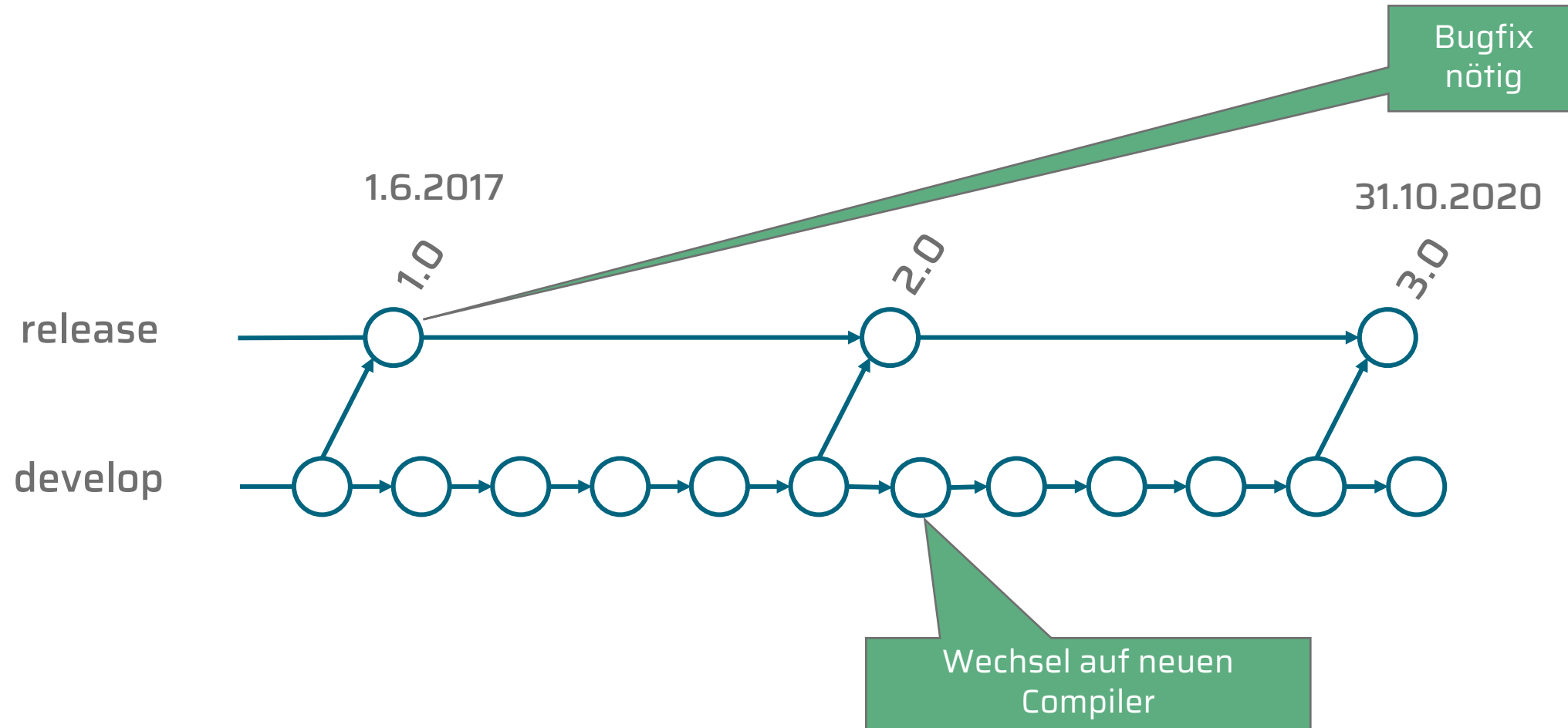


Versionierte Toolchains für reproduzierbare CI

Inkompatible Änderungen erlauben und
dennoch alte Stände nutzbar halten

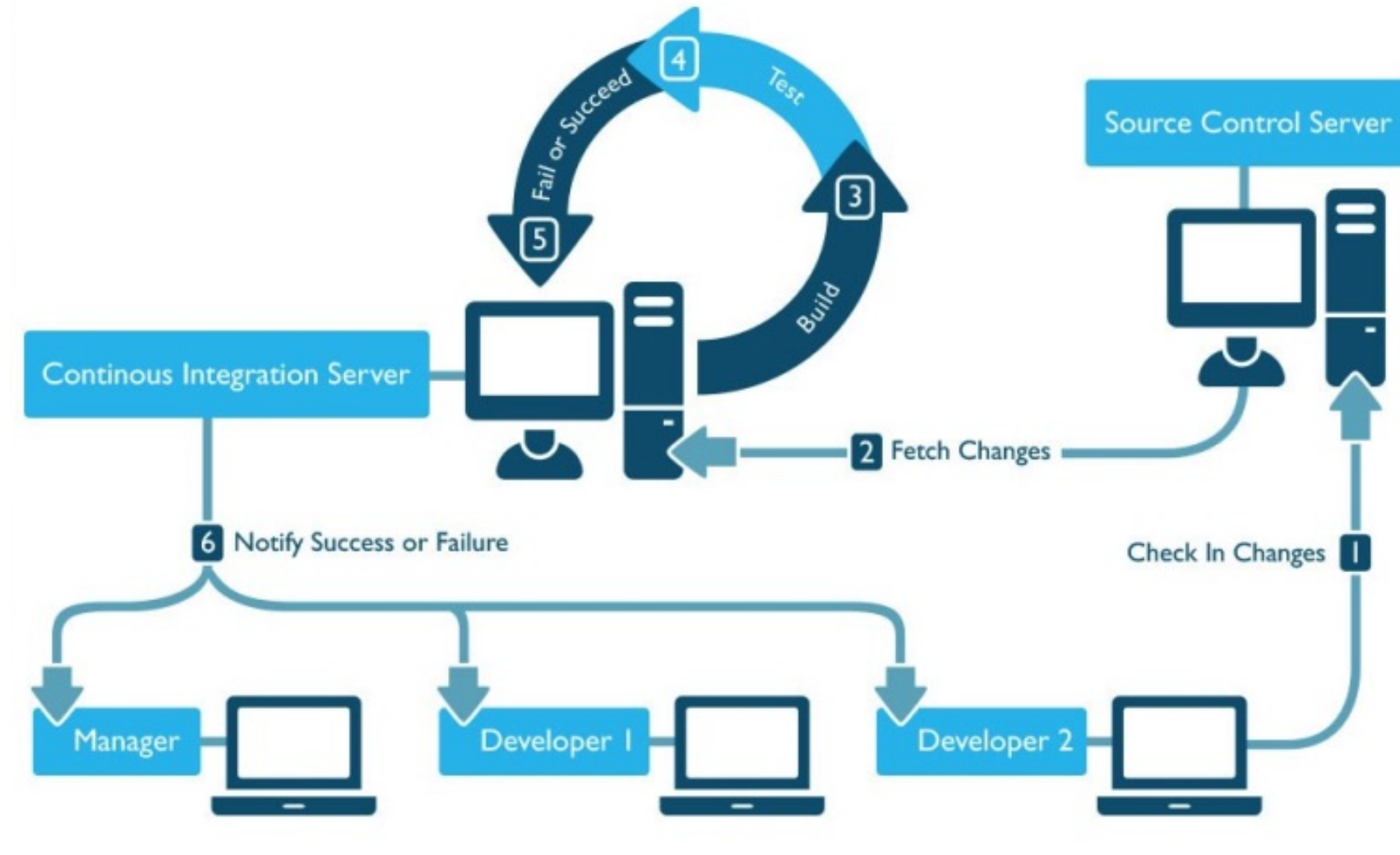
Daniel Penning, embeff GmbH

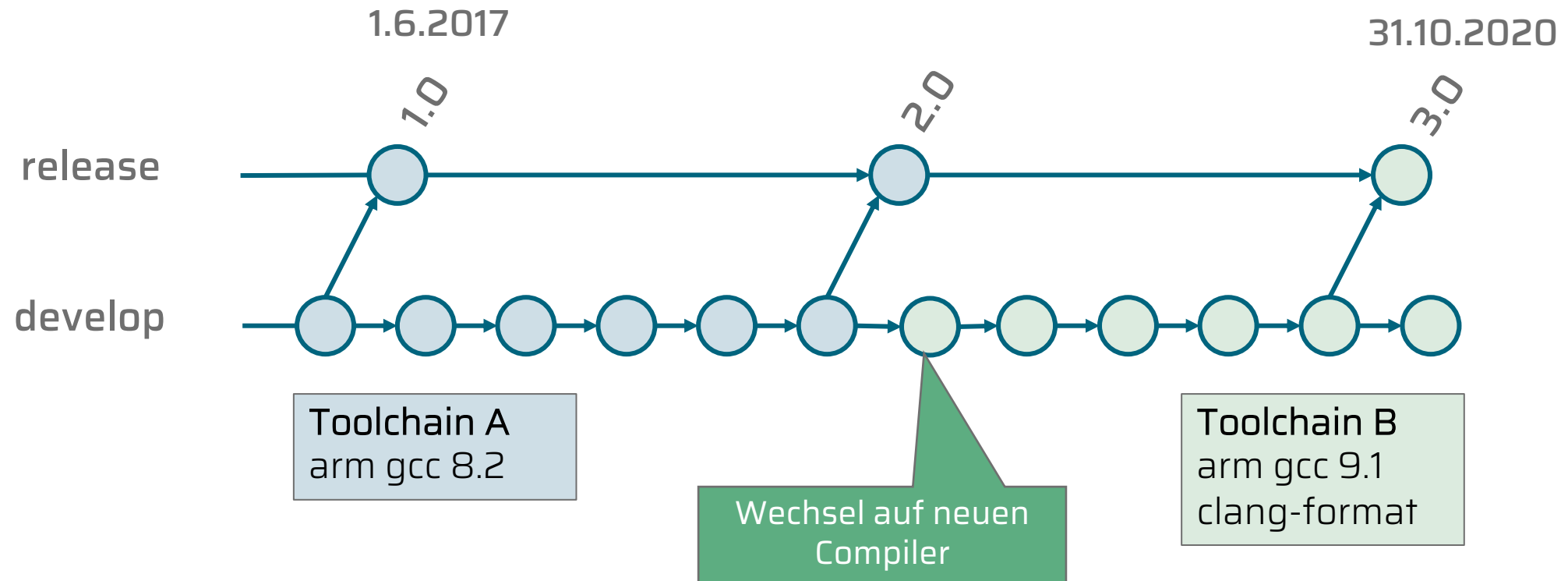
