

Langlebige Softwarearchitekturen - technische Schulden beherrschen und abbauen

Dr. Carola Lilienthal, cl@wps.de, [@cairolali](https://www.instagram.com/cairolali)
www.wps.de

WPS - Workplace Solutions GmbH Hans-Henny-Jahnn-Weg 29 22085 HAMBURG



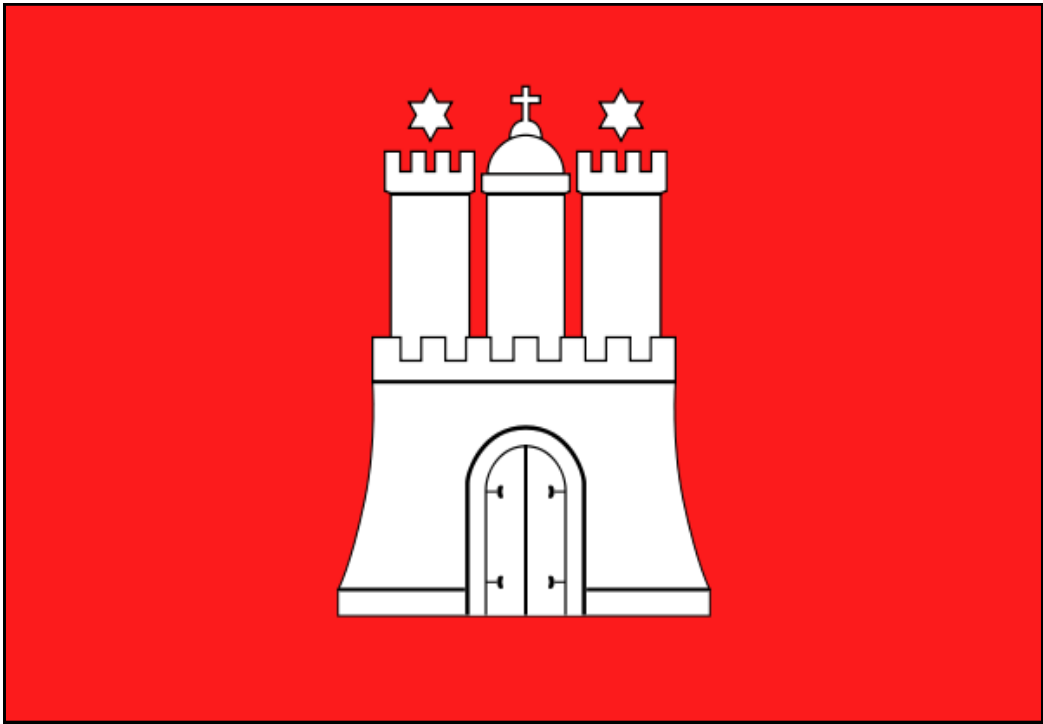


„Business-Software
die Spaß macht“

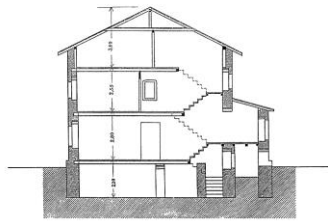




„Business-Software
die Spaß macht“



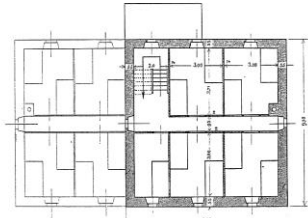
Neugersdorfer Mühle
an Krimmler Säuen
Der Sektion Wagnsdorf D. u. O. U.



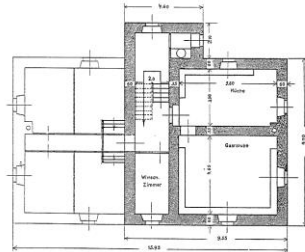
Querschnitt



Ansicht



Obergeschoss



Erdgeschoss

Foto: H. Gmehlich/Wikipedia





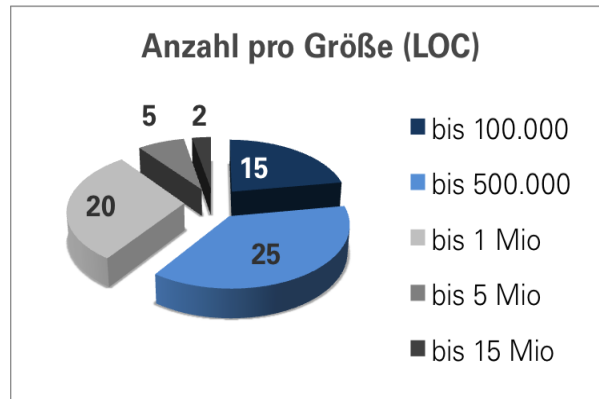


	System-Informationen
	Dr. Carola Lilienthal Geschäftsführerin
	@caiolali
	WPS - Workplace Solutions GmbH Hans-Henny-Jahn-Weg 29 22765 Hamburg Telefon +49 40 239 489-0 Telefax +49 40 239 489-299 Mobil +49 170 184 77 11 cl@wps.de www.wps.de



@caiolali
 cl@wps.de
www.ilsa.de

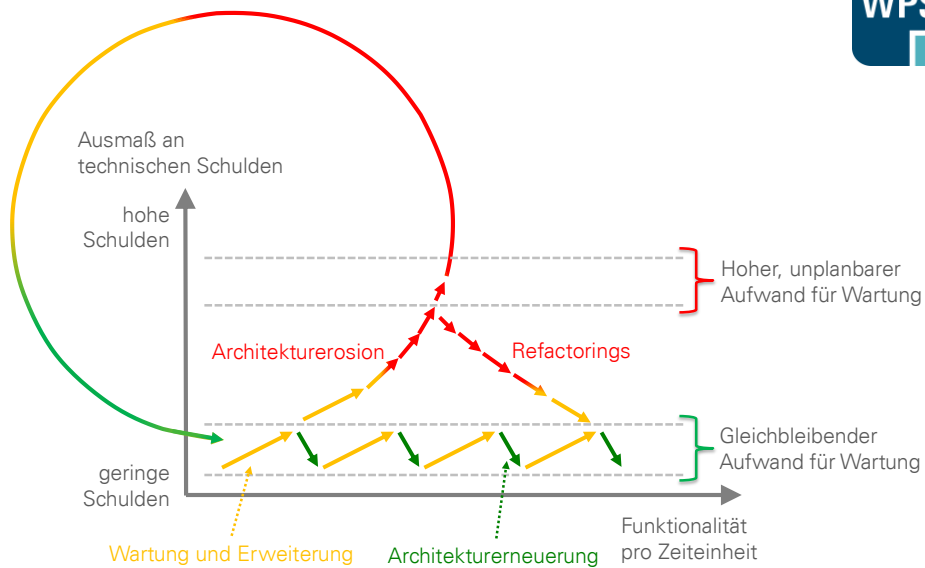




BEST PRACTICES IN APPLICATION ARCHITECTURE

TODAY: USE LAYERS TO DECOUPLE

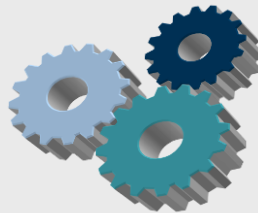




Architekturziele für Langlebigkeit

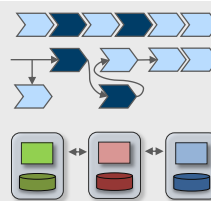
Architekturziel 1: Wartbarkeit

- schnelle Fehleranalyse
- schnelle Anpassungen
- Handlungssicherheit
- Stabilität



Architekturziel 2: Flexibilität

- Varianten von Geschäftsprozessen
- Geänderte Anforderungen
- Serviceorientierung und Skalierbarkeit
- Baukastenprinzip

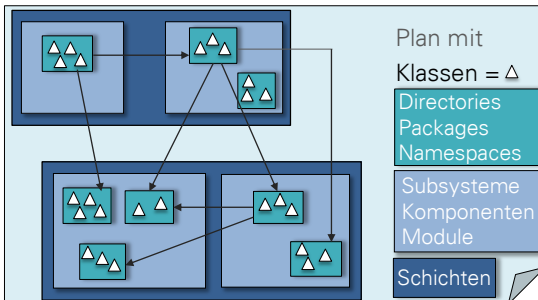


Architekturanalyse: Was ist das?

