

# **SOFTWARE ARCHITEKTUREN GEZIELT ANALYSIEREN, DOKUMENTIEREN UND VERBESSERN MIT SOFTWARE ANALYTICS**

**Dev Day**

**24.05.2022**

**Stephan Pirnbaum @spirnbaum**

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

**BUSCHMAIS**

## ^ Wer wir sind

- ^ Dresdner IT-Beratungsunternehmen, gegründet im Jahre 2008

## ^ Unsere Schwerpunkte

- ^ Architekturberatung und Entwicklung moderner Geschäftsanwendungen
- ^ Software-Qualitätsanalysen und -Sicherung
- ^ Köpfe hinter jQAssistant: Entwicklung, Trainings, Workshops

## ^ Unsere Kunden

- ^ Kleine Auswahl: ITZBund, Sächsische Aufbaubank-Förderbank, ASML, GlobalFoundries, Thyssenkrupp Steel, Deutsche Telekom, COOP Schweiz

Wenn ich diesen Code ändere, weiß ich nicht,  
was ich sonst noch einreiße!

Eine Neuimplementierung ist schneller als  
eine Änderung des bestehenden Codes!

Die Komplexität der Umsetzung eines Features  
ist höher als die fachliche Komplexität!

## WARUM?

Test- und Releaseprozesse sind  
langwierig und fehleranfällig!

Die Anwendung lässt sich nicht  
horizontal skalieren!

Das Anlernen neuer Kollegen  
bindet zu viele Ressourcen!

Die Entwicklung verschiedener Domänen lässt sich  
nicht auf unterschiedliche Teams verteilen!

**UM DIE PROBLEME ZU LÖSEN,  
MÜSSEN WIR DIE URSACHEN VERSTEHEN!**

**ES WAR EINMAL...**

**EIN GREENFIELD-PROJEKT**

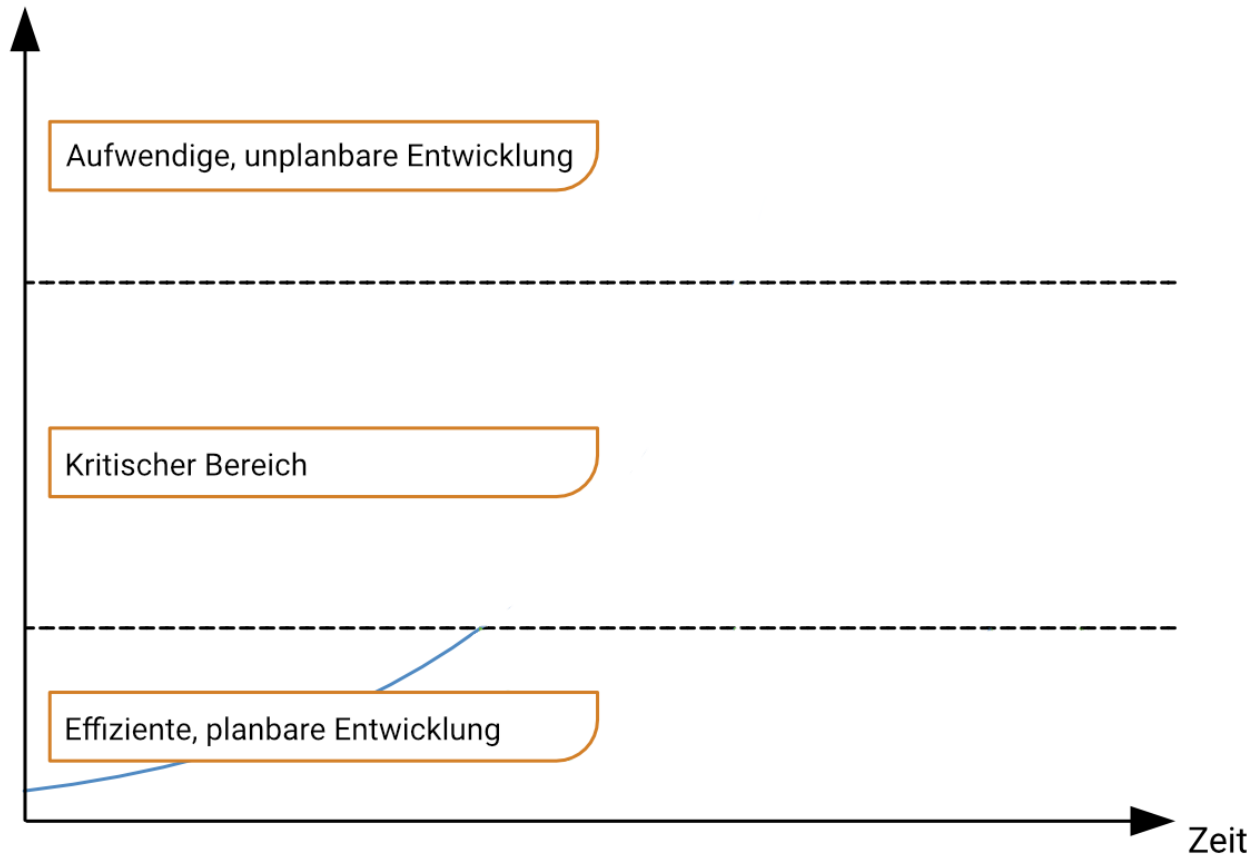
## **Chancen**

- ▲ Geringe fachliche Komplexität
- ▲ Kleine Source Code Basis
- ▲ Leichte Änderbarkeit des Codes

## **Risiken**

- ▲ Geringe Erfahrung mit Use Cases
- ▲ Unbekannte Anforderungen
- ▲ Notwendige Architekturdefinition

