

Miteinander  
Nebeneinander  
Durcheinander

# Kommunikation und Koordination in Softwareprojekten

buschmais

Beratung . Technologie . Innovation

Dirk Mahler  
buschmais GbR

Inhaber

Torsten Busch, Frank Schwarz,  
Dirk Mahler und Tobias Israel

dirk.mahler@buschmais.com  
<http://www.buschmais.de/>

TU Dresden, 17.11.2014

# Agenda

- Das Kommunikations-Problem
- Architekten, Entwickler und die Code-Basis
- Zusammenfassung

Miteinander, Nebeneinander, Durcheinander – Kommunikation und Koordination in Softwareprojekten

# Das Kommunikations- Problem

# Das Kommunikations-Problem

- Kommunikation
  - Übertragung von Informationen
  - Quelle – Kanal – Senke

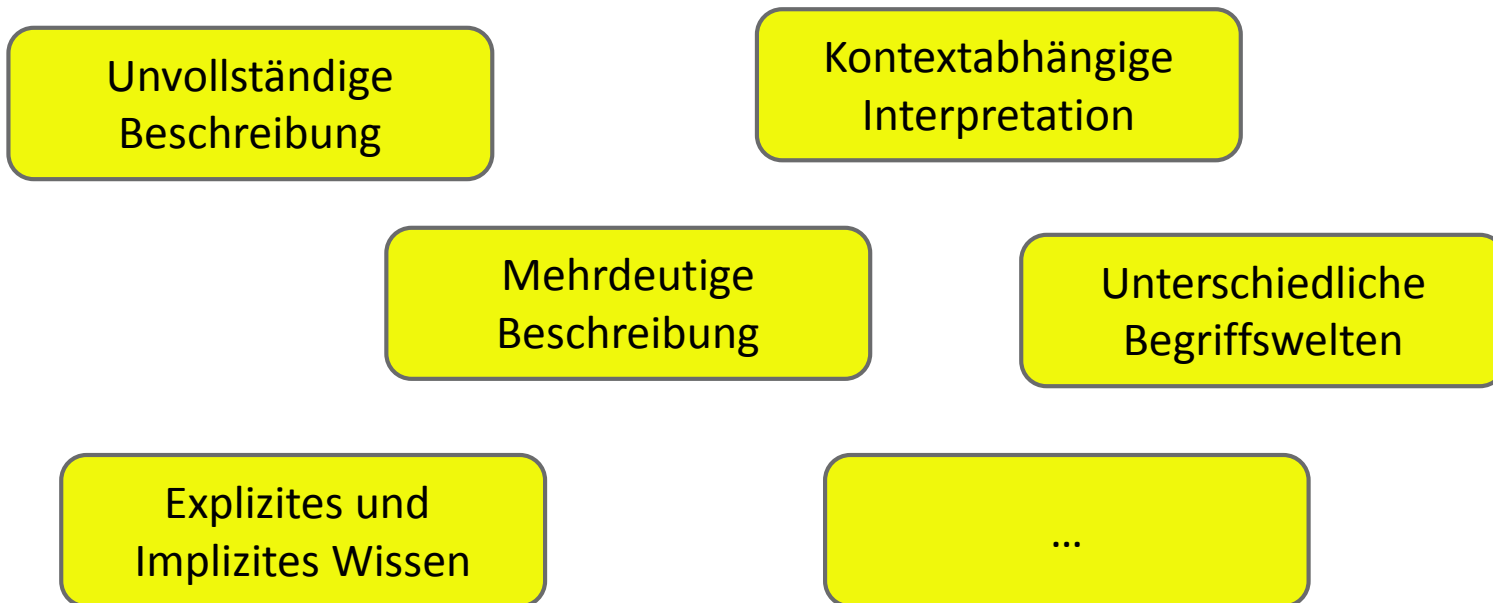
- Beispiel: Mensch zu Mensch



# Experiment

# Das Kommunikations-Problem

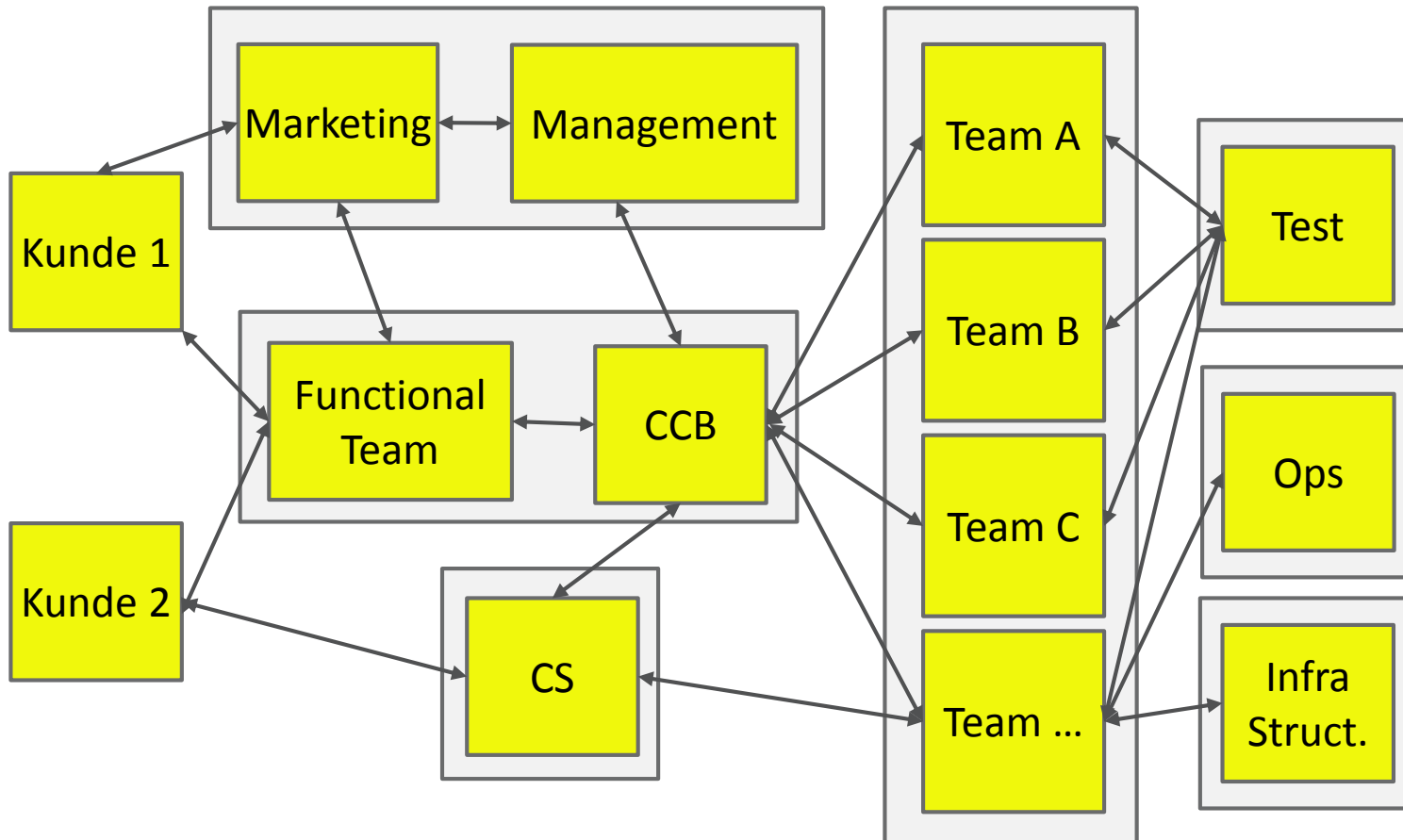
- Welche Problem haben Sie erkannt?



- Kommunikation ist unsicher!



