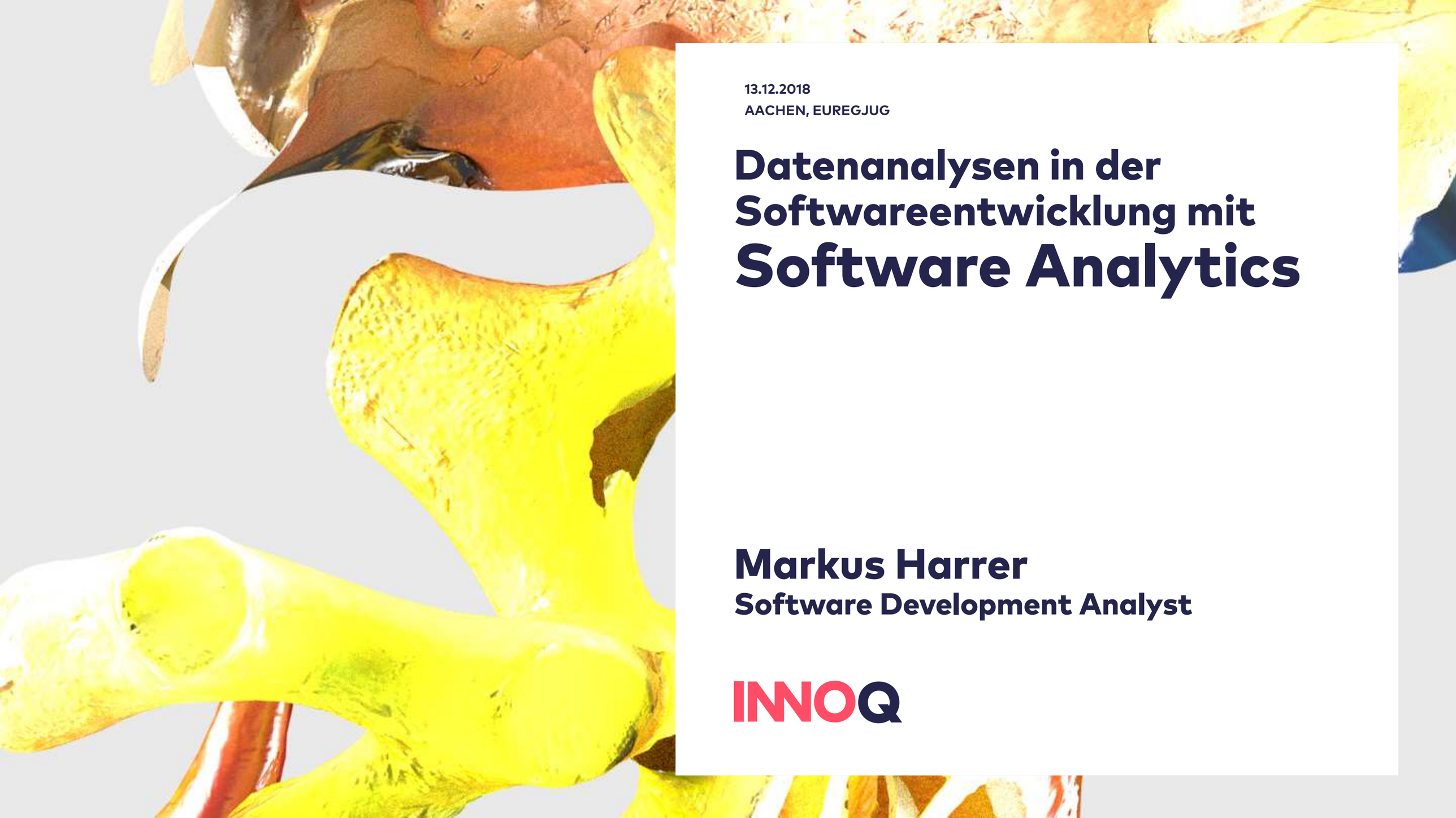


13.12.2018
AACHEN, EUREGJUG

Datenanalysen in der Softwareentwicklung mit Software Analytics

Markus Harrer
Software Development Analyst

INNOQ

Markus Harrer
@feststelltaste

Antwort an @rotnroll666

Thanks for the kind words! But I'm just hacking my way through all the data with pure stubbornness 😇

Tweet übersetzen

10:25 - 20. Apr. 2017

1 „Gefällt mir"-Angabe

I ♥ legacy code!

Markus Harrer

Software Development Analyst

@feststelltaste

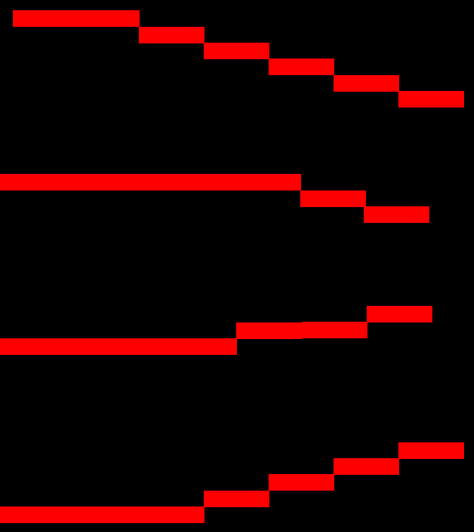
Blog: feststelltaste.de

Web: markusharrer.de

Work: INNOQ

Agenda

- Warum?
- WTF?
- Wie?
- Was?



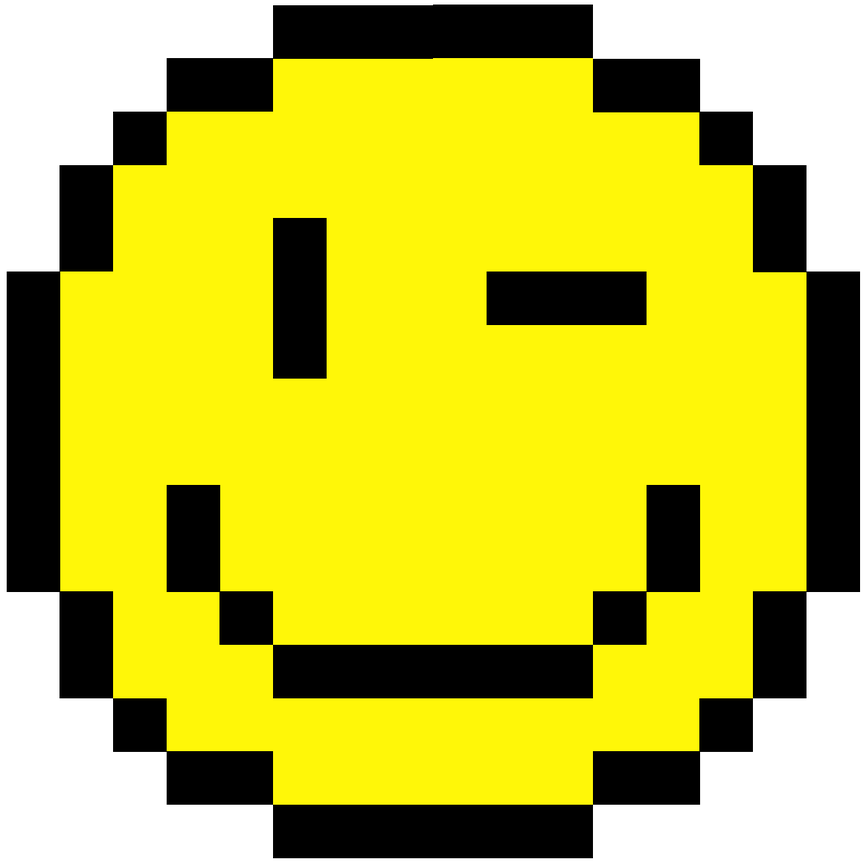
Datenanalysen in der Softwareentwicklung

Warum?

@feststelltaste

Management

Entwickler



Abends in der Kneipe...

Wir haben jetzt eine 7-Schichten-architektur!

Wie kommt es denn zu sowas?

Jedes Jahr gibt es ne' neue Schicht, um den Mist vom letzten Jahr zu überdecken!

Symptombehebungen

Gestern ist unser Flaggschiffprodukt in Produktion abgeraucht!

Schrecklich! Was ist denn passiert?

Nichts! Niemand verwendet es...

Wahrnehmungsdiskrepanz


Management

Software History repeating

```
10 REWRITE
20 SLEEP 3
30 GOTO 10
```

...the only <u>iterative</u> software development process that works every single time!

WTF?

