

Codeballast über Bord?!

Erfahrungen aus der Identifikation von nutzlosem Code
und dem Umgang damit in der Praxis

Dr. Tobias Röhm (@langelot)
CQSE GmbH





Freelancer

TUM

CQSE

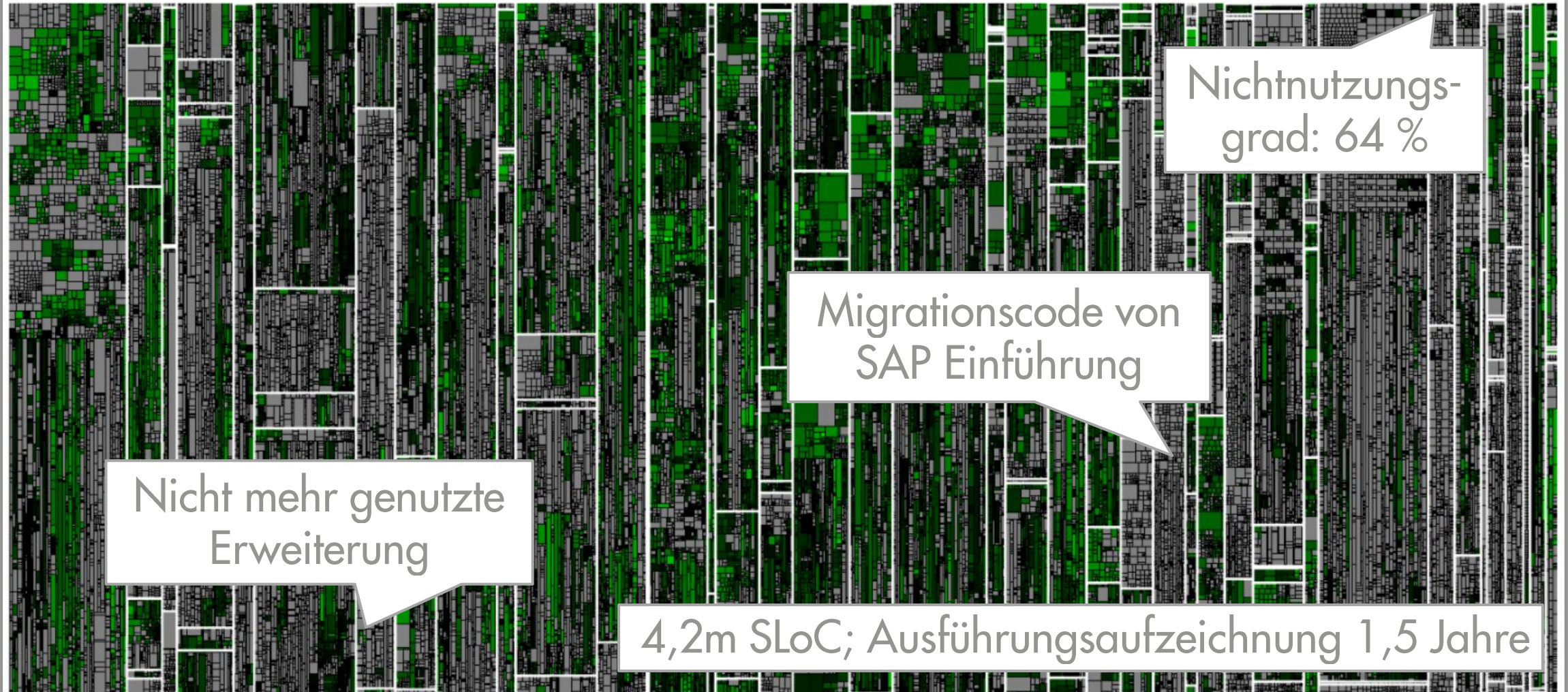


2004 - 2010

2010 - 2015

2015 - Heute

Execution Count: 0 1 10³ 10⁶ 10¹⁰ | Unexecuted: 64.09%



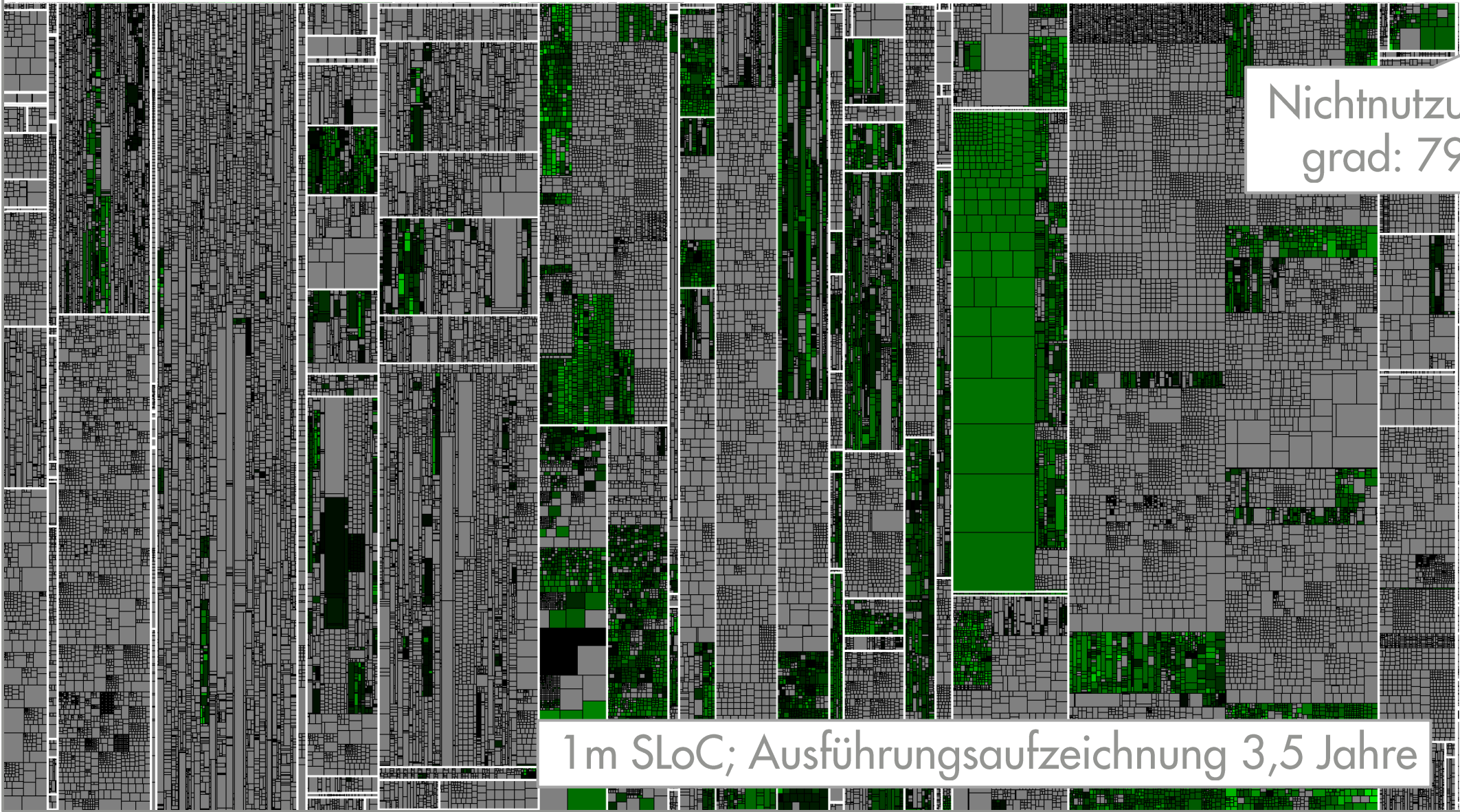
Nichtnutzungsgrad: 64 %

Migrationscode von SAP Einführung

Nicht mehr genutzte Erweiterung

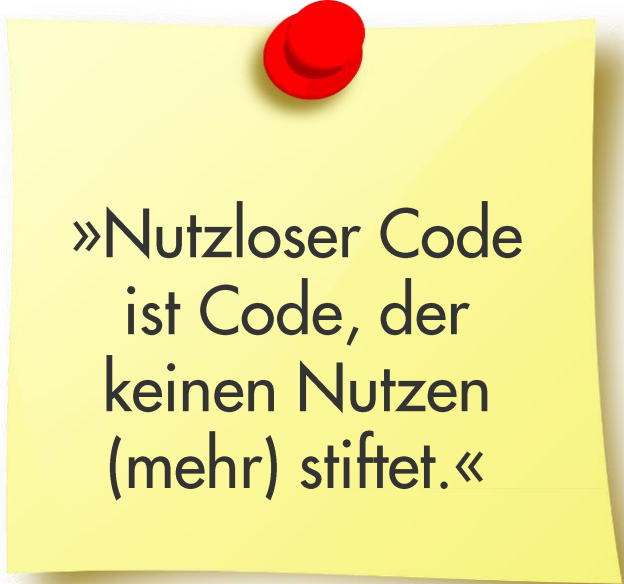
4,2m SLoC; Ausführungsaufzeichnung 1,5 Jahre

Sep 30 2016 01:00 - Now | Execution Count: ■ 0 ■ 1 ■ 10⁴ ■ 10⁸ ■ 10¹² | Unexecuted: 78.94%

