

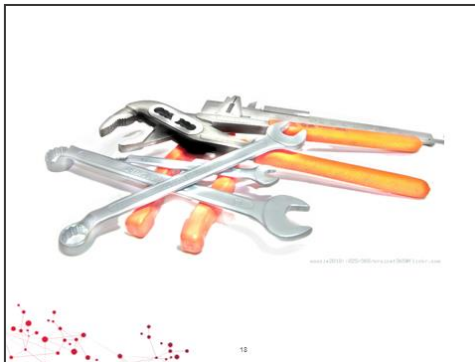
McCabe, Halsted, WTFPS Software Metriken zum Mitnehmen

Tobias Getrost
Mai 2014

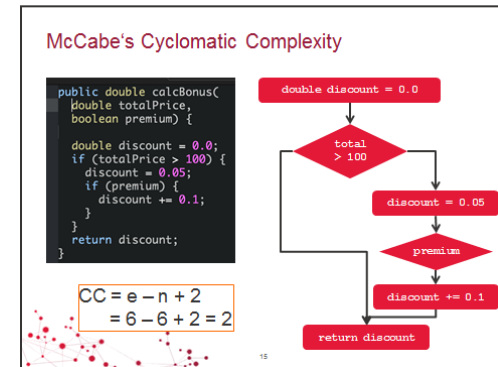
Agenda



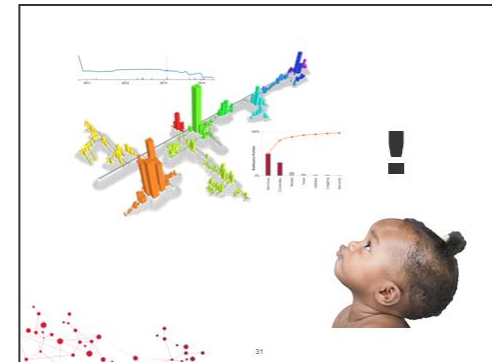
Einführung



Tool



Metriken