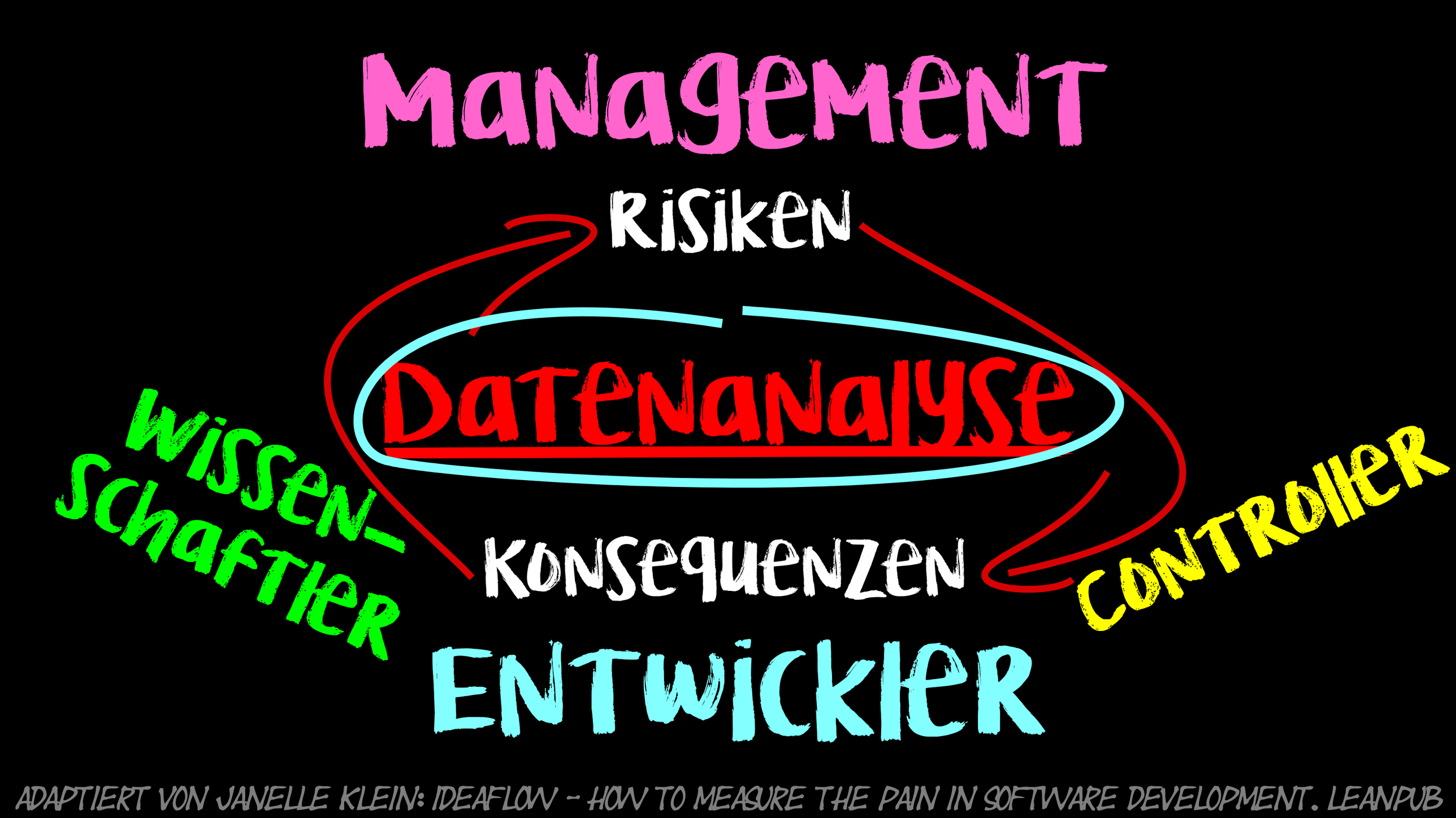
Management

Risiken

Kommunikationsbarriere

Konsequenzen

Entwickler



Adaptiert von Janelle Klein: Ideaflow - How to Measure the Pain in Software Development. Leanpub

Wie?

THE EMPIRIC STRIKES BACK

Not a long time ago, from brains not far, far away....

SOFTWARE ANALYTICS

A definition of

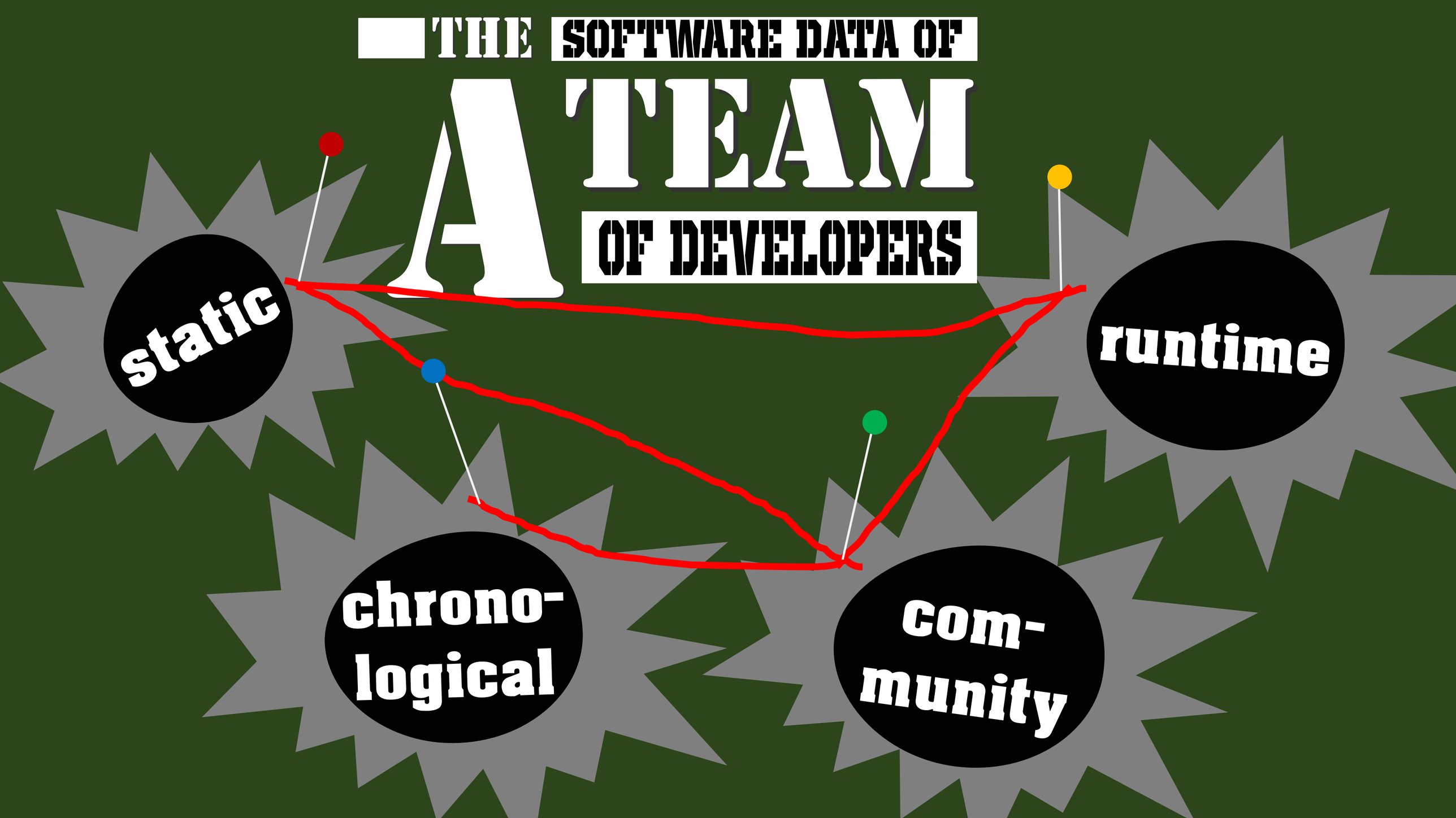
MENZIES & ZIMMERMANN

Software Analytics

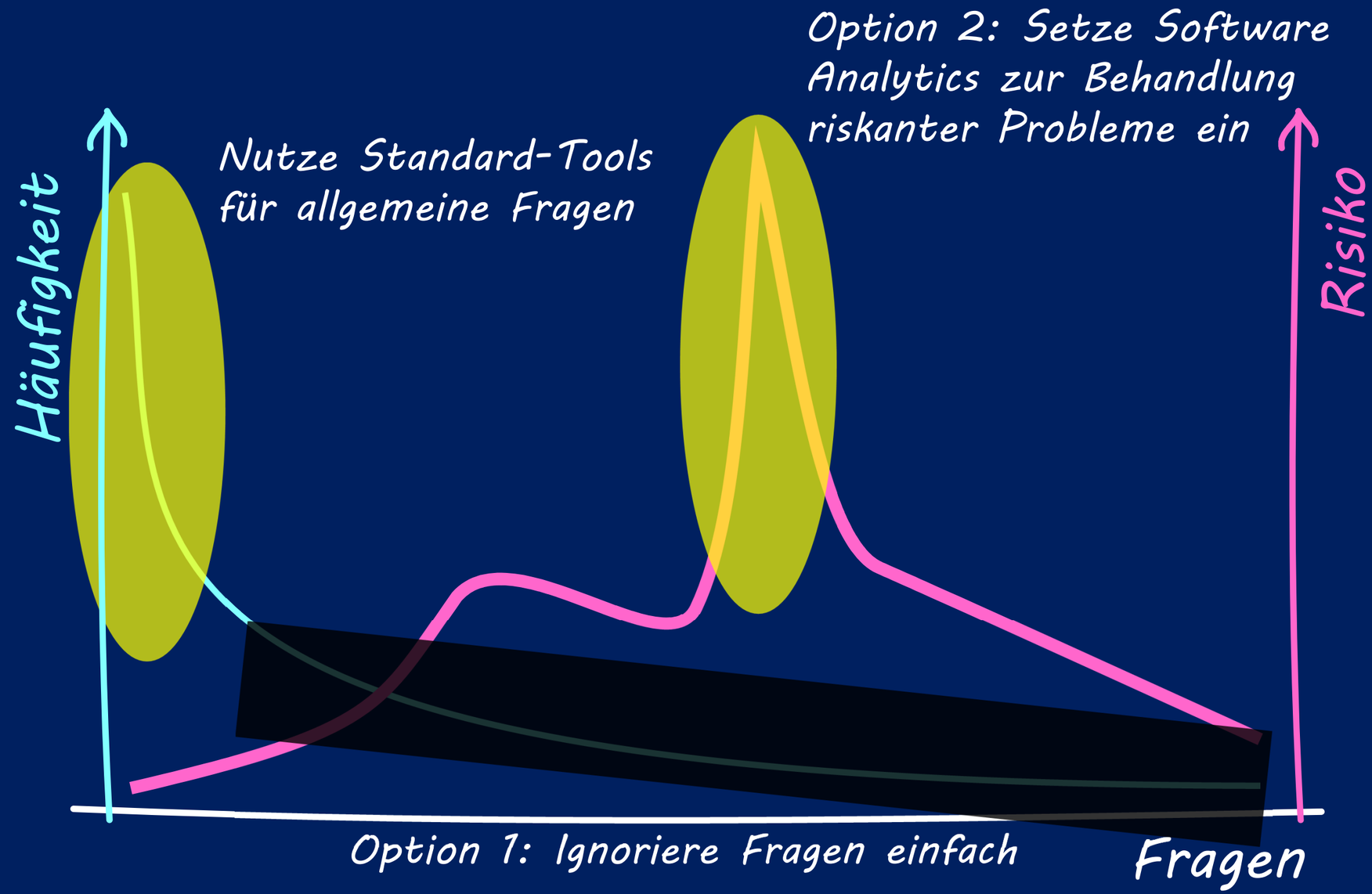
is analytics on software data for
managers and software engineers

with the aim of empowering
software development individuals
and teams

to gain and share insight from
their data to make better
decisions.

