



09.02.2021

OOP2021 DIGITAL

Software Analytics

Softwaresysteme datengetrieben analysieren

Markus Harrer
Software Development Analyst
@feststelltaste

INNOQ

**// Tools only
find, people
have to find
out!"**



Markus Harrer

Senior Consultant, Roth

- Architektur-, Design- und Code-Reviews
- Softwaremodernisierung und -sanierung
- Datenanalysen in der Softwareentwicklung

INNOQ



Cards for **A**nalyzing and
Reflecting on **D**oomed
Software



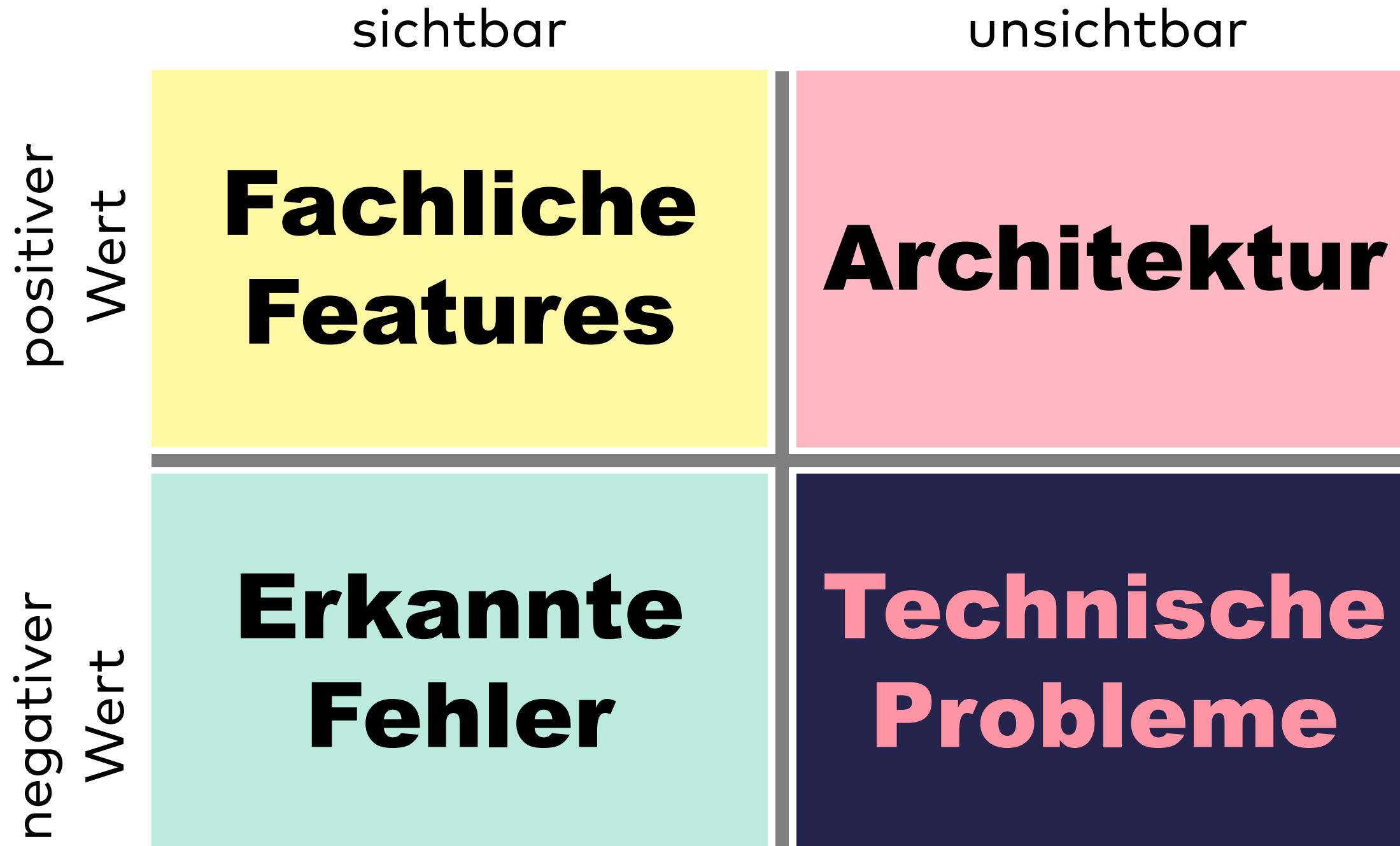
Foundation & IMPROVE



Software Analytics?

@feststelltaste

Problem der technischen Probleme



Management

Risikoappetit

Kommunikationsbarriere

*Software
Analytics*

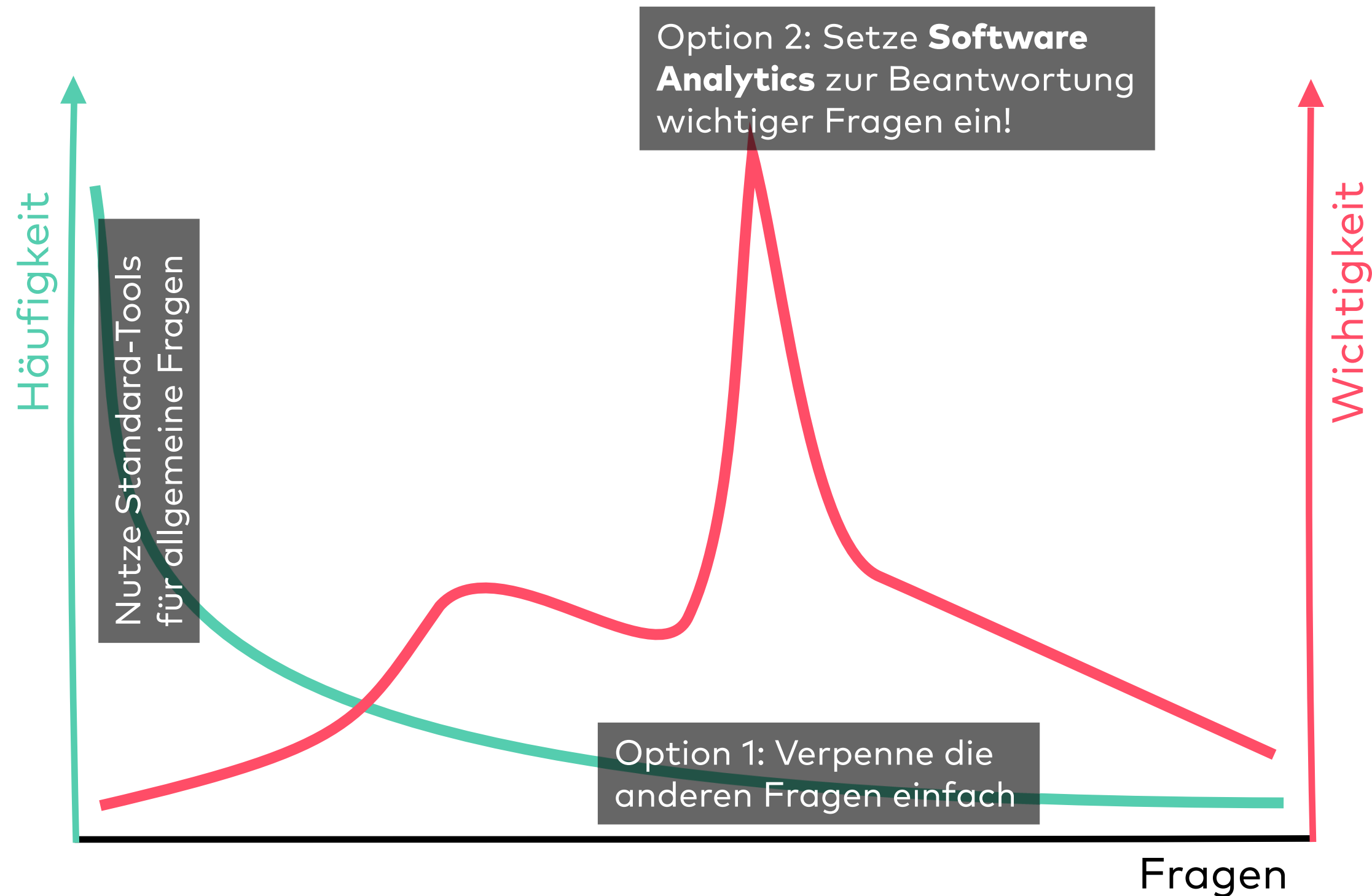
Technische Probleme

Entwicklung

Software Analytics...