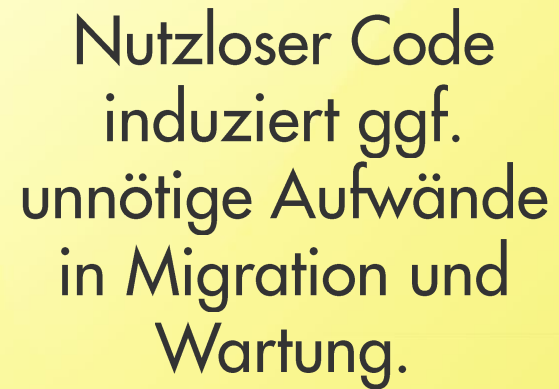


Nichtnutzungsgrad: 79 %

1 m SLoC; Ausführungsaufzeichnung 3,5 Jahre



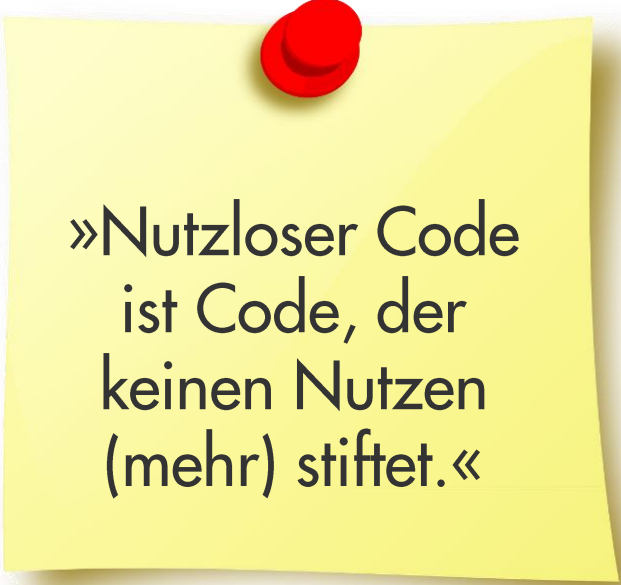
»Nutzloser Code
ist Code, der
keinen Nutzen
(mehr) stiftet.«



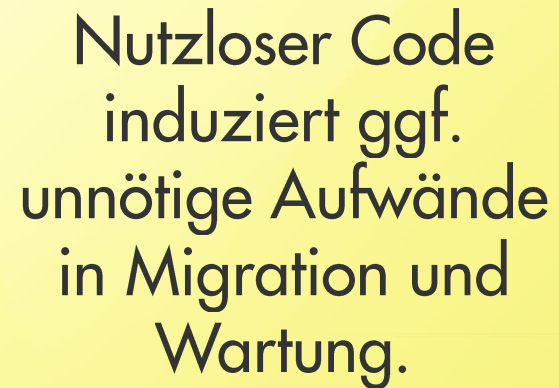
Nutzloser Code
induziert ggf.
unnötige Aufwände
in Migration und
Wartung.

Nutzloser Code ...

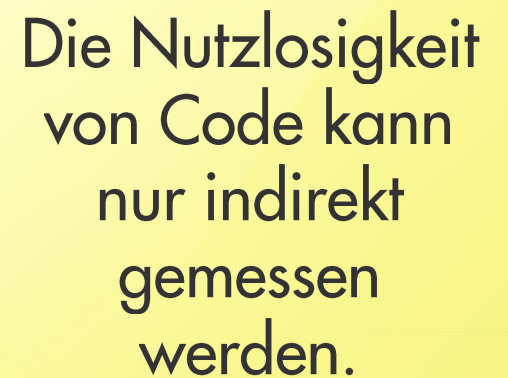
- ↑ ... vergrößert ggf. das Executable
- ↑ ... verlängert ggf. Kompilierzeiten und kostet Buildressourcen
- ↑ ... verlängert ggf. Testlaufzeiten und kostet Testressourcen
- ↑ ... vergrößert ggf. den Aufwand für Wartung
- ↑ ... vergrößert ggf. den Aufwand für Migration
- ↑ ... vergrößert ggf. Angriffsfläche für Hacker



»Nutzloser Code
ist Code, der
keinen Nutzen
(mehr) stiftet.«



Nutzloser Code
induziert ggf.
unnötige Aufwände
in Migration und
Wartung.



Die Nutzlosigkeit
von Code kann
nur indirekt
gemessen
werden.

Agenda

- 1) Identifikation von nutzlosem Code: Analysen und Erfahrungen
 - Via Unerreichbarkeit
 - Via Änderungsstabilität
 - Via Nichtausführung
- 2) Umgang mit (vermutlich) nutzlosem Code