Technische  
Schulden



**Legende:**

-----  
Aufbau technischer Schulden

**DAS WUCHS SEHR SCHNELL...**

**UND WURDE ZUM BROWNFIELD-PROJEKT**





<http://www.quickmeme.com/p/3vpkh8>

Funktionelles Wachstum

Organisatorisches Wachstum

Technisches Wachstum

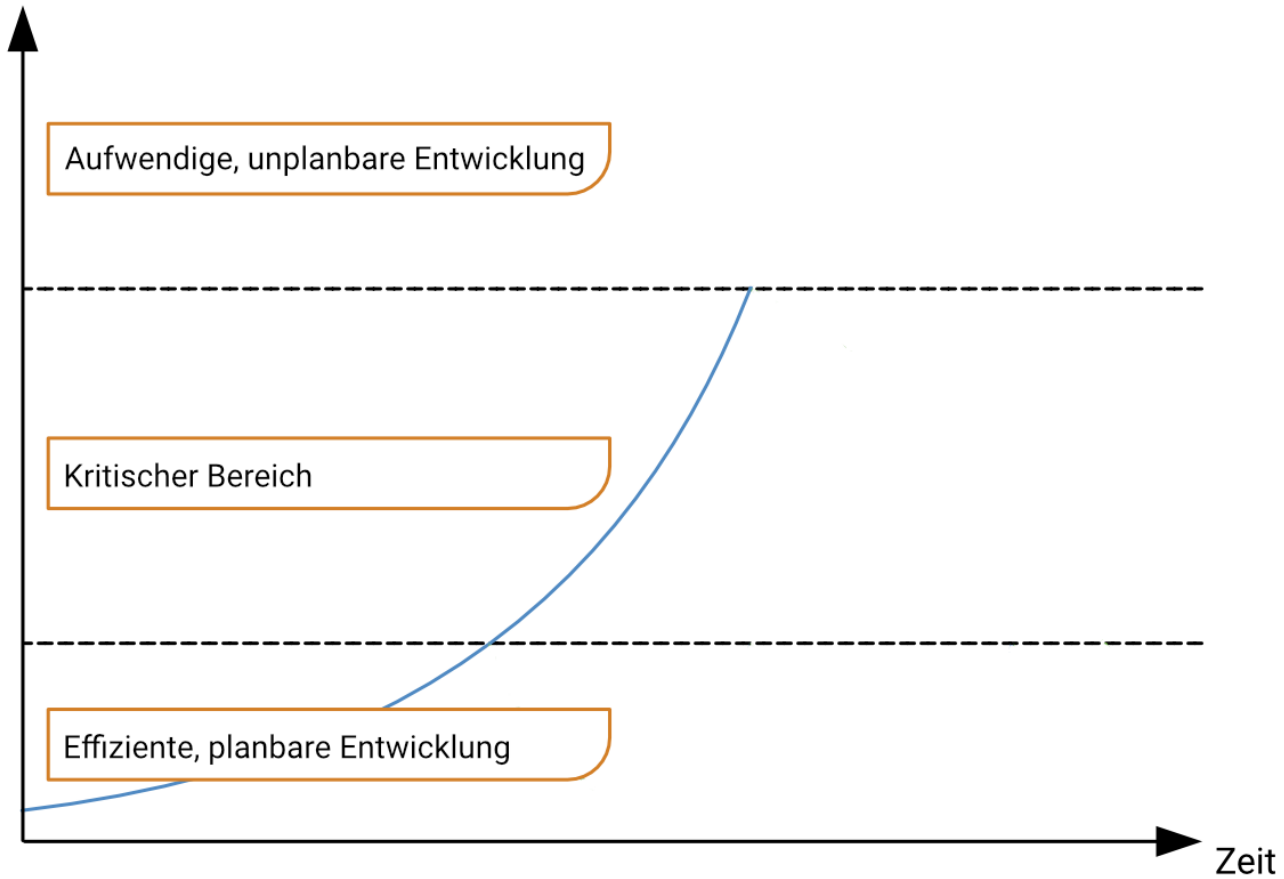


- ▲ Langsames, qualitatives Wachstum
- Geringere Architekturerosion



- ▲ Schnelles, quantitatives Wachstum
- Stärkere Architekturerosion

Technische  
Schulden



**Legende:**

-----  
Aufbau technischer Schulden

**MIT DER ZEIT WURDE ES NICHT BESSER**

## ▲ Innovationszwang durch externe Faktoren

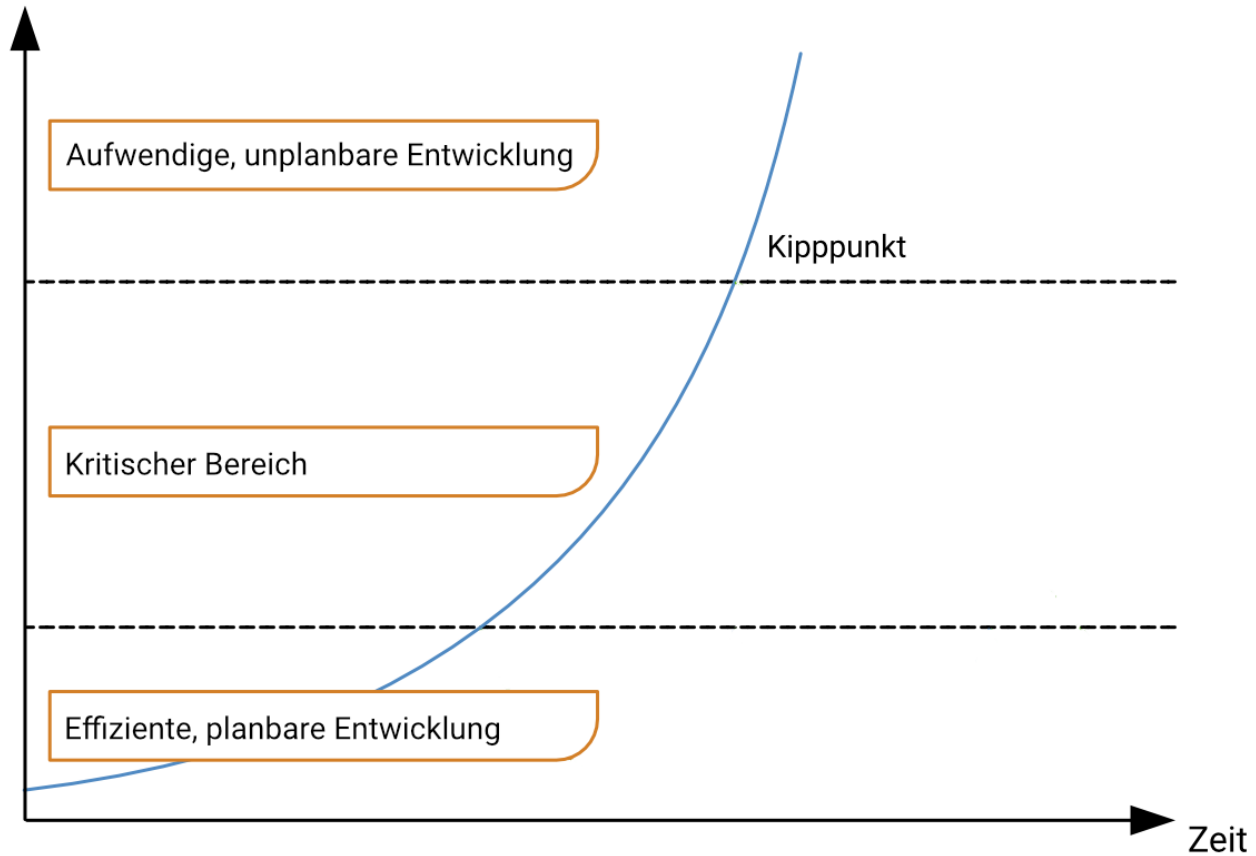
- ▲ Neue Konkurrenten

- ▲ Steigende Kundenanforderungen

- ▲ Andere Geschäftsmodelle

**Kann unser System folgen?**

Technische  
Schulden



**Legende:**

-----  
Aufbau technischer Schulden



<https://imgflip.com/i/3db8rk>



Dokumentierte Architektur (Wiki, PowerPoint, ...)

