Continuous Integration

für ein gutes Zusammenspiel

Kevin Kessenich
Software Architekt @CofinproAG

Kevin Kessenich

Software Architekt

> Ne echte Kölsche Jung

COFINPRO

Management-, Fach- und Technologieberatung für Deutschlands führende Banken und Kapitalverwaltungsgesellschaften. Als Experten für Kredit und Wertpapier begleiten und navigieren wir unsere Kunden durch die Herausforderungen von Digitalisierung, neuen Marktanforderungen und Regulatorik.

Warum

Wie wir Software entwickeln

Business Value

Schnell an den Markt anpassen

Build
Ideen in Produkte umwandeln

Measure
Reaktionen der Kunden messen

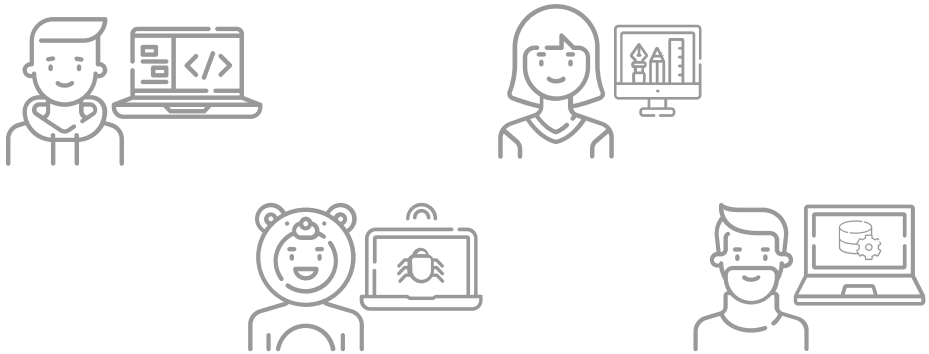
Learn
Aus Kundenverhalten lernen

https://sniederm.files.wordpress.com/2018/06/eric-ries-lean-startup.jpeg

Im Team Software erstellen

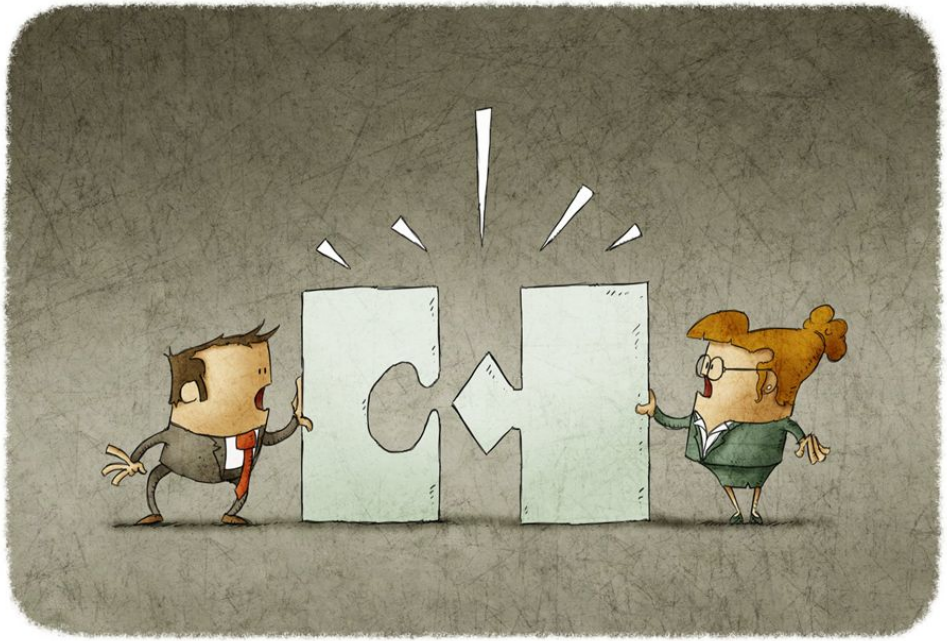
Verschiedene Rahmenbedingungen

- SCRUM, XP,...
- Cross-Functional
- Mehere Teams
- Verteilt
- Mehrsprachig



Software ist ein großes Puzzle

Die Puzzlestücke müssen zusammen passen



https://www.allbusiness.com/asset/2016/10/Puzzles-dont-fit..jpg

Agile Manifest

Prinzipien hinter dem Agilen Manifest

Unsere höchste Priorität ist es, den Kunden durch frühe und **kontinuierliche Auslieferung wertvoller Software** zufrieden zu stellen.

Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen **Veränderungen** zum Wettbewerbsvorteil des Kunden.

Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.

Fachexperten und Entwickler müssen während des Projektes täglich **zusammenarbeiten**.

Funktionierende Software ist das wichtigste Fortschrittsmaß.

...

https://agilemanifesto.org/iso/de/principles.html

Was

Was CI und CD ist

Continuous X

Definitionen

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. - *Martin Fowler*

Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time. - *Martin Fowler*

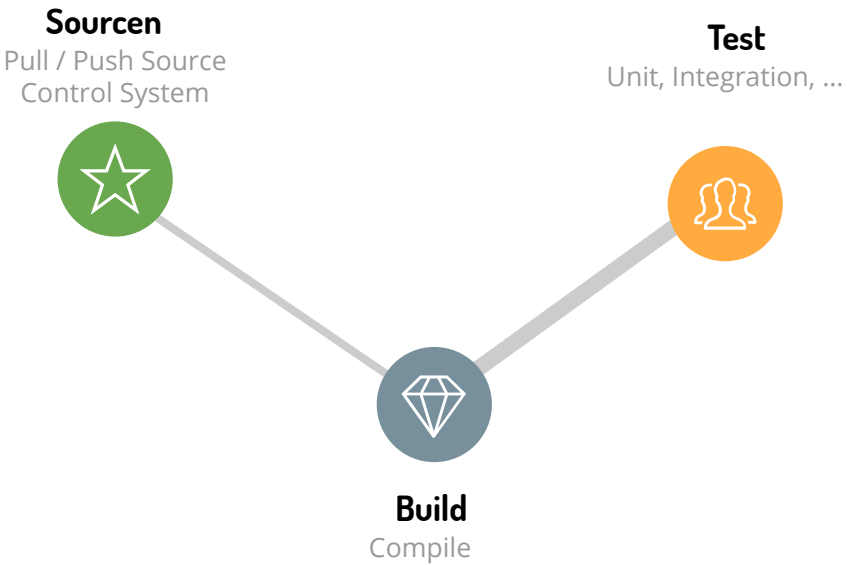
Continuous Deployment means that every change goes through the pipeline and automatically gets put into production, resulting in many production deployments every day. - *Martin Fowler*