# Das Kommunikations-Problem





# Das Kommunikations-Problem

## □ Hürden

Indirektion

Sprachliche  
Barrieren

Veränderungen  
über die Zeit

Erfahrungsabh.  
Interpretation

Kulturelle  
Differenzen

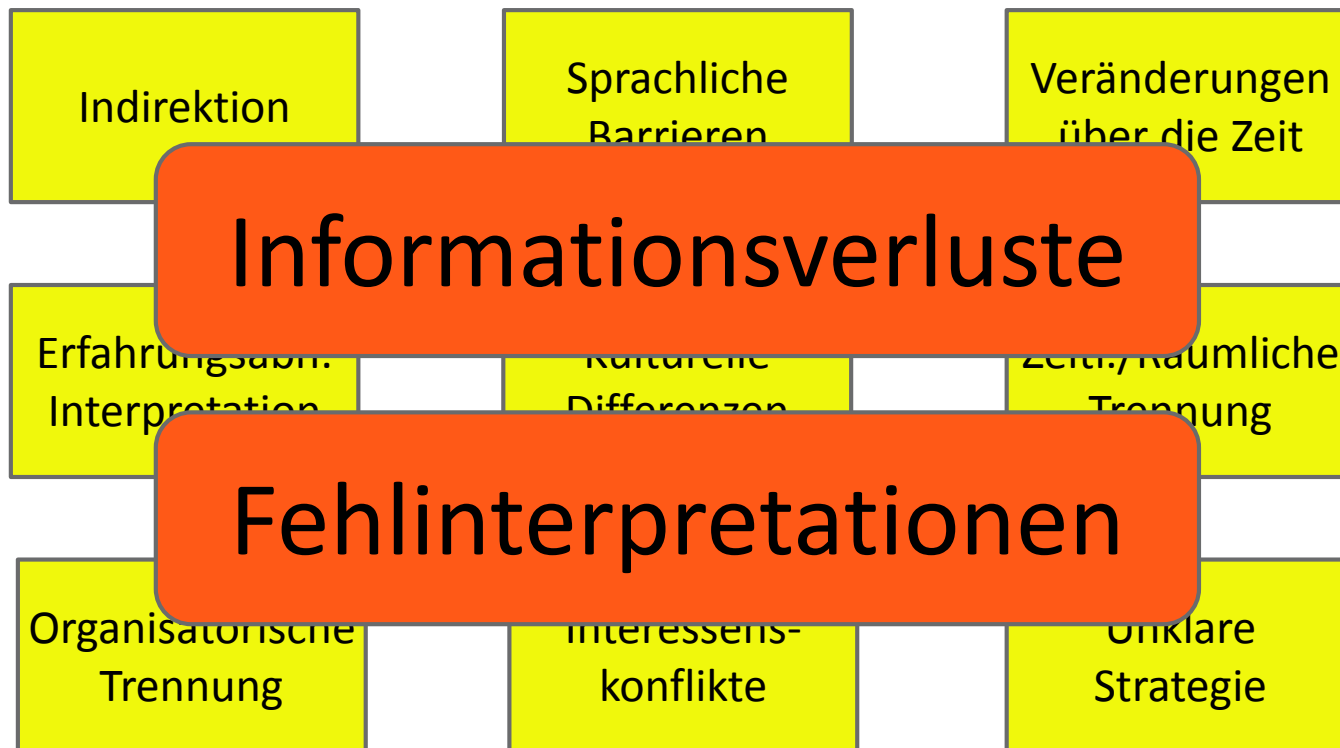
Zeitl./Räumliche  
Trennung

Organisatorische  
Trennung

Interessens-  
konflikte

Unklare  
Strategie

## □ Hürden



# How Projects Really Work

Quelle: <http://www.projectcartoon.com/cartoon/2>

# Das Kommunikations-Problem



How the customer explained it