<>

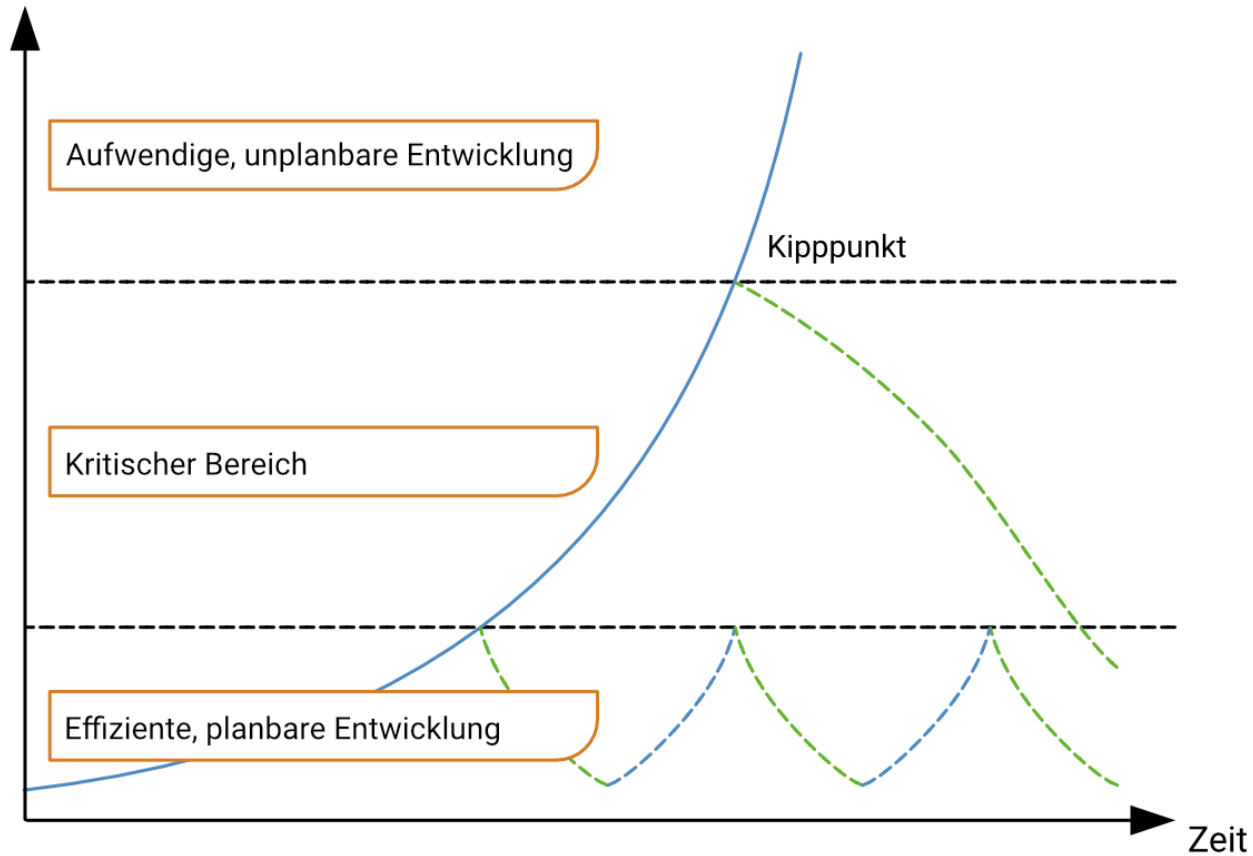
Gefühlte Architektur (Entwickler)

<>

Reale Architektur (Code)