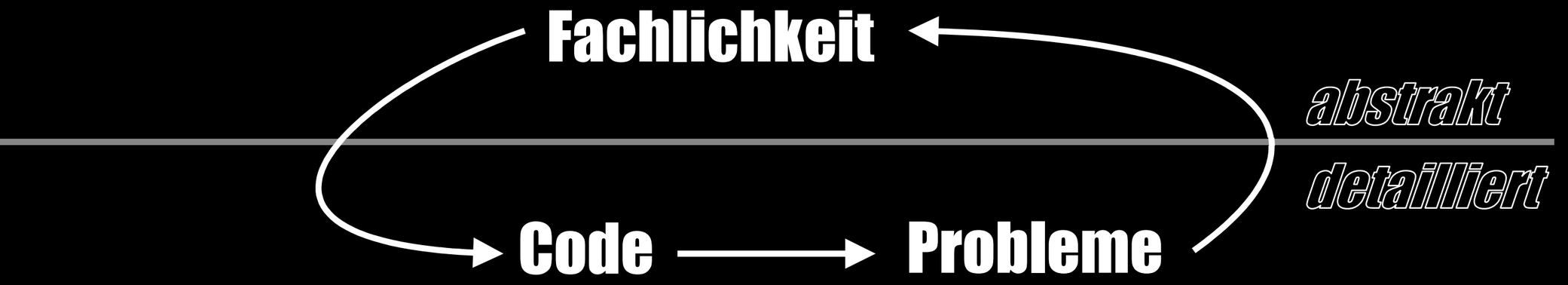

THE SOFTWARE DATA OF
A TEAM
OF DEVELOPERS
static
runtime
chrono-
logical
com-
munity

Software Analytics fokussiert sich auf wichtige Fragen



MINE THE GAP!

- Code verschmilzt mit Fachlichkeit
- Data Science bringt datenorientierte Problemanalyse zu Entwicklern
- Werkzeuge bilden und vernetzen verschiedene Perspektiven

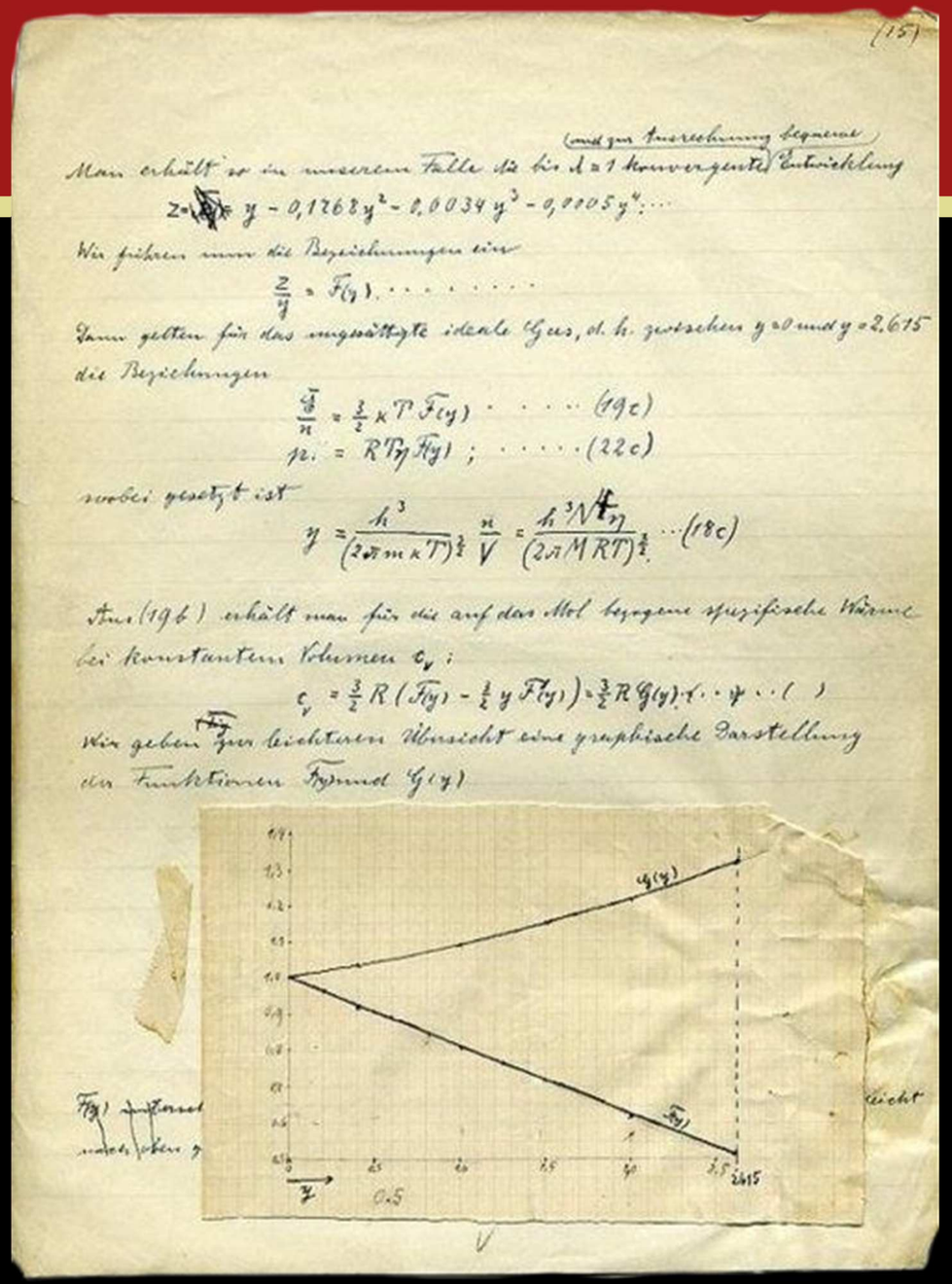


@feststelltaste

Notebook

- Kontext dokumentiert
- Ideen, Daten, Annahmen und Vereinfachungen aufgeführt
- Berechnungen verständlich dargelegt
- Zusammenfassungen erklärt

- Komplett automatisiert


(15)

(und zur Ausrechnung bequemer)

Man erhält so in unserem Falle die bis $\delta = 1$ konvergente Entwicklung

$$z = y - 0{,}1768\,y^2 - 0{,}0034\,y^3 - 0{,}0005\,y^4 \cdots$$

Wir führen nun die Bezeichnungen ein

$$\frac{z}{y} = F(y) \quad \cdots\cdots$$

Dann gelten für das ungesättigte ideale Gas, d. h. zwischen $y = 0$ und $y = 2{,}615$ die Beziehungen

$$\frac{c_v}{n} = \frac{3}{2} \kappa T F(y) \quad \cdots\cdots (19c)$$

$$p = RT\eta F(y) \quad ; \quad \cdots\cdots (22c)$$

wobei gesetzt ist

$$y = \frac{h^3}{(2\pi m \kappa T)^{\frac{3}{2}}} \frac{n}{V} = \frac{h^3 N \eta}{(2\pi M RT)^{\frac{3}{2}}} \cdots (18c)$$

Aus (19b) erhält man für die auf das Mol bezogene spezifische Wärme bei konstantem Volumen c_v:

$$c_v = \frac{3}{2} R \left(F(y) - \frac{1}{2} y F'(y) \right) = \frac{3}{2} R \varphi(y) \quad \cdots\cdots ( )$$

Wir geben zur leichteren Übersicht eine graphische Darstellung der Funktionen $F(y)$ und $\varphi(y)$

V

jupyter Production Coverage Demo Notebook Last Checkpoint: 2 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Markdown

Context

John Doe remarked in #AP1432 that there may be too much code in our application that isn't used at all. Before migrating the application to the new platform, we have to analyze which parts of the system are still in use and which are not.

Idea

To understand how much code isn't used, we recorded the executed code in production with the coverage tool JaCoCo. The measurement took place between 21st Oct 2017 and 27st Oct 2017. The results were exported into a CSV file using the JaCoCo command line tool with the following command:

```
java -jar jacococli.jar report "C:\Temp\jacoco.exec" --classfiles \
C:\dev\repos\buschmais-spring-petclinic\target\classes --csv jacoco.csv
```

The CSV file contains all lines of code that were passed through during the measurement's time span. We just take the relevant data and add an additional `LINES` column to be able to calculate the ratio between covered and missed lines later on.

In [1]:

```
1 import pandas as pd
2 coverage = pd.read_csv("../input/spring-petclinic/jacoco.csv")
3 coverage = coverage[['PACKAGE', 'CLASS', 'LINE_COVERED' ,'LINE_MISSED']]
4 coverage['LINES'] = coverage.LINE_COVERED + coverage.LINE_MISSED
5 coverage.head(1)
```

Out[1]:

	PACKAGE	CLASS	LINE_COVERED	LINE_MISSED	LINES
0	org.springframework.samples.petclinic	PetclinicInitializer	24	0	24

0	org.springframework.samples.petclinic	PetclinicInitializer	24	0	24

Analysis

It was stated that whole packages wouldn't be needed anymore and that they could be safely removed. Therefore, we sum up the coverage data per class for each package and calculate the coverage ratio for each package.

In [2]:
```
grouped_by_packages = coverage.groupby("PACKAGE").sum()
grouped_by_packages['RATIO'] = grouped_by_packages.LINE_COVERED / grouped_by_packages.LINES
grouped_by_packages = grouped_by_packages.sort_values(by='RATIO')
grouped_by_packages
```

Out[2]:

PACKAGE	LINE_COVERED	LINE_MISSED	LINES	RATIO
org.springframework.samples.petclinic.repository.jdbc	0	152	152	0.000000
org.springframework.samples.petclinic.util	13	17	30	0.433333
org.springframework.samples.petclinic.web	75	40	115	0.652174
org.springframework.samples.petclinic.model	75	33	108	0.694444
org.springframework.samples.petclinic.repository.jpa	21	7	28	0.750000
org.springframework.samples.petclinic.service	16	2	18	0.888889
org.springframework.samples.petclinic	24	0	24	1.000000

We plot the data for the coverage ratio to get a brief overview of the result.