Agenda

1) Identifikation von nutzlosem Code: Analysen und Erfahrungen

Via Unerreichbarkeit

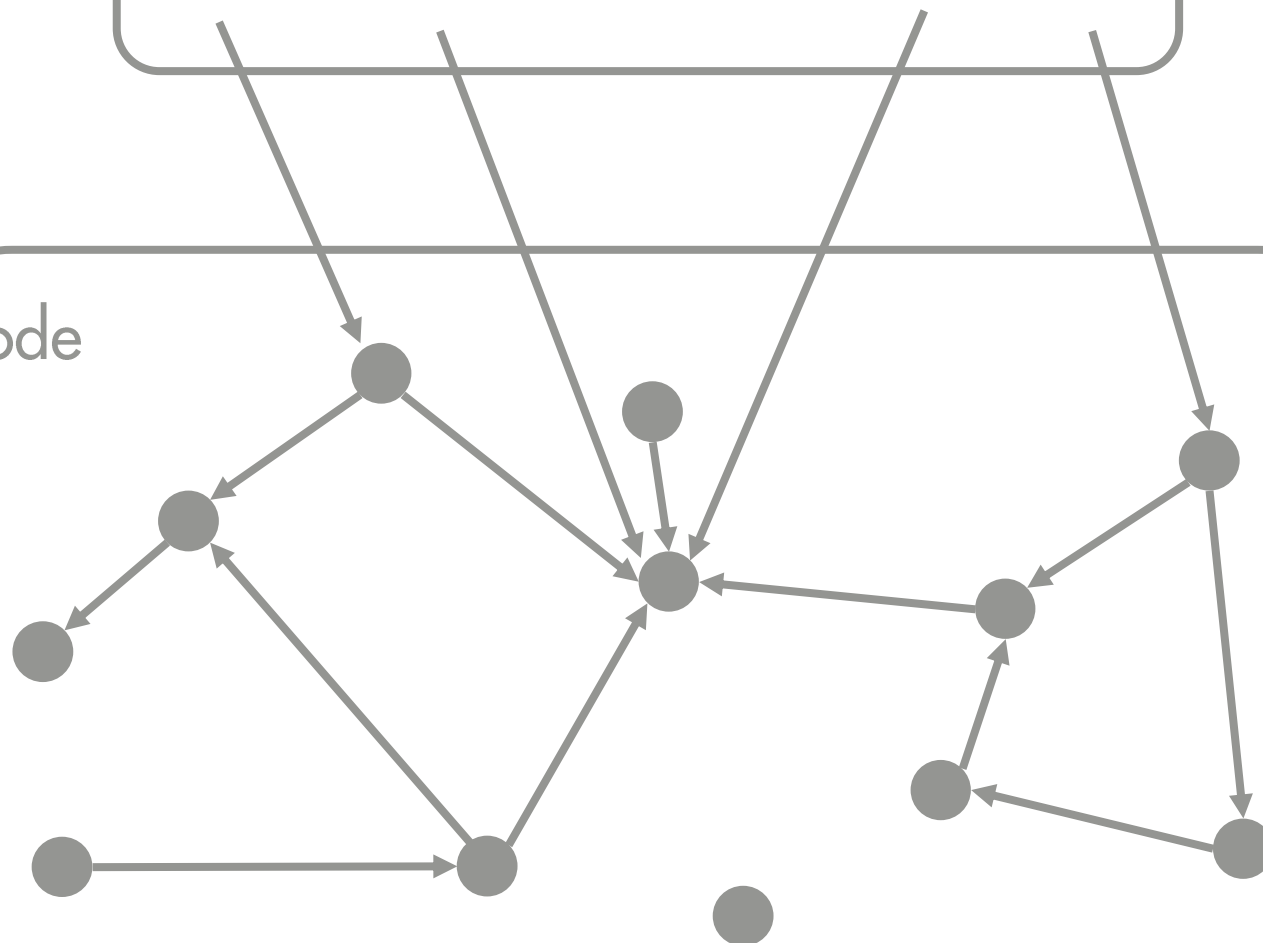
Via Änderungsstabilität

Via Nichtausführung

2) Umgang mit (vermutlich) nutzlosem Code

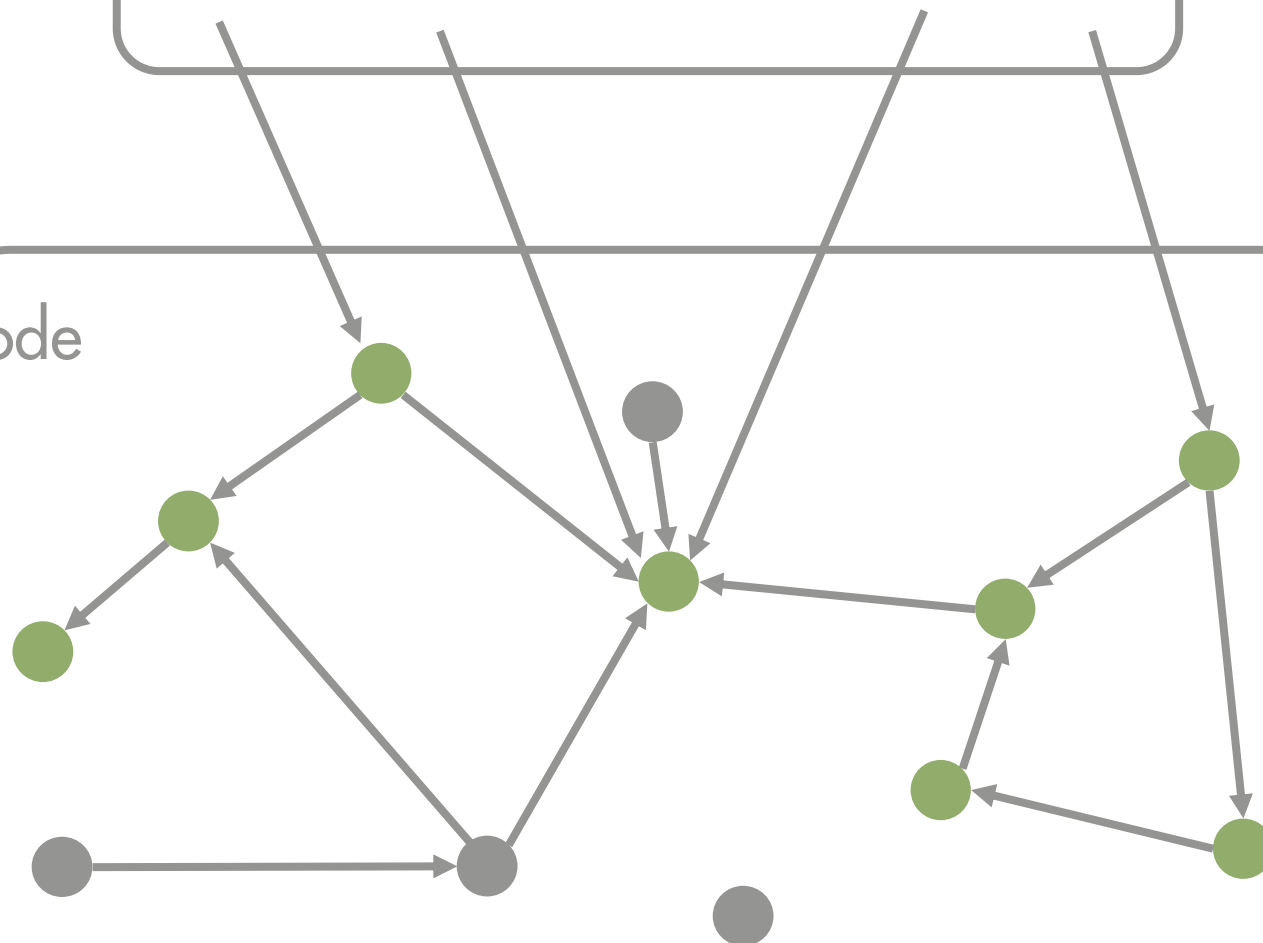
Einsprungpunkt(e)

Code



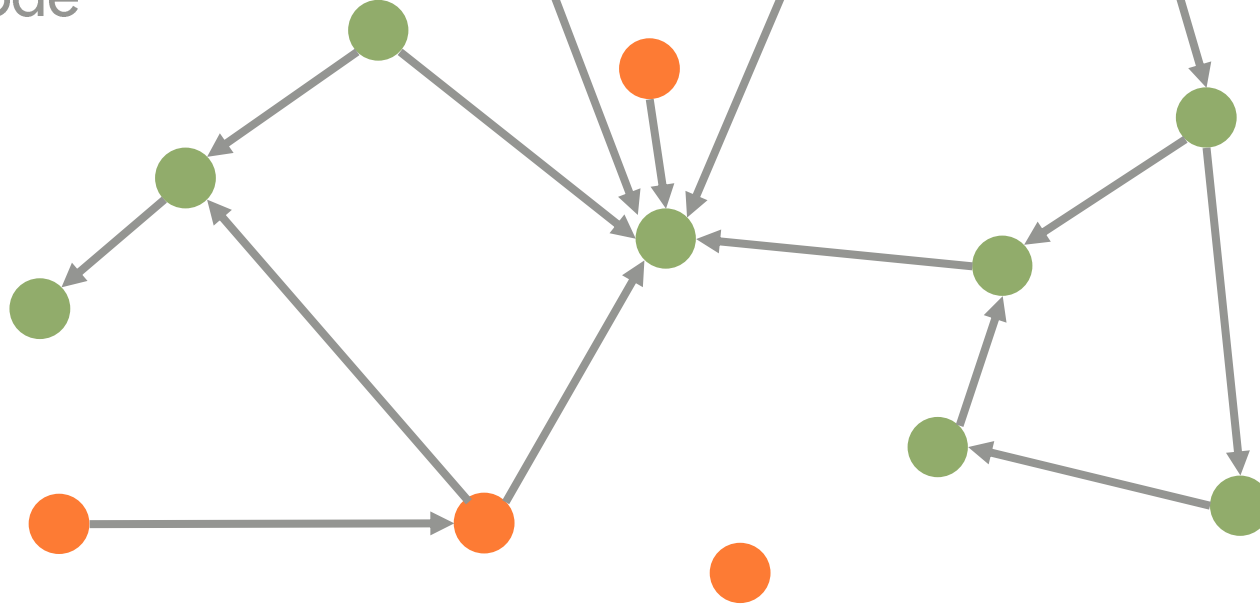
Einsprungpunkt(e)

Code



Einsprungpunkt(e)

Code



Knoten

Module/ Pakete

Klassen

Methoden


Kanten

Imports


Referenzen

Aufrufe



```
19
20 |     private void unusedPrivateMethod () {
21       System.out.println("Hello World!");
22     }
23
```

 **Private method `unusedPrivateMethod` is unused**

```
public class X {
    void foo() {}
    void bar() {
        if (false)
            foo();
    }
}
```

 Dead code

1 quick fix available:

-  [Remove \(including condition\)](#)

Press 'F2' for focus

Jenkins-Code:
13 ungenutzte private Methoden in 217k LoC.

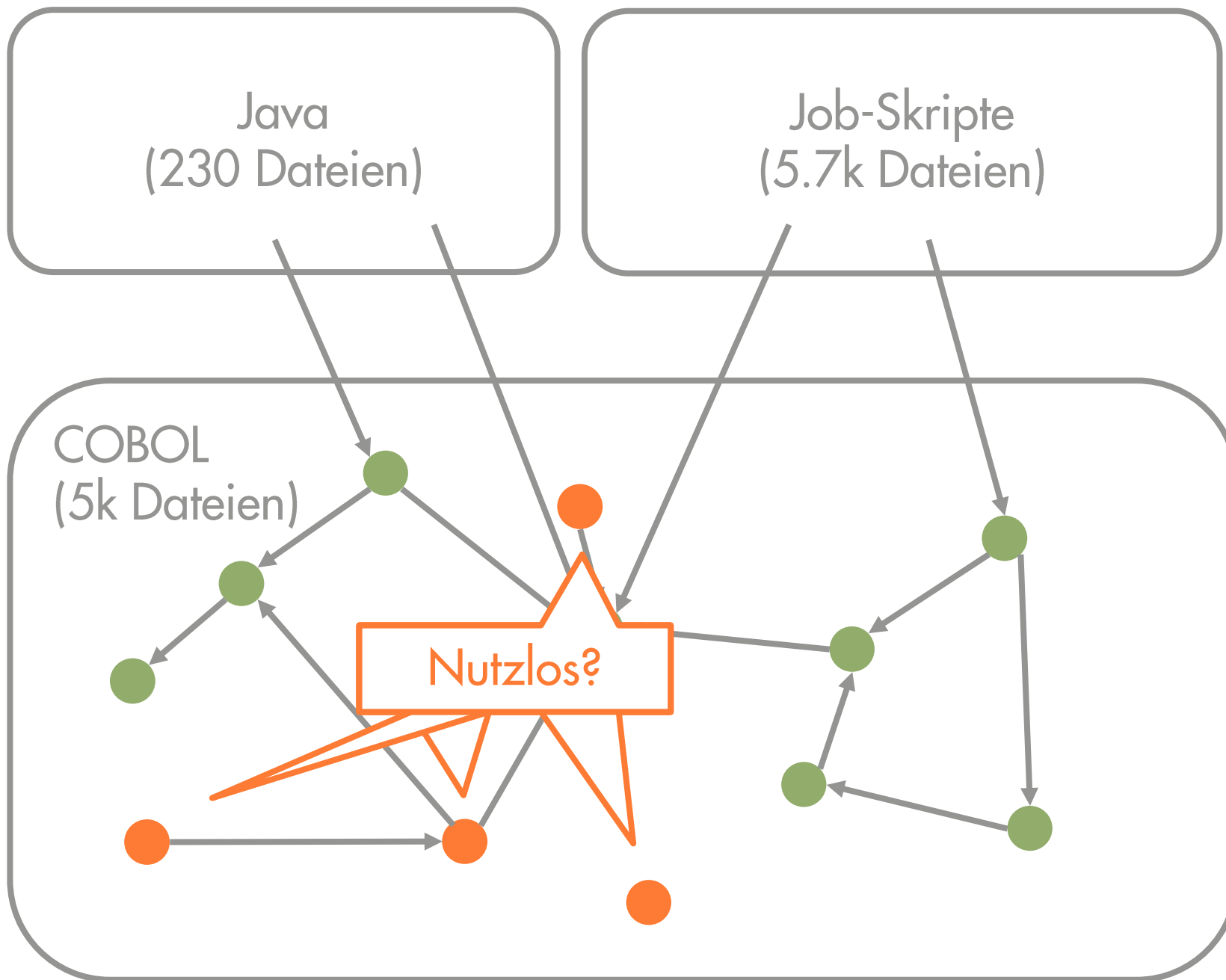
Java
(230 Dateien)

Job-Skripte
(5.7k Dateien)

COBOL
(5k Dateien)

Nutzlos?