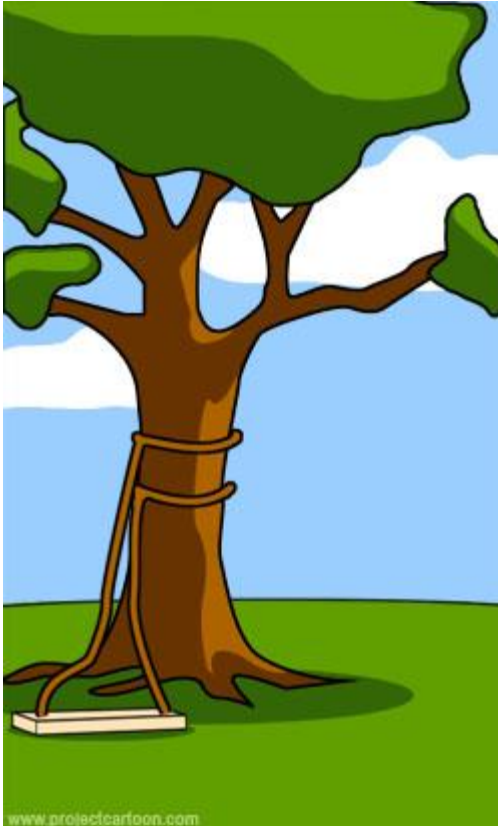


How the project leader understood it



How the analyst designed it

# Das Kommunikations-Problem



How the programmer wrote it

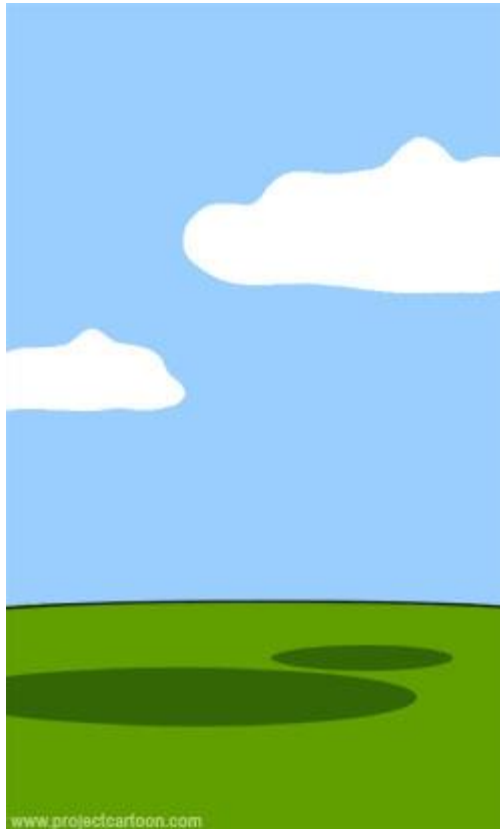


What the beta testers received

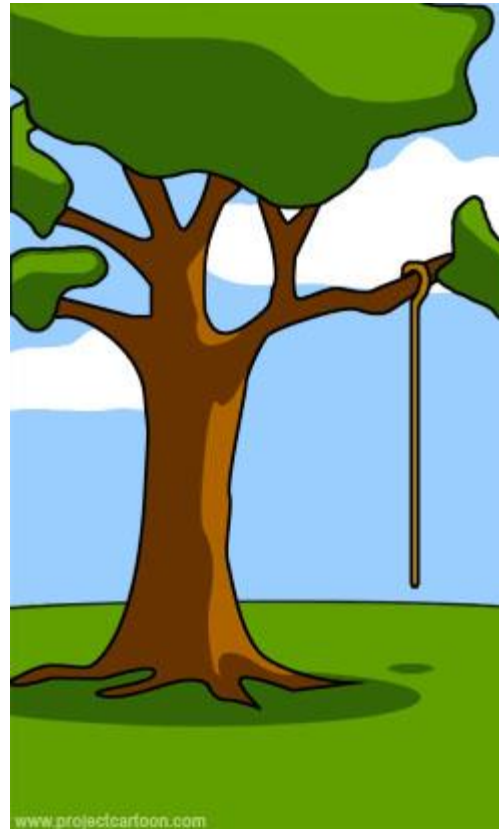


How the business consultant described it

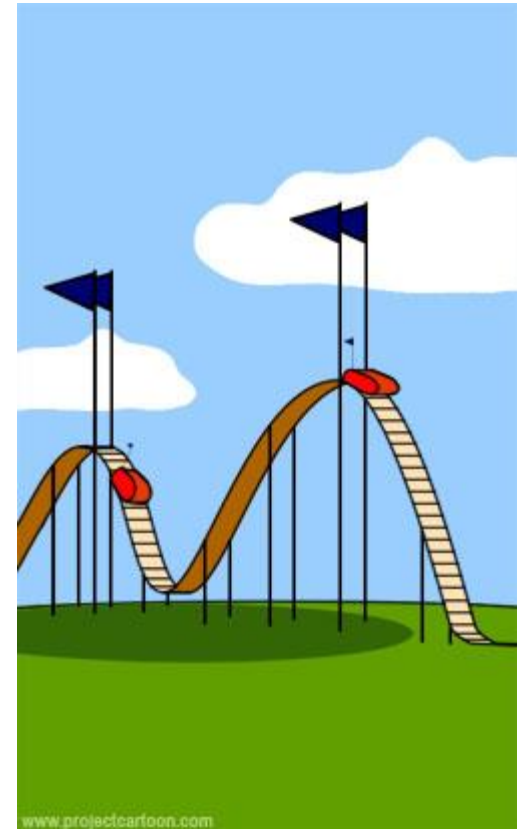
# Das Kommunikations-Problem



How the project was documented

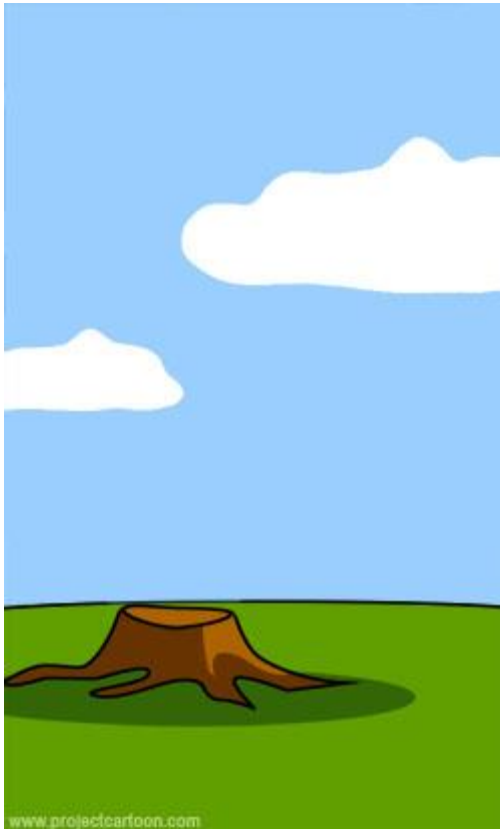


What operations installed