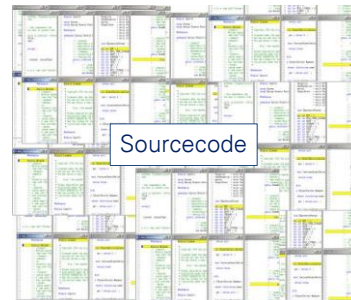


Findet sich die geplante Architektur (Soll-Architektur) in der Strukturen der implementierten Software (Ist-Architektur) wieder?

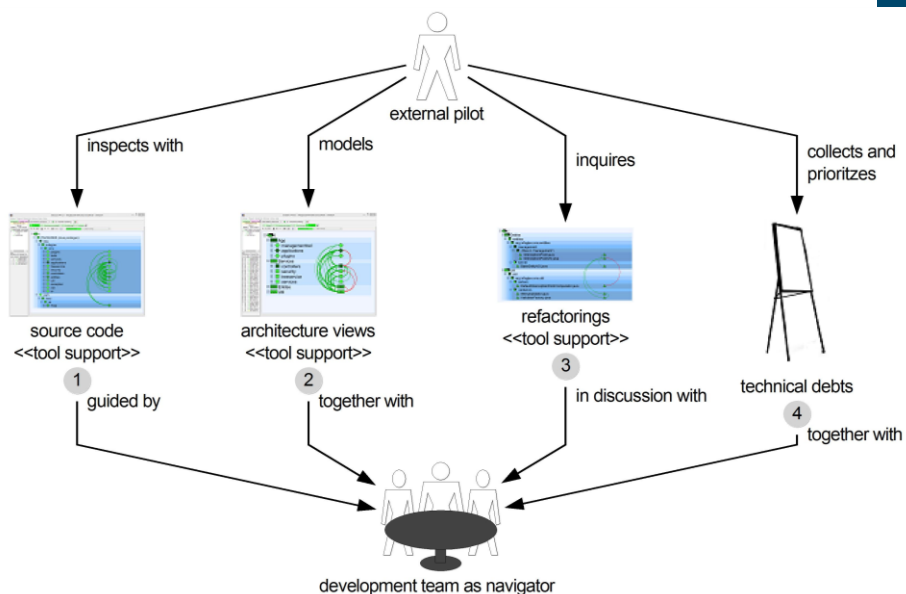
Soll-Architektur



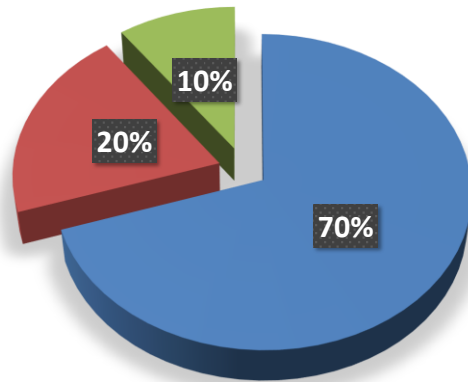
Ist-Architektur



Mob Architecting – Architekturanalyse im Team

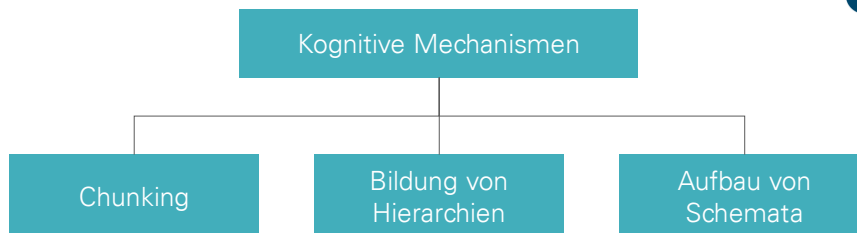


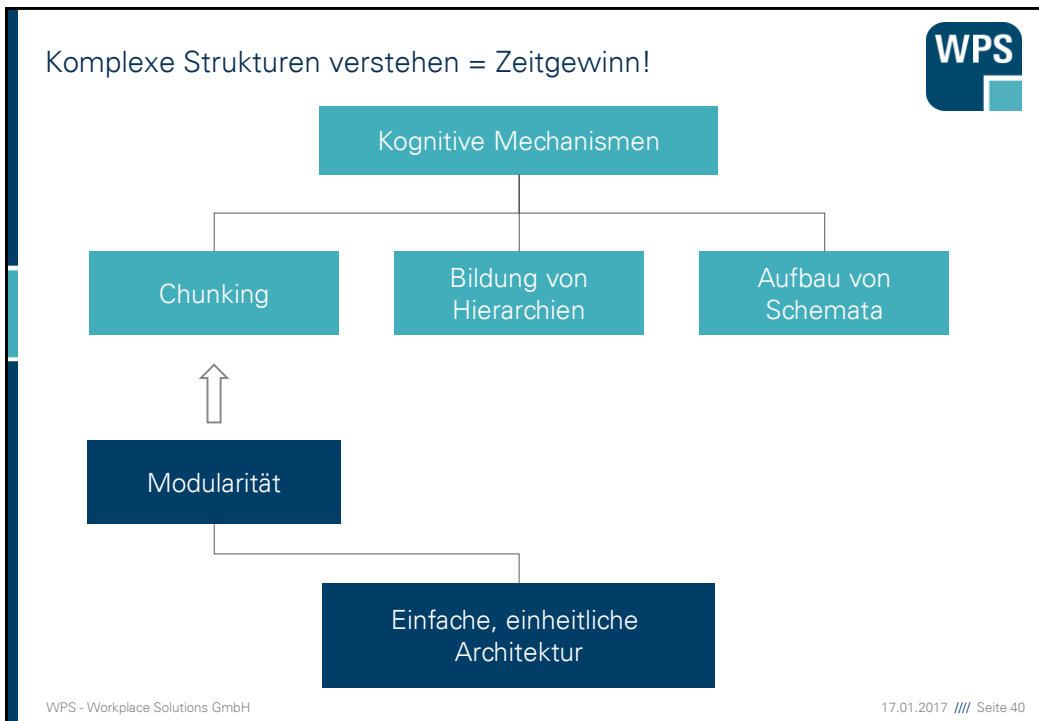
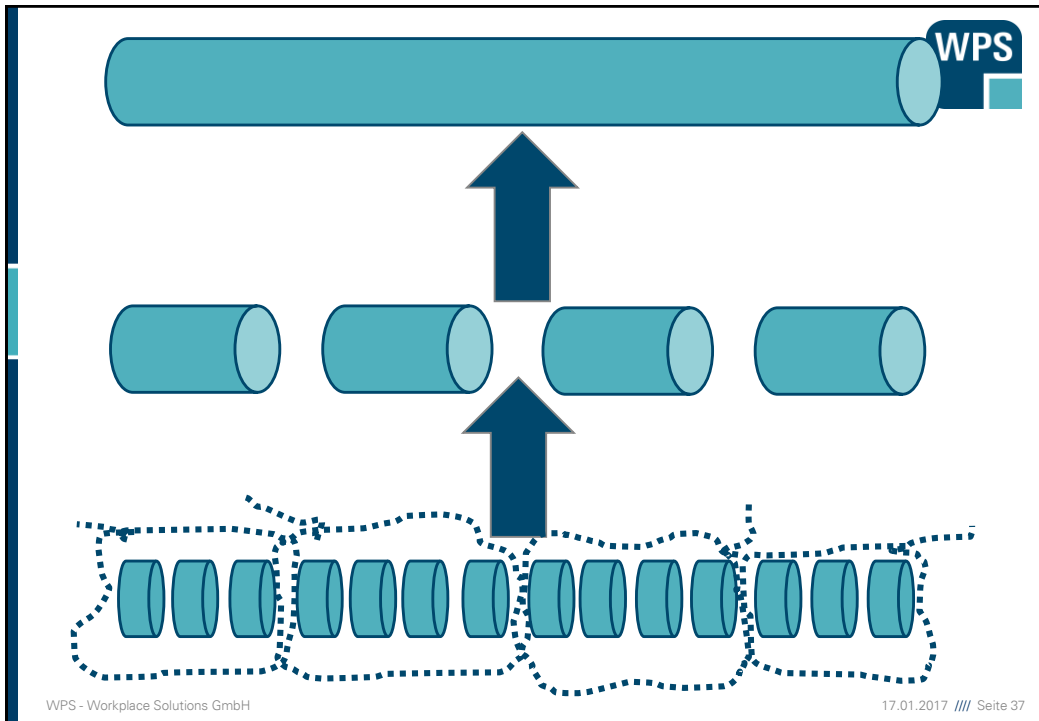
Womit verbringen wir unsere Zeit? - revisited