173k von 911k SLoC
(19%) unerreichbar



Agenda

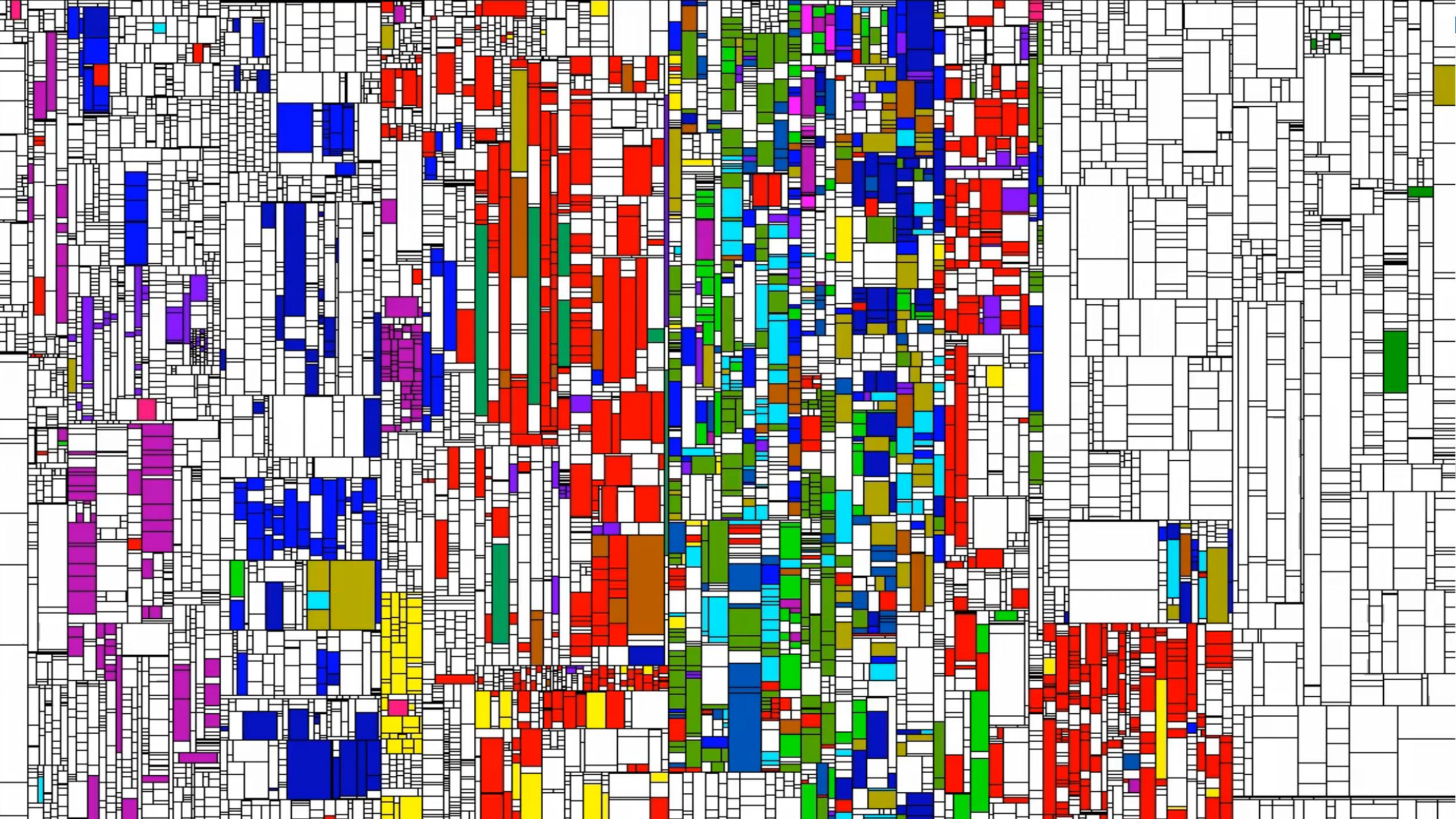
1) Identifikation von nutzlosem Code: Analysen und Erfahrungen