**JEDOCH GIBT ES EINEN AUSWEG**

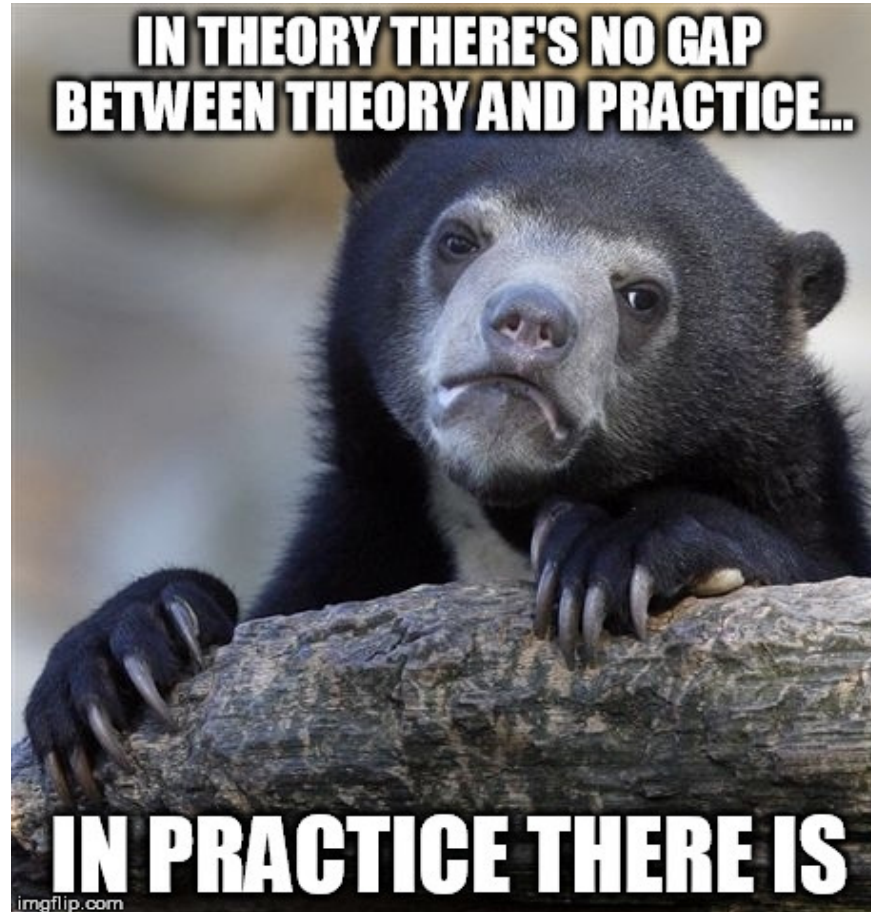
Technische Schulden



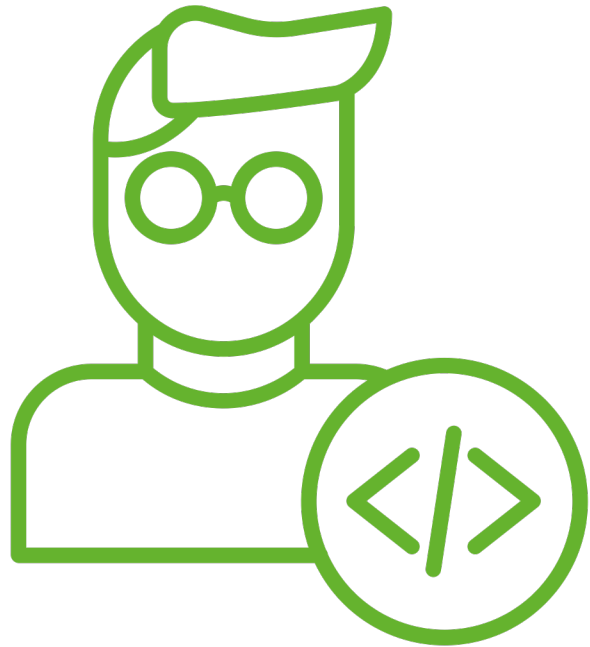
**Legende:**

- Aufbau technischer Schulden
- Abbau technischer Schulden

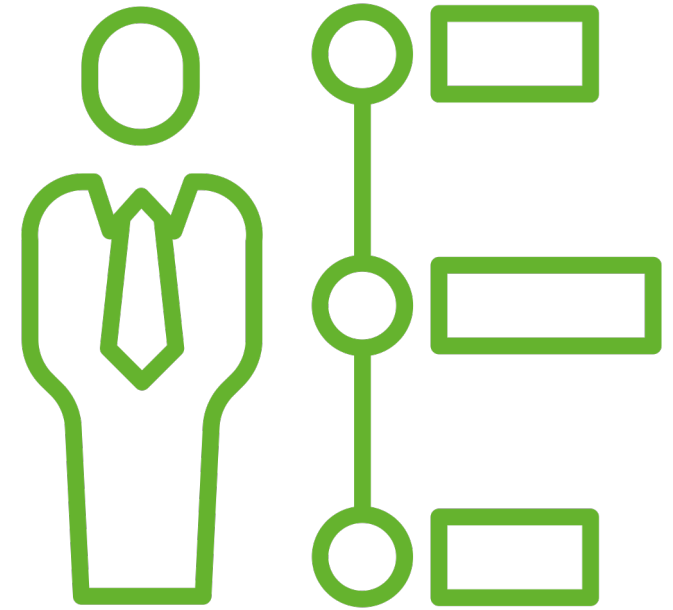
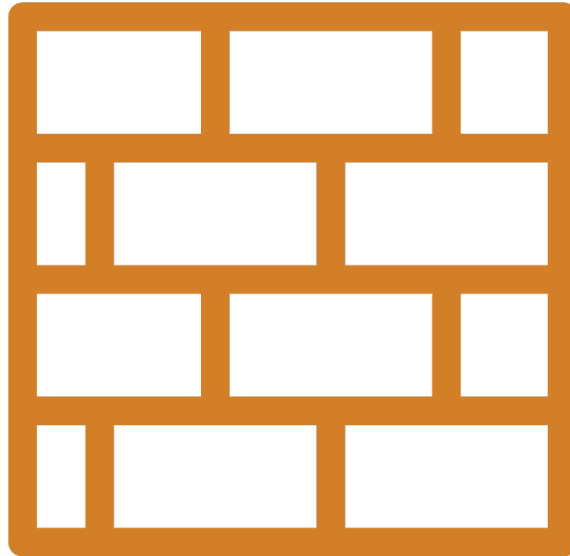
- ▲ Aktives Management und Abbau technischer Schulden
  - ▲ Unter Berücksichtigung der kurz-, mittel- und langfristigen Ziele
  - ▲ Als fester Bestandteil der Projektplanung
  - ▲ Gezielt bei kritischen Bereichen
  - ▲ Mit gleichzeitiger Re-Dokumentation



<https://imgflip.com/i/23eyxo>



Wall of Ignorance



Icons made by [Kiranshastry](#) from [Flaticon](#)

**WICHTIG**

**Proper Preparation Prevents Poor Performance**



<https://builder.cheezburger.com/Builder/RenderPreview/65f84df1-bae0-4cb8-9602-5205a7caf98d>



# SOFTWARE ANALYTICS

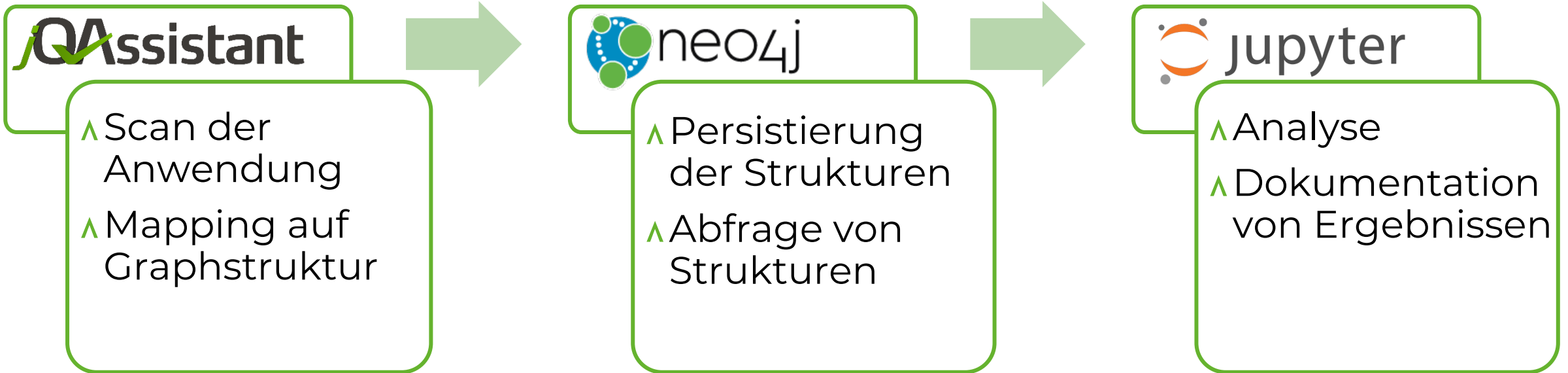
- ▲ Data Analytics für Software Systeme
  - ▲ Extraktion und Zusammenführung von Informationen aus
    - ▲ Source Code
    - ▲ Statischen und dynamischen Eigenschaften
    - ▲ Entwicklungshistorien
  - ▲ Schlussfolgern von neuen Informationen zum Aufbau von Wissen