How the customer was billed

# Das Kommunikations-Problem



How it was supported



What marketing advertised



What the customer really needed

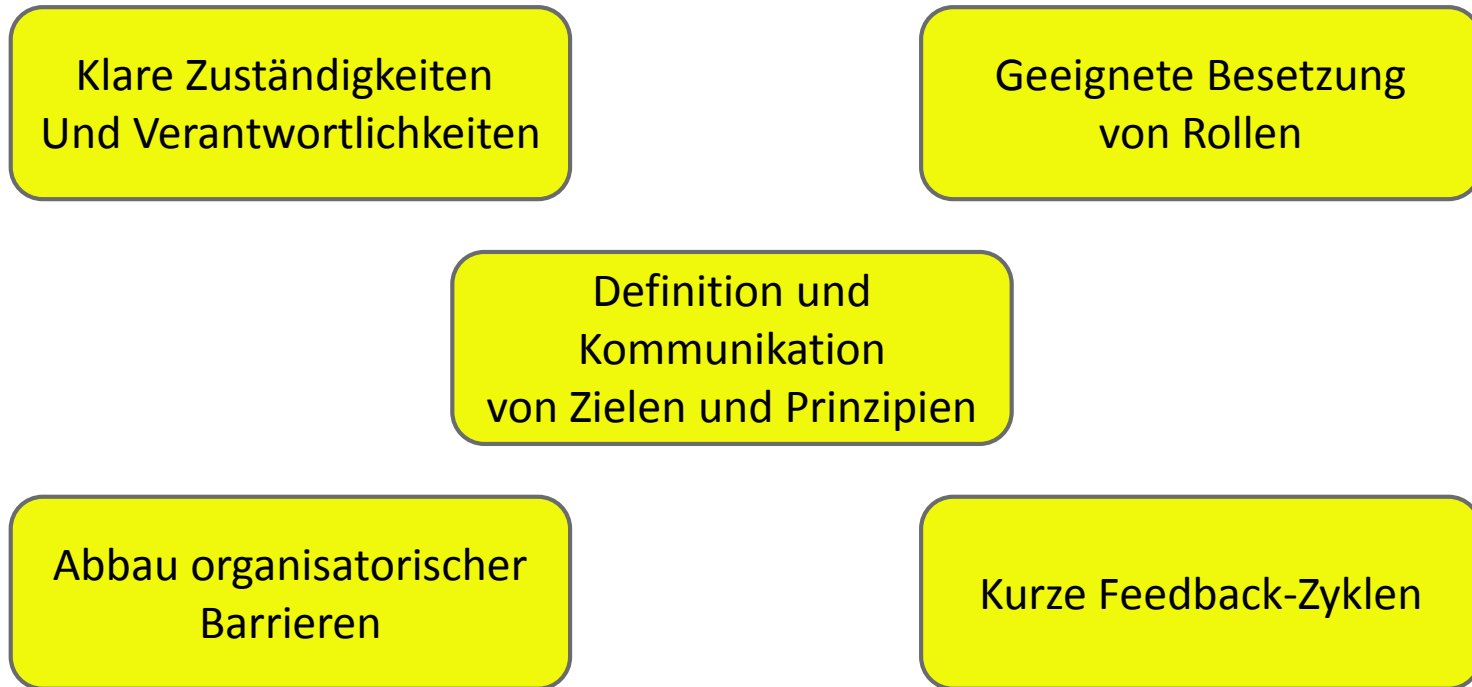
Sind Software-Projekte  
von vornherein  
zum Scheitern verurteilt?



- Software-Projekt = Beherrschung von Komplexität



## □ Ansätze

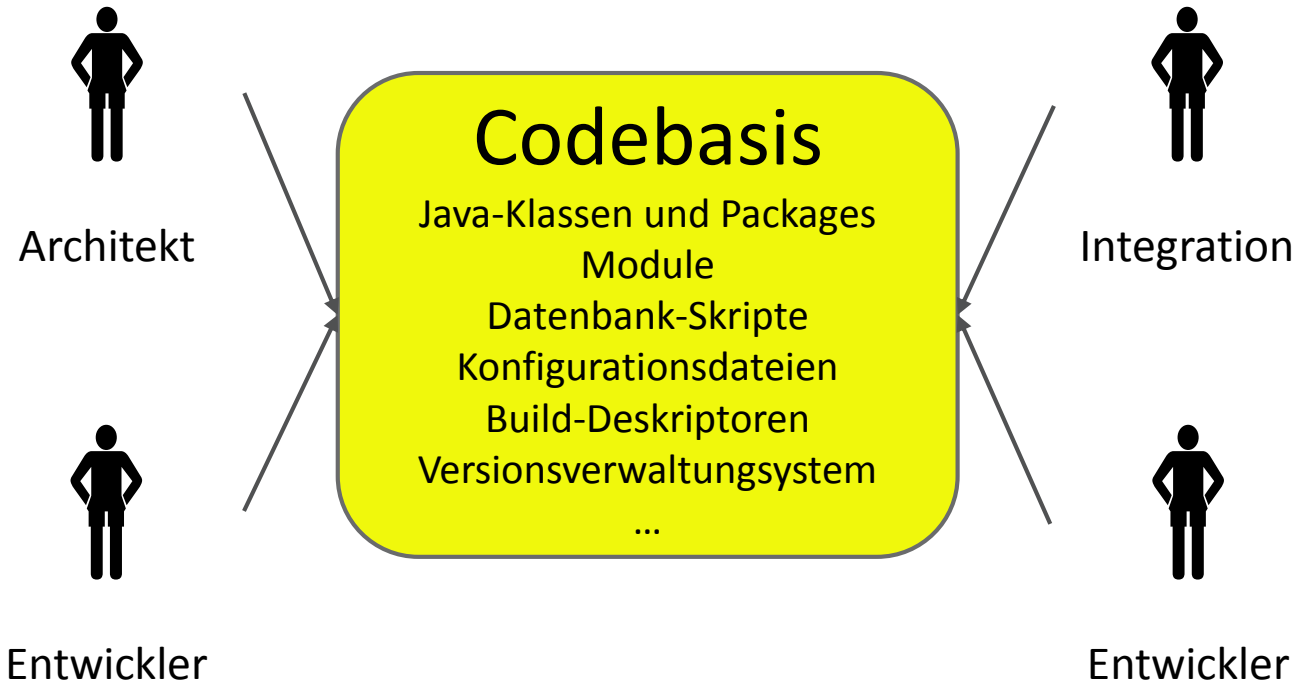


## □ Grundbegriffe agiler Ansätze

Miteinander, Nebeneinander, Durcheinander – Kommunikation und Koordination in Softwareprojekten