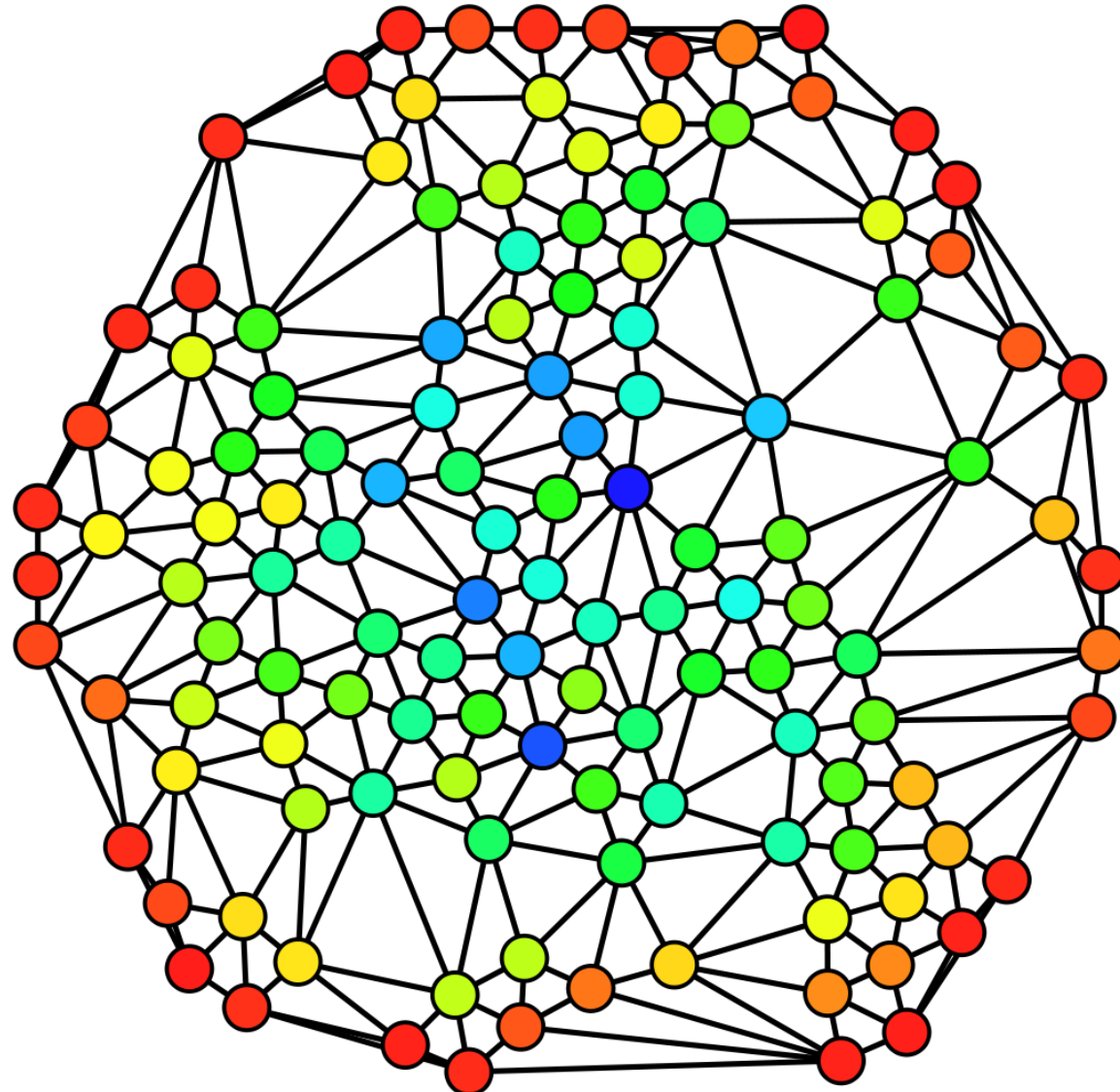
Via Unerreichbarkeit

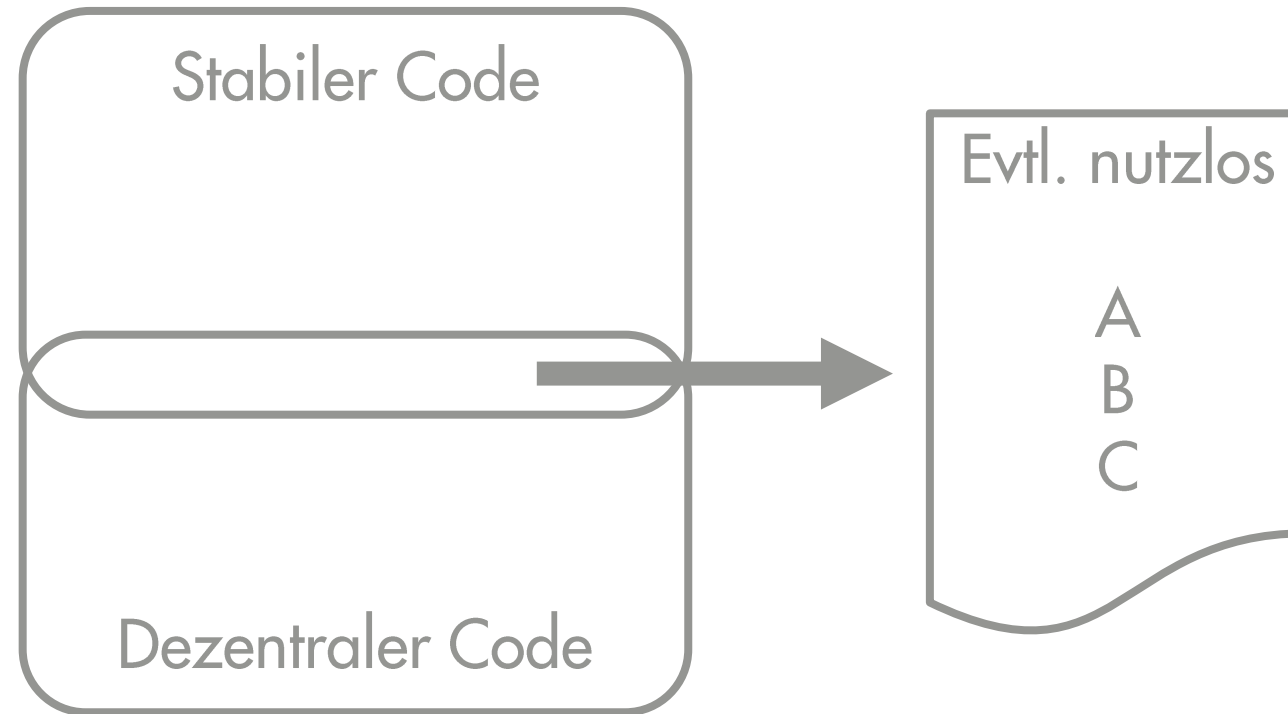
Via Änderungsstabilität

Via Nichtausführung

2) Umgang mit (vermutlich) nutzlosem Code







20 von 50 Vorschlägen tatsächlich nutzlos (40 %),
10 von 50 Vorschlägen gelöscht (20 %)

Agenda

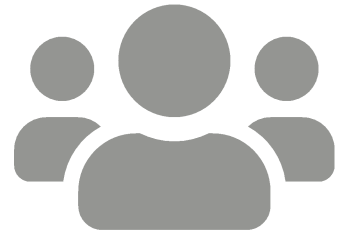
1) Identifikation von nutzlosem Code: Analysen und Erfahrungen

Via Unerreichbarkeit

Via Änderungsstabilität

Via Nichtausführung

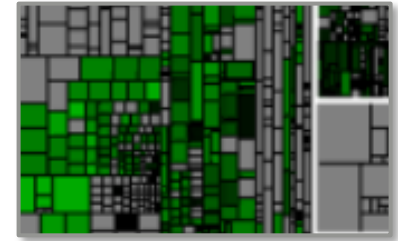
2) Umgang mit (vermutlich) nutzlosem Code



Nutzer



Produktivsystem



2019

January

	Mo	Tu	We	Th	Fr	Sa	Su
1		1	2	3	4	5	6
2	7	8	9	10	11	12	13
3	14	15	16	17	18	19	20
4	21	22	23	24	25	26	27
5	28	29	30	31			

February

	Mo	Tu	We	Th	Fr	Sa	Su
5					1	2	3
6	4	5	6	7	8	9	10
7	11	12	13	14	15	16	17
8	18	19	20	21	22	23	24
9	25	26	27	28			

March

	Mo	Tu	We	Th	Fr	Sa	Su
9					1	2	3
10	4	5	6	7	8	9	10
11	11	12	13	14	15	16	17
12	18	19	20	21	22	23	24
13	25	26	27	28	29	30	31

April

	Mo	Tu	We	Th	Fr	Sa	Su
14	1	2	3	4	5	6	7
15	8	9	10	11	12	13	14
16	15	16	17	18	19	20	21
17	22	23	24	25	26	27	28
18	29	30					

May

	Mo	Tu	We	Th	Fr	Sa	Su
18			1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26

June

	Mo	Tu	We	Th	Fr	Sa	Su
22						1	2
23	3	4	5	6	7	8	9
24	10	11	12	13	14	15	16
25	17	18	19	20	21	22	23
26	24	25	26	27	28	29	30

July

	Mo	Tu	We	Th	Fr	Sa	Su
27	1	2	3	4	5	6	7
28	8	9	10	11	12	13	14
29	15	16	17	18	19	20	21
30	22	23	24	25	26	27	28
31	29	30	31				

August

	Mo	Tu	We	Th	Fr	Sa	Su
31				1	2	3	4
32	5	6	7	8	9	10	11
33	12	13	14	15	16	17	18
34	19	20	21	22	23	24	25
35	26	27	28	29	30	31	

Ausbildungsstart

Inventur

September

	Mo	Tu	We	Th	Fr	Sa	Su
35							1
36	2	3	4	5	6	7	8
37	9	10	11	12	13	14	15
38	16	17	18	19	20	21	22
39	23	24	25	26	27	28	29
40	30						

October

	Mo	Tu	We	Th	Fr	Sa	Su
40		1	2	3	4	5	6
41	7	8	9	10	11	12	13
42	14	15	16	17	18	19	20
43	21	22	23	24	25	26	27
44	28	29	30	31			

November

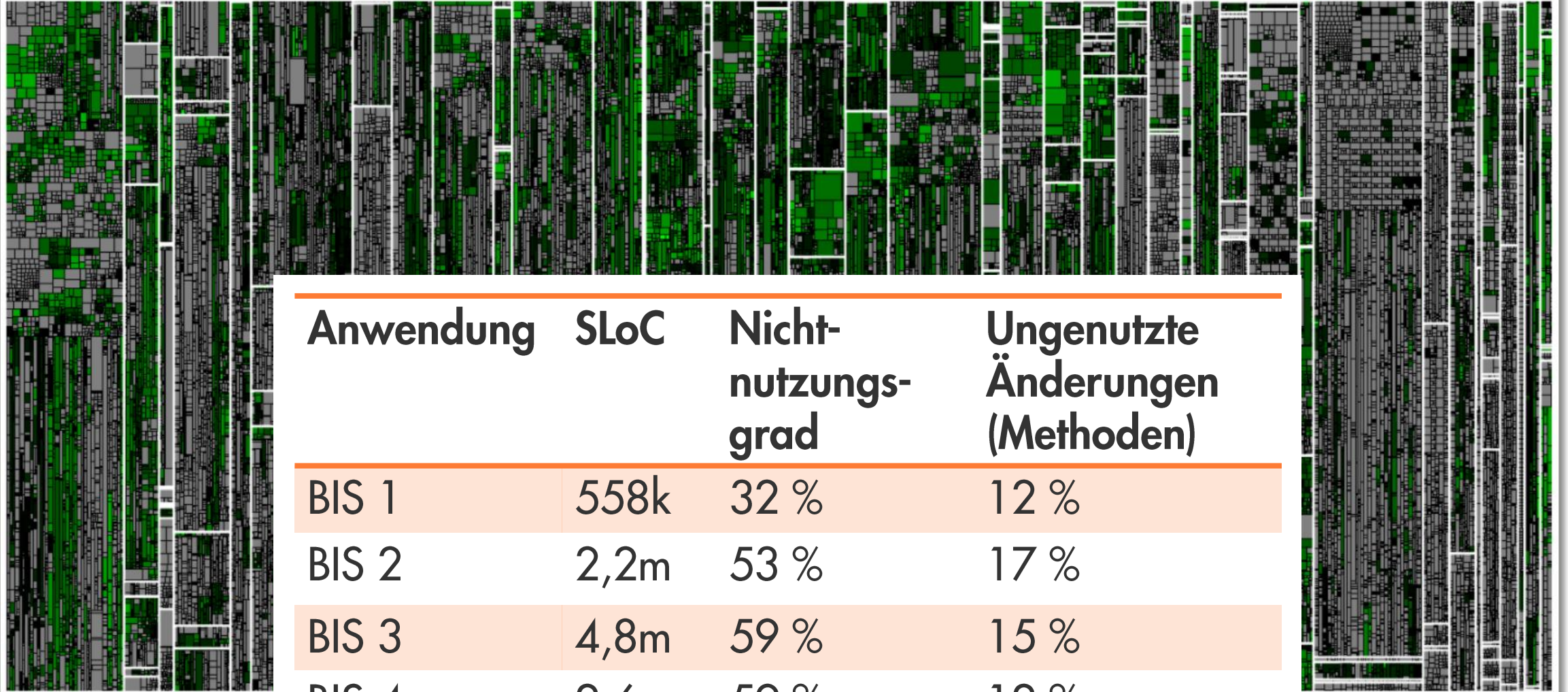
	Mo	Tu	We	Th	Fr	Sa	Su
44					1	2	3
45	4	5	6	7	8	9	10
46	11	12	13	14	15	16	17
47	18	19	20	21	22	23	24
48	25	26	27	28	29	30	