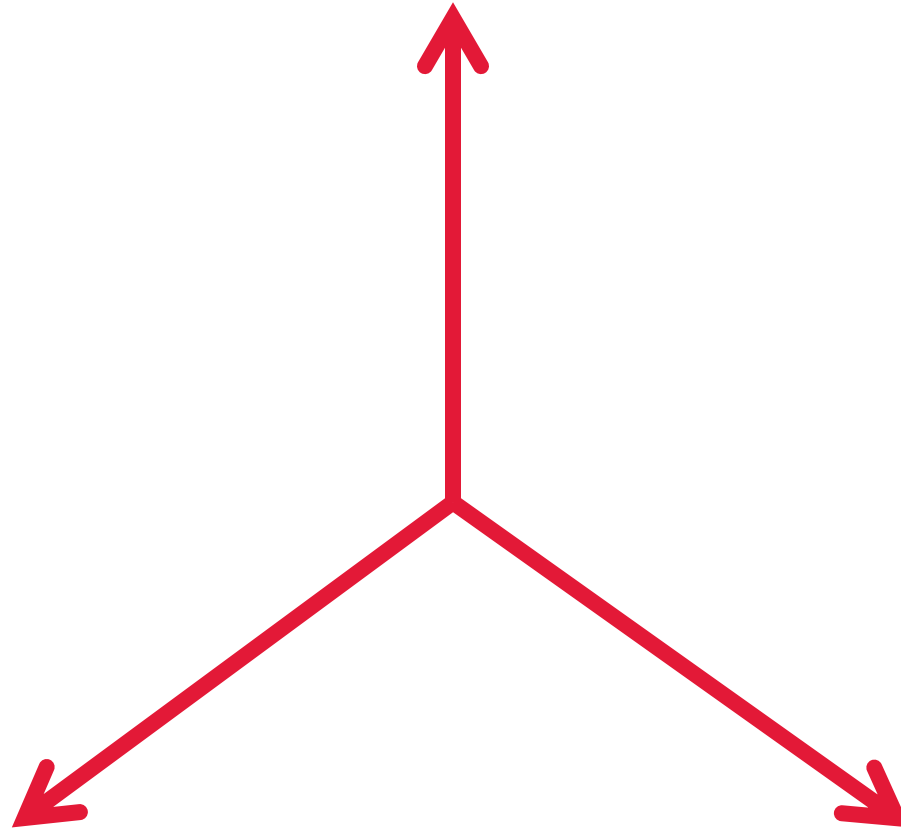
Visualisierung

“
*you can't control what
you don't measure*
”

Tom de Marco



Quantität



Qualität

Komplexität



Objektiv



Robust

Vergleichbar

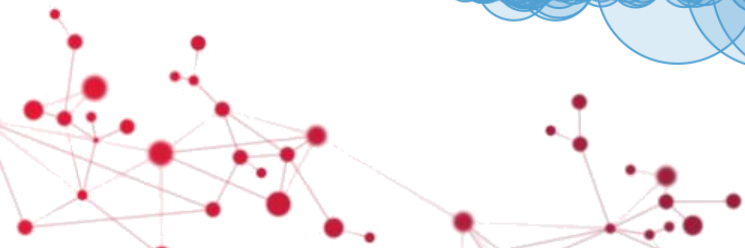
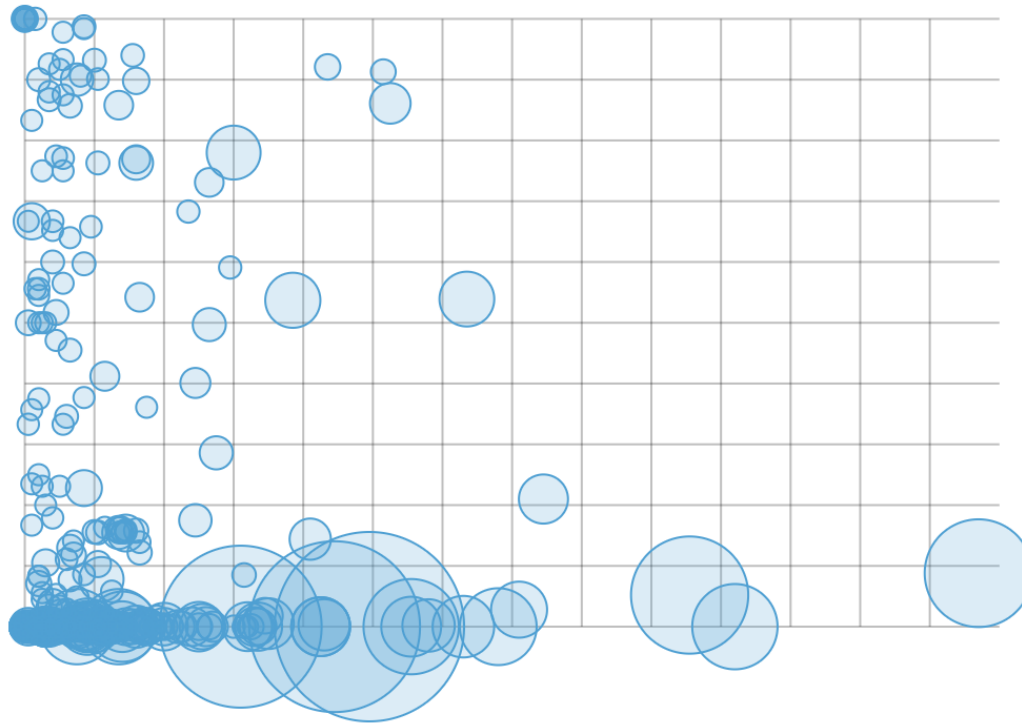