Was heißt Reproduzierbarkeit?





Zusammenhang mit Continuous Integration





Versionierte Toolchains mit Containern

“A Docker **container** image is a lightweight, **standalone**, executable package of software that **includes everything needed to run an application**: code, runtime, system tools, system libraries and settings.”

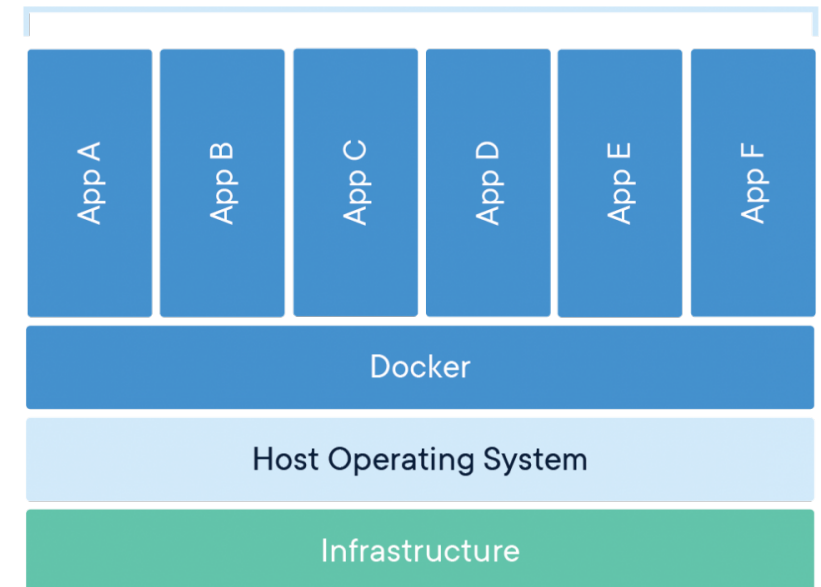
Toolchain A arm gcc 8.2

```
# Dockerfile A
FROM ubuntu:18.04
RUN apt-get install git
RUN apt-get install gcc-arm-none-eabi-8.2.0
```