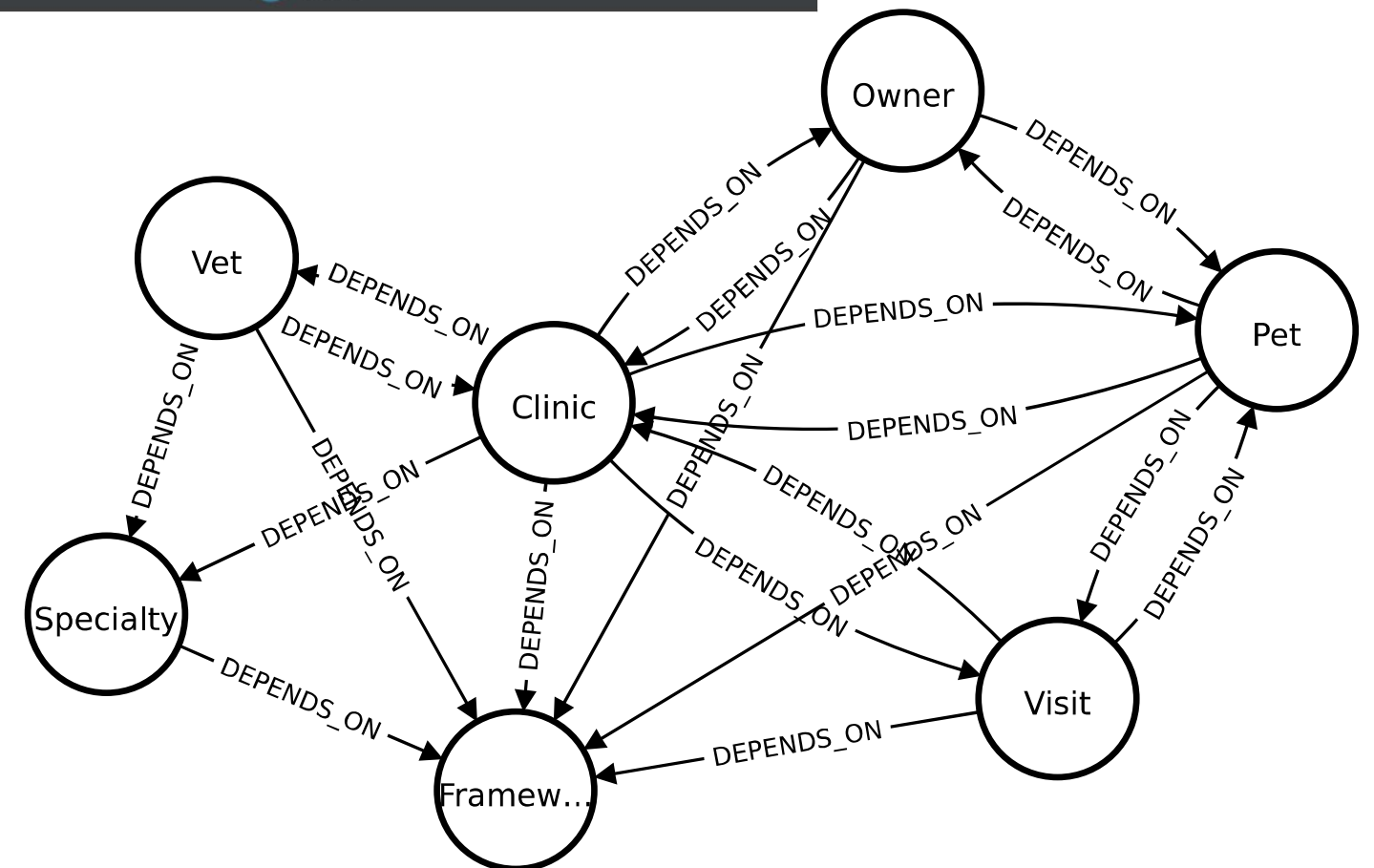
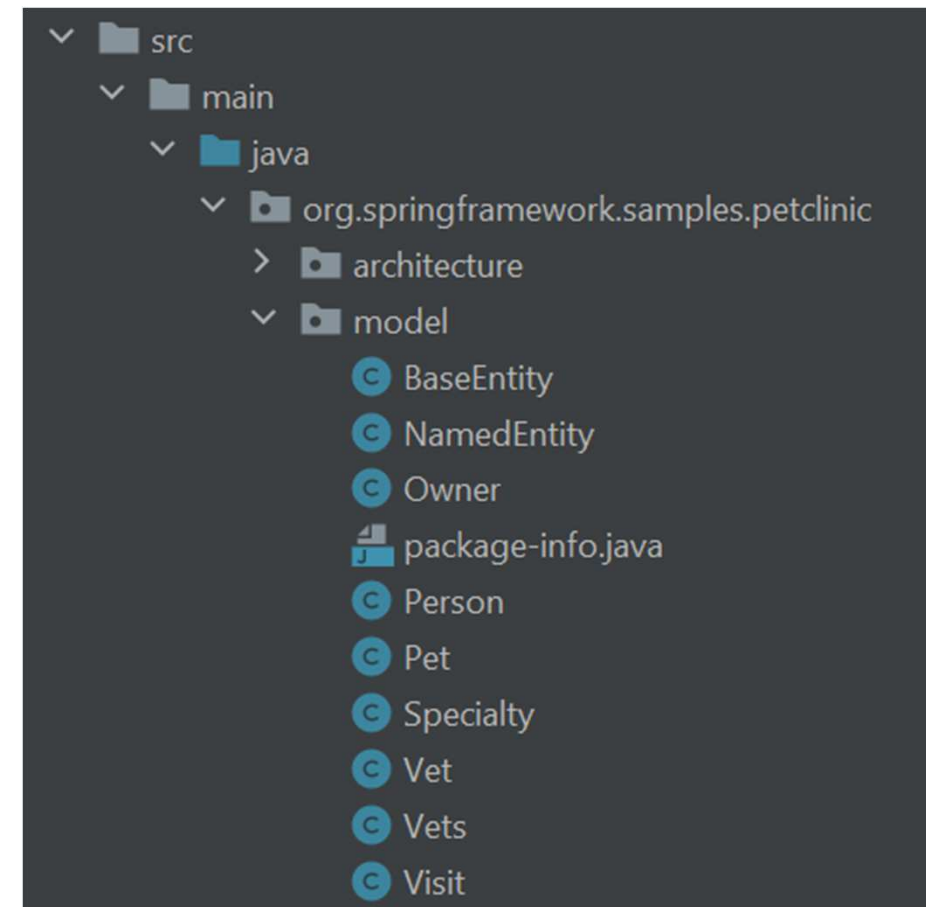
"...ist die Analyse von Softwaredaten
für **Manager** und **Softwareentwickler**
mit dem Ziel,
es allen an der Entwicklung Beteiligten zu ermöglichen,
neue Einsichten aus ihren **Daten** zu erhalten,
um **bessere Entscheidungen** zu treffen."

Spezifische Fragen beantworten



Warum jetzt?