December

	Mo	Tu	We	Th	Fr	Sa	Su
48							
49							
50							
51	16	17	18	19	20	21	22
52	23	24	25	26	27	28	29
1	30	31					

Jahresabschluss

Execution Count: 0 1 10³ 10⁶ 10¹⁰ | Unexecuted: 64.09%



Anwendung	SLoC	Nicht-nutzungs-grad	Ungenutzte Änderungen (Methoden)
BIS 1	558k	32 %	12 %
BIS 2	2,2m	53 %	17 %
BIS 3	4,8m	59 %	15 %
BIS 4	2,6m	59 %	12 %
BIS 5	4,2m	64 %	17 %

Agenda

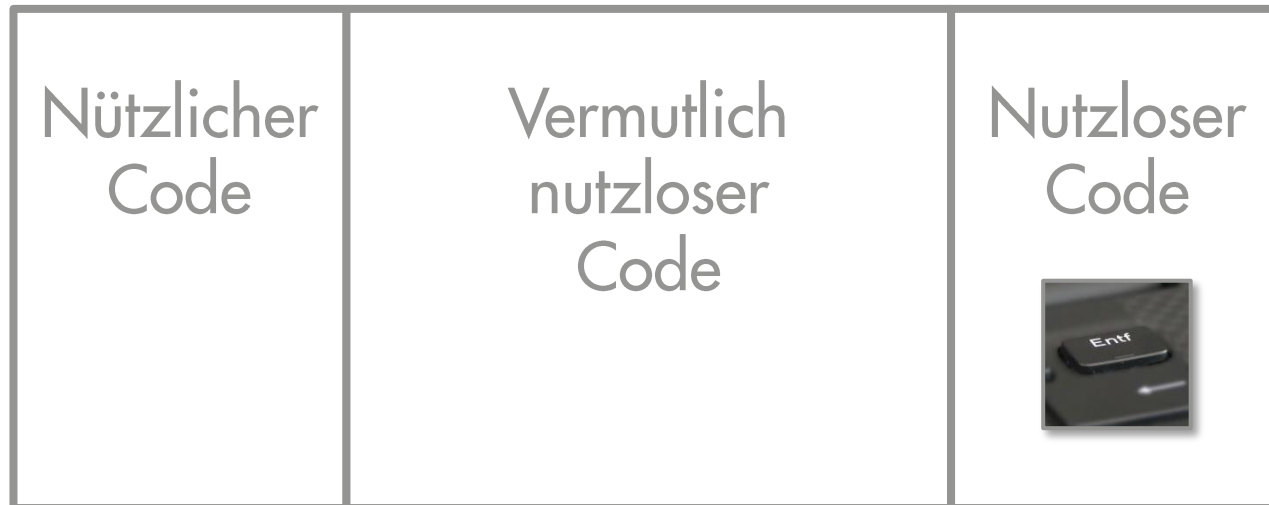
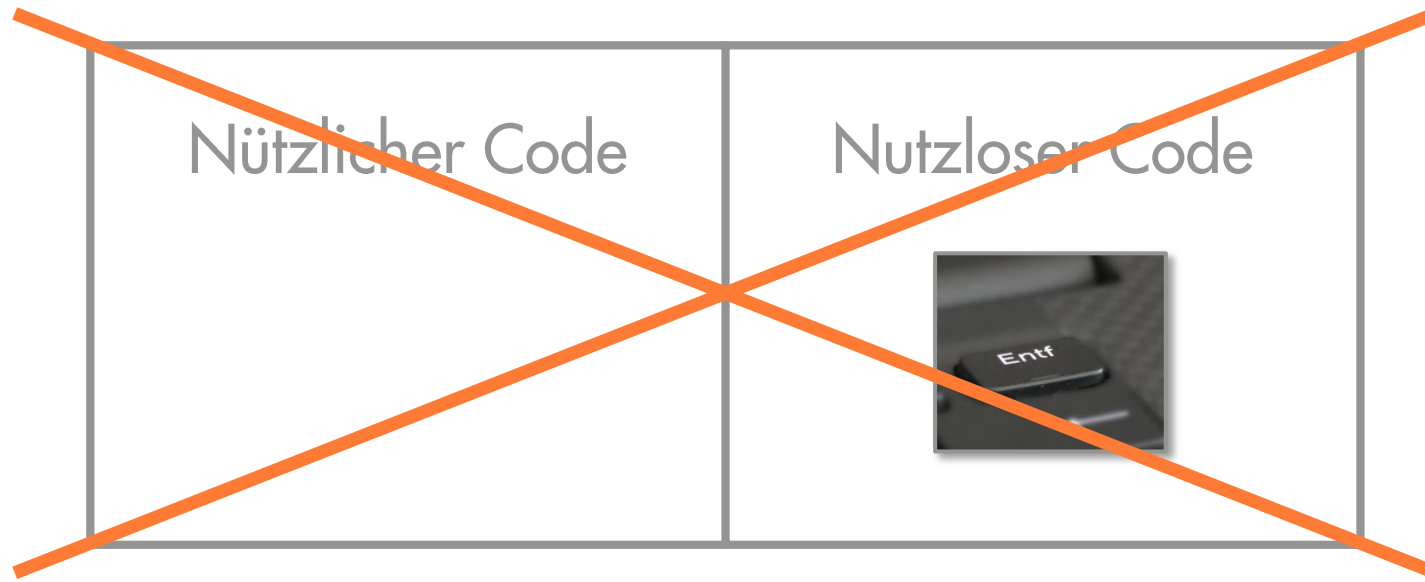
1) Identifikation von nutzlosem Code: Analysen und Erfahrungen

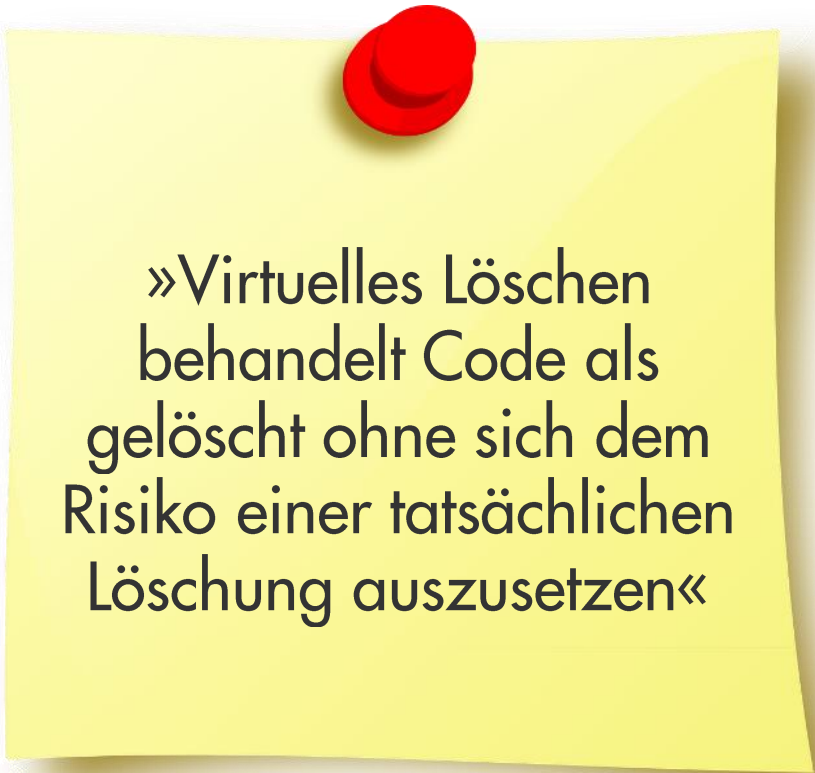
Via Unerreichbarkeit

Via Änderungsstabilität

Via Nichtausführung

2) Umgang mit (vermutlich) nutzlosem Code





»Virtuelles Löschen
behandelt Code als
gelöscht ohne sich dem
Risiko einer tatsächlichen
Löschung auszusetzen«