In [3]:
```
%matplotlib inline
grouped_by_packages[['RATIO']].plot(kind="barh", figsize=(8,2))
```

Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x1874cdde9e8>

org.springframework.samples.petclinic.util	13	17	30	0.433333
org.springframework.samples.petclinic.web	75	40	115	0.652174
org.springframework.samples.petclinic.model	75	33	108	0.694444
org.springframework.samples.petclinic.repository.jpa	21	7	28	0.750000
org.springframework.samples.petclinic.service	16	2	18	0.888889
org.springframework.samples.petclinic	24	0	24	1.000000

We plot the data for the coverage ratio to get a brief overview of the result.

In [3]:
```
%matplotlib inline
grouped_by_packages[['RATIO']].plot(kind="barh", figsize=(8,2))
```

Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x1874cdde9e8>

Conclusion

The JDBC package `org.springframework.samples.petclinic.repository.jdbc` isn't used at all and can be left out safely when migrating to the new platform.

https://www.feststelltaste.de/production-coverage-demo-notebook/

STANDARDWERKZEUGE

Jupyter
Interaktives Notizbuch: Zentrale Stelle für Datenanalysen und Dokumentation

Python
Data Scientist's best friend: Einfache, effektive, schnelle Programmiersprache

Pandas
Pragmatisches Datenanalyse-Framework: Großartige Datenstrukturen und gute Integration mit Machine Learning Tools

matplotlib
Plotting-Bibliothek für einfache Visualisierungen (Linien-, Balken-, Tortendiagramme etc.)

TOOLS advanced level

Structural Code Analysis Framework

1. Scanne Softwarestrukturen
2. Speichere in Graphdatenbank
3. Analysiere und erstelle Verbindungen
4. Füge eigene Konzepte hinzu
5. Finde Antworten

jQAssistant

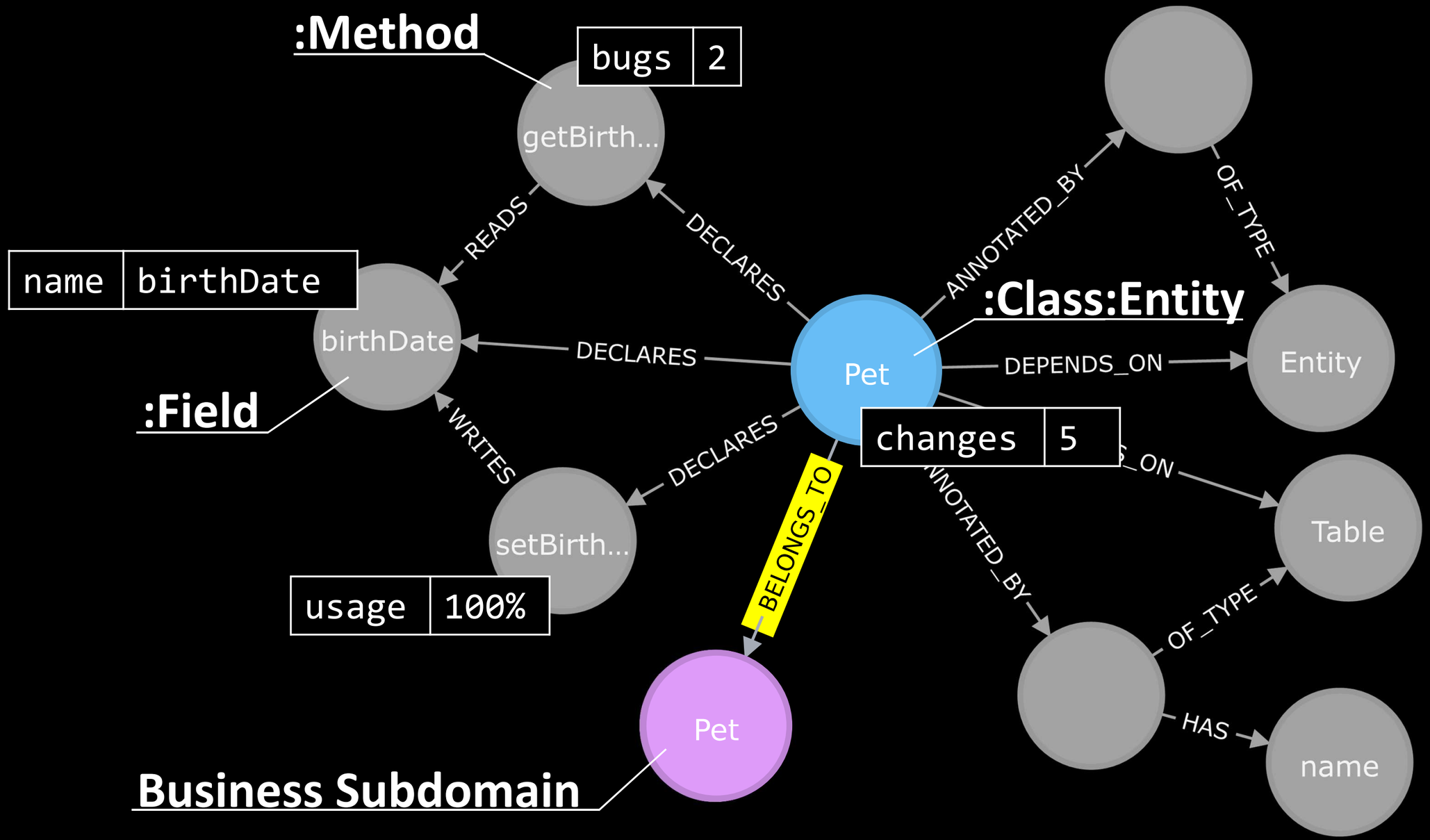
neo4j

GPL Free Software

Free as in Freedom

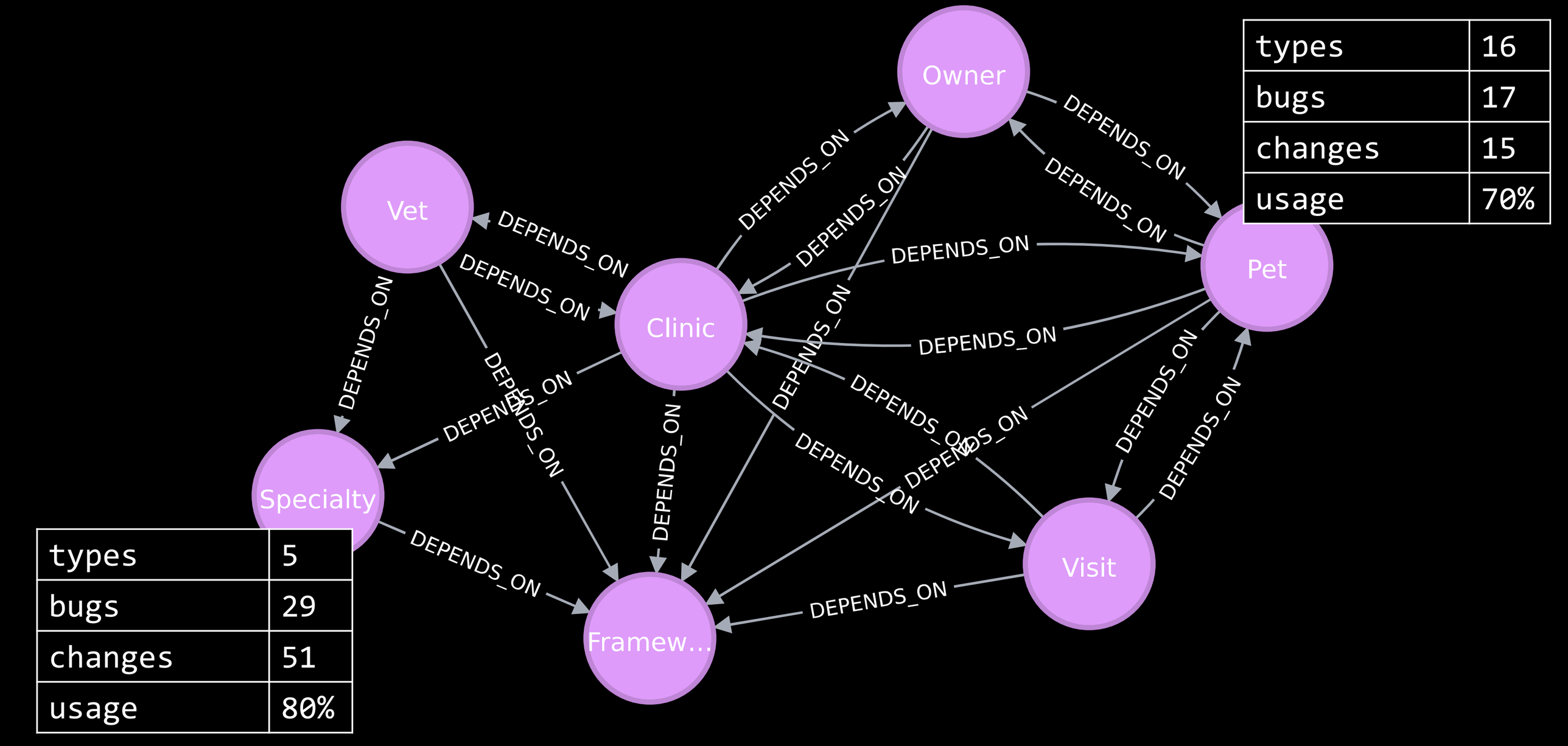
jQAssistant – Die komplexe Softwarelandschaft als Graph

https://github.com/buschmais/spring-petclinic



jQAssistant – Die komplexe Softwarelandschaft als Graph

https://github.com/buschmais/spring-petclinic



types	16
bugs	17
changes	15
usage	70%

types	5
bugs	29
changes	51
usage	80%

Analysen zum Terminieren von Problemen

- Quantifizierung des Wissensverlusts bei Entwicklerfluktuation
- Verprobung von Modularisierungsvarianten (“virtuelles Refactoring”)
- Erstellung von Codeinventar / Musterkatalog / Behebungslisten
- Ermittlung von Performance-Hotspots über Call-Tree-Analyse
- Identifikation von besonders fehlerbehafteten Code-Bereichen
- ...