https://martinfowler.com/articles/continuousIntegration.html

https://martinfowler.com/bliki/ContinuousDelivery.html

Continuous Integration Pipeline

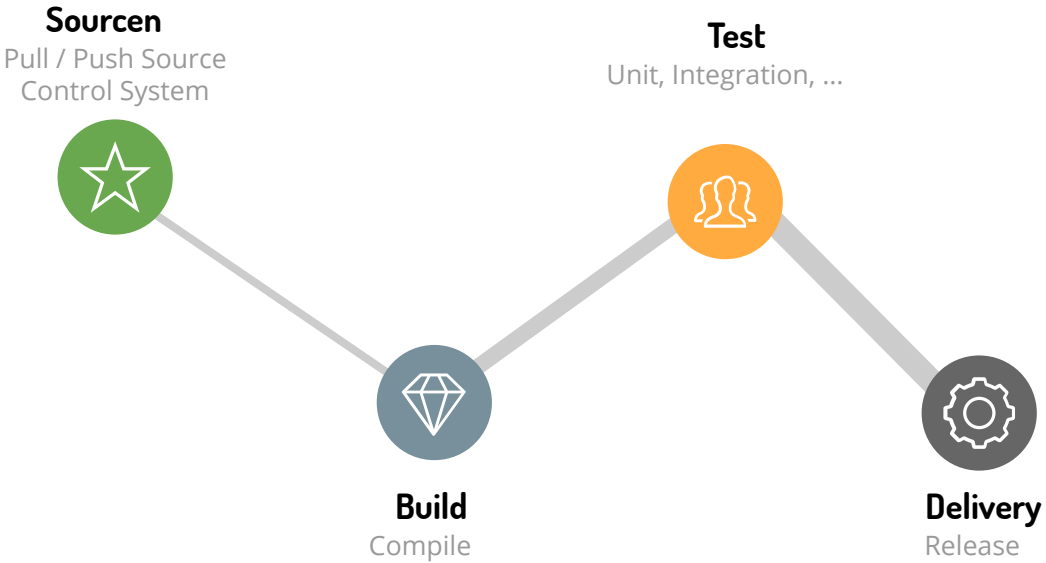
Merge-Konflikte vermeiden



https://www.redhat.com/de/topics/devops/what-is-ci-cd

Continuous Delivery Pipeline