- Code verstehen
- Problem lösen
- Code schreiben

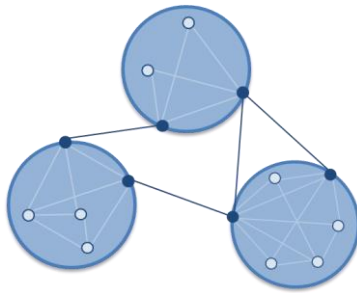
Komplexe Strukturen verstehen = Zeitgewinn!



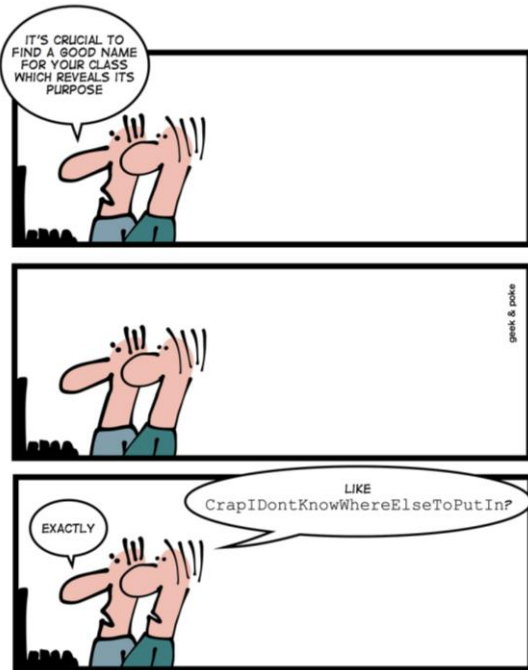


Modularität

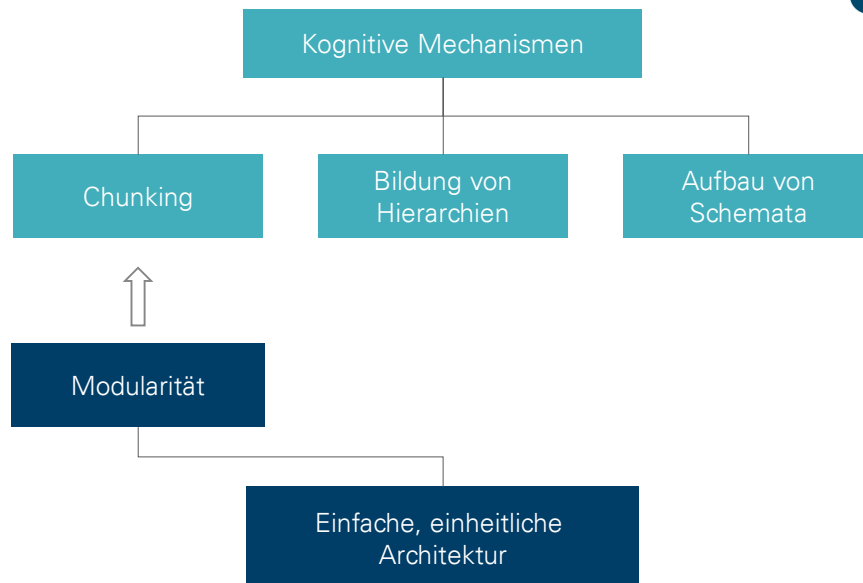
- Hohe Kohäsion und lose Kopplung
- Separation of Concerns
- Single Responsibility Principle



WPS - Workplace Solutions GmbH



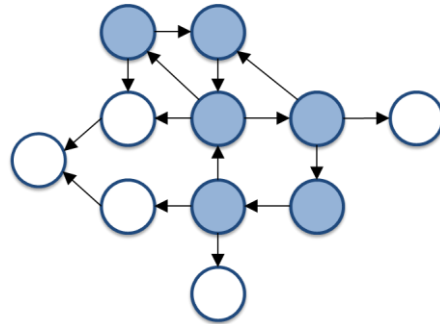
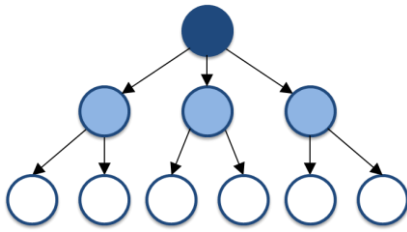
Komplexe Strukturen verstehen = Zeitgewinn!