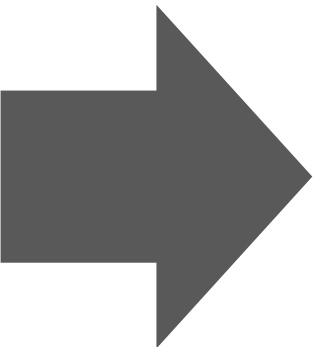
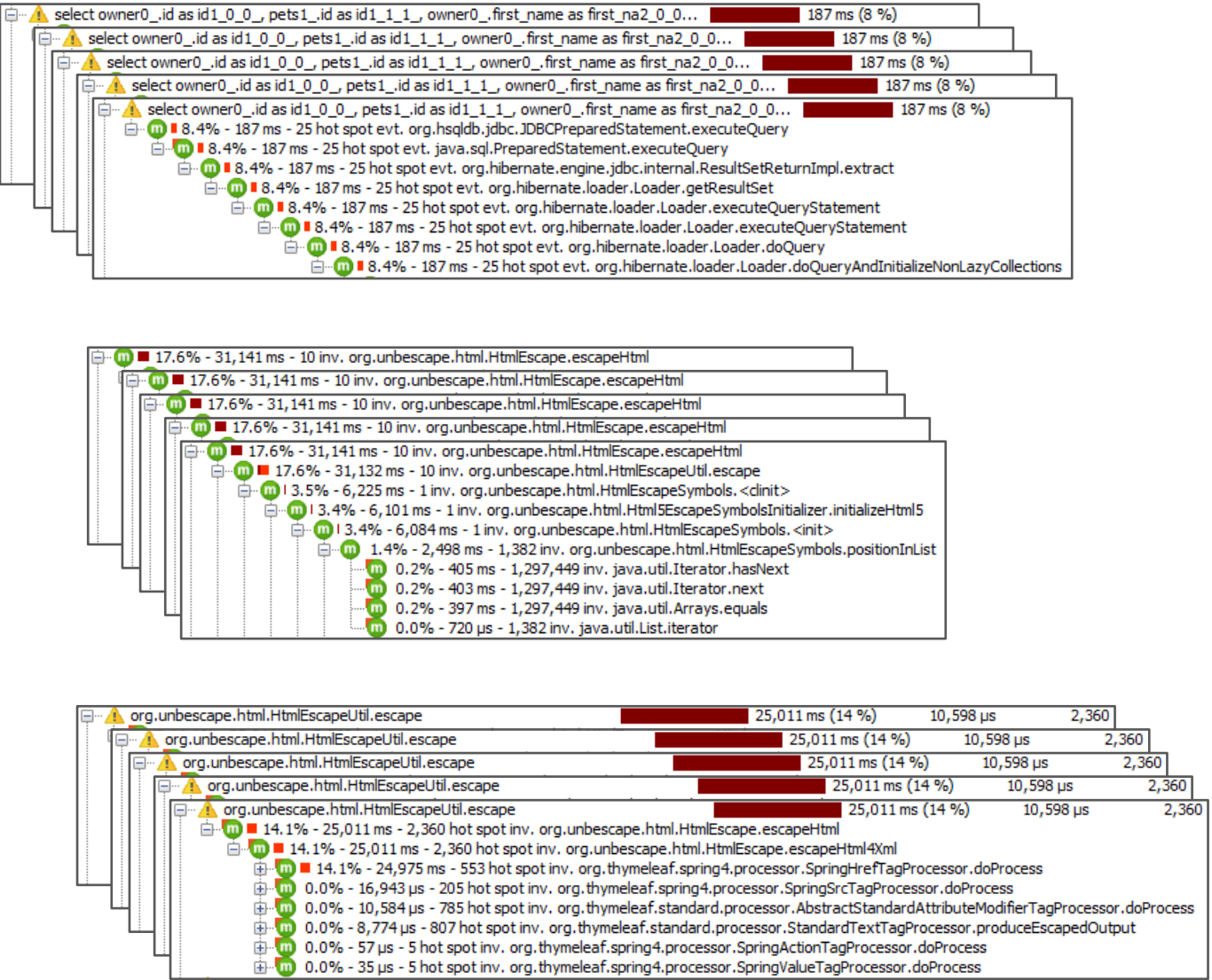
- Bezeichner im Code werden immer fachlicher
- Analysen können sich mit der Fachlichkeit verbinden
- Analysewerkzeuge können nun auch mit stark vernetzten Daten umgehen
- Datenanalysen sind einfacher durchzuführen