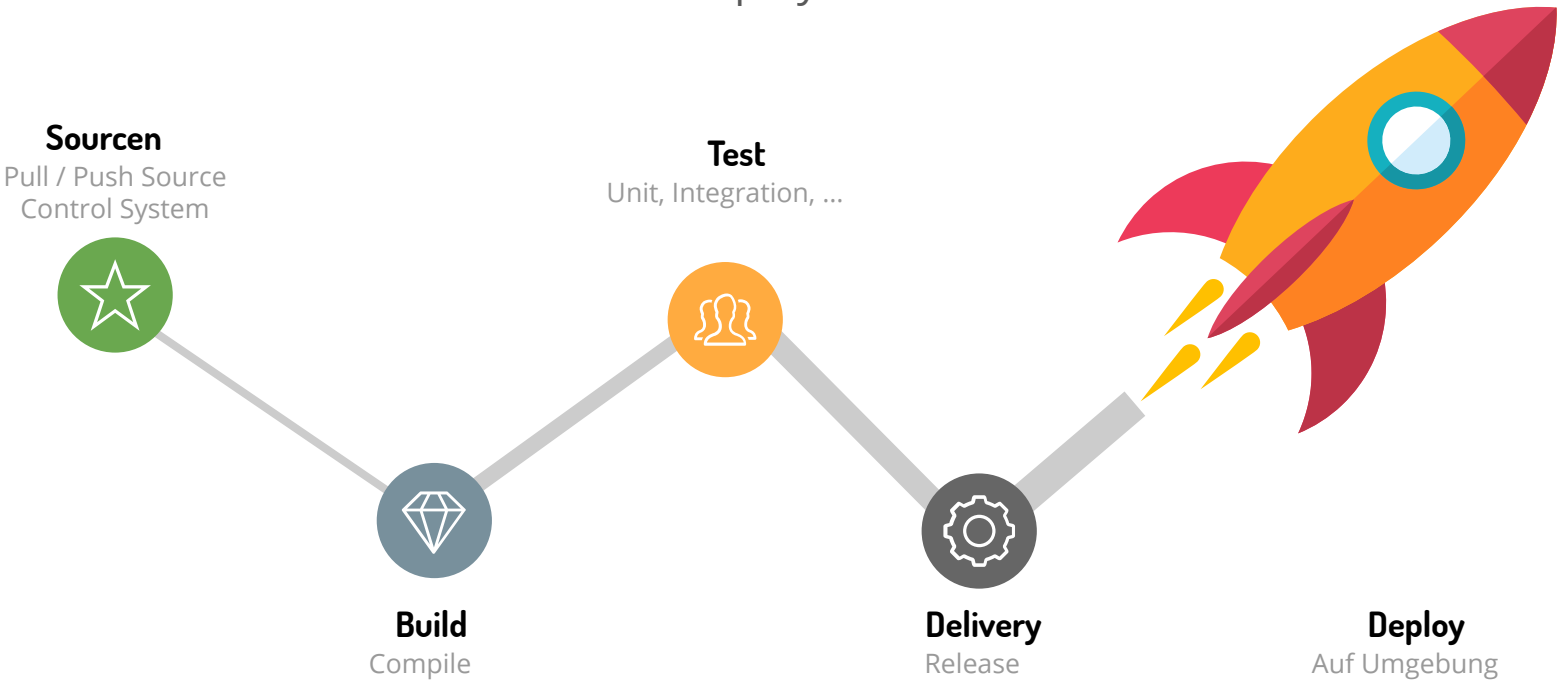
Automatisches Release in Repository



https://www.redhat.com/de/topics/devops/what-is-ci-cd

Continuous Deployment Pipeline

Automatisches Deployment auf Produktion



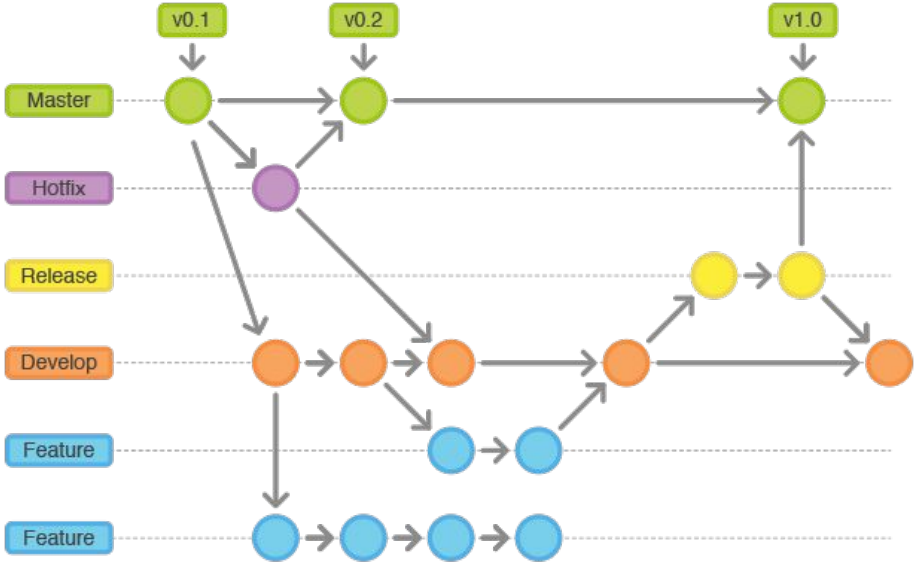
https://www.redhat.com/de/topics/devops/what-is-ci-cd