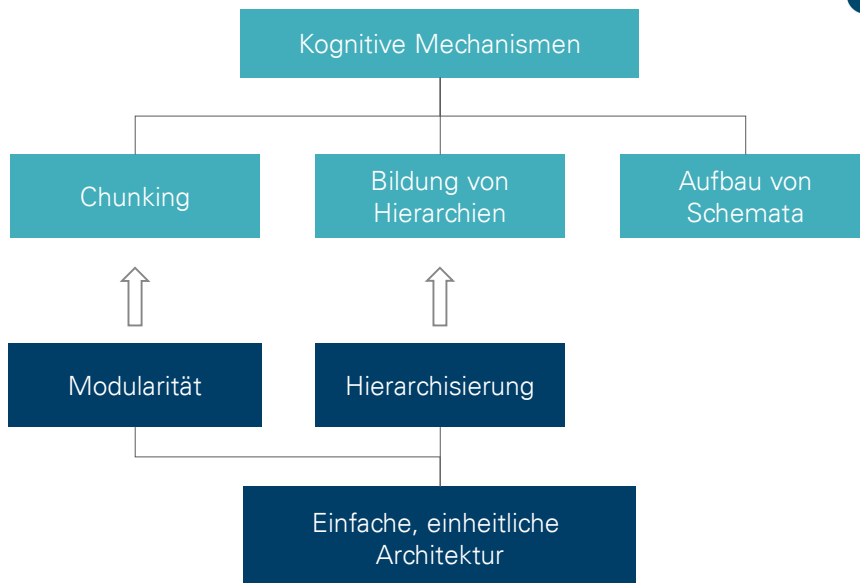
WPS - Workplace Solutions GmbH

17.01.2017 /// Seite 43

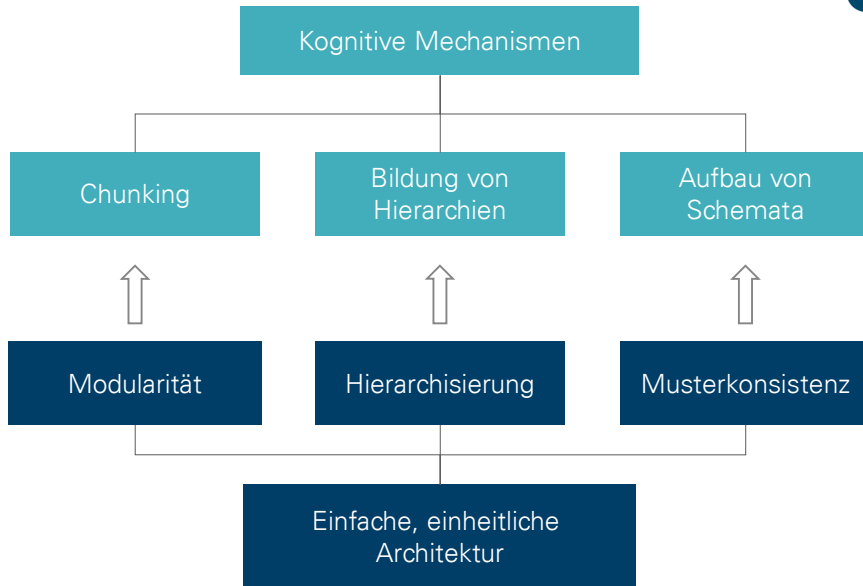
Hierarchien



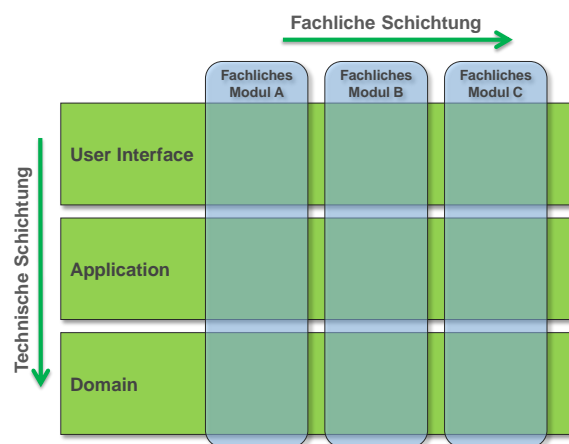
Komplexe Strukturen verstehen = Zeitgewinn!



Komplexe Strukturen verstehen = Zeitgewinn!



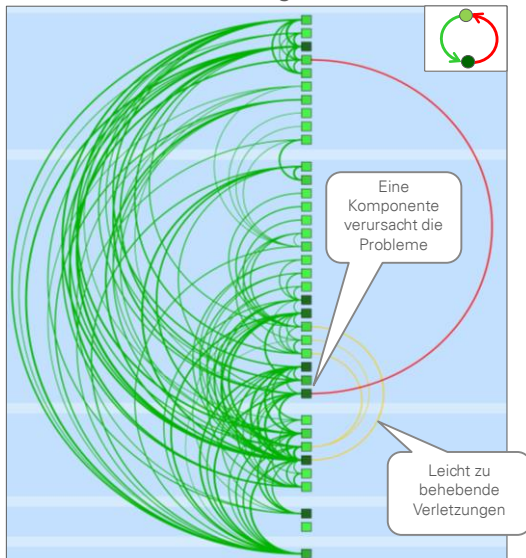
Hierarchien in Architekturebene: Schichten und Module