Individuelle Probleme im eigenen Softwaresystem **sichtbar** machen!

e. g. race conditions, architecture smells, build breaker, programming errors, dead code, ...

Was?

DEMOS

No-Go-Areas

Code Smells

Strategic Redesign

DEMOS

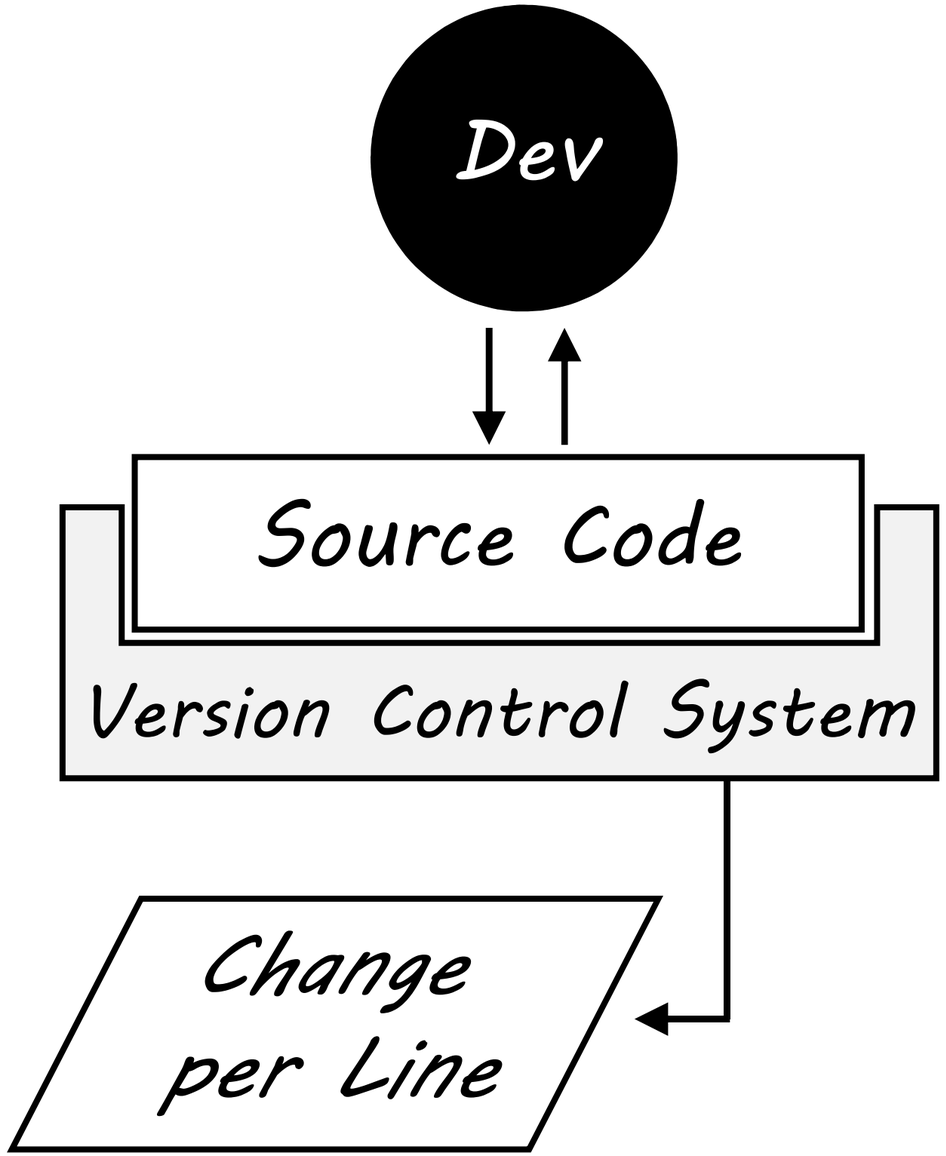
No-Go-Areas

Code Smells

Strategic Redesign

Identification of No-Go areas using version control data

Demo
No-Go Areas in the Linux kernel



https://www.feststelltaste.de/identifying-lost-knowledge-in-the-linux-kernel-source-code/

Identification of No-Go Areas: Starting Point

```
 void rb532_mask_and_ack_irq(struct irq_data *d)

2_disable_irq(d);
local_irq(group_to_ip(irq_to_group(d->irq)));



 int rb532_set_type(struct irq_data *d, unsigned type)

gpio = d->irq - GPIO_MAPPED_IRQ_BASE;
group = irq_to_group(d->irq);

group != GPIO_MAPPED_IRQ_GROUP)
```

Source Code

Identification of No-Go Areas: Idea

```
static void rb532_mask_and_ack_irq(struct irq_data *d)
{
  rb532_disable_irq(d);
  ack_local_irq(group_to_ip(irq_to_group(d->irq)));
}

static int rb532_set_type(struct irq_data *d, unsigned type)
{
  int gpio = d->irq - GPIO_MAPPED_IRQ_BASE;
  int group = irq_to_group(d->irq);

  if (group != GPIO_MAPPED_IRQ_GROUP)
```

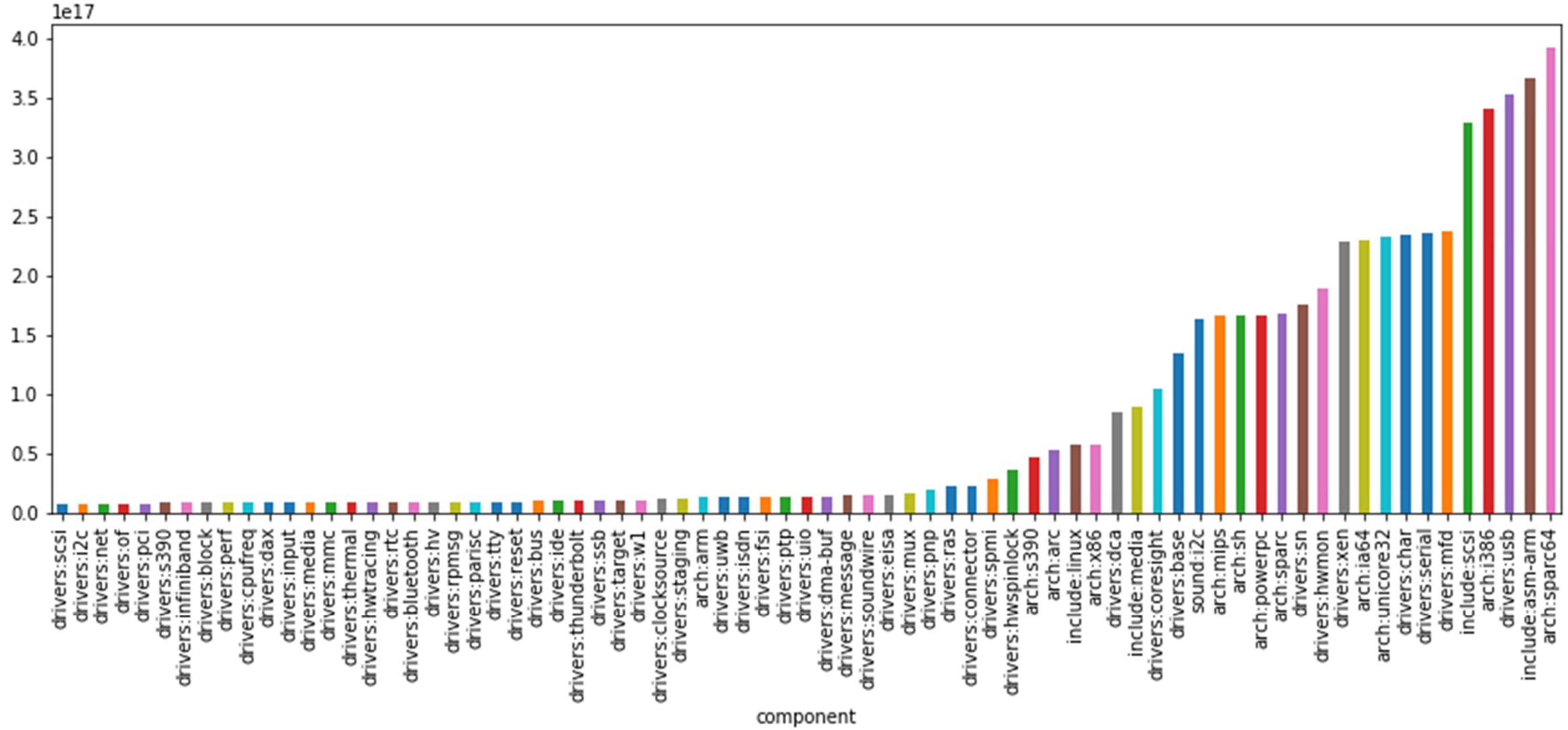
Change per Line

Identification of No-Go Areas: Idea

```
efb02d arch/irq.c (Thomas Gleixner  2011-03-23 21:09:10 +0000 164) stati
73b439 arch/irq.c (Ralf Baechle     2008-07-16 16:12:25 +0100 165) {
efb02d arch/irq.c (Thomas Gleixner  2011-03-23 21:09:10 +0000 166)   rb5
efb02d arch/irq.c (Thomas Gleixner  2011-03-23 21:09:10 +0000 167)   ack
73b439 arch/irq.c (Ralf Baechle     2008-07-16 16:12:25 +0100 168) }
73b439 arch/irq.c (Ralf Baechle     2008-07-16 16:12:25 +0100 169)
efb02d arch/irq.c (Thomas Gleixner  2011-03-23 21:09:10 +0000 170) stati
4aa0f4 arch/irq.c (Phil Sutter      2008-11-28 20:45:10 +0100 171) {
efb02d arch/irq.c (Thomas Gleixner  2011-03-23 21:09:10 +0000 172)   int
efb02d arch/irq.c (Thomas Gleixner  2011-03-23 21:09:10 +0000 173)   int
4aa0f4 arch/irq.c (Phil Sutter      2008-11-28 20:45:10 +0100 174)
efb02d arch/irq.c (Thomas Gleixner  2011-03-23 21:09:10 +0000 175)   if
```