# Architekten, Entwickler und die Codebasis

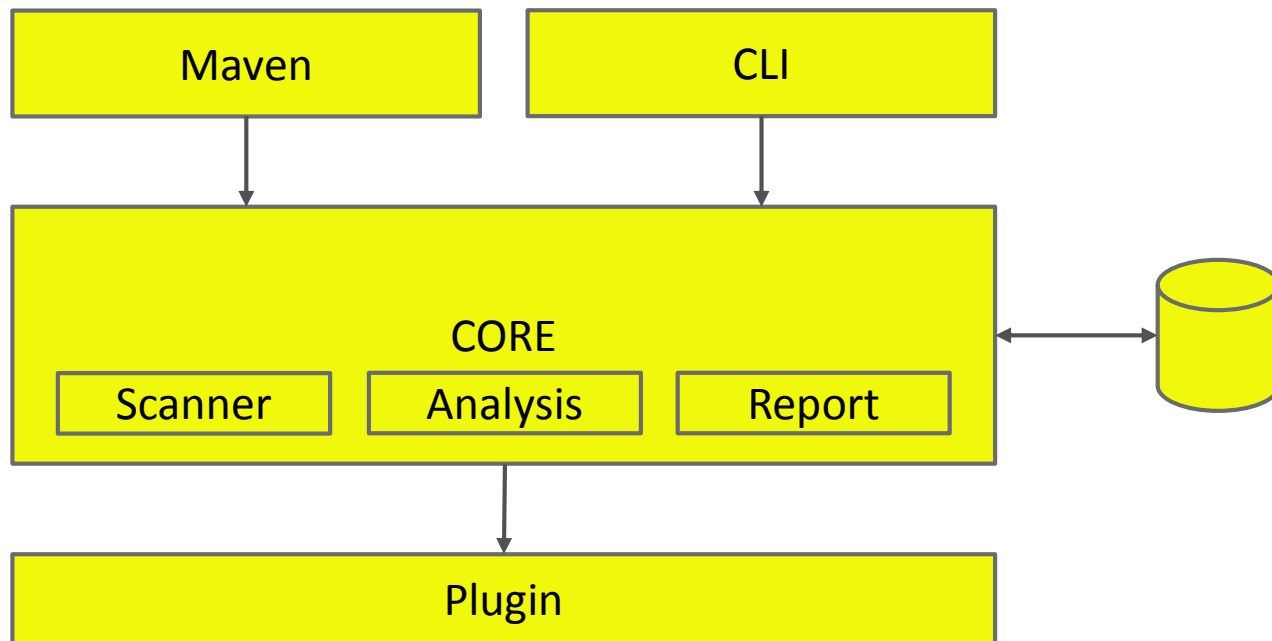




- Großer Umfang (10.000-100.000 Dateien, Mio. LoC)
- Unterschiedliche Sichten und Interessen

- Architekt
  - Betrachtung auf abstrakter Ebene
    - funktionale Einheiten und deren Beziehungen
    - Schnittstellen
  
- Entwickler
  - Betrachtung auf Detail-Ebene
    - Klassen, Methoden, Felder, Annotationen
    - Frameworks, Bibliotheken
    - Design
  
- Quellcode umfasst mehrere semantische Ebenen
  - Explizit und implizit → Kommunikationsproblem!