Ökonomisch



Korrelierbar



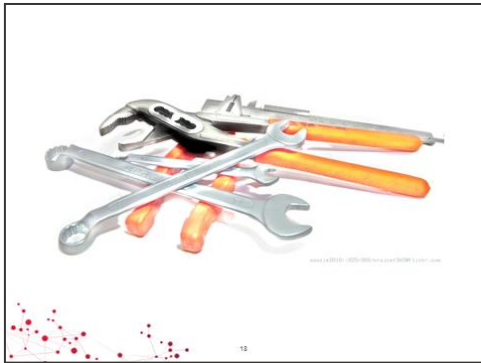
Verwertbar



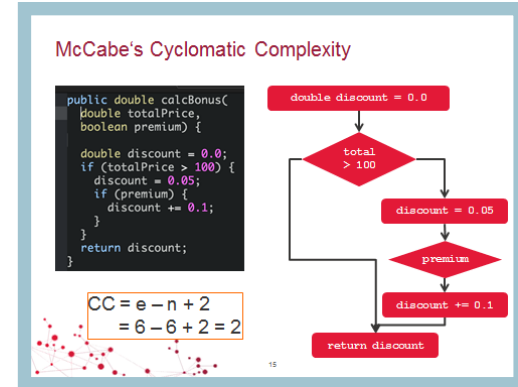
Agenda



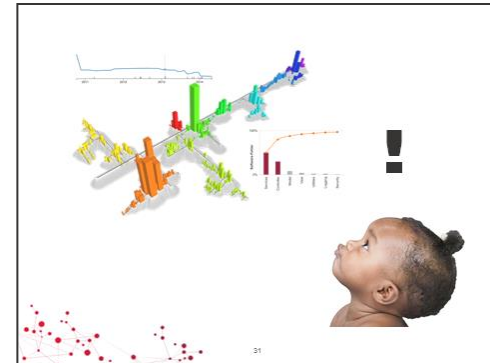
Einführung



Tool



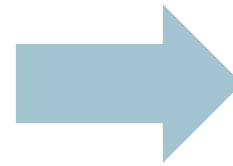
Metriken



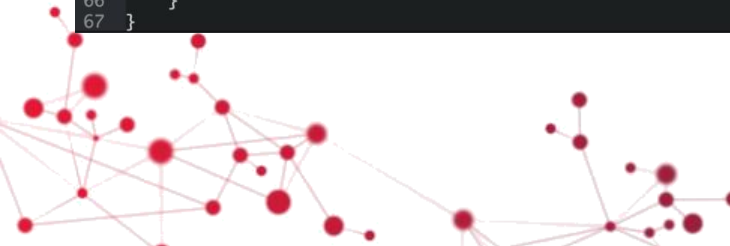
Visualisierung

Lines of Code (LOC)

```
26 * the underlying function.
27 * </p>
28 * @version $Id$
29 * @since 3.1
30 */
31 public class JacobianFunction implements MultivariateMatrixFunction {
32
33     /** Underlying vector-valued function. */
34     private final MultivariateDifferentiableVectorFunction f;
35
36     /** Simple constructor.
37      * @param f underlying vector-valued function
38      */
39     public JacobianFunction(final MultivariateDifferentiableVectorFunction f) {
40         this.f = f;
41     }
42
43     /** {@inheritDoc} */
44     public double[][] value(double[] point) {
45
46         // set up parameters
47         final DerivativeStructure[] dsX = new DerivativeStructure[point.length];
48         for (int i = 0; i < point.length; ++i) {
49             dsX[i] = new DerivativeStructure(point.length, 1, i, point[i]);
50         }
51
52         // compute the derivatives
53         final DerivativeStructure[] dsY = f.value(dsX);
54
55         // extract the Jacobian
56         final double[][] y = new double[dsY.length][point.length];
57         final int[] orders = new int[point.length];
58         for (int i = 0; i < dsY.length; ++i) {
59             for (int j = 0; j < point.length; ++j) {
60                 orders[j] = 1;
61                 y[i][j] = dsY[i].getPartialDerivative(orders);
62                 orders[j] = 0;
63             }
64         }
65         return y;
66     }
67 }
```