## ^ Dokumentation von Erkenntnissen

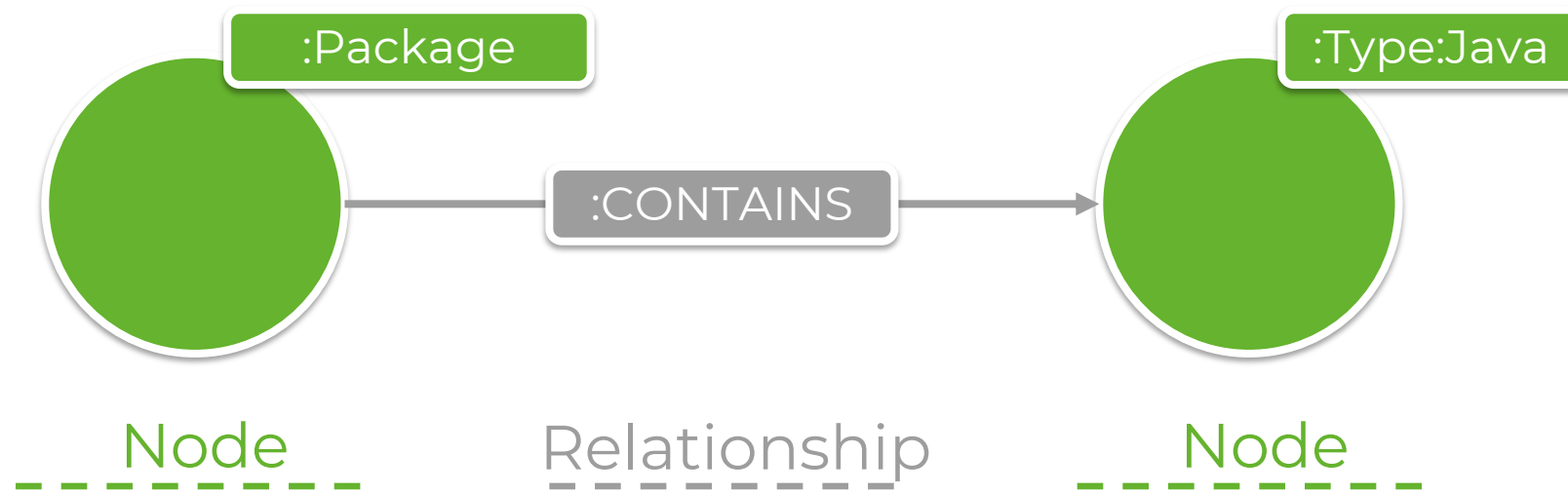
- ^ Unterstützung von Planung und Refaktorisierung
- ^ Referenz für (neue) Entwickler
- ^ Kommunikationsgrundlage für Entwickler und Entscheider



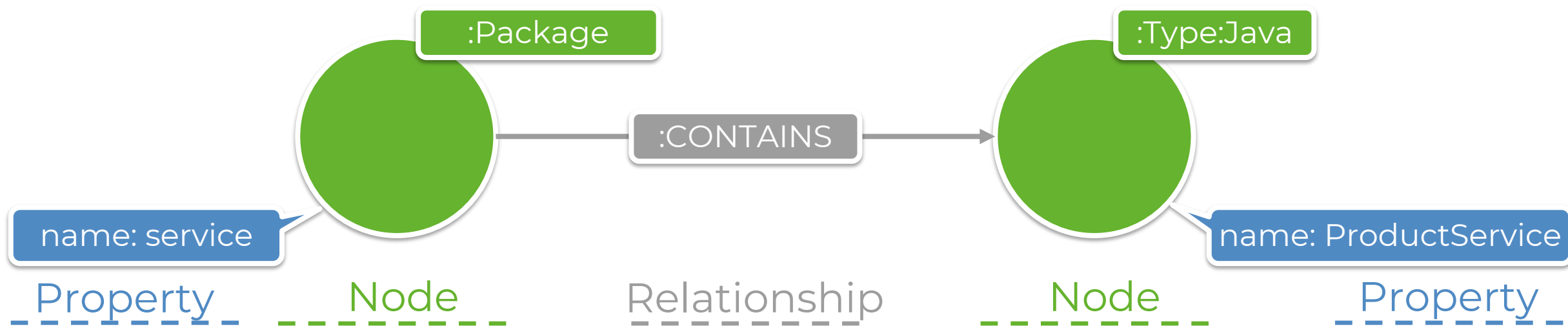
<https://www.kino.de/serie/hoer-mal-wer-da-haemert-1999/news/hoer-mal-wer-da-haemert-tim-allen-bringt-neue-folgen-ins-gespraech/>



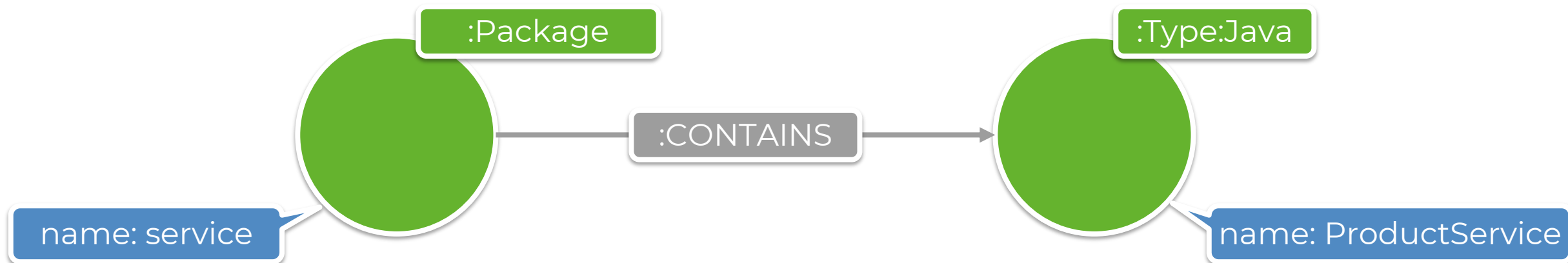
# ... MIT JQASSISTANT UND NEO4J



# ... MIT JQASSISTANT UND NEO4J



# ... MIT JQASSISTANT UND NEO4J



```
MATCH (p:Package)-[:CONTAINS]->(t:Type:Java)
WHERE p.name = "service" AND t.name = "ProductService"
RETURN p, t
```