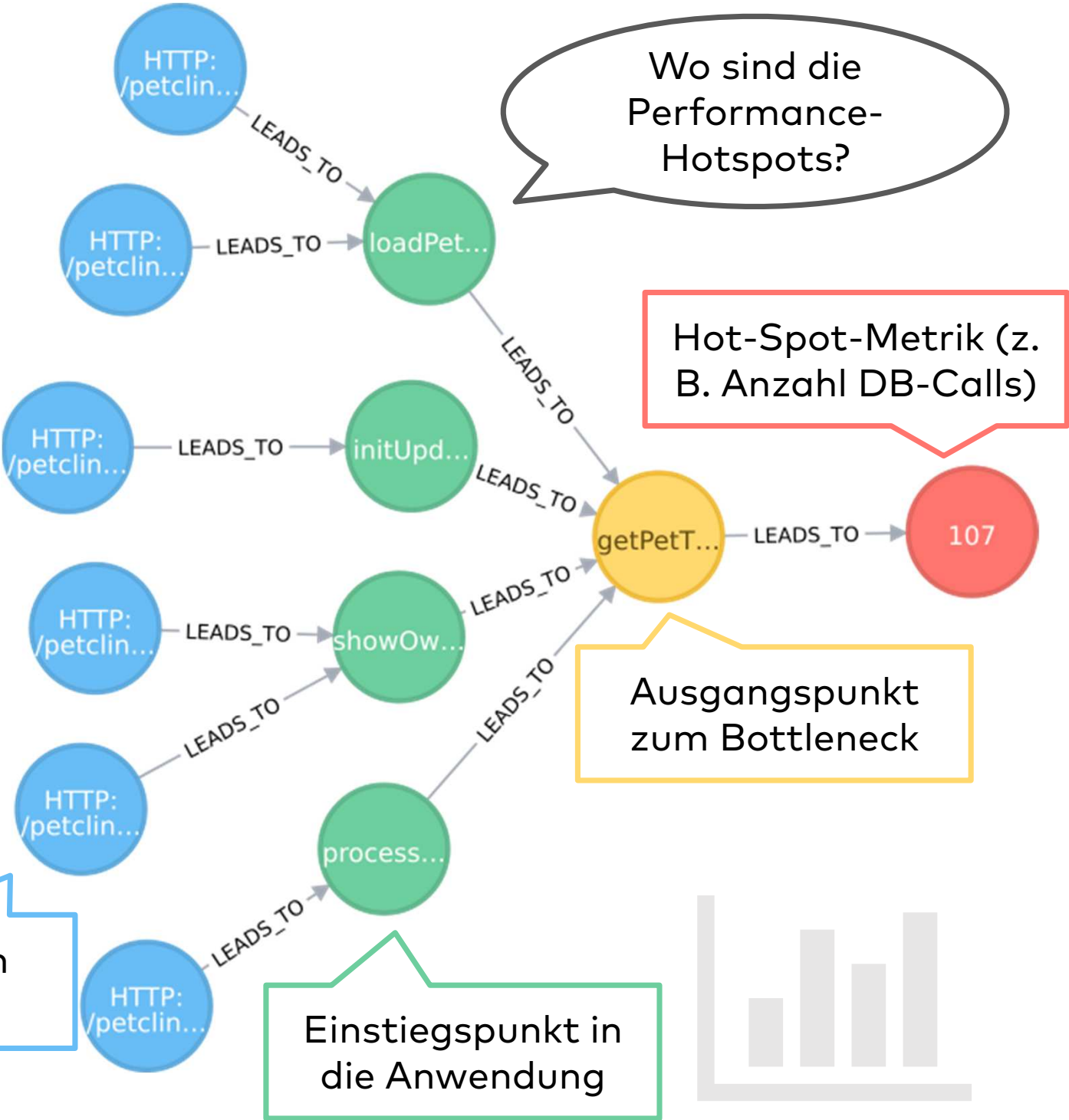
Anwendungsbeispiele

@feststelltaste

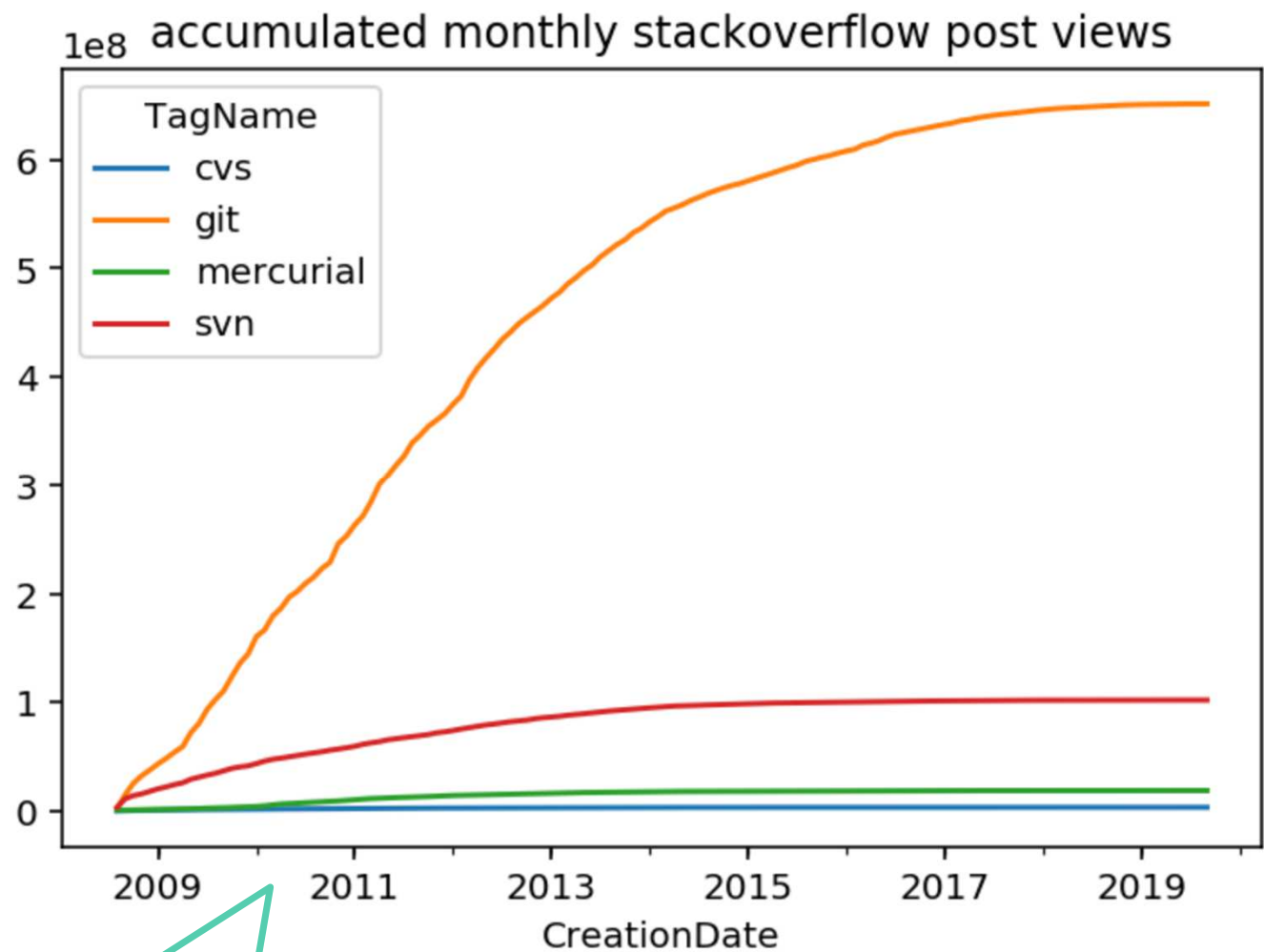
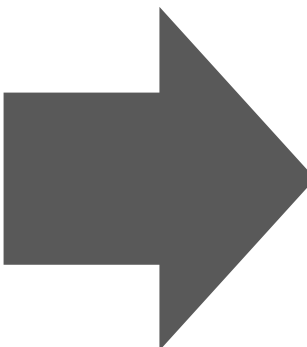
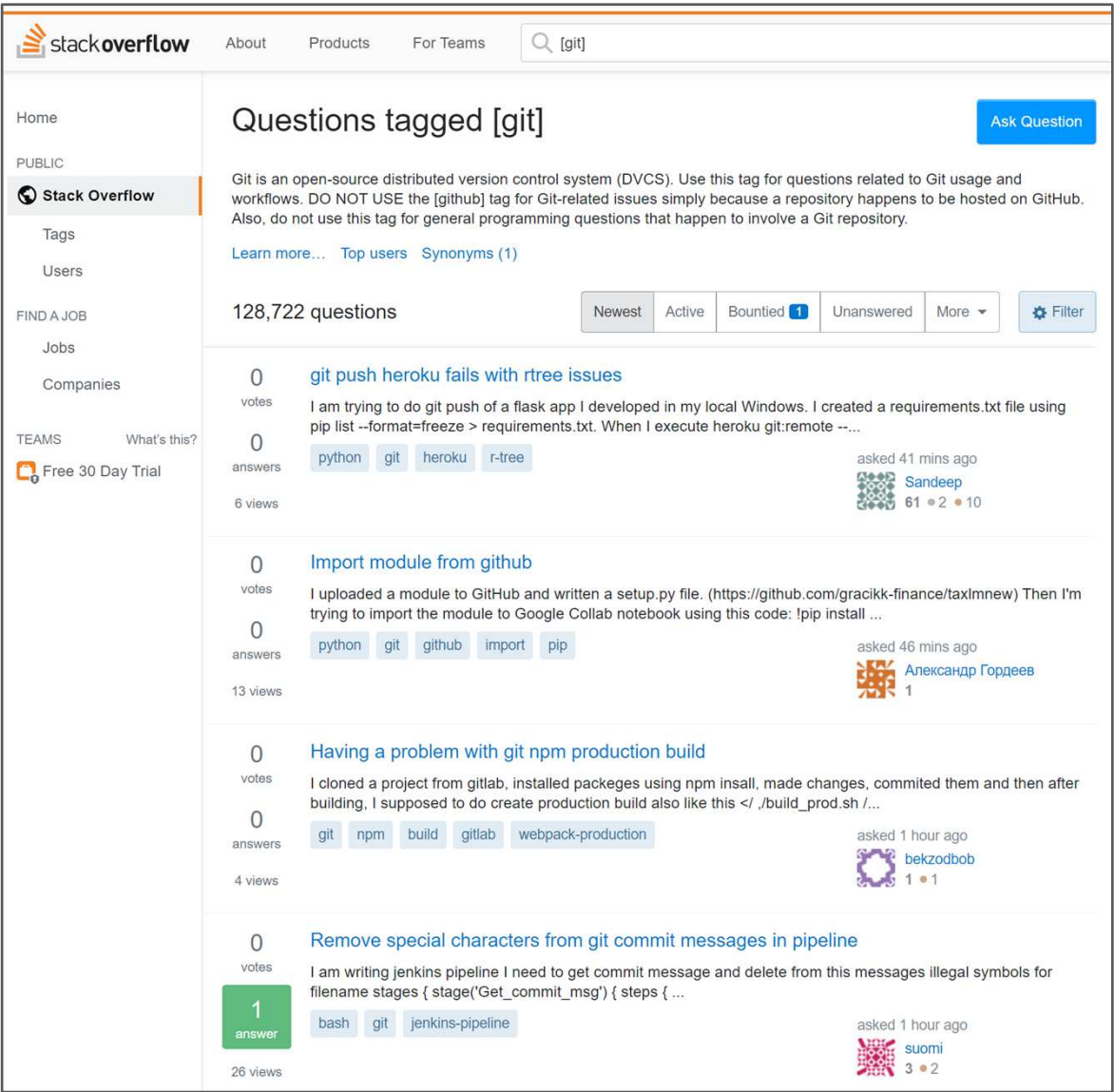
Ermittlung von Performance-Hotspots über Call-Tree-Analyse



API-Calls zum Backend



Analyse der Community-Aktivitäten um Software-Tools



Wie gut wird das Tool durch die Community supported?



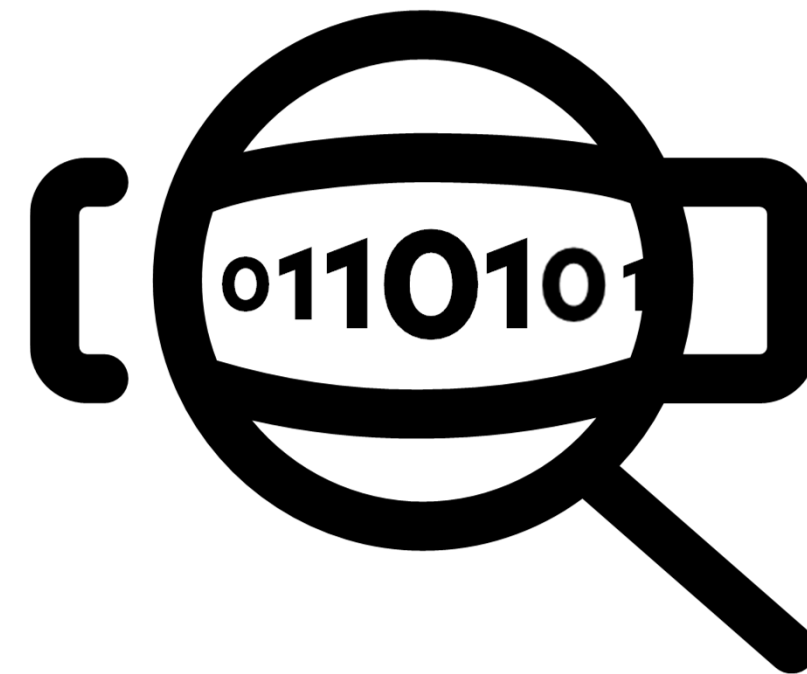
Community

(M)eine Umsetzung von Software Analytics

Analyse von Softwaresystemen*

mit Vorgehen, Methoden
und **Standardwerkzeugen**

aus (**Graph**) **Data Science**



* wird sehr weit aufgefasst

Data Science als Fundament

@feststelltaste

Data Science und Softwareentwicklung?