Change per Line

Identification of No-Go Areas: Result



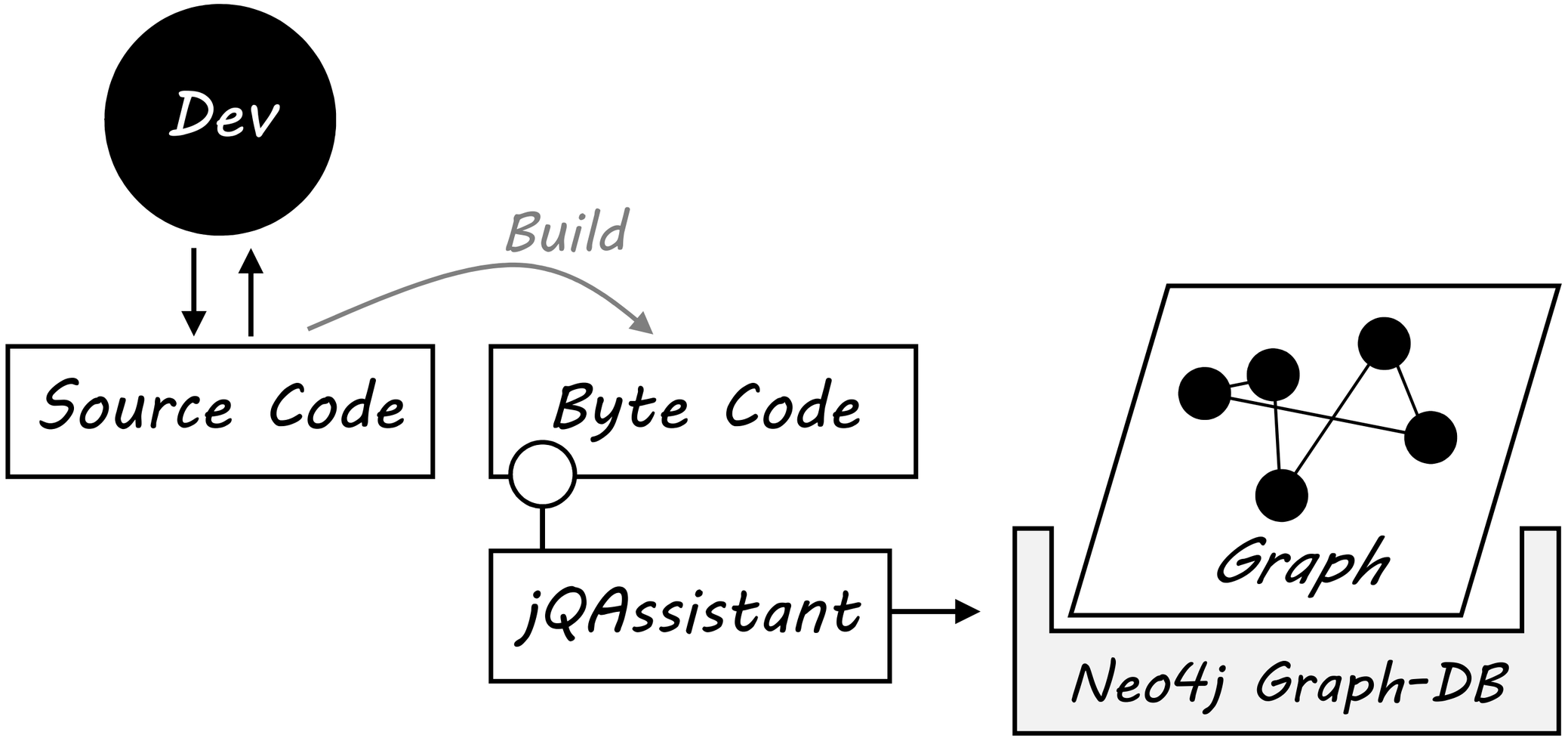
DEMOS

No-Go-Areas

Code Smells

Strategic Redesign

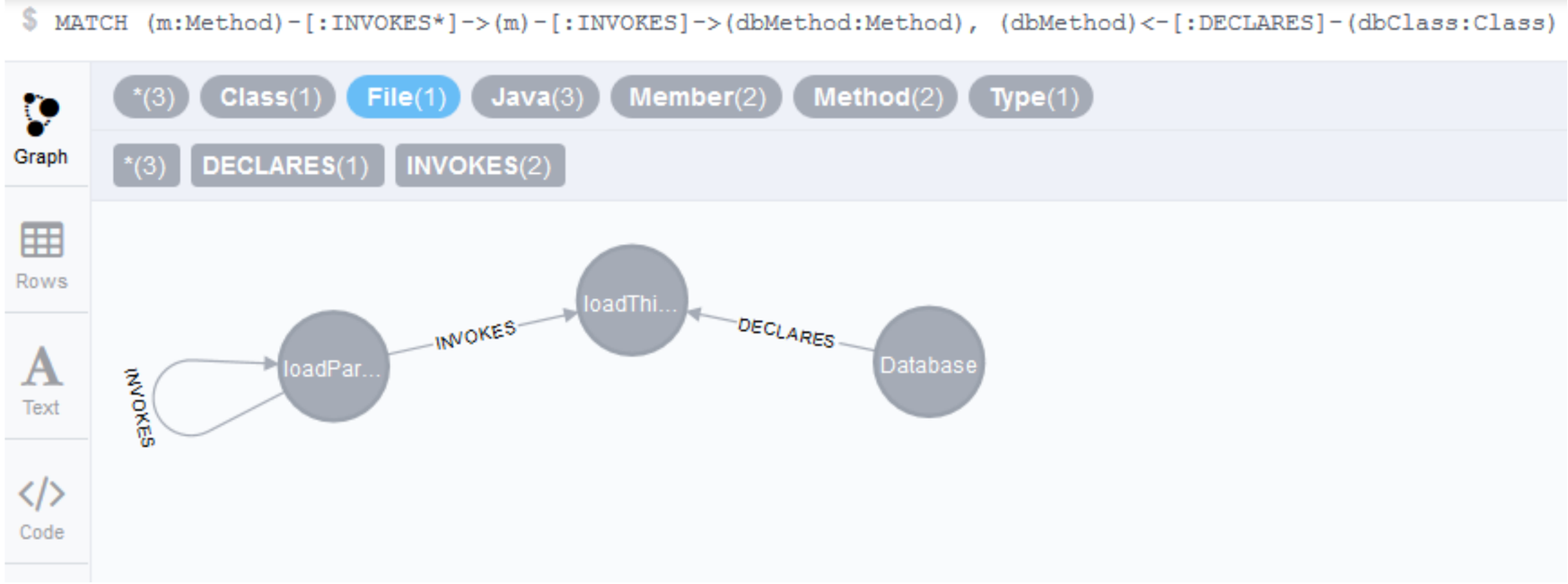
Code Smells: Seeing Software as a Graph



https://git.io/f49KO

Code Smells: Seeing Software as a Graph

```
1 // Rekursive Aufrufe zur Datenbank
2 MATCH
3   (m:Method)-[:INVOKES*]->(m)-[:INVOKES]->(dbMethod:Method),
4   (dbMethod)<-[:DECLARES]-(dbClass:Class)
5 WHERE dbClass.name ENDS WITH "Database"
6 RETURN m, dbMethod, dbClass
```