# ... MIT JUPYTER

jupyter Software Analytics Intro Last Checkpoint: letzten Freitag um 14:21 Uhr (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 O

11 bounded\_context\_connections = bcRelations.get\_dataframe()  
52 rows affected.

```
In [27]: 1 text = Template(open('vis/chord/chord-diagram.html', 'r').read().replace("\n", "").substitute({
2         'chord_data': bounded_context_connections.to_csv(index = False).replace("\r\n", "\n").replace("\n", "\\n"),
3         'container': 'bc-chord-diagram'})
4
5 HTML(text)
```

Out[27]:

Slide Type | Slide

Vorbereitung der Repository-Analysen

Slide Type |










<https://makeameme.org/meme/possibilities-are-endless-59fa8f>

1. Welche Frage möchte ich beantworten?
  2. Mit welchen Daten kann ich diese ausreichend beantworten?
  3. Welche Annahmen vereinfachen die Analyse?
  4. Für wen und wie müssen die Daten aufbereitet werden?
  5. Wie kann die Analyse umgesetzt werden?
- Beantwortung vor der eigentlichen Analyse

6. Was sind die wichtigsten Ergebnisse der Analyse?
7. Was sind die nächsten Schritte?
- Beantwortung nach der Analyse

# Software Analytics Canvas

Project: \_\_\_\_\_

<p> <b>1. Question</b></p> <p><i>What is it that we want to know about the software / processes / usage / organization / etc.?</i></p>	<p> <b>2. Data Sources</b></p> <p><i>Which data can possibly answer our question? What information do we need?</i></p>	<p> <b>3. Heuristics</b></p> <p><i>Which assumptions do we want to make to simplify the answer to our question?</i></p>	<p> <b>4. Validation</b></p> <p><i>What results do we expect from our analysis, how are they reviewed and presented in an understandable way?</i></p>
<p> <b>5. Implementation</b></p> <p><i>How can we implement the analysis step by step and in a comprehensible way?</i></p>	<p> <b>6. Results</b></p> <p><i>What are the main insights from our analysis?</i></p>	<p> <b>7. Next Steps</b></p> <p><i>What follow-up actions can we derive from the findings? Who or what do we need to address next?</i></p>	

Software Analytics Canvas v1.0 designed by Markus Harrer. Visit <https://www.feststelltaste.de/software-analytics-canvas/> for more information. CC BY-SA 4.0

# SOFTWARE ANALYTICS

**Ein Beispiel**

^ Open Source e-Commerce System „shopizer“

^ Warenkorb

^ Katalog

^ Suche

^ Bestellung

^ ...

^ Fork: <https://github.com/StephanPirnbaum/shopizer/tree/2.12.0-jQA>

- ▲ Szenario (Organisatorisch)
  - ▲ Shop soll stark anwachsen (Funktionsumfang, Nutzerbasis)
    - ▲ z.B. Verkauf digitaler Produkte
  - ▲ Onboarding zahlreicher neuer Entwickler
    - ▲ mitsamt der Strukturierung in fachlich fokussierte Teams
  - ▲ Vorbereitung soll eingeplant und umgesetzt werden



## ^ Ziel

- ^ Identifikation von Wissensträgern
- ^ Abbau und aktives Management von technischen Schulden
- ^ Refaktorisierung der Strukturen hin zu fachlichen Schnitten
- ^ Erstellung von Dokumentation

## ^ Problem

- ^ Verteiltes und fehlendes Wissen über Anwendungsstruktur
- ^ Architektur- und Designentscheidungen nicht nachvollziehbar
- ^ Unbekanntes Nichtwissen („Überraschungen“)
- ^ Zeitrestriktionen
- ^ ...

# SOFTWARE ANALYTICS

**Vorbereitung der Anwendung**

```
<plugin>
  <groupId>com.buschmais.jqassistant</groupId>
  <artifactId>jqassistant-maven-plugin</artifactId>
  <version>1.9.1</version>
  <executions><execution>
    <goals>
      <goal>scan</goal>
      <goal>analyze</goal>
    </goals>
  </execution></executions>
  <.../>
</plugin>
```

```
mvn clean install -DskipTests
```

```
mvn jqassistant:server
```

```
→ localhost:7474
```



<https://www.memecreator.org/meme/lets-get-to-work81>

## ▲ Mögliche Fragestellungen

1. In welche fachlichen Komponenten strukturiert sich die Anwendung und wie hängen diese voneinander ab?
2. Welcher Entwickler ist der Experte für bestimmte Domänen?
3. Welche Code-Teile werden im Vergleich besonders häufig geändert?
4. Welche Module sind besonders instabil/änderungskritisch?
5. Eure eigene Frage? ;)

# SOFTWARE ANALYTICS

**In welche fachlichen Komponenten strukturiert sich die Anwendung und wie hängen diese voneinander ab?**



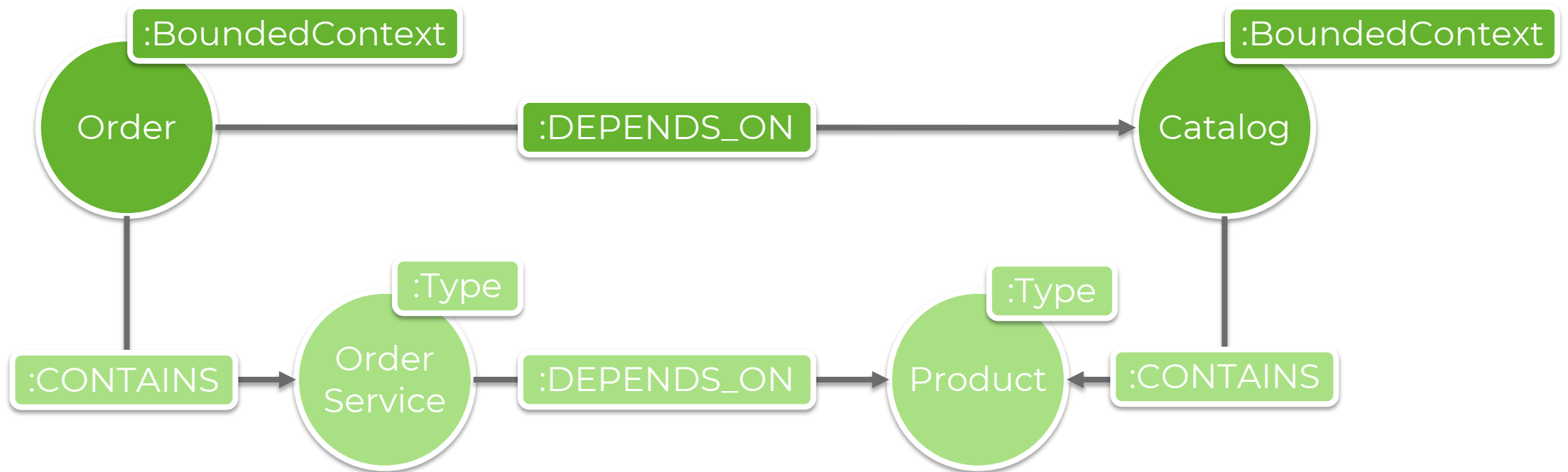
## ▲ Analyse der fachlichen Strukturierung