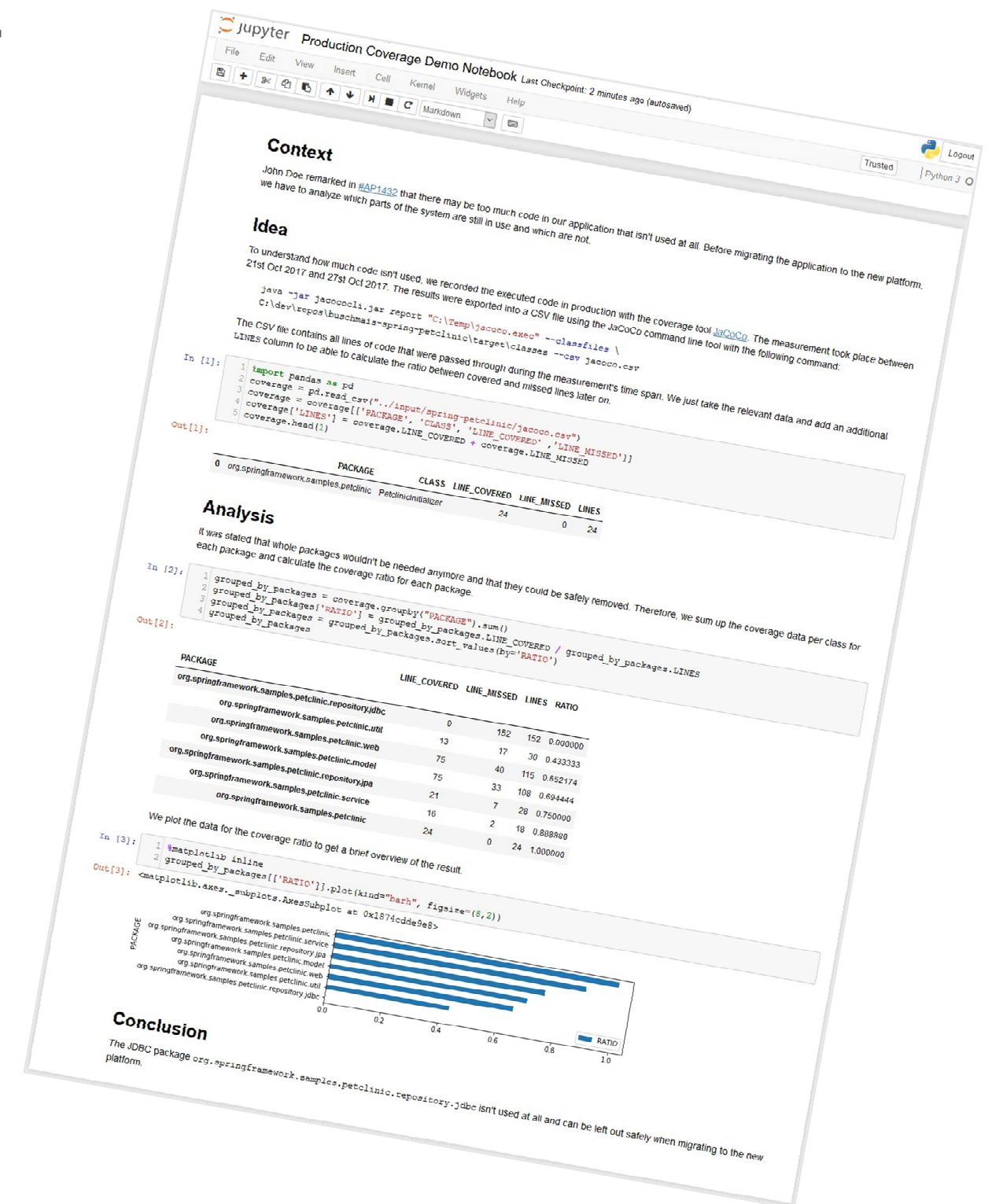
**"Ein Data Scientist ist besser in Statistik
als ein Softwareentwickler
und besser in Softwareentwicklung
als ein Statistiker."**

Data Science Tools



Interaktives Notebooksystem

- Dokumentenzentrierte Analysen
- Ausführbare Codeblöcke
- Direkt sichtbare Visualisierungen



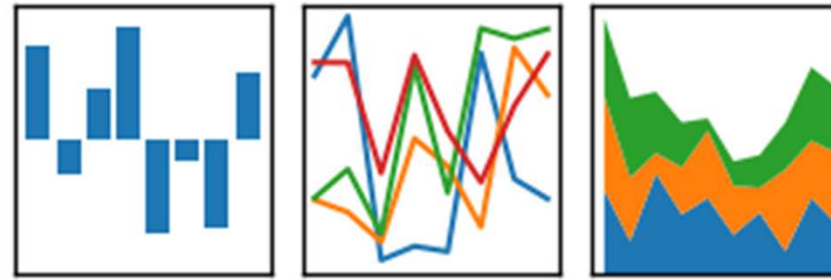


Die Programmiersprache für Data Science

- ✓ Einfach
- ✓ Effektiv
- ✓ Schnell

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

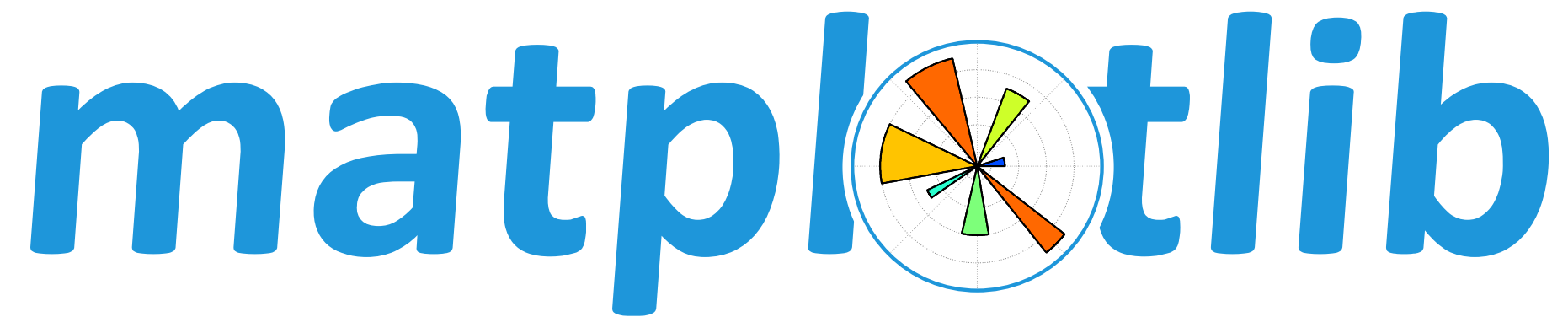


Pragmatisches Datenanalysewerkzeug

✓ Das programmierbare Excel-Arbeitsblatt

- Richtig schnell
- Flexibel
- Ausdrucksstark

➔ Sehr gute Integration mit anderen Bibliotheken



Visualisierungsbibliothek

Ermöglicht die programmatische Erstellung von Grafiken

- Erstellung von Balken-, Linien-Diagrammen und mehr
- Gute Integration mit **pandas** & Co.

➔ Direkte Ausgabe in **Jupyter** Notebooks

Demo „Production Coverage“

Welcher Code in welchen Packages wird nicht verwendet?