Toolchain B arm gcc 9.1 clang-format

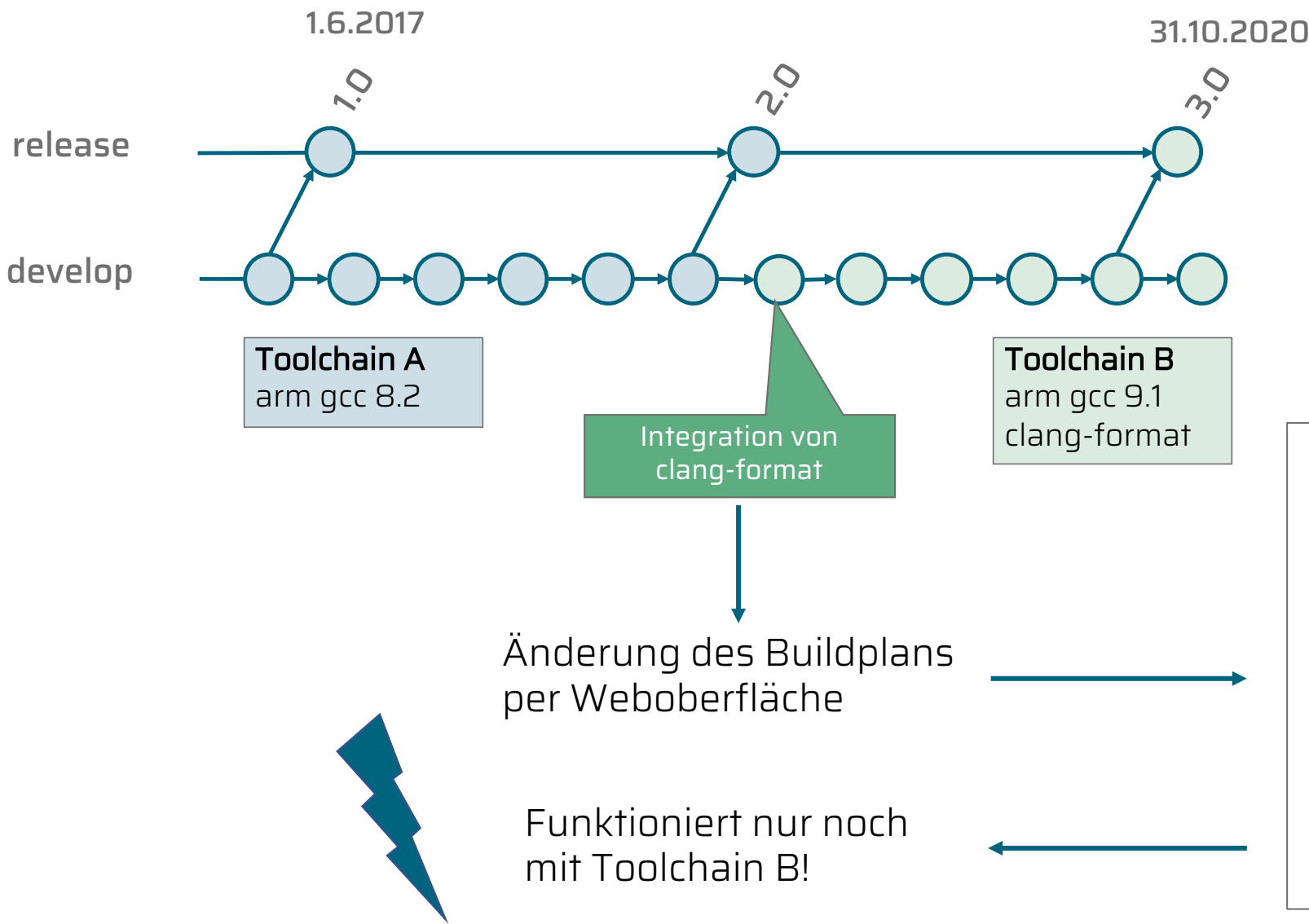
```
# Dockerfile B
FROM ubuntu:18.04
RUN apt-get install git clang-format
RUN apt-get install gcc-arm-none-eabi-9.1.0
```