Markierung für Entwickler



Markierung für Nutzer



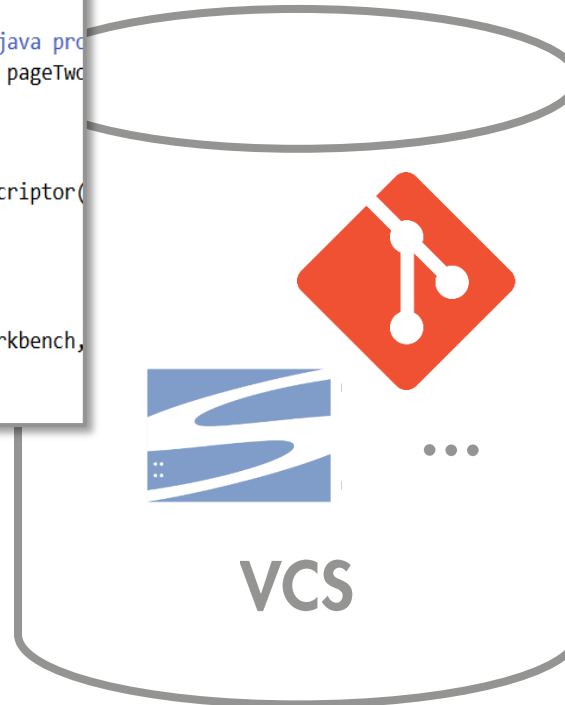
Rückgängig-Funktion



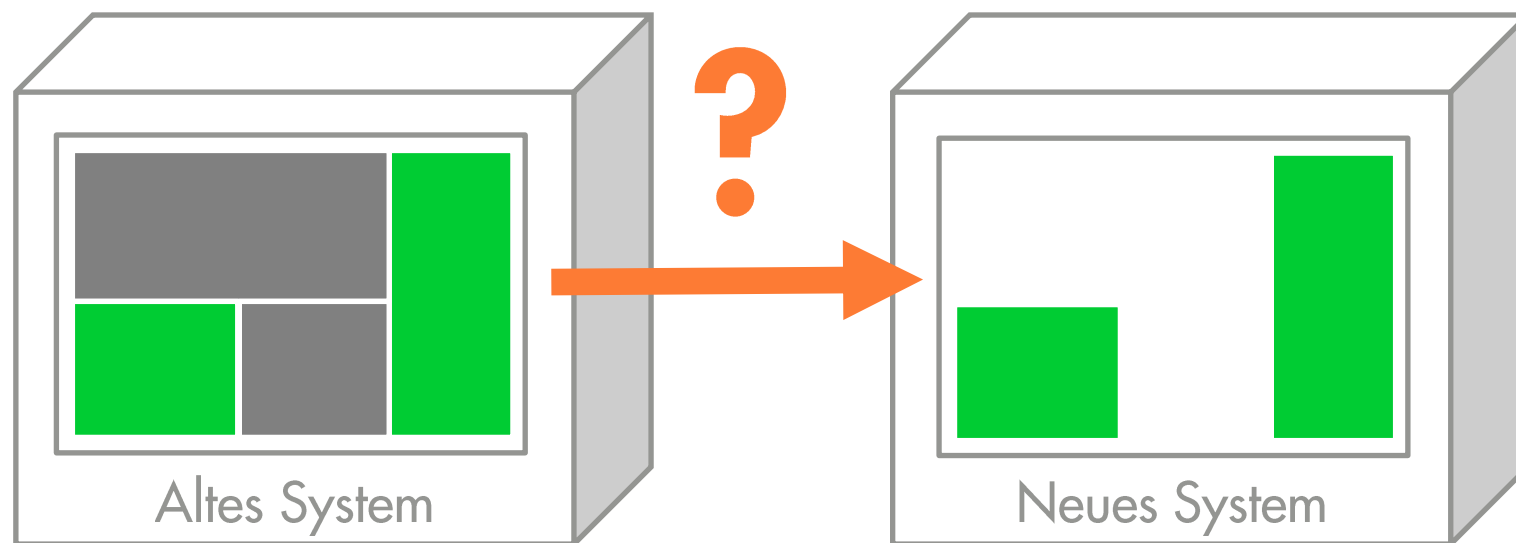
Wiedervorlage-Funktion

Strategie 1: Löschen und ggf. aus Versionskontrollsystem wiederherstellen

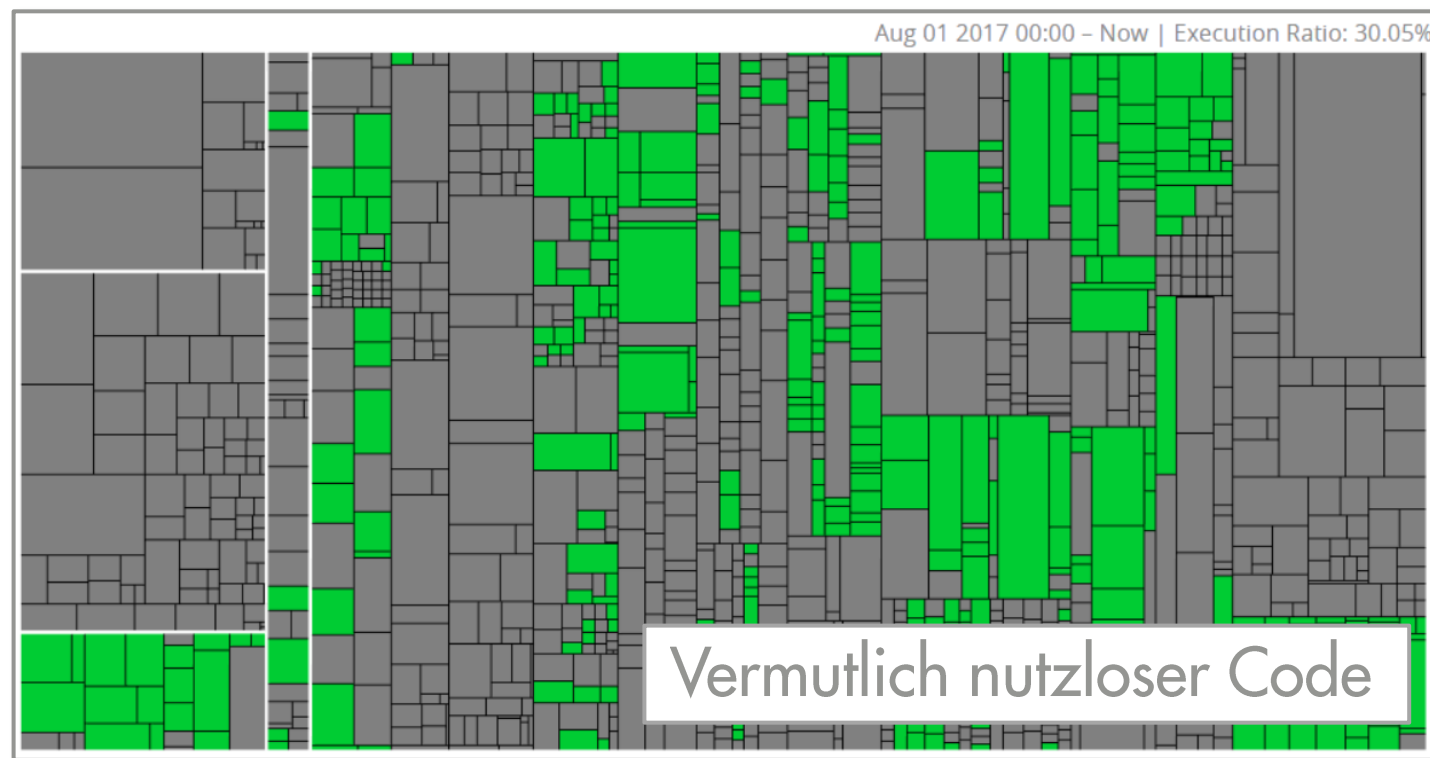
```
33  /**
34   * The wizard class for creating new bundle projects
35   *
36   * @author
37   * @author
38   */
39  public class ... extends Wizard implements ...
40
41  // ...
42  // ...
43  // ...
44  // ...
45  // ...
46  // ...
47  // ...
48
49  /** Empty implementation */
50  @Override
51  public void init(IWorkbench workbench,
52                // Does nothing
53  }
```