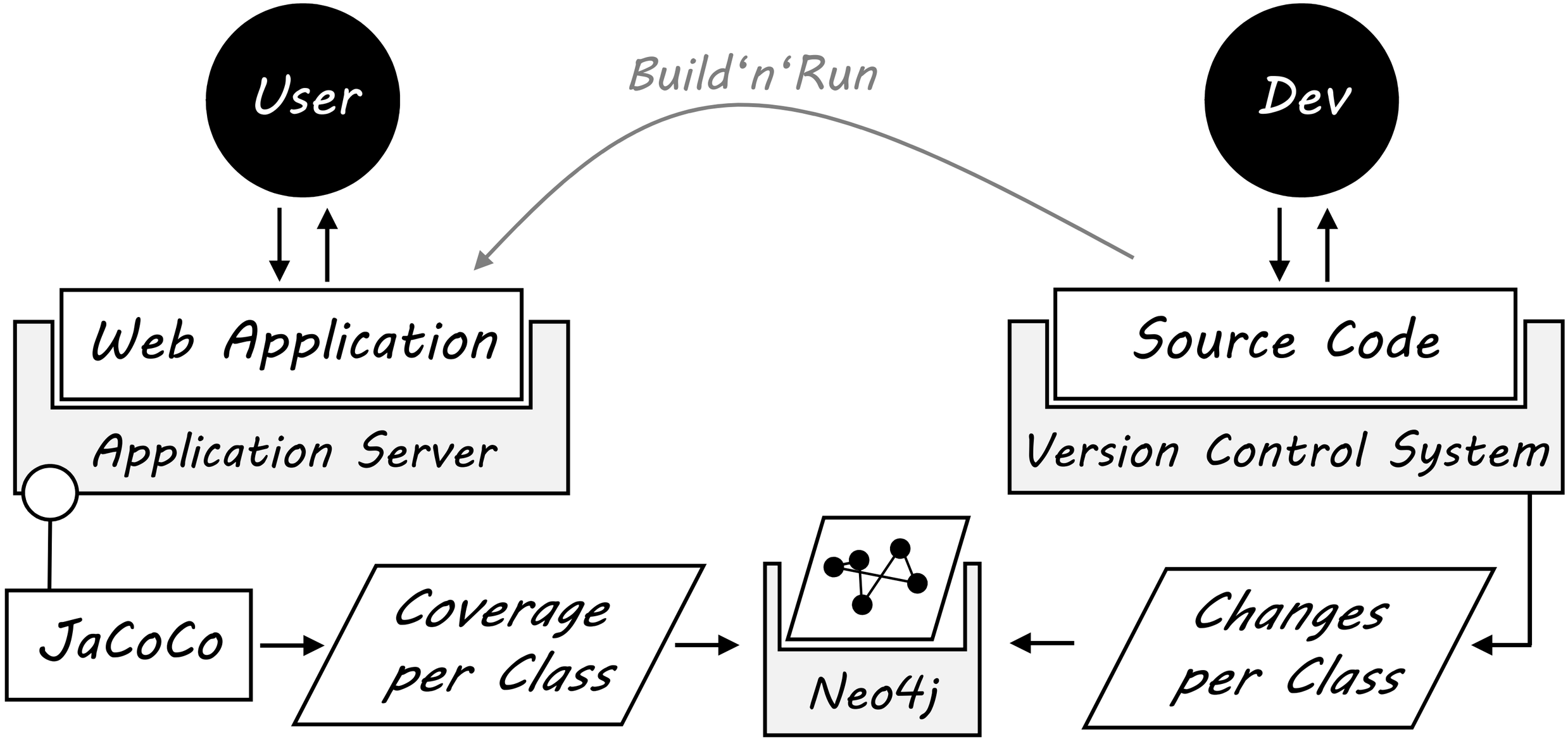
DEMOS

No-Go-Areas

Code Smells

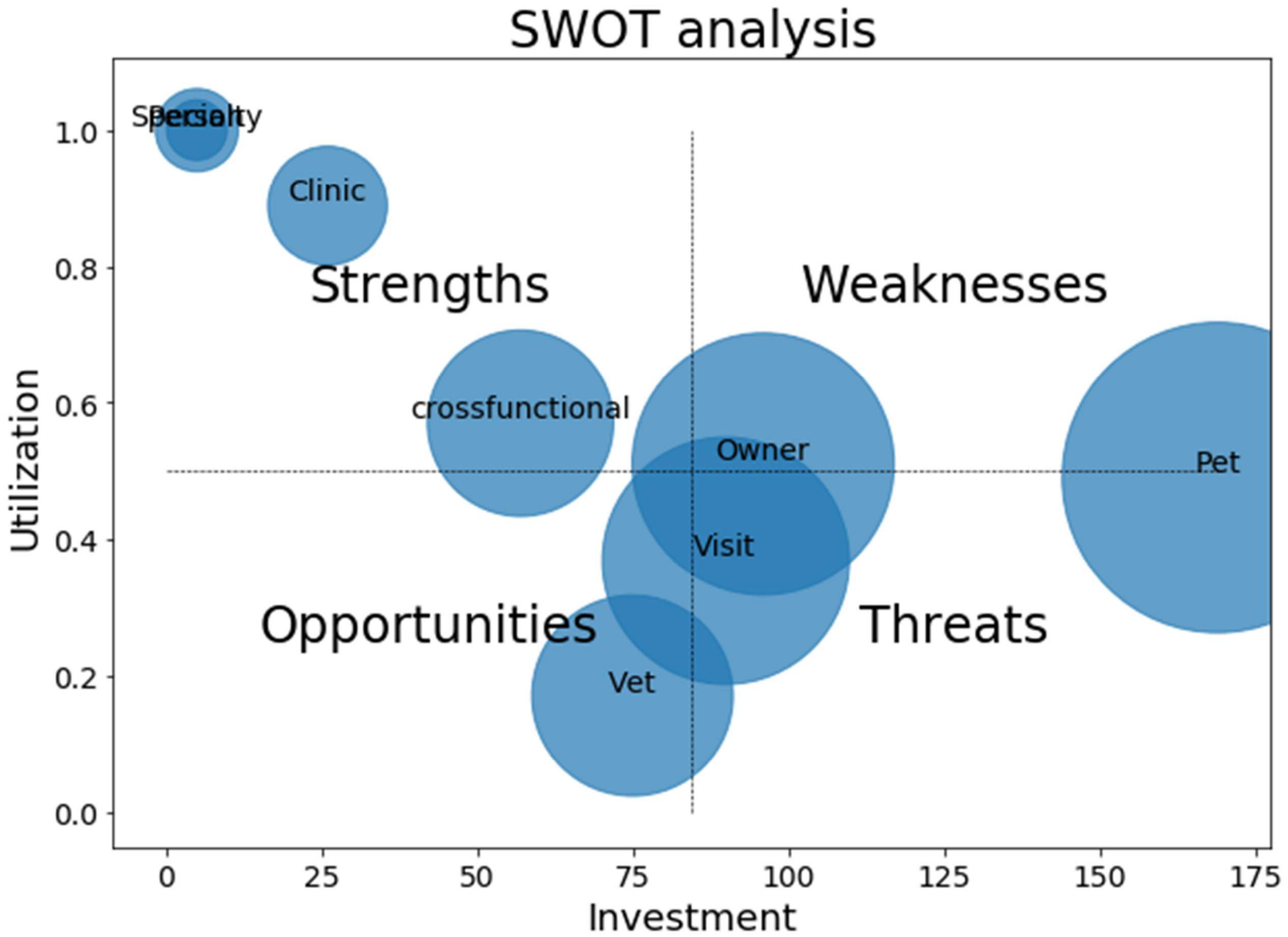
Strategic Redesign

Strategic Redesign: Fixing code that's actually used

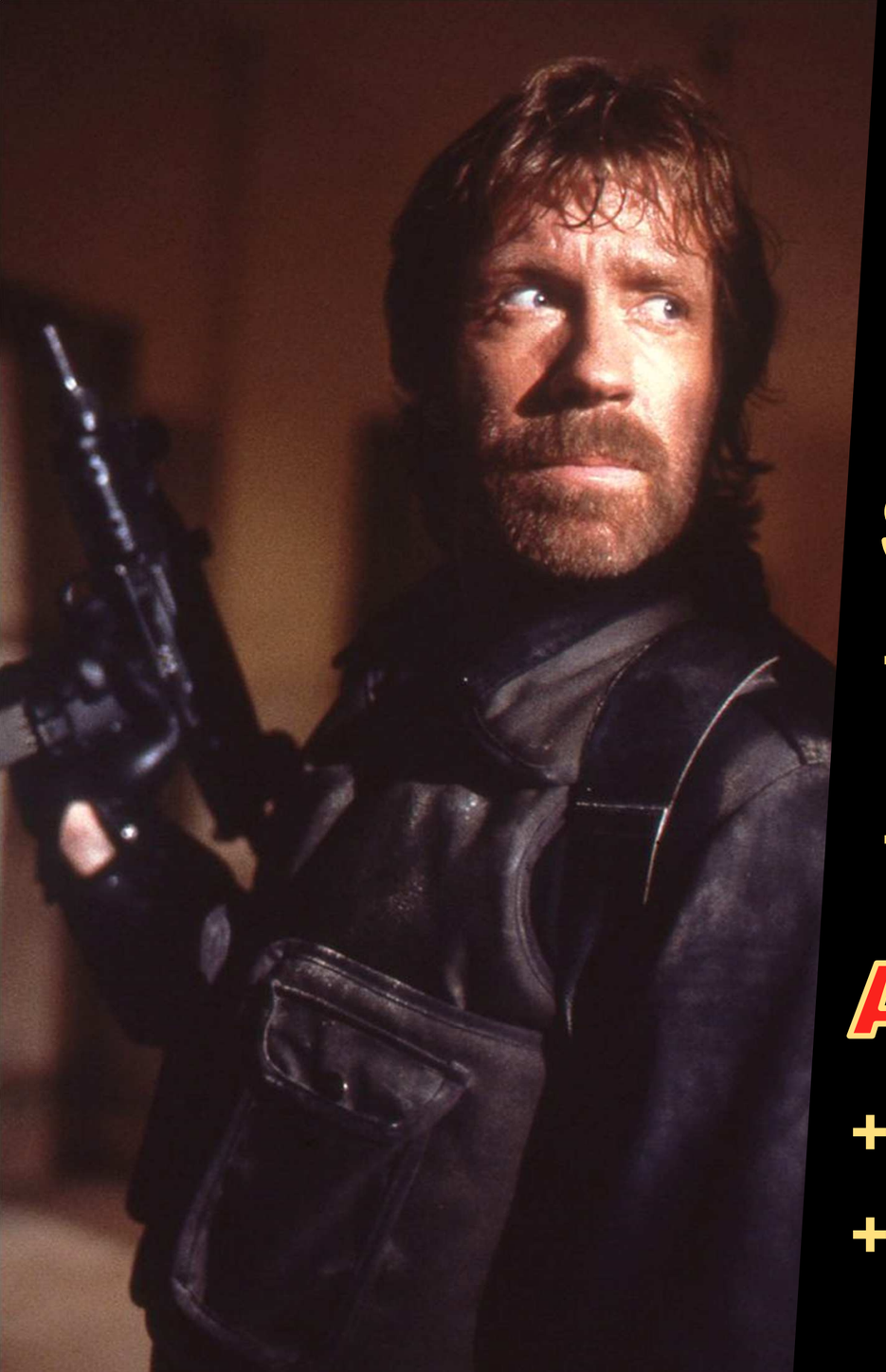


https://www.feststelltaste.de/swot-analysis-for-spotting-worthless-code/

Strategic Redesign: Fixing code that's actually used



Zusammenfassung



Werkzeuge sind da!

+ Erste Schritte sind einfach durchführbar
+ Tiefergehende Analysen sind es wert

Sichtbarkeit möglich!

+ Probleme im Code können identifiziert, aggregiert und kommuniziert werden
+ Sichten für Nicht-Techniker erstellbar

Analysen wirken!

+ Risiken neutral darstellen und adressieren
+ Ursachen beheben statt Symptome fixen!