Containerized Applications



<https://www.docker.com/resources/what-container>

Die Build-Anleitung



Source Code Checkout

Script
clang-format

Script
Build with arm-gcc

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

Versionierte Build-Anleitung

“A Jenkinsfile is a text file that contains the definition of a Jenkins Pipeline and is checked into source control.”

<https://www.jenkins.io/doc/book/pipeline/jenkinsfile/>



Toolchain A

arm gcc 8.2

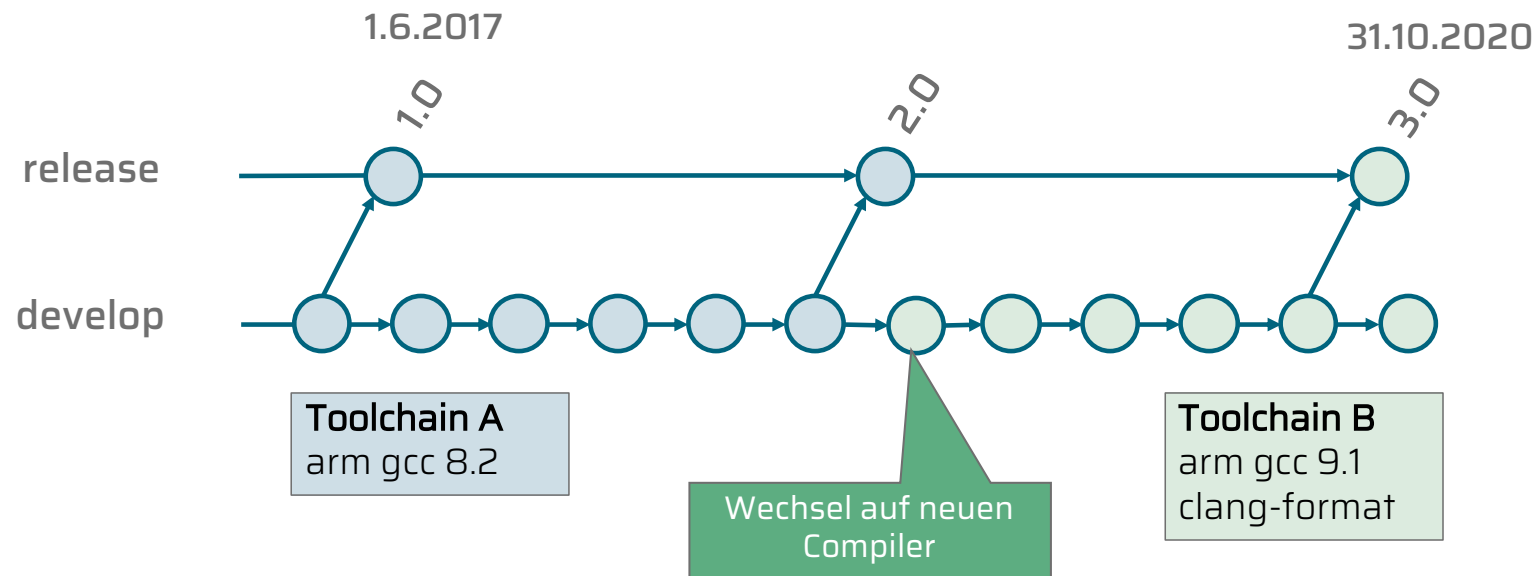
```
# Jenkinsfile A
pipeline {
  stages {
    stage('Build') {
      steps { make all }
    }
  }
}
```