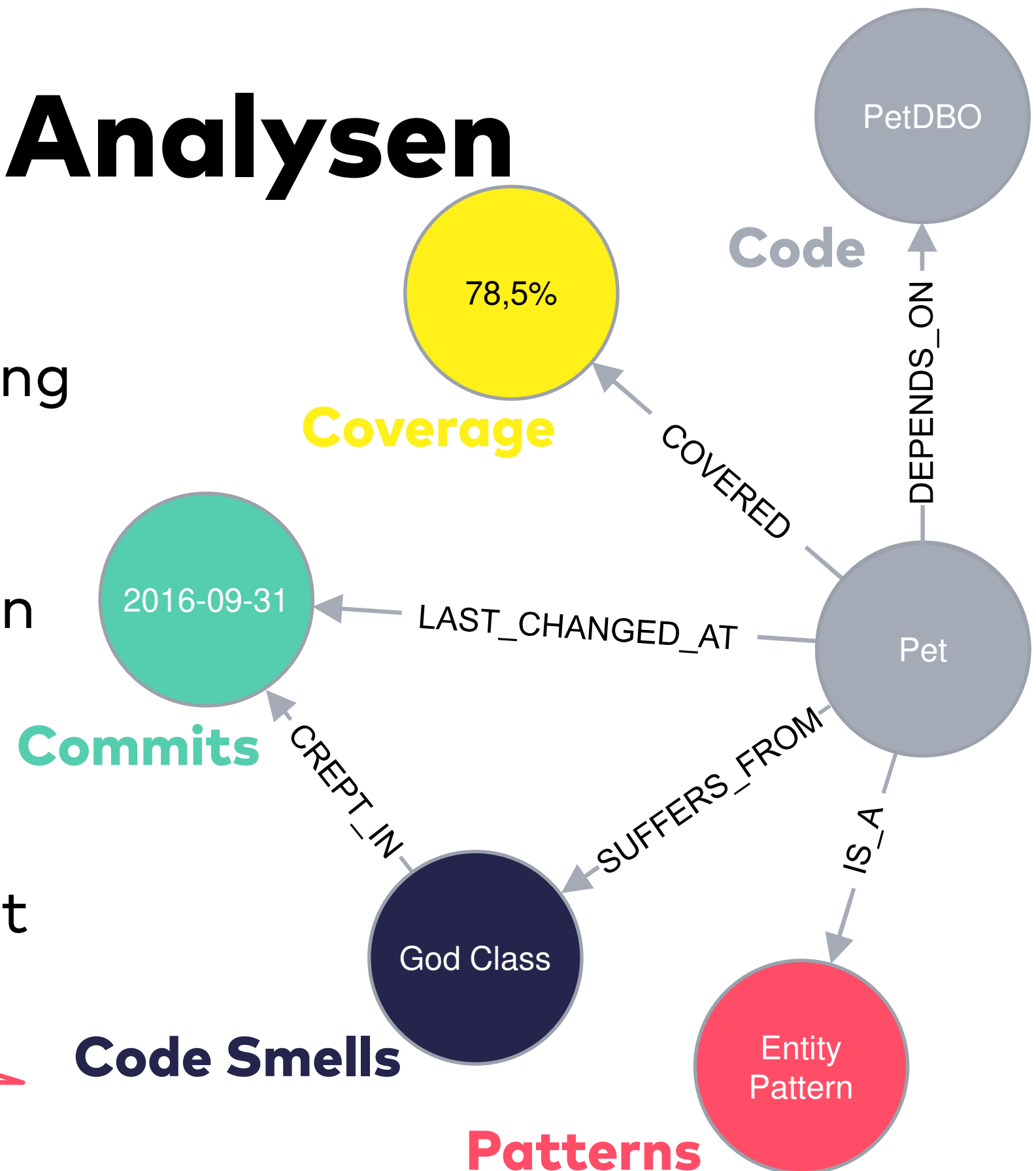
Graph Data Science

@feststelltaste

Graph-orientierte Analysen

- ✓ Verschiedene Datenquellen werden miteinander in Beziehung gesetzt
- ✓ Neue Konzepte / Perspektiven werden auf feingranulare Daten projiziert
- ✓ Vielen Einzelheiten werden zu einer besser interpretierbaren Gesamtsicht zusammengefasst

„Unsere Entitys tendieren dazu, God Classes zu werden“



Graph Data Science Tools

Die Spezialisten für vernetzte Daten



i Framework zur statischen Architektur- und Code-Analyse auf Basis von Softwaredaten

[:SPEICHERT_IN]



i Graph-Datenbank zur Ablage und Analyse stark vernetzter Daten

Neo4j, Cypher & Jupyter Notebook

Cypher-Extension

<https://github.com/versae/ipython-cypher>

Alternative: Cypher-Kernel

https://github.com/HelgeCPH/cypher_kernel

Analysis

With the following Cypher queries, we can spot some kind of race conditions that are declared public and not static fields and are written by some methods.

In [1]: `%load_ext cypher`

In [2]: `%%cypher
MATCH (c:Class)-[:DECLARES]->(f:Field)<-[w:WRITES]-(m:Method)
WHERE
 EXISTS(f.static) AND NOT EXISTS(f.final)
RETURN
 c.name as InClass,
 m.name as theMethod,
 w.lineNumber as writesInLine,
 f.name as toStaticField`

3 rows affected.

Out[2]:

InClass	theMethod	writesInLine	toStaticField
OwnerController	processFindForm	112	ownersIndexes
OwnerController	processFindForm	112	ownersIndexes
BrokenSingleton	getInstance	11	INSTANCE

Jupyter Notebook + py2neo

```
import pandas as pd
from py2neo import Graph
```

```
graph = Graph()
```

```
query="""
MATCH (e:Element)-[:HAS_ATTRIBUTE]->(a:Attribute)
WHERE a.value = "SELECT id, name FROM types ORDER BY name"
WITH e as node
MATCH (node)-[:HAS_ATTRIBUTE]->(all:Attribute)
RETURN all.name, all.value
"""
pd.DataFrame(graph.run(query).data())
```

Out[3]:

	all.name	all.value
0	count	107
1	time	78386
2	value	SELECT id, name FROM types ORDER BY name
3	leaf	false

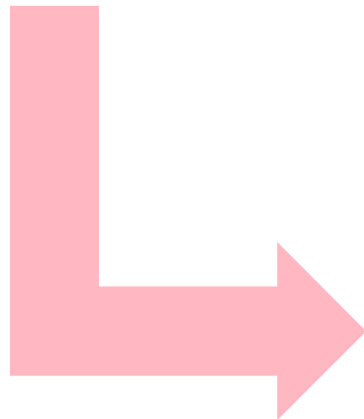
1. Bibliotheken importieren
2. Verbindung zur laufenden Neo4j Instanz herstellen
3. Abfrage in Cypher absetzen
4. Ergebnis in DataFrame umwandeln
5. Ergebnis von der Graph-Datenbank wird dargestellt

Roundtrip pandas ↔ Neo4j

	fqn	ratio
0	org.springframework.samples.petclinic.PetclinicInitializer	1.0
1	org.springframework.samples.petclinic.model.NamedEntity	0.8
2	org.springframework.samples.petclinic.model.Specialty	1.0
3	org.springframework.samples.petclinic.model.PetType	1.0
4	org.springframework.samples.petclinic.model.Vets	0.0

	class	coverage
0	org.springframework.samples.petclinic.PetclinicInitializer	1.0
1	org.springframework.samples.petclinic.model.NamedEntity	0.8
2	org.springframework.samples.petclinic.model.Specialty	1.0
3	org.springframework.samples.petclinic.model.PetType	1.0
4	org.springframework.samples.petclinic.model.Vets	0.0

pandas DataFrame as input



```
import py2neo
graph = py2neo.Graph()

query="""
    UNWIND {coverage_data} as coverage
    MATCH (t:Type {fqn : coverage.fqn})
    MERGE (t)-[:HAS_MEASURE]->(m)
    SET
        m:Measure:Coverage,
        m.ratio = coverage.ratio
    RETURN t.fqn as class, m.ratio as coverage
"""

coverage_dict = coverage.to_dict(orient='records')
result = graph.run(query, coverage_data=coverage_dict).data()
pd.DataFrame(result).head()
```