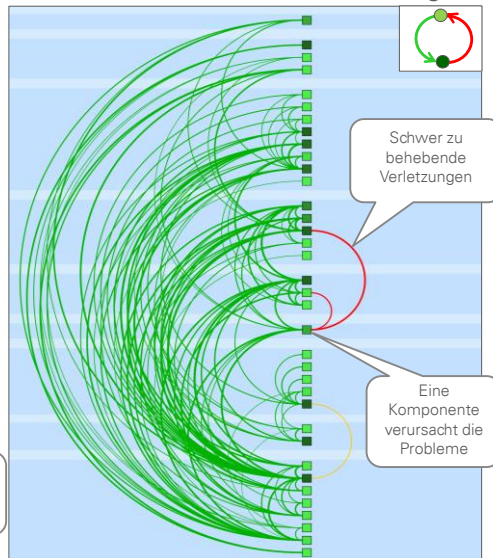
Zwei Dimensionen einer Architektur



Technische Schichtung



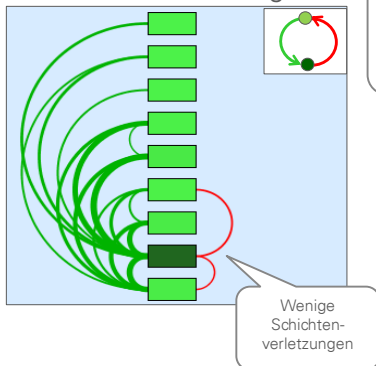
Fachliche Schichtung



Fachliche Schichtung misslungen

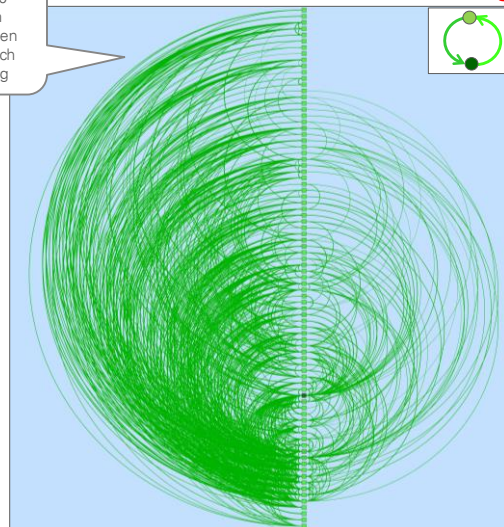


Technische Schichtung



Fast alle 90 fachlichen Komponenten brauchen sich gegenseitig

Keine fachliche Schichtung

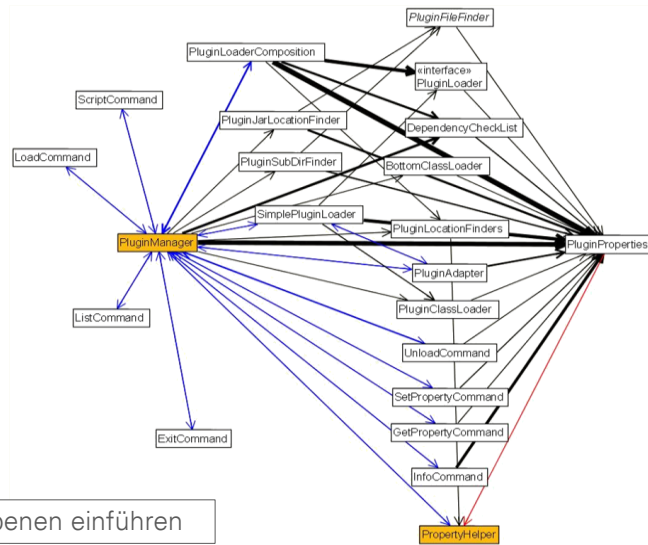


Hierarchien auf Klassenebene

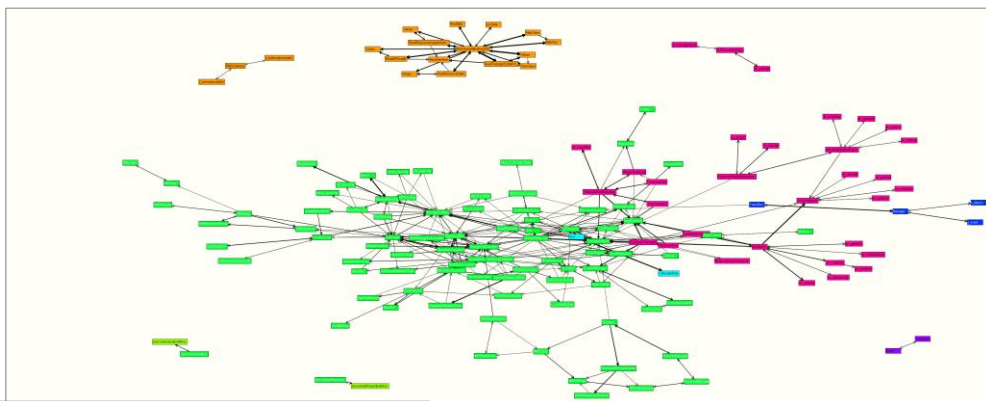


Auswirkung auf:

- Wartbarkeit
- Austauschbarkeit
- Testbarkeit
- Modularität
- Musterkonsistenz

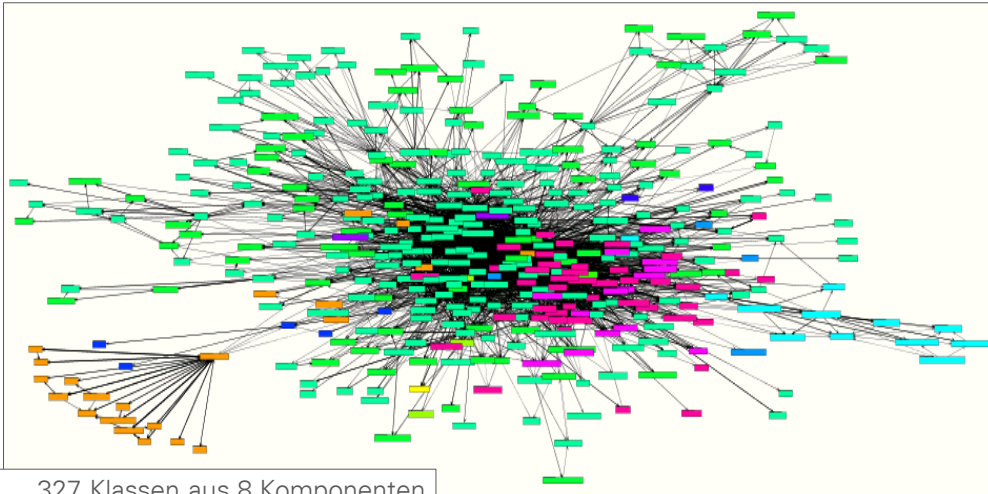


→ Hierarchien auf allen Ebenen einführen

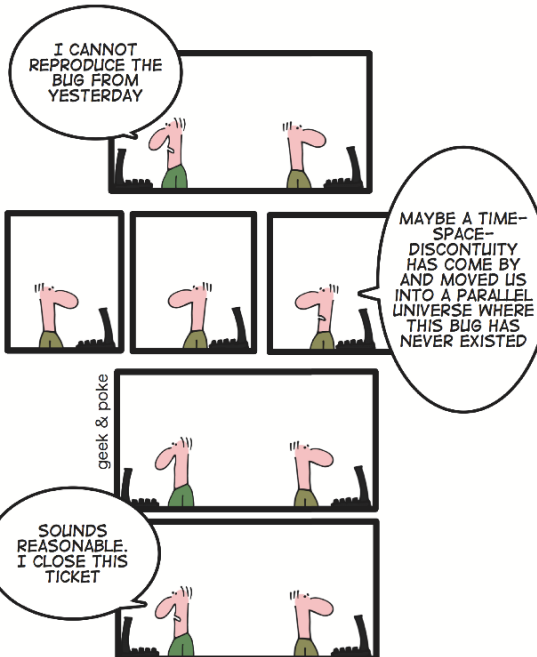


119 Klassen aus 4 Komponenten
+ 28 weitere Klassen

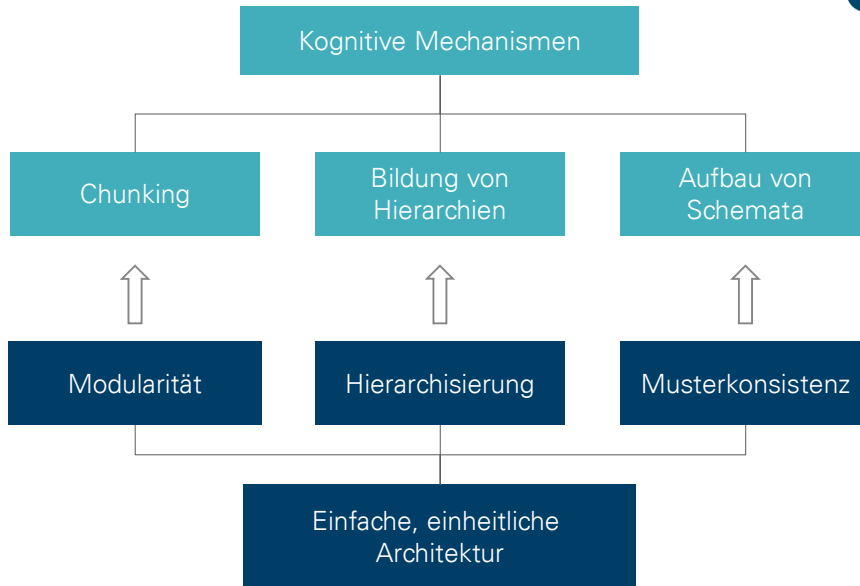
Große Zyklen sichtbar machen



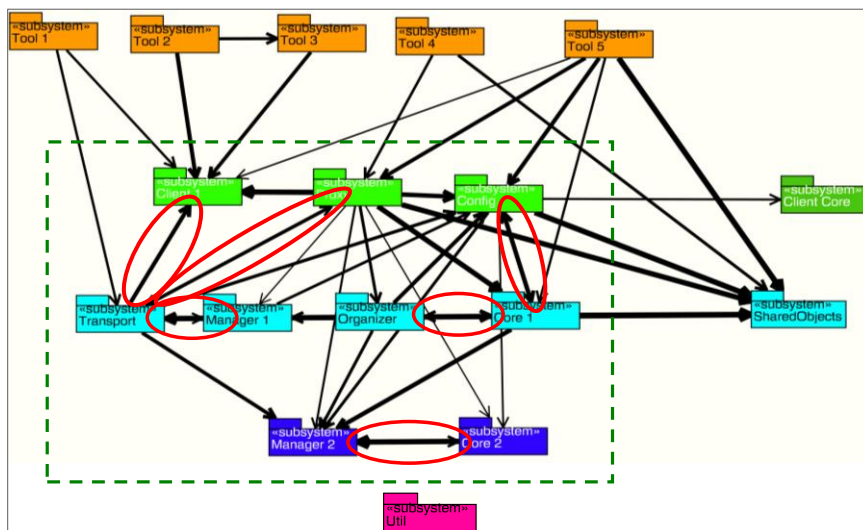
327 Klassen aus 8 Komponenten
brauchen sich gegenseitig



Komplexe Strukturen verstehen = Zeitgewinn!