Was Genau

Ein Blick auf die einzelnen Steps

Source Control System

Zentraler Ablageort für Sourcen und Konfigurationen



git

SUBVERSION

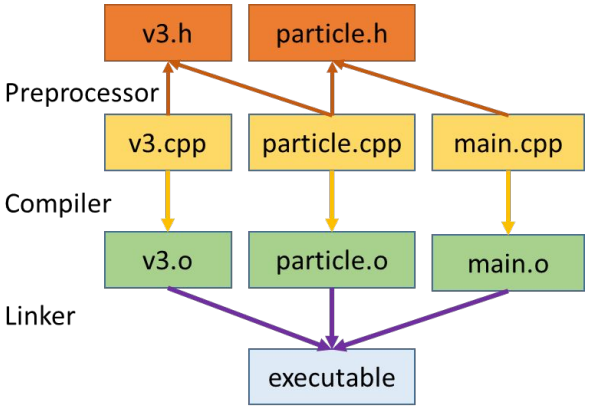
mercurial

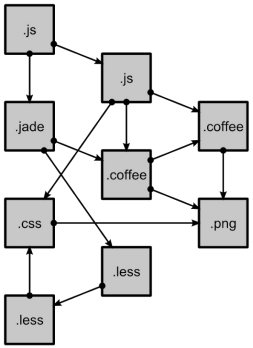
TFS

https://leanpub.com/git-flow/read

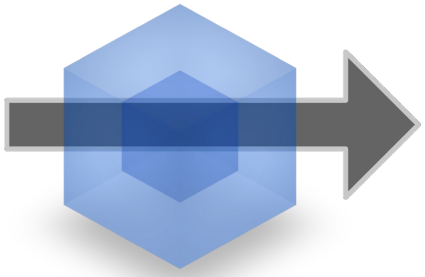
Build

Komilieren der Sourcen

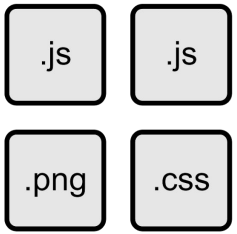




modules
with dependencies



webpack
MODULE BUNDLER



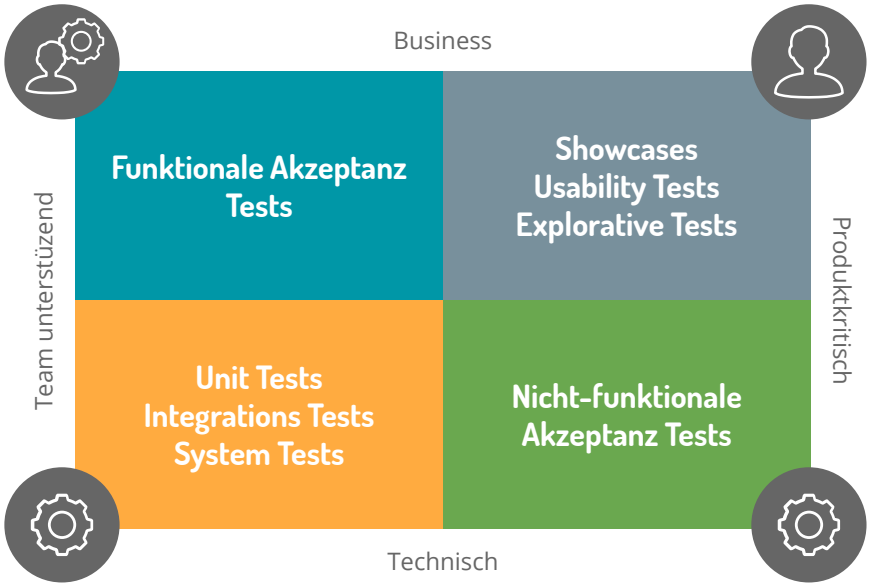
static
assets

https://devblogs.nvidia.com/separate-compilation-linking-cuda-device-code/

http://webpack.github.io/

Test

Brian Marick's Test Quadrant



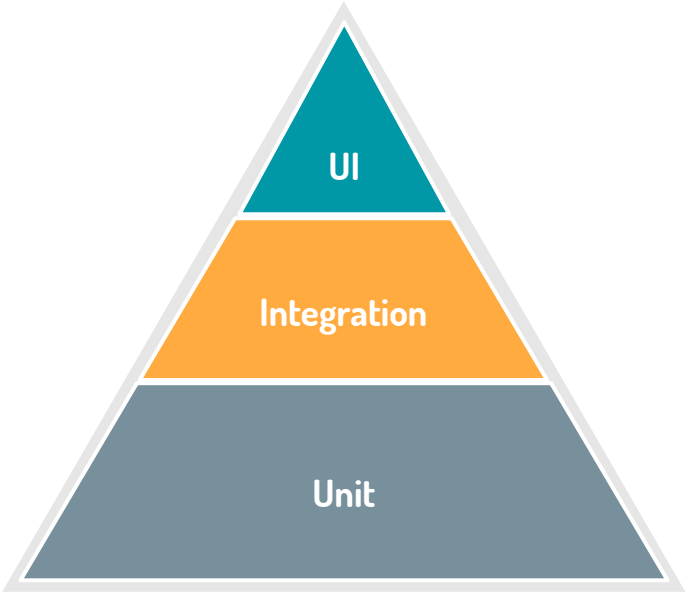
https://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants