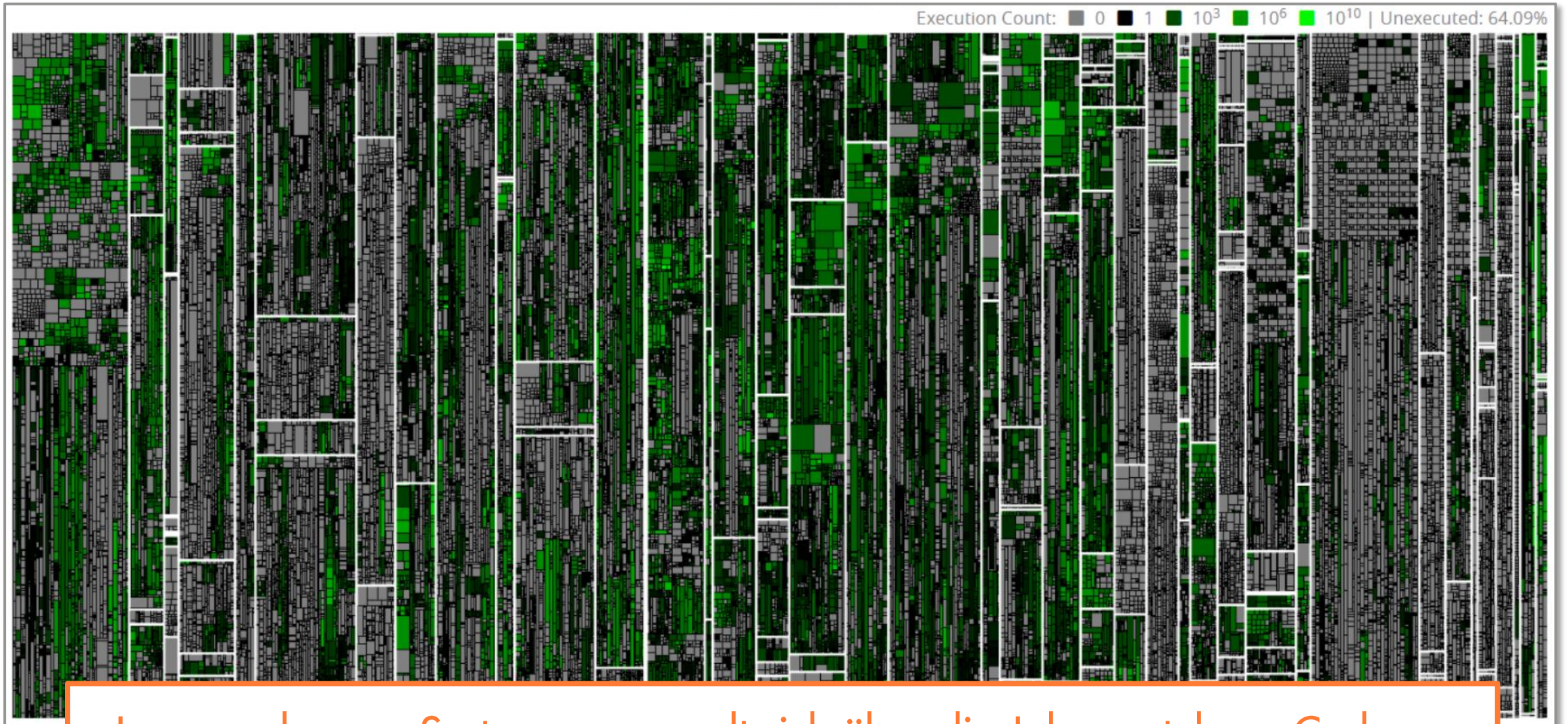


Strategie 2: Ausführungsbasierte (HANA-) Migration

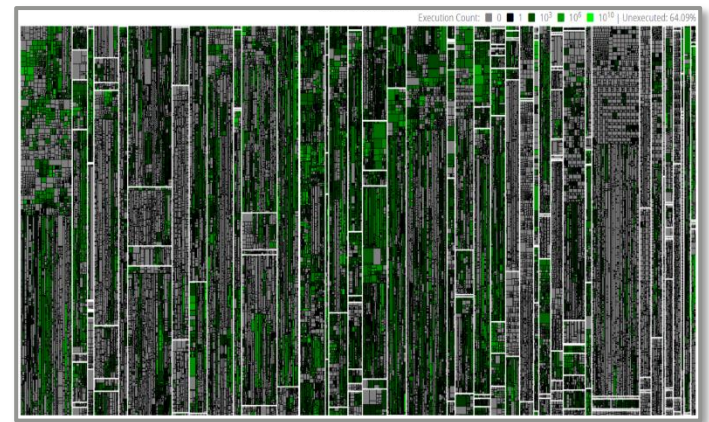
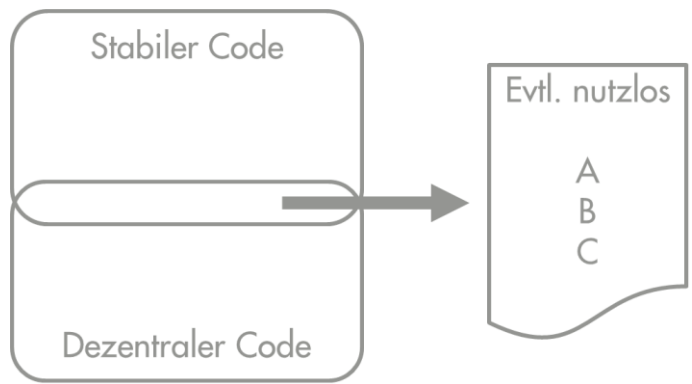
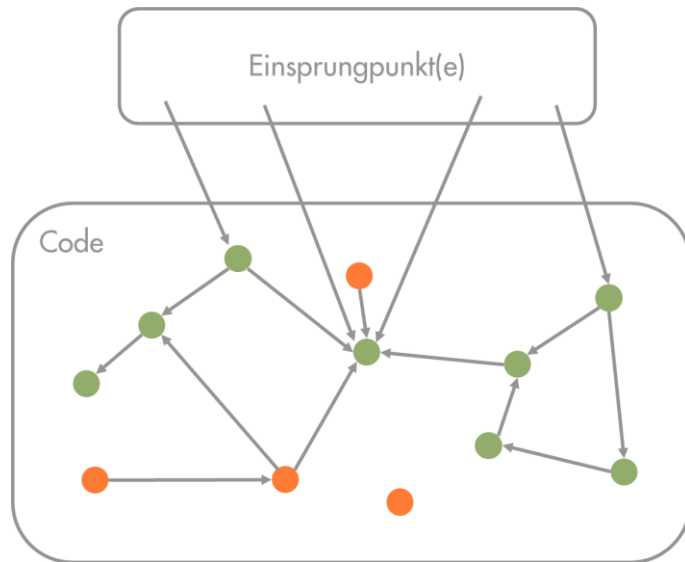


Strategie 3: Markierung und Nachverfolgung über Paket »Obsoleter Code«





In gewachsenen Systemen sammelt sich über die Jahre nutzloser Code an und dieser verursacht ggf. unnötige Kosten bei Migration und Wartung.



Nutzlosigkeit kann nicht direkt gemessen werden, aber es gibt die indirekten Näherungsmaße Unerreichbarkeit, Änderungsstabilität und Nichtausführung.

Nützlicher Code	Vermutlich nutzloser Code	Nutzloser Code 
--------------------	---------------------------------	--

Ein großer Teil der Codebasis ist vermutlich nutzlos,
wird aber aufgrund von Unsicherheit nicht gelöscht.



Markierung für Entwickler



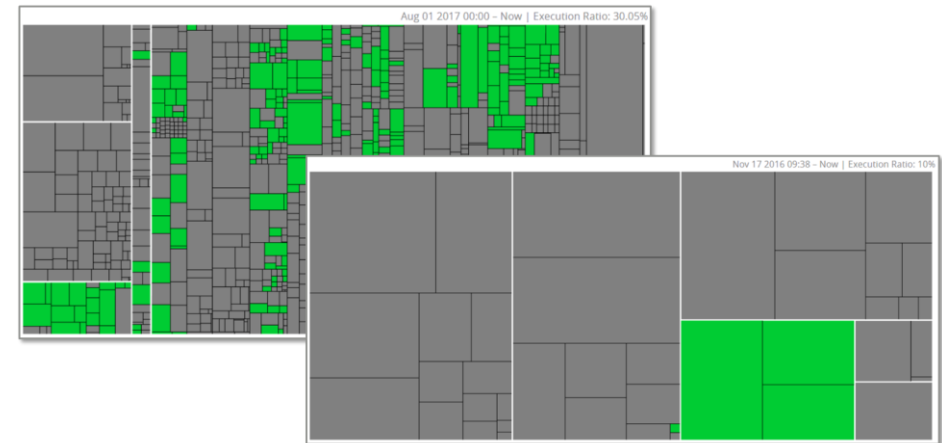
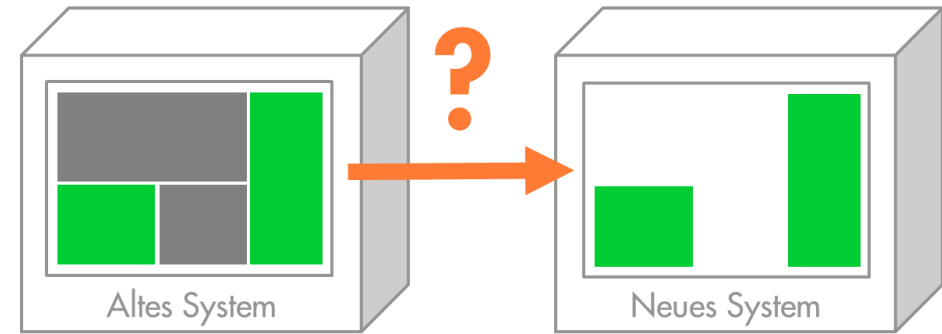
Markierung für Nutzer



Rückgängig-Funktion



Wiedervorlage-Funktion



Deshalb behandelt »Virtuelles Löschen« diesen Code als gelöscht ohne sich dem Risiko einer tatsächlichen Löschung auszusetzen.



Diskussionsmöglichkeiten:

- Jetzt via JFS-Chat
- In Pausen via Google Meet: <http://cqse.eu/jfs2020>
- Immer via Mail: roehm@cqse.eu



Kontakt – Ich freue mich auf Diskussionen 😊



Dr. Tobias Röhm

@langelot · roehm@cqse.eu · +49 159 04330842

Danke an Alexander von Rhein, Elmar Jürgens und Roman Haas für ihre Beiträge!