- Definition und Dokumentation einer fachlichen Strukturierung
- Separierung von Source Code nach fachlicher Zugehörigkeit
- Fokussierung der Entwickler auf kleinere, kohäsive Anwendungsteile
- Planung und Umsetzung konkreter Refaktorisierungsschritte

## ▲ Analyse der fachlichen Strukturierung



## ▲ Analyse der fachlichen Strukturierung



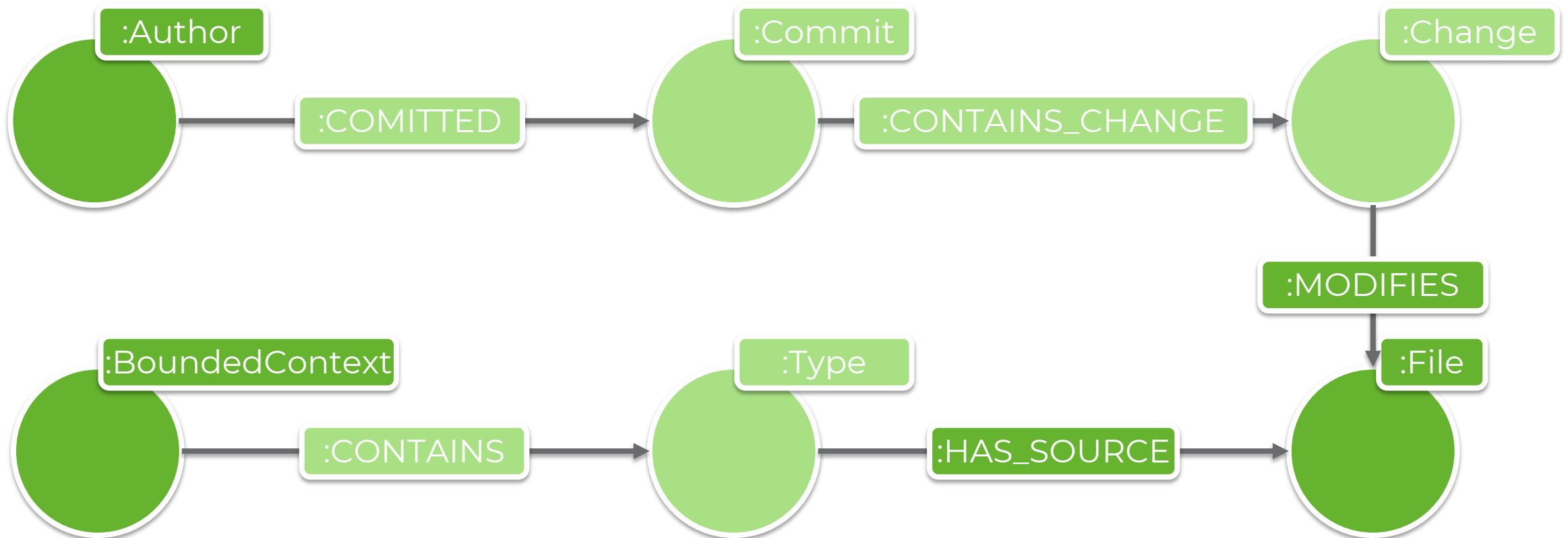
# SOFTWARE ANALYTICS

**Welcher Entwickler ist der Experte für bestimmte Domänen?**

## ▲ Analyse der Code-Ownership

- Definition von fachlichen und technischen Ansprechpartnern
- Dokumentation des vorhandenen Wissens
- Aufbau von Wissen in unbekanntem Bereichen
  - Verstehen und dokumentieren, ggf. Neuimplementierung