Test

Test Pyramide

Langsam

€€€ Teuer



Schnell

€ Günstig

https://martinfowler.com/bliki/TestPyramid.html

Test

Code Analyse

sonarqube

DEPENDENCY-CHECK

ESLint

stylelint

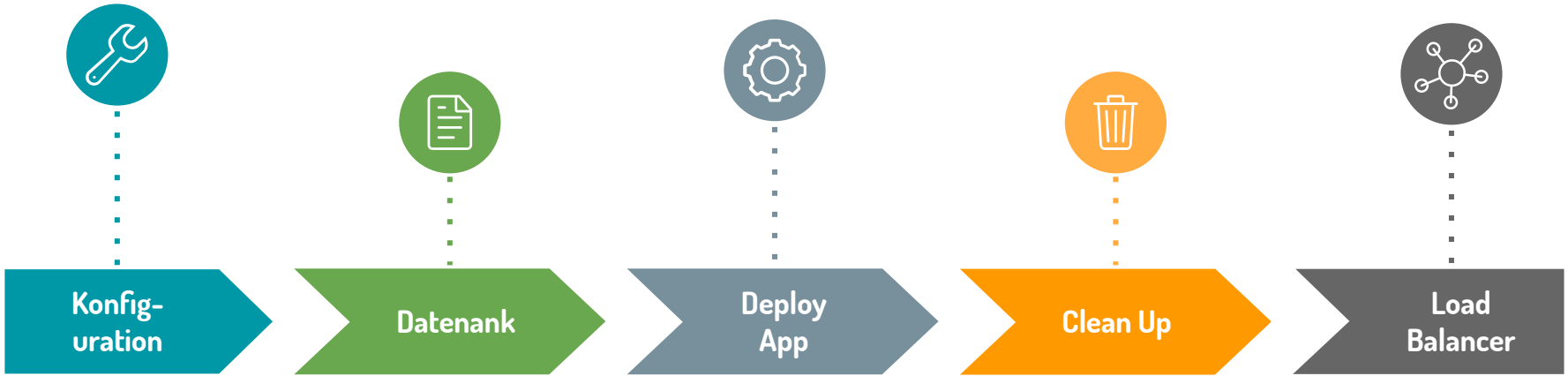
Delivery

Paketierung und Release

- Paketierung der Anwendung und allen benötigten Dateien für das Deployment
- Configuration Management; jegliche Konfiguration der Umgebung, die benötigt wird. Bsp.: Datenbank Scripte
- Release in ein Repository / zentrale Stelle

Deploy

Deployment besteht aus mehreren Schritten



Wie

Wie eine Build Pipeline definiert ist

Workflow / Pipeline

Aufbau des Build Workflow

Workflow / Pipeline

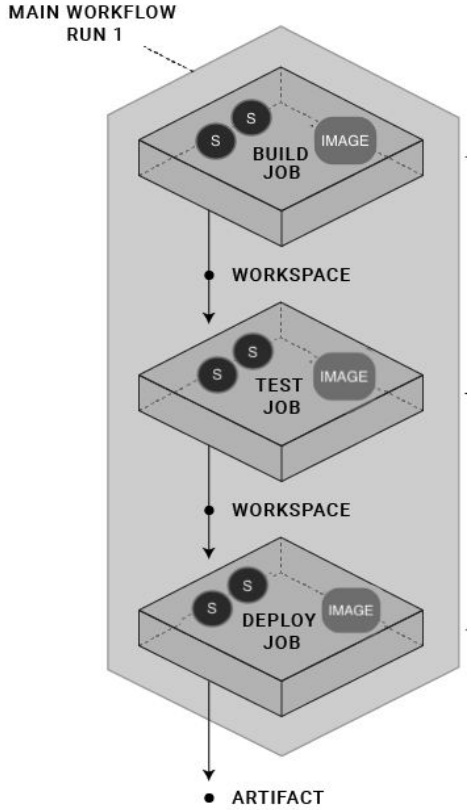
- Beschreibt den gesamten Build Prozess
- Besteht aus Jobs/Stages die sequentiell, parallel, zeitgesteuert oder “manuell” laufen

Jobs / Stages

- Sammlung von Steps
- Beispiele: Build, Test, Deploy

Steps

- Scripte (sh) / Befehle die ausgeführt werden



https://circleci.com/docs/2.0/concepts/#section=getting-started

https://jenkins.io/doc/book/pipeline/#pipeline-concepts

Datenhaltung

Storage Varianten

Workspace

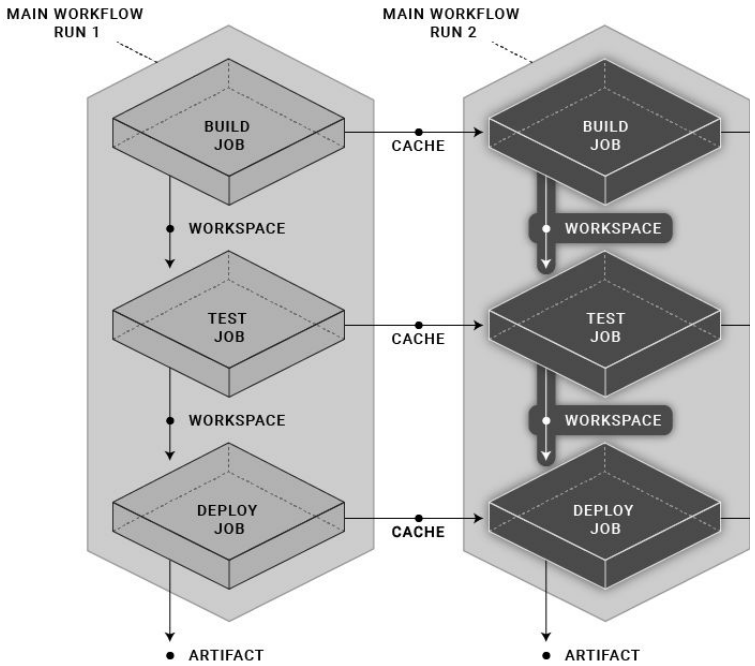
- Dauer: 1 Build
- Daten für den jeweiligen Build