Import code into Neo4j with py2neo

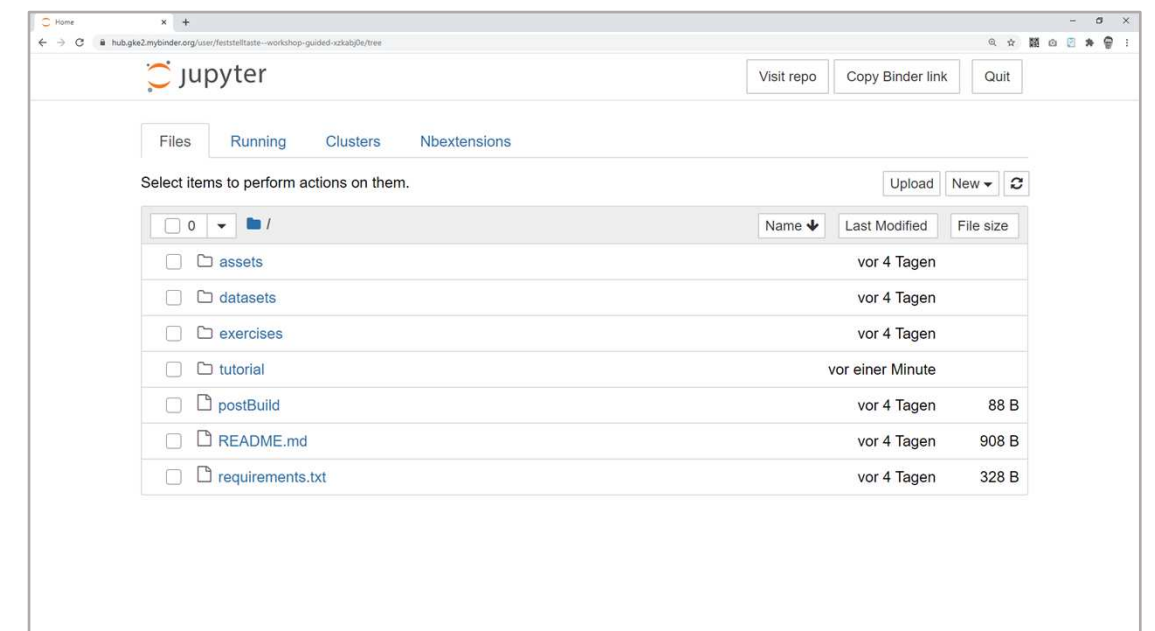
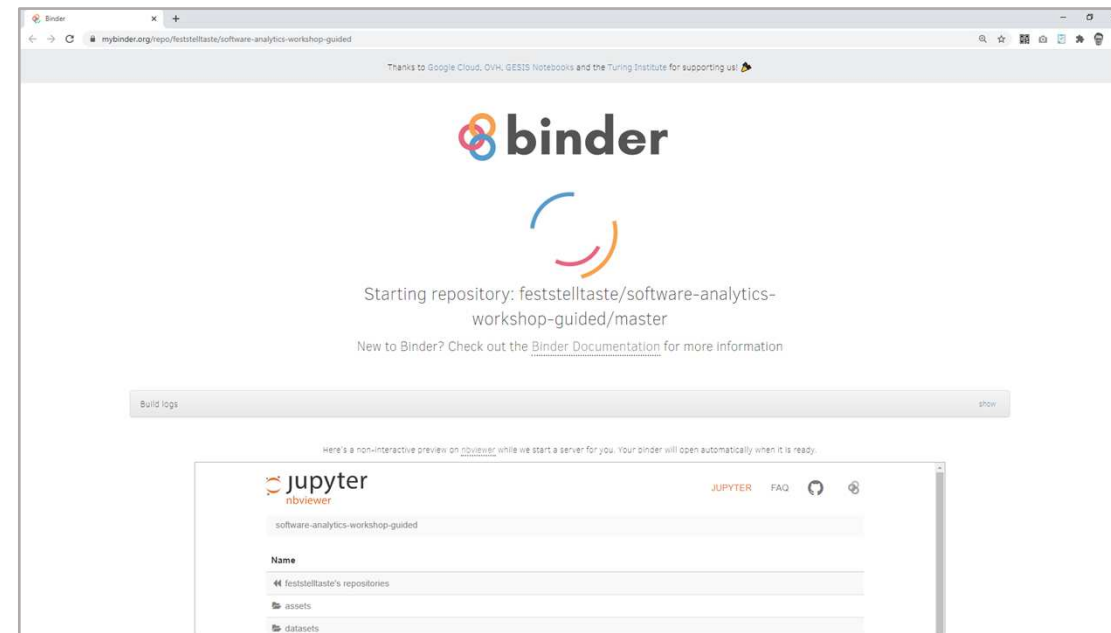
Neo4j output as DataFrame



Das war's schon wieder :-)

@feststelltaste

Ihr Weg zum Mini-Tutorial



Klick auf

<https://mybinder.org/repo/feststelltaste/software-analytics-workshop-guided>

Dann Klick auf Ordner „tutorial“

Weitere Infos

Einstiegshilfen von mir

Sammlung von Ressourcen über Software Analytics

<https://github.com/feststelltaste/awesome-software-analytics>

Awesome Software Analytics

Curated list of awesome resources and links about Software Analytics.

This list is an open community project. Feel free to [contribute your ideas](#) to it.

What is "Software Analytics"?

Software analytics is analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions.

-- Tim Menzies, Thomas Zimmermann

Contents

ordered by "from theory to practice"

- [Influential Papers](#)
- [Systematic Literature Reviews](#)
- [Academic Courses](#)
- [Books](#)
- [Blog Posts](#)

Einstiegshilfen von mir

TOP 5 Software Analytics

<https://www.feststelltaste.de/top-5-software-analytics/>

Mein Software Analytics Repository

<https://github.com/feststelltaste/software-analytics>

Mini-Tutorial und mehr zu Software Analytics

<https://github.com/feststelltaste/software-analytics-workshop>

Weiteres über das Thema von mir

Podcast-Folge „Software Analytics - Mit Data Science Probleme in der eigenen Software finden“

<https://www.innoq.com/de/podcast/083-software-analytics/>

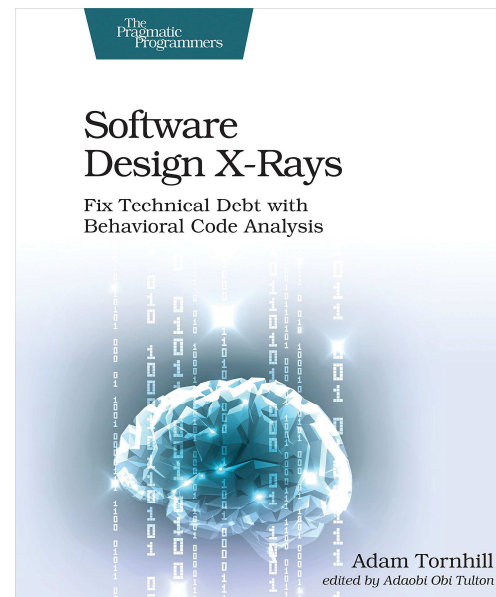
Vortrag „Software-Systeme datengetrieben analysieren“

<https://www.youtube.com/watch?v=v73iPtJEkls>

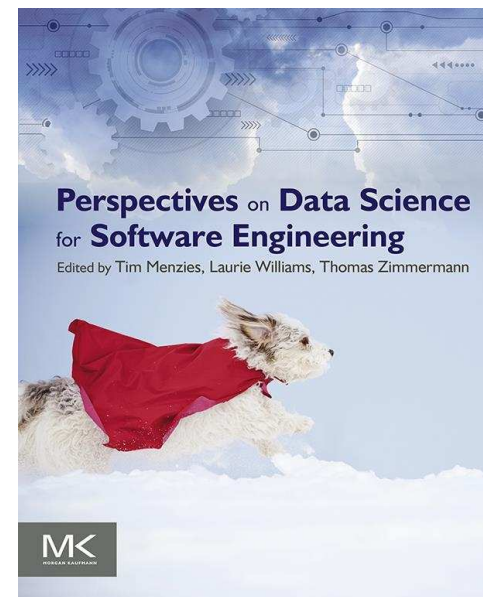
Mein Blog

<https://feststelltaste.de>

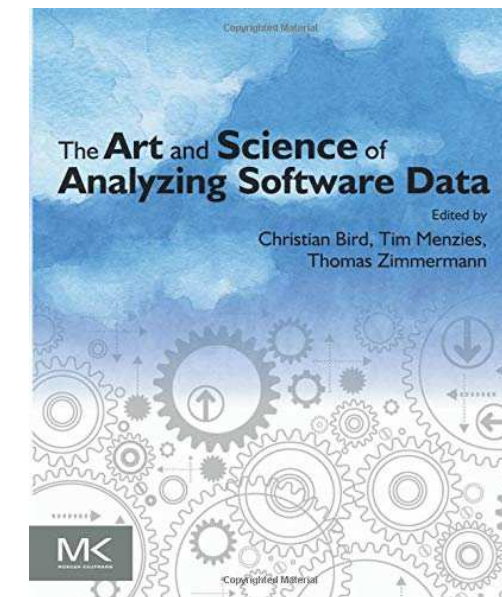
Bücher zu Software Analytics



Adam Tornhill:
Software X-Ray



Tim Menzies, Laurie Williams,
Thomas Zimmermann:
*Perspectives on Data Science for
Software Engineering*



Christian Bird, Tim Menzies,
Thomas Zimmermann:
*The Art and Science of Analyzing
Software Data*

Weitere Details zu jQAssistant/Neo4j

<https://easychair.org/publications/preprint/893N>

Towards an Open Source Stack to Create a Unified Data Source for Software Analysis and Visualization

Richard Müller*, Dirk Mahler†, Michael Hunger‡, Jens Nerche§ and Markus Harrer¶

*Leipzig University, Germany

Email: rmueller@wifa.uni-leipzig.de

†buschmais GbR, Dresden, Germany

Email: dirk.mahler@buschmais.com

‡Developer Relations, Neo4j Inc., Malmö, Sweden

Email: michael.hunger@neo4j.com

§Application Development, Kontext E GmbH, Dresden, Germany

Email: j.nerche@kontext-e.de

¶Software Development Analyst, Freelancer, Roth, Germany

Email: contact@markusharrer.de

Abstract—The beginning of every software analysis and visualization process is data acquisition. However, there are various sources of data about a software system. The methods used

Creating, storing, and querying the data captured by such graphs is very challenging. Diehl et al. summarize the most important questions in this respect [2].