Mehr Informationen

Literatur

- Christian Bird, Tim Menzies, Thomas Zimmermann:
  The Art and Science of Analyzing Software Data
- Tim Menzies, Laurie Williams, Thomas Zimmermann:
  Perspectives on Data Science for Software Engineering
- Wes McKinney: Python For Data Analysis
- Adam Tornhill: Software Design X-Ray

Software

- Python Data Science Distribution: anaconda.com
- DataCamp: https://projects.datacamp.com/projects/111
- jQAssistant: github.com/JavaOnAutobahn/spring-petclinic
- My Repo: github.com/feststelltaste/software-analytics


THE END

QUESTIONS

ASK ´ EM ALL

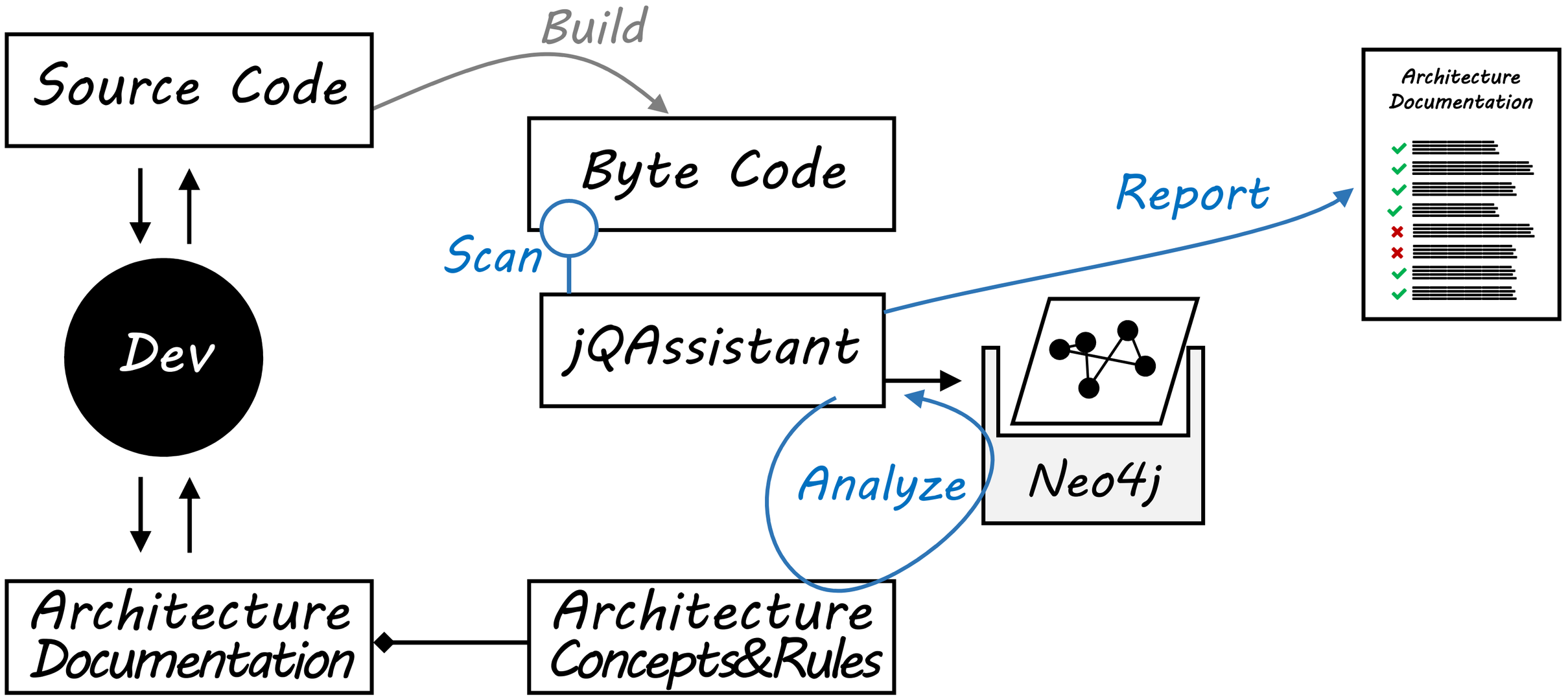
Was?

Bonus Demos

Bonus Demo

Living, self-validating architecture documentation

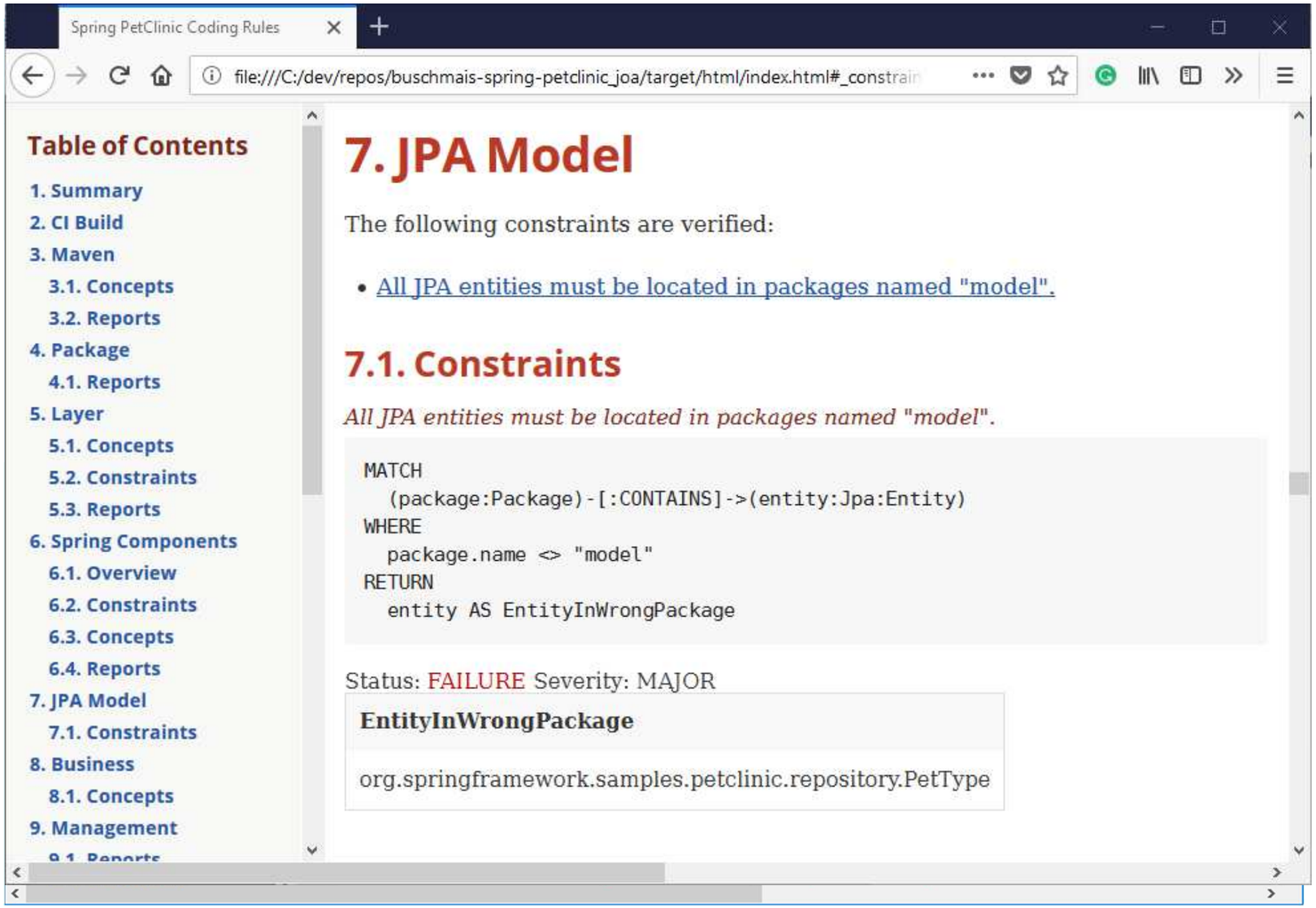
Living, self-validating architecture documentation



https://git.io/f49KO

Living, self-validating architecture documentation

- Architecture documentation with AsciiDoc
- Definition of concepts aka pattern language
- Rule validation by checking constraints
- Redocumentation of the actual design



Danke!

www.innoq.com

Markus Harrer
markus.harrer@innoq.com

+49 175 5753640
@feststelltaste

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

INNOQ

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

Markus Harrer

Software Development Analyst

bei innoQ Deutschland GmbH

- Datenanalysen in der Softwareentwicklung
- Architektur-, Design- und Code-Reviews
- Reverse- und Re-Engineering von Legacy-Code

Bildnachweise

Emoji One

License: CC BY-SA 4.0
Source: Wikimedia Commons (https://commons.wikimedia.org/wiki/File:Emojione_1F37A.svg)

Michelangelo: Creation of Adam

License: Public Domain
Source: Wikimedia Commons (https://en.wikipedia.org/wiki/The_Creation_of_Adam#/media/File:Creaci%C3%B3n_de_Ad%C3%A1n_(Miguel_%C3%81ngel).jpg)

Edvard Munch: The Screams

License: Public Domain
Source: Wiki Commons (https://commons.wikimedia.org/wiki/File:The_Scream.jpg)

Albert Einstein: Abhandlung

Citation: Einstein, Albert: Quantentheorie des einatomigen idealen Gases – Zweite Abhandlung. In: Sitzungsberichte der preussischen Akademie der Wissenschaften, page 14, Reichsdruckerei
Source: Lorentz Archive (https://www.lorentz.leidenuniv.nl/history/Einstein_archive/Einstein_1925_publication/Pages/paper_1925_12.html)

Python Logo

Adopted based on work by www.python.org (www.python.org)
License: GPL (http://www.gnu.org/licenses/gpl.html)
Source: Wikimedia Commons (https://commons.wikimedia.org/wiki/File:Python-logo-notext.svg)

Yoni S. Hamenahem: Chuck Norris - The Delta Force 1986

License: CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0/deed.en)
Source: Wikimedia Common (https://commons.wikimedia.org/wiki/File:Chuck_Norris,_The_Delta_Force_1986.jpg)