Cache

- Dauer: Monate
- Dependencies, etc.

Artifact

- Dauer: Monate
- Output vom Build Prozess (Logs, Exec,..)



https://circleci.com/docs/2.0/concepts/#section=getting-started

DEMO

Jenkins / CircleCI

Wie Richtig

Vorteile und Best Practices

Vorteile von CI/CD

Schnell qualitative passende Software liefern

Integration

Prozess

Tests

Scripte

Feedback

Hohe Qualität und Produktivität

Fehler werden schnell erkannt
Das Team kann sich auf seine Arbeit konzentrieren

Deploybare Software

Builds und Releases sind nachvollziehbar
Ein Deployment ist jederzeit möglich (theoretisch)

Projekt Sichtbarkeit

Build Status zeigt Qualität und Fortschritt

CI Best Practices

Ziel ist ein stabiler Master Branch

Commit code frequently

Fix broken builds immediately

Run private builds

Don't commit broken code

Write automated developer test

Avoid getting broken code

Keep the build fast

All tests and inspections must pass

Don't go home before master passed

Backup CI Server

Disk Space CI Server

Don't schedule multiple jobs at the same time

Ganzheitlicher Ansatz

Gemeinsam Richtung Ziel

> To successfully implement continuous delivery, you need to change the culture of how an entire organization views software development efforts.
>
> Tommy Tynjä

https://dzone.com/articles/my-18-favorite-quotes-on-agile-devops-and-continuo

CI / CD

THANK YOU

Link zur Präsentation

https://github.com/kessenich/webdev-2020

Die Beispiel App gibt es hier:

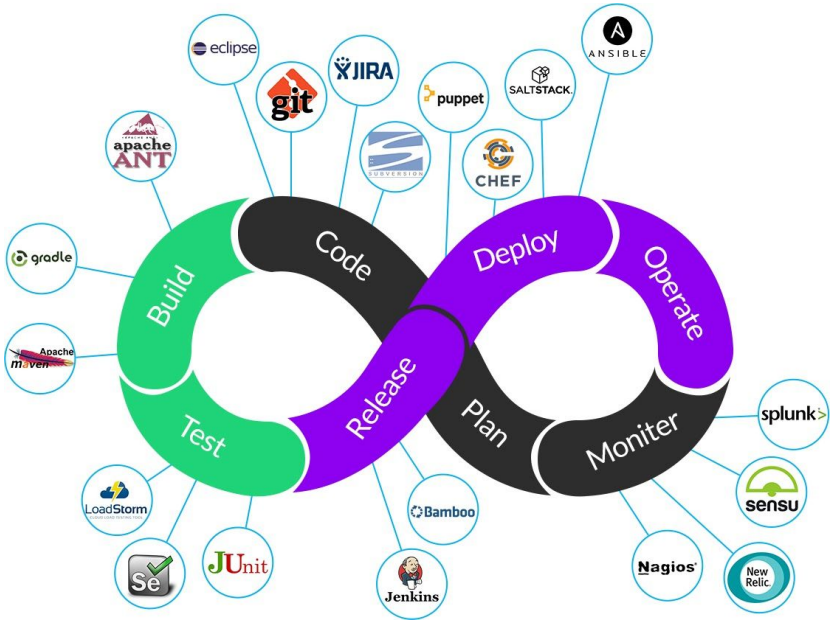
https://github.com/kessenich/webdev-2020-app

CI / CD

Back Up

DevOps

DevOps Cycle



https://evontech.com/component/easyblog/why-is-devops-the-future-of-software-production-cycles.html?Itemid=159