Software Analytics Training

@feststelltaste

Software Analytics Training



Öffentliche Termine ab März 2021 remote, firmenintern nach Abstimmung

Nächste Termine

1. März – 4. März 2021

4 Nachmittage

13:30 bis 16:30 Uhr

Deutsche Version

3. Mai – 6. Mai 2021

4 Abende

18:00 bis 21:00 Uhr

Englische Version

Website: <https://www.innoq.com/de/trainings/software-analytics/>

Rabatt-Code: OOP2021

Remote-Tools zur Durchführung

Zoom

- Videokonferenzlösung
- Nutzung für Theorie und Übungen
- In Übungen meist Arbeit in Kleingruppen (Breakout-Sessions)

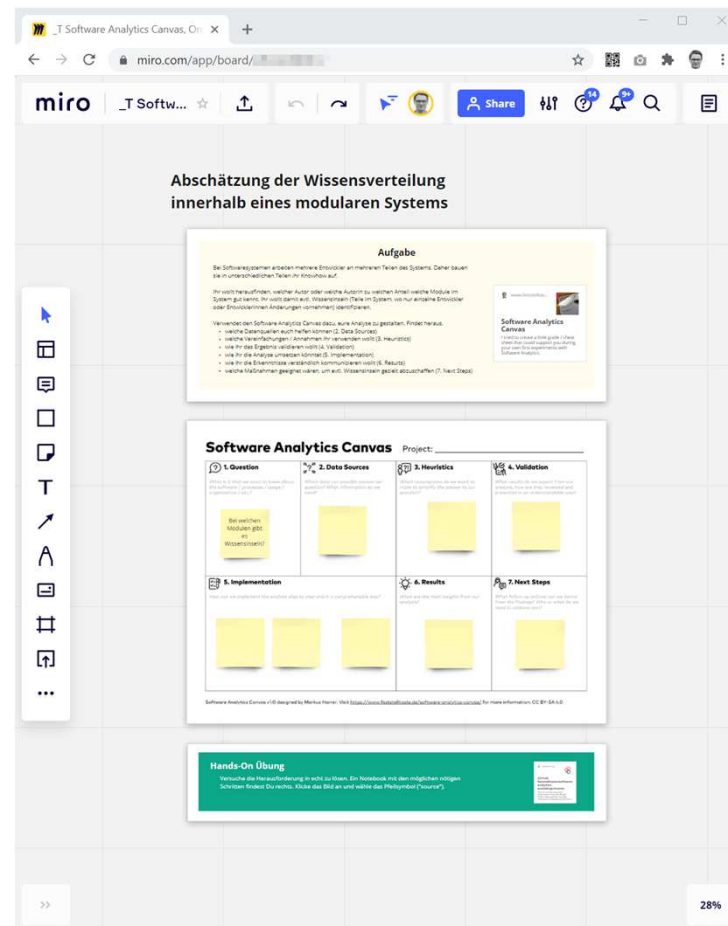
Miro

- Online-Whiteboard
- Nutzung über anonyme Gastzugänge
- Interaktive Übungen zu diversen Themen

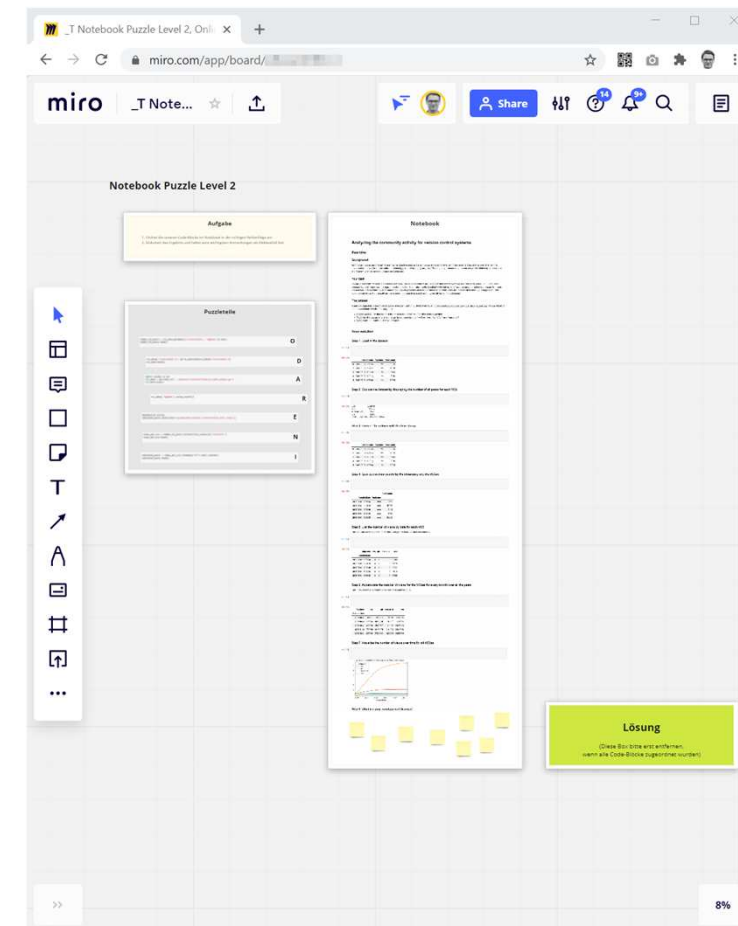
Zusätzlich verwenden wir eine Online-Datenanalyse-Plattform für Pandas, Neo4j & Co.

Übungen mit Online-Whiteboard

Kollaborative Übungen in Kleingruppen mit Miro



Übung zu Software Analytics Canvas



Übung zum Aufbau von Datenanalysen

Kurzbeschreibung

Nutzen Sie datengetriebene Softwareanalysen, um Entscheidungen bei der Weiterentwicklung Ihrer Softwaresysteme auf eine solide Faktenbasis zu stellen!

Dieser Kurs bietet Ihnen dazu einen kompletten Einstieg in das Thema Software Analytics. Lernen Sie die Methodik, Vorgehensweisen und Werkzeuge, um eigenständig nachvollziehbare Datenanalysen in der Softwareentwicklung durchzuführen.

Lernziele

- Standardwerkzeuge aus dem Data-Science- und Graphdatenbanken-Bereich für die Analyse von Softwaredaten einsetzen
- Probleme in der Softwareentwicklung datengetrieben, methodisch und strukturiert herausarbeiten
- Handlungsorientierte Schlüsse aus den Analyseergebnissen ableiten
- Analysen und Erkenntnisse verständlich kommunizieren

Inhalte Tag 1

- Grundprobleme der Softwareentwicklung
- Einführung in Software Analytics
- Datenquellen für Analysen
- Herausforderungen bei Analysen im Softwarebereich
- Einführung in Reproducible Data Science
- Datenanalysen mit Jupyter, Python, pandas & Co.
- Visualisierungen mit matplotlib

Inhalte Tag 2

- Interaktive Visualisierung mit D3 und pypal
- Graph-basierte Softwareanalysen mit Neo4j
- Integration von Software Analytics in die Softwareentwicklung
- Ausblick in das Maschinelle Lernen auf Basis von Softwaredaten

Fragen!?

Diskussionen!?

Anmerkungen!?

Feedback!?

Vielen Dank!



Markus Harrer
markus.harrer@innoq.com

 @feststelltaste

<https://feststelltaste.de>

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin
Germany

Ludwigstr. 180E
63067 Offenbach
Germany

Kreuzstr. 16
80331 München
Germany

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
+41 41 743 01 11

Albulastr. 55
8048 Zürich
Switzerland