## □ Sicht eines Architekten



## □ Sicht eines Entwicklers

### Plugin

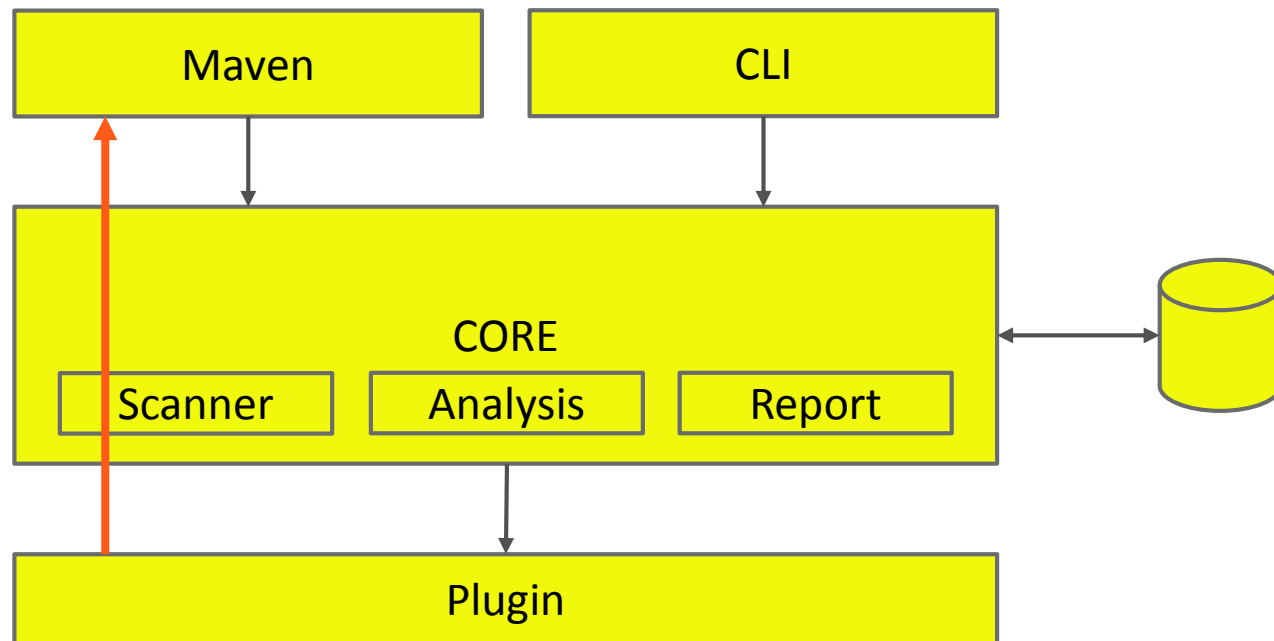
```
public class CSVFileScannerPlugin extends
    AbstractScannerPlugin<FileResource, CSVFileDescriptor> {

    @Override
    public boolean accepts(FileResource item, String path, Scope scope)
        throws IOException {
        return path.toLowerCase().endsWith(".csv");
    }

    @Override
    public CSVFileDescriptor scan(FileResource item, String path,
        Scope scope, Scanner scanner) throws IOException {
        // Open the input stream for reading the file.
        try (InputStream stream = item.createStream()) {
```



## □ Verletzung der Architektur



- i.d.R. unbeabsichtigt, z.B. fehlendes Wissen
- Sehr teuer, wenn zu spät erkannt

## □ Entwickler != Entwickler

- Gleiches Problem – Verschiedene Lösungen

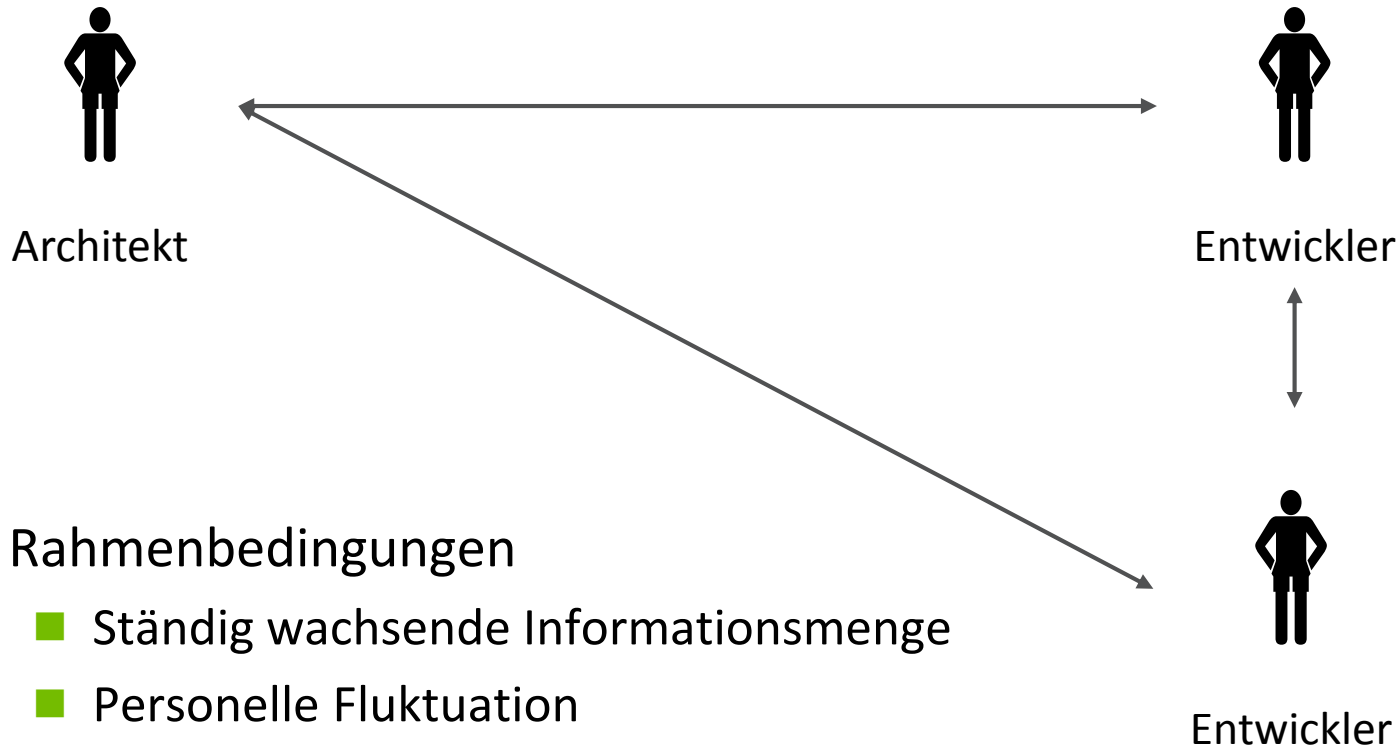
```
System.out.println()  
LOGGER.info()
```