Toolchain B

arm gcc 9.1
clang-format

```
# Jenkinsfile B
pipeline {
  stages {
    stage('Check format') {
      steps { echo 'Run clang-format....' }
    }
    stage('Build') {
      steps { make all }
    }
  }
}
```

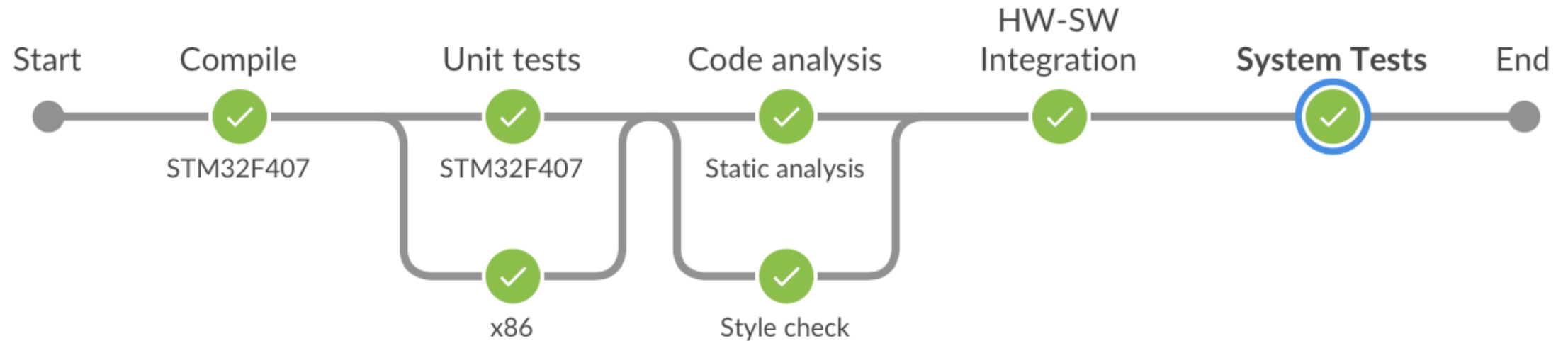

Wo stehen wir mit dieser Lösung?



- ✓ Beschreibung des Prozesses in strukturierter Form
- ✓ Lebende Dokumentation

Problem	Lösung	Reproduzierbarkeit
Compiler / Tools verändern sich	Änderung im Dockerfile	Jederzeit möglich
Build-Prozess verändert sich	Änderung im Jenkinsfile	Jederzeit möglich

Eine Toolchain zum Ausprobieren



Selbst ausprobieren und als Template nutzen
<https://github.com/embeff/embedded-ci-demo>

On-Target Tests automatisieren?

Die embeff ExecutionPlatform

- Unterstützt wichtige Test-Frameworks
- Unterstützt direkte On-Target Unittests
- Unterstützt Ihre spezifische MCU
- Keine weitere Hardware nötig
- Einfache Integration in CI

<https://embeff.com/executionplatform/>



Danke für die Aufmerksamkeit!



Kontakt

daniel.penning@embeff.com

Direkt +49 (451) 16088698

**Hochwertige
Embedded-Software
alltäglich machen.**

Embedded Software

- Echtzeit
- Vielzahl von Zielsystemen
- Hardwarenaher Code
- Qualitätsanspruch

Klassische Software

- Automatisierung
- Direktes Feedback
- Wiederverwendbare Bibliotheken
- Ausgereiftes Tooling