Uneven Modules

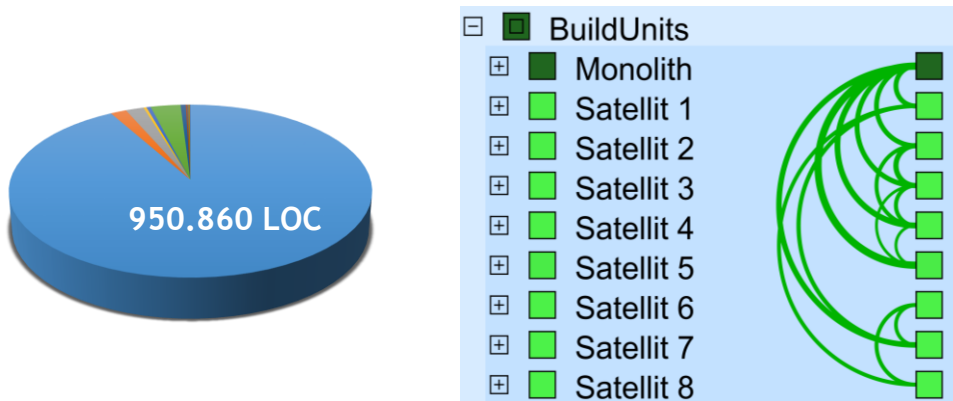


9 Komponenten = 17 Subsysteme

Uneven Modules

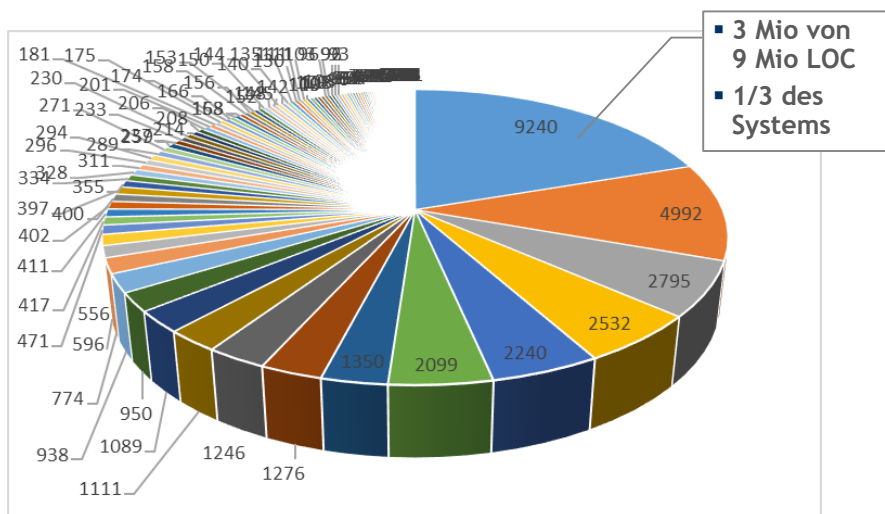


Ein großer Monolith mit vielen kleinen Satelliten

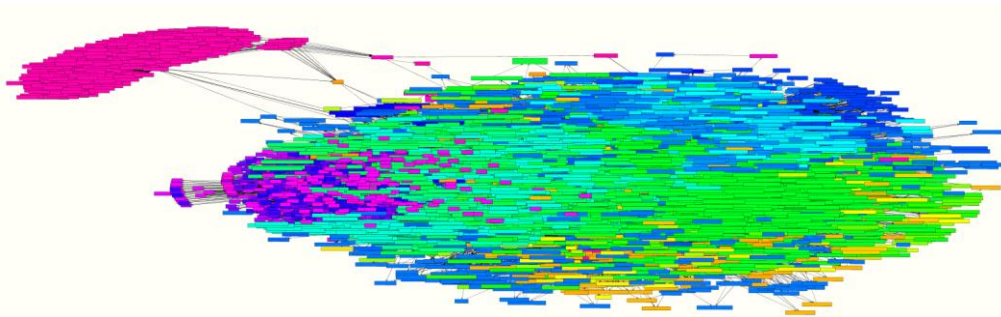


Häufige Ursache: Build-System, das Zyklen verbietet

128 Build Units mit Anzahl Klassen



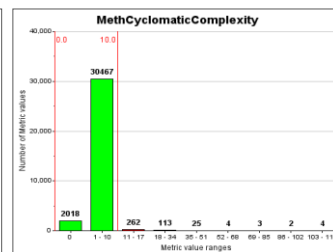
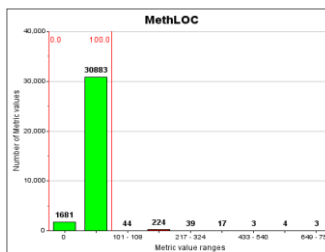
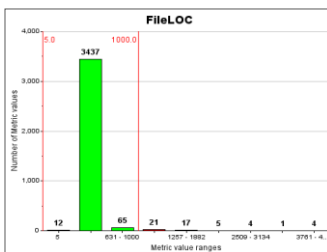
5.270 Klassen in einem Zyklus in einer BuildUnit



Modularität: Ausgewogene Größenverhältnisse



- Ist das System auf den verschiedenen Ebenen ausgewogen?
- Welche Code-Abschnitte fallen durch ihre Größe besonders auf?

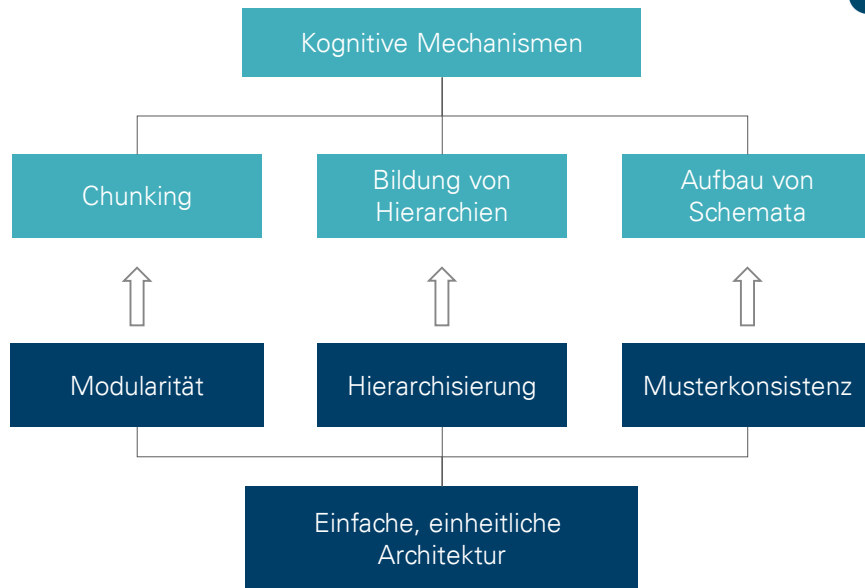


Typische Metriken:

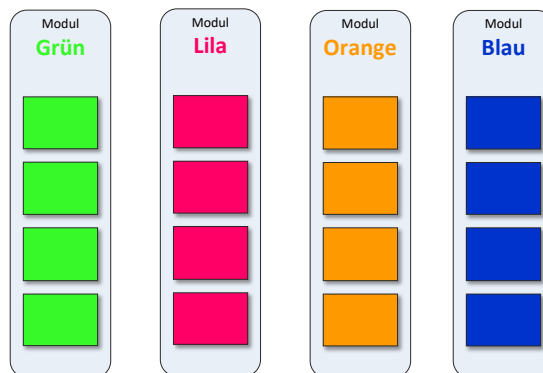
- LOC pro Methode, Klasse, Package, Komponenten
- Duplizierter Code
- Zyklomatische Komplexität



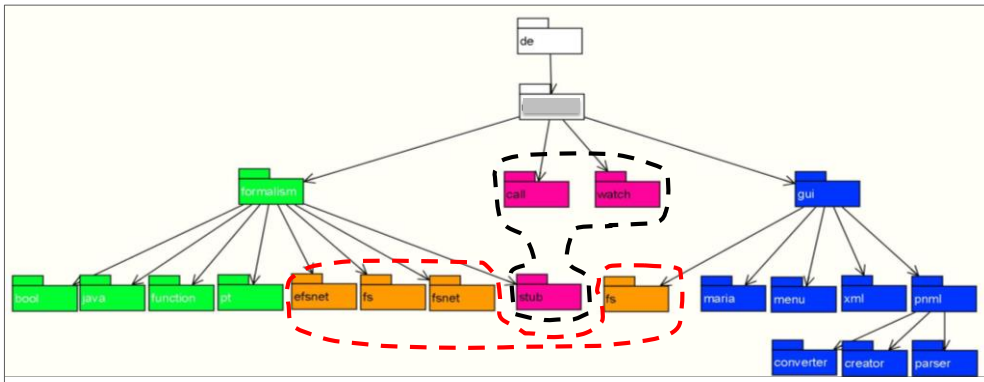
Komplexe Strukturen verstehen = Zeitgewinn!



Muster auf Architekturebene: Vier Module

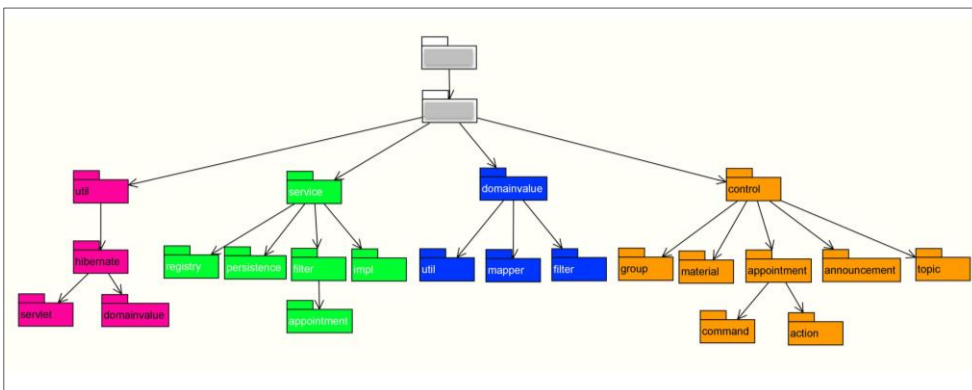


Musterkonsistenz: Was finden wir?

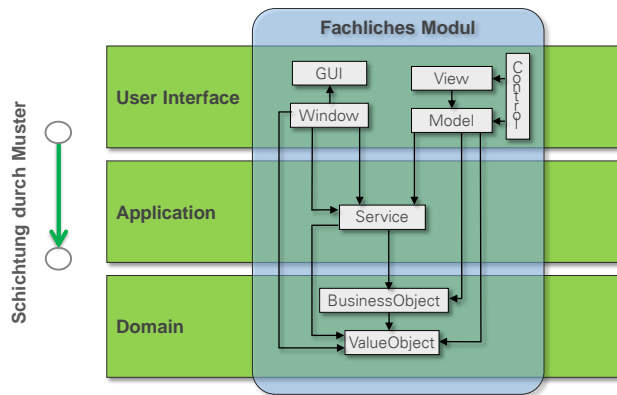


- Ist die Abbildung der Architektur in der Struktur des Codes zu erkennen?

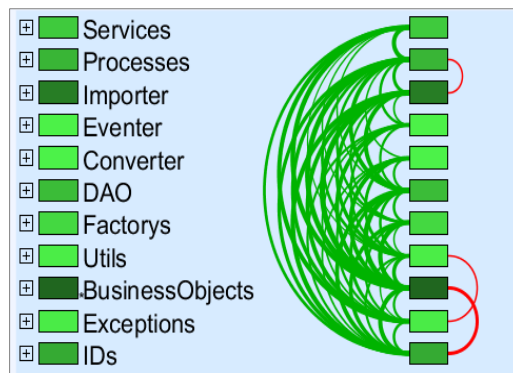
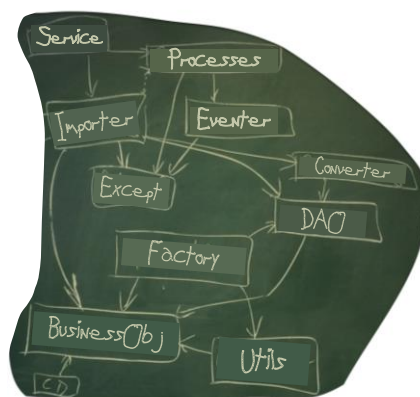
Muster sinnvoll eingesetzt



Muster auf Klassenebene: Entwurfsmuster und Mustersprachen

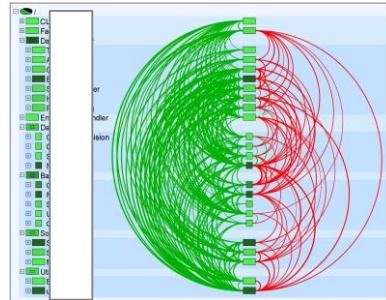
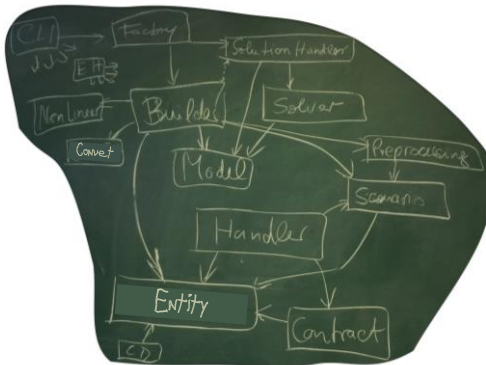


Gute umgesetzte Mustersprache



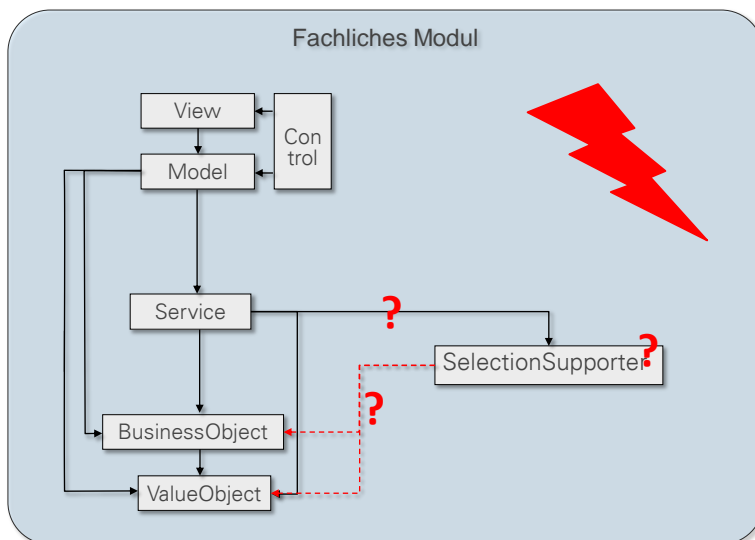
- ☺ 90% des Sourcecodes lässt sich den Mustern zuordnen
- ☺ 0,1% Verletzungen in den Mustern