## ▲ Analyse der Code-Ownership



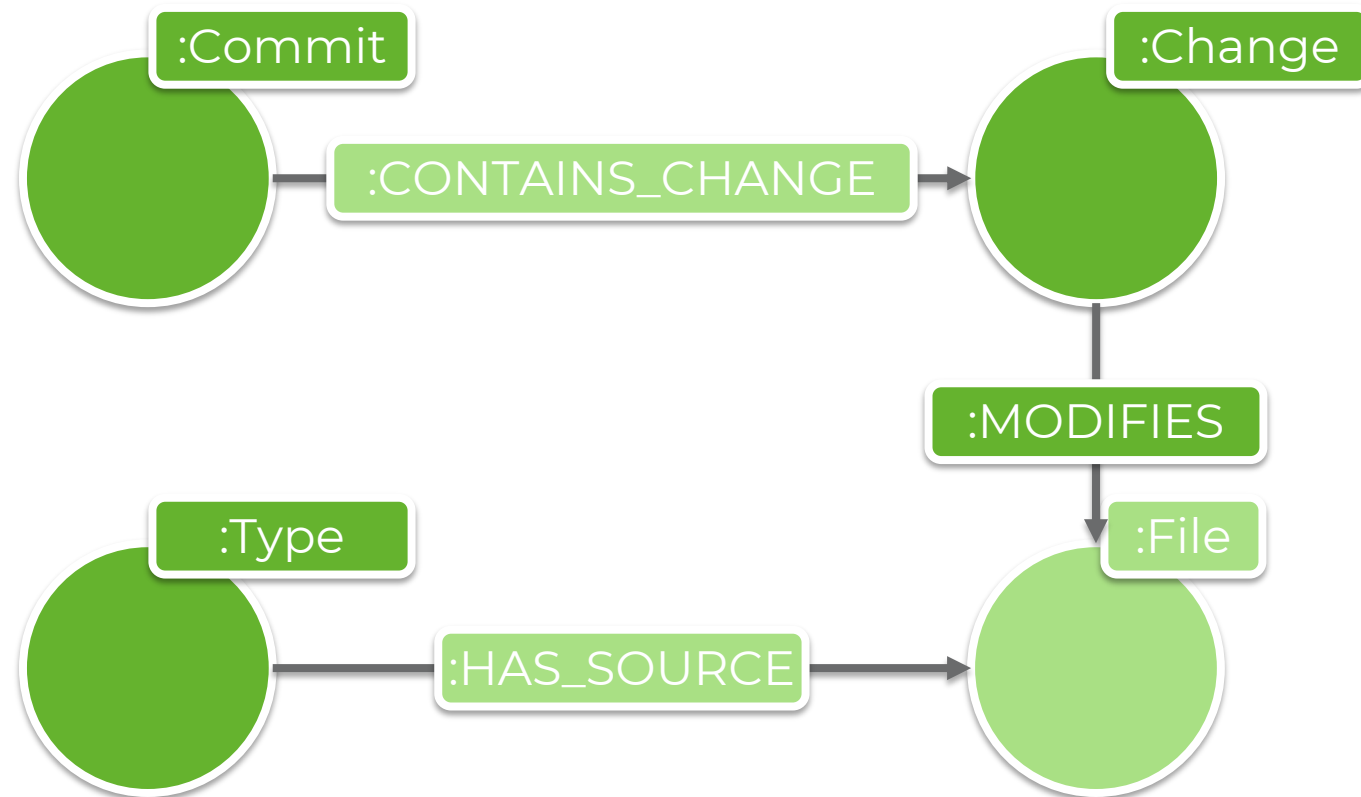
# SOFTWARE ANALYTICS

**Welche Code-Teile werden im Vergleich besonders häufig geändert?**

- ^ Analyse von häufig geändertem Code
  - ^ Identifikation von wartungsintensiven Hot Spots
  - ^ Auffinden von fehleranfälligem Code
  - ^ Aufdecken von Technischen Schulden
  - ^ Finden von schlecht abgegrenzten Verantwortlichkeiten



## ▲ Analyse der Code-Ownership



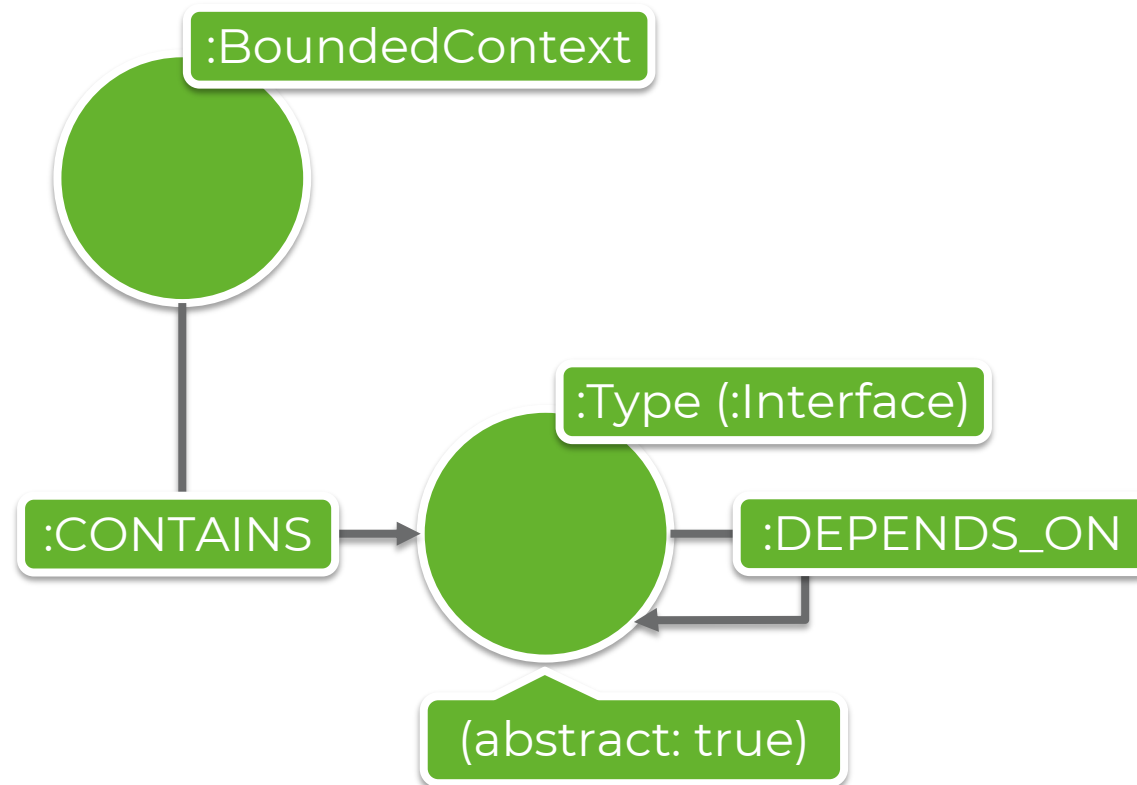
# SOFTWARE ANALYTICS

**Welche Module sind besonders instabil/änderungskritisch?**

## ▲ Analyse von änderungskritischem Code

- ▲ Identifikation von Code, welcher bei Änderung zahlreiche Änderungen in abhängigem Code erfordert
  - ▲ Zeigt fehlende Abstraktheit von Klassen/Packages/Modulen
  - ▲ Deutet auf Instabilen Code hin, also Code, welcher häufig aufgrund anderer Änderungen geändert werden muss

## ▲ Analyse von änderungskritischem Code



# SOFTWARE ANALYTICS

**Eure eigene Frage? ;)**

# LIVE DEMO

<https://github.com/buschmais/software-analytics-starter>

**VIELEN DANK!**

## KONTAKT

Stephan Pirnbaum

[stephan.pirnbaum@buschmais.com](mailto:stephan.pirnbaum@buschmais.com)

+49 351 320923-22



@spirnbaum



stephan-pirnbaum

## BUSCHMAIS GbR

Leipziger Straße 93

01127 Dresden

Tel. +49 351 3209230

[info@buschmais.com](mailto:info@buschmais.com)

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





BUSCHMAIS ist ein Dresdner IT-Beratungsunternehmen, gegründet im Jahre 2008. Unsere Schwerpunkte liegen in der Architekturberatung und der Entwicklung moderner Geschäftsanwendungen.

Gemeinsam mit unseren Kunden analysieren, planen und optimieren wir IT-gestützte Prozesse und unterstützen sie bei der Umsetzung neuer Anforderungen. Dabei arbeiten wir branchenunabhängig und technologiefokussiert in effizienten Teams.

**BUSCHMAIS GbR**

Leipziger Straße 93

01127 Dresden

Tel. +49 351 3209230

info@buschmais.com

www.buschmais.de