- Unverständliche Lösung

```
for (int i=10; i>0; i--) {  
    ...  
}
```

## □ „Principle of least Surprise“

- Festlegung, Kommunikation und Durchsetzung von Regeln
  - Architektur
  - Design
  - Coding-Konventionen
  
- Ziel
  - Kohärente Strukturen und Namen
    - Schnelle Orientierung
  - Einheitliche Design-Muster
    - Befriedigung von Erwartungshaltungen
    - Erleichterung von Refactorings
  - Implementierungen, die zum technologischen Umfeld passen



## □ Rahmenbedingungen

- Ständig wachsende Informationsmenge
- Personelle Fluktuation
- Unterschiedliche Skills und Einstellungen
- Zeitdruck

- Dokumentation
  - Word, UML, Wiki, ...
  - Ist immer veraltet
  - Oft nicht einem Softwarestand zuzuordnen
  
- Zusammenführung mit Code
  - Generierung von Code aus Dokumentation
    - Modellgetriebene Ansätze
  
  - Generierung von Dokumentation aus Code
    - Entwickler-Dokumentation
    - Architektur-Dokumentation
    - Anwender-Dokumentation?

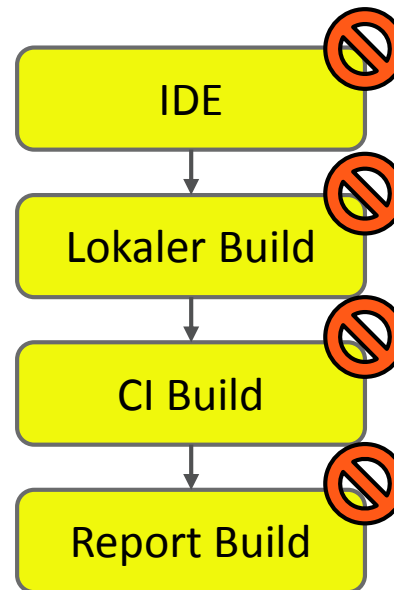
- Definition von Verantwortlichkeiten
  - Nach Funktionalität
    - Teams mit kompletter Rollenverteilung
  - Nach Aspekt
    - Architektur (fachlich)
    - Architektur (technisch)
    - Build-/Integrationsmanagement
  
- Pair-Programming und Reviews
  - Direkte Verteilung von Wissen
  - Aber: erreicht die Information alle Beteiligten?

## □ Einsatz von Werkzeugen

- Statische und dynamische Code-Analysen



Entwickler



- Schnellstmögliches Feedback
- Quality Gates

Miteinander, Nebeneinander, Durcheinander – Kommunikation und  
Koordination in Softwareprojekten

# Zusammenfassung



- Kommunikation ist fehleranfällig
  - Verluste, Interpretation, Interessen, ...
- Softwareprojekte sind komplex
  - Fachlich, technisch, organisatorisch
- Organisatorische Rahmenbedingungen sind wichtigster Faktor für Erfolg/Misserfolg eines Projektes
  - Abgleich von Organisations- und Projektstrukturen
  - Nähe zu allen Projektbeteiligten (Kunde bis Operations) schaffen
  - Schnelle Feedback-Zyklen ermöglichen

- Quellcode umfasst verschiedene semantische Ebenen
  - Nicht alle sind direkt sichtbar
- Verschiedene Sichten auf die selben Artefakte
  - Rollenabhängig
- Schaffung von Kohärenz ist elementar
  - Allen Rollen muss benötigtes Kontextwissen zugänglich sein
  - Kommunikationsproblem
- Ansätze
  - Dokumentation
  - Pair-Programming, Reviews
  - Code-Generierung oder automatisiertes Feedback

# Vielen Dank!



[buschmais.de](http://buschmais.de)



[facebook.com/buschmais](https://facebook.com/buschmais)



[twitter.com/buschmais](https://twitter.com/buschmais)