

# Softwaretechnik-Praktikum

## Vorlesung 02

Sommersemester 2023  
Jens Kosiol,  
Adrian Kunz, et al.

# Softwareentwicklung im Team

Jens Kosiol

# Motivation

Produktivität:  $\approx$  100 Lines of Code pro Tag (20 KLoc pro Jahr)

Große Programme  $>$  1 mio LoC z.b.  $>$  50 Personenjahre

$\Rightarrow$  Software sollte in Teams entwickelt werden

# Wie entwickle ich Software in Teams

1980er:

- Programm in Stücke aufteilen /  
Komponenten / Module / Dateien
- Eine Person pro Komponente
  
- Architektur / Blaupause für  
Komponente
- Komponenten bauen und testen
- Komponenten integrieren (Big  
Bang)

```
Alice's PC  
c:/pm/  
GroupApp.java
```

```
Bob's PC  
c:/pm/  
Party.java
```

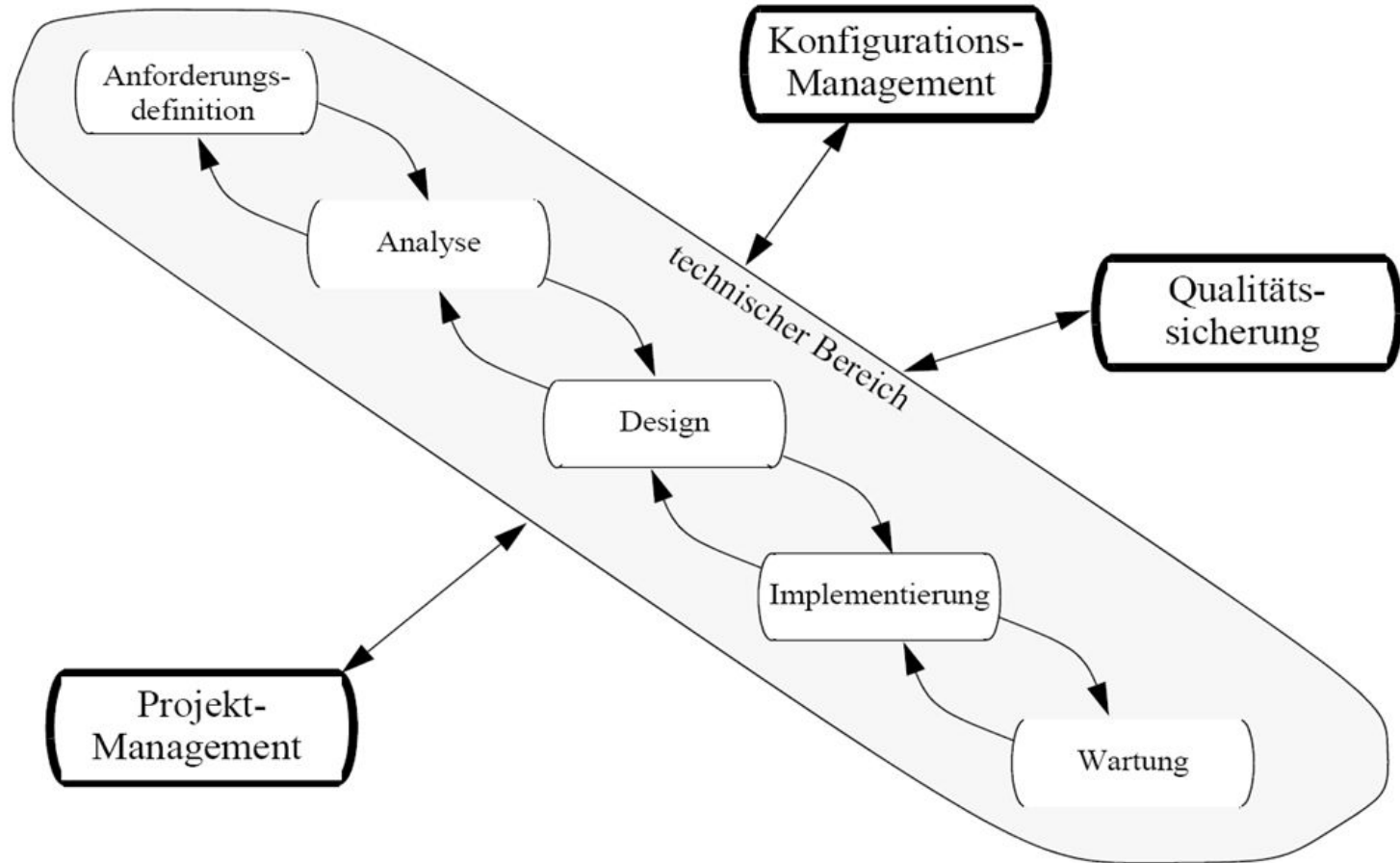
```
Integration PC  
c:/pm/  
GroupApp.java  
Party.java  
Item.java
```

```
Carli's PC  
c:/pm/  
Item.java
```