Entdeckung einer Mustersprache

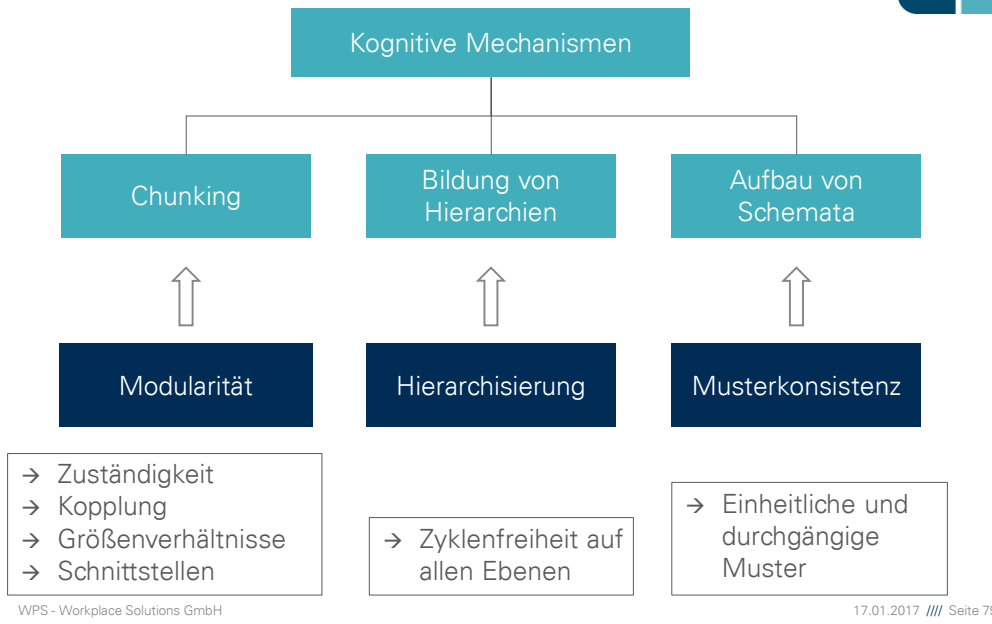


- 😊 80% des Sourcecodes lässt sich den 23 Mustern zuordnen
- 😊 4% Verletzungen in den Mustern

Fehlende und unbekannte Muster



Komplexe Strukturen verstehen = Zeitgewinn!



Kostenfreie Werkzeuge

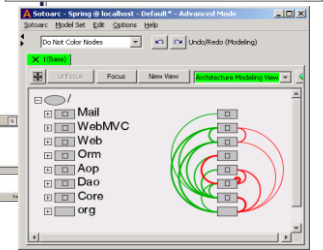
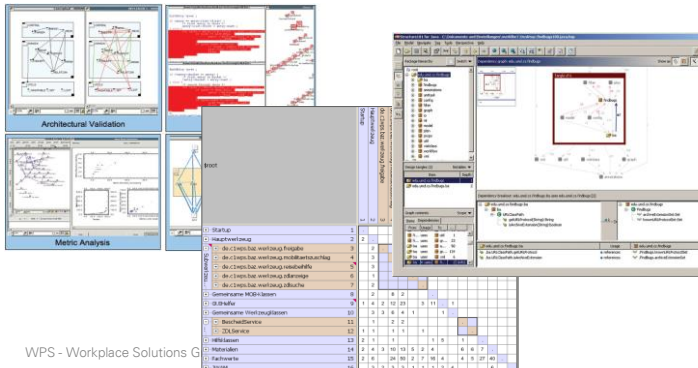


- SonarQube:
 - Leitstand für Qualitätsmetriken
 - Plattform für vielfältige Plugins
- JDepend:
 - wenige Metriken
 - einfache Abhängigkeitsanalyse
- JDepend + Google Architecture Rules:
 - einfache Architekturbeschreibung
- Ndepend/CDepend:
 - Metriken
 - Abhängigkeitsanalyse
- XRadar:
 - Analyse von Java-Projekten via maven
 - Reports bezüglich Komplexität und Architekturverletzungen

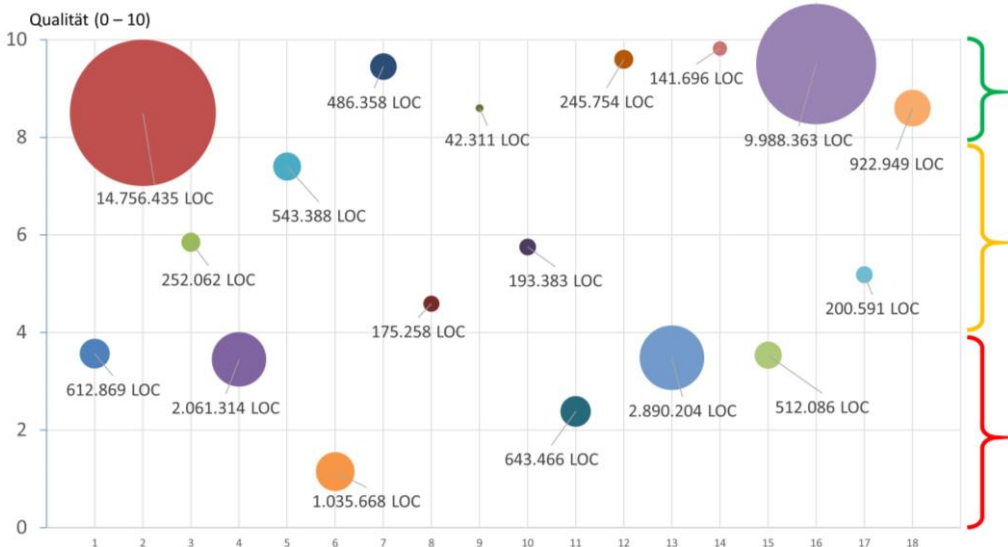


Kommerzielle Produkte

- Axivion Bauhaus: Java, C#, C/C++, Ada, VB und Cobol
- Lattix: Java, .Net, C/C++, Ada, Delphi und DB-Systeme
- Structure101: Java, C++, Ada
- SotoArc und Sonargraph: Java, C#, C/C++, ABAP, PHP
- Teamscale: C#, Java
- Seerene: Java, C#, C/C++, ABAP



Qualität von Softwarearchitekturen - Java, C#, C++, ABAP, PHP -





WPS WORKPLACE SOLUTIONS

Diplom-Informatikerin
Dr. Carola Lilienthal
Geschäftsführerin

WPS - Workplace Solutions GmbH
Hans-Henny-Jahns-Weg 29
22385 Hamburg
Telefon +49 40 229 499-0
Telefax +49 40 229 499-299
Mobil +49 170 184 77 11
cl@wps.de
www.wps.de

@caiolali

@caiolali
 cl@wps.de
www.lisa.de

Schulungen zu Architektur und Domain-Driven Design



wps.de/schulung