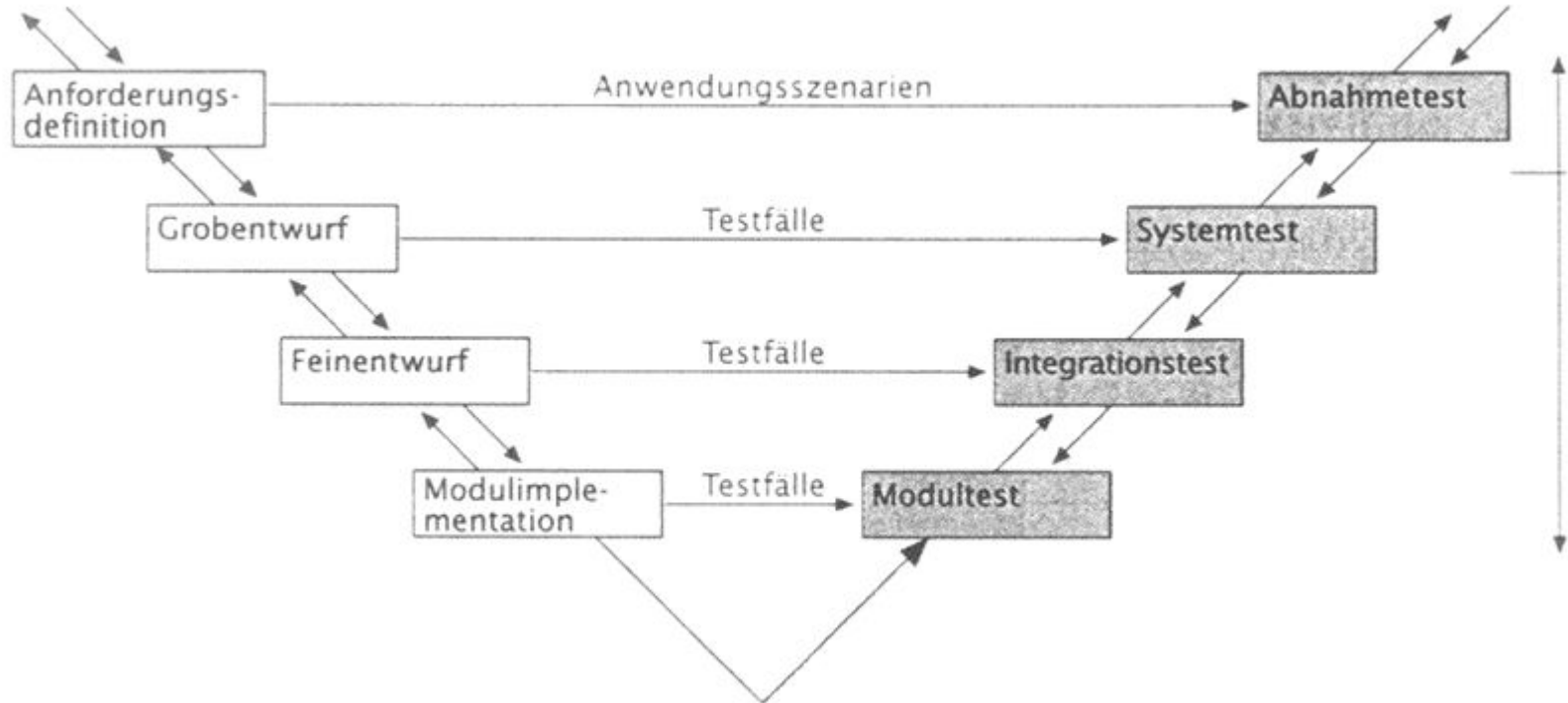
# Kommunikationsaufwand $n * (n - 1) / 2$

- 60 Komponenten / 60 Teammitglieder
- Tägliche Meetings: Diskussion 5 Minuten pro Komp.  $\Rightarrow$  5 Stunden Meeting
  - $\Rightarrow$  Bessere Arbeit / Komponentenbeschreibung
  - $\Rightarrow$  Komponenten zu Packages gruppieren / Teams in Subteams
  - $\Rightarrow$  Diskussionen aufteilen nach Packages

# Wasserfallmodell



# V - Modell



# Komponentenübergreifende Features $\Leftrightarrow$ Versionsverwaltung

1990er:

- SCCS, CVS, Subversion, GIT (2005), ...
- Ein Feature / Eine Person / Mehrere Komponenten
- Reduziert Meetingdauer



# Komponentenübergreifende Features $\Leftrightarrow$ Versionsverwaltung

2000er

- Iterativer Softwareentwicklungsprozess / Auslieferung nach Features
- eXtreme Programming
- Scrum

# Komponentenübergreifende Features $\Leftrightarrow$ Versionsverwaltung

2010er

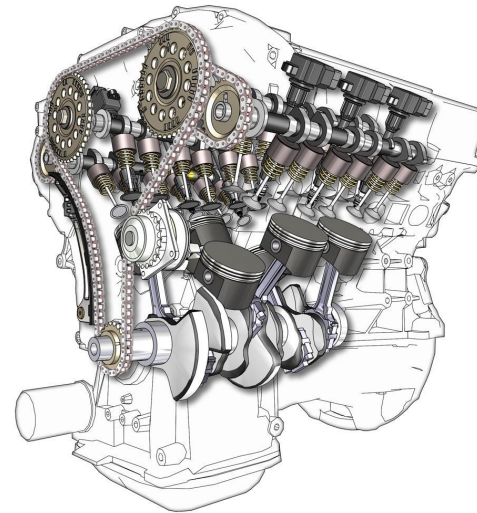
- Cloud Computing
- Dev Ops

# Software Engineering

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD
  XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/
  xhtml1-transitional.dtd">
3
4 <html xmlns="http://www.w3.org/1999/
  xhtml">
5   <head>
6     <meta http-equiv="Content-
  Type" content=
7     "text/html; charset=us-
  ascii" />
8     <script type="text/
  javascript">
9       function reDo() {top.
  location.reload();}
10      if (navigator.appName ==
  'Netscape') {top.onresize = reDo;}
11      dom=document.
  getElementById;
12      </script>
13    </head>
14    <body>
15    </body>
16 </html>
```



=



?

# Manifest für agile Softwareentwicklung

Wir erschließen bessere Wege, Software zu entwickeln,  
indem wir es selbst tun und anderen dabei helfen.

Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

**Individuen und Interaktionen** mehr als Prozesse und Werkzeuge  
**Funktionierende Software** mehr als umfassende Dokumentation  
**Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung  
**Reagieren auf Veränderung** mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden,  
schätzen wir die Werte auf der linken Seite höher ein.

# Zusammenfassung und Ausblick

- Komplexe Software kann nur in Teams entwickelt werden
- Versionsverwaltungssysteme ermöglichen iteratives und featureorientiertes Arbeiten im Team
- Agile Prozesse passen (oft) besser zum Wesen von Software als starrere Prozesse aus anderen Ingenieurwissenschaften

**Nächste Woche:** Scrum und Extreme Programming als konkrete Beispiele für agile Softwareentwicklung

# Git

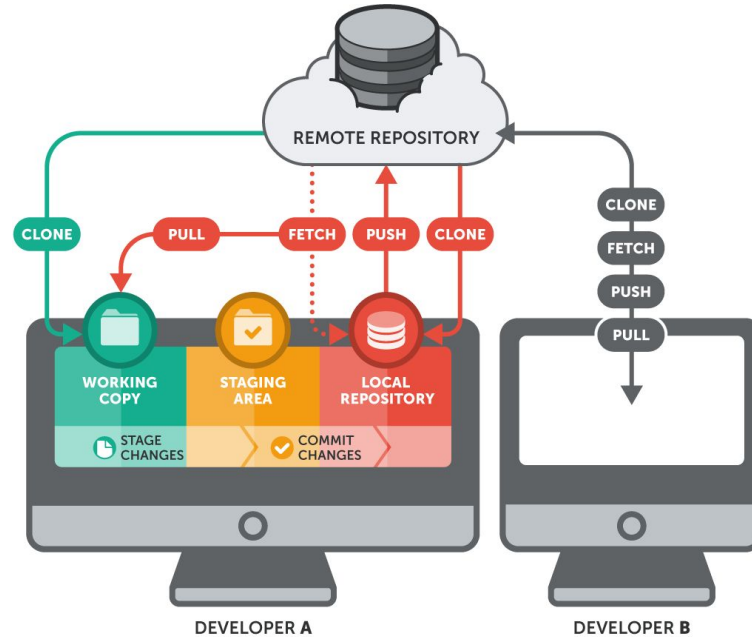
Jens Kosiol

# Git Wiederholung

Commit / Push / Pull / Fetch

Branching / Merging

# Commit / Push / Fetch / Pull





# Branching / Merging