68

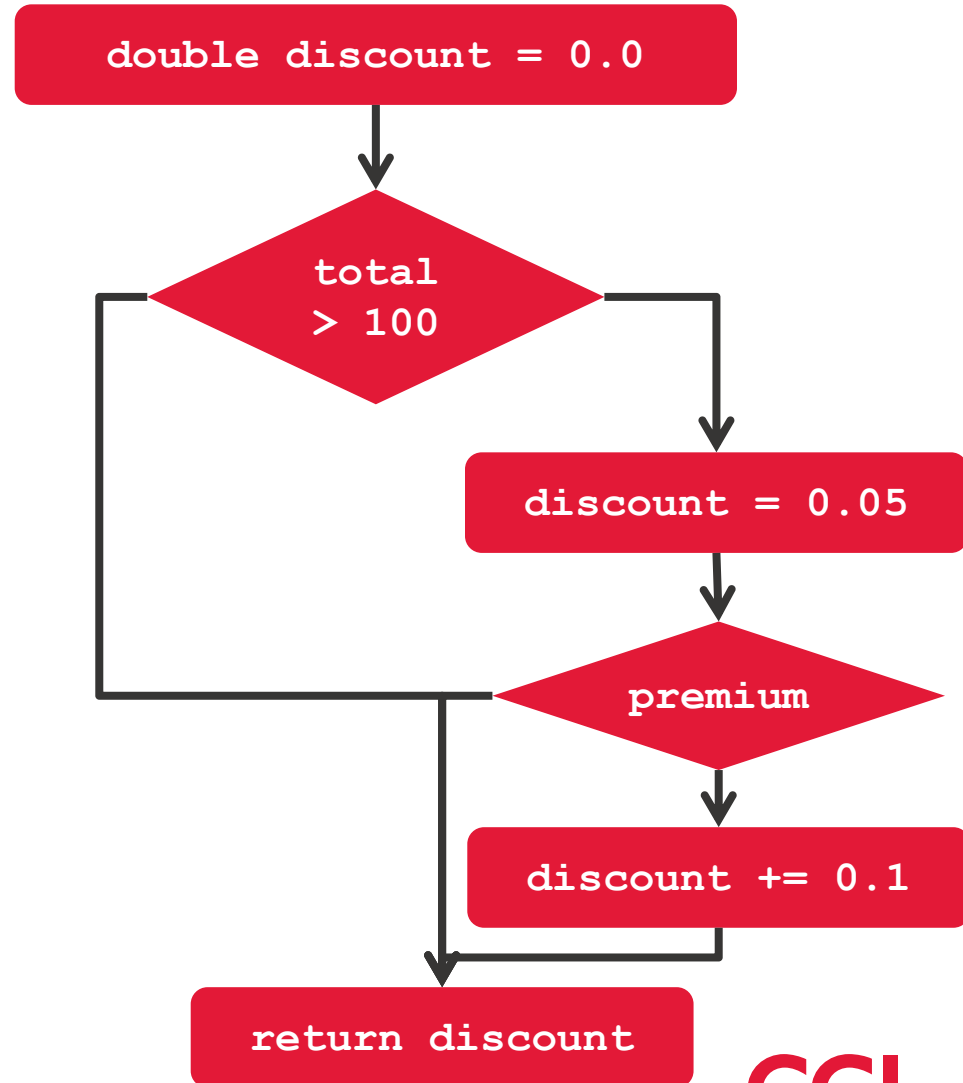


```
1 public class Robot {
2     private int x;
3     private int y;
4
5     public void move(int dx, int dy) {
6         x += dx;
7         y += dy;
8     }
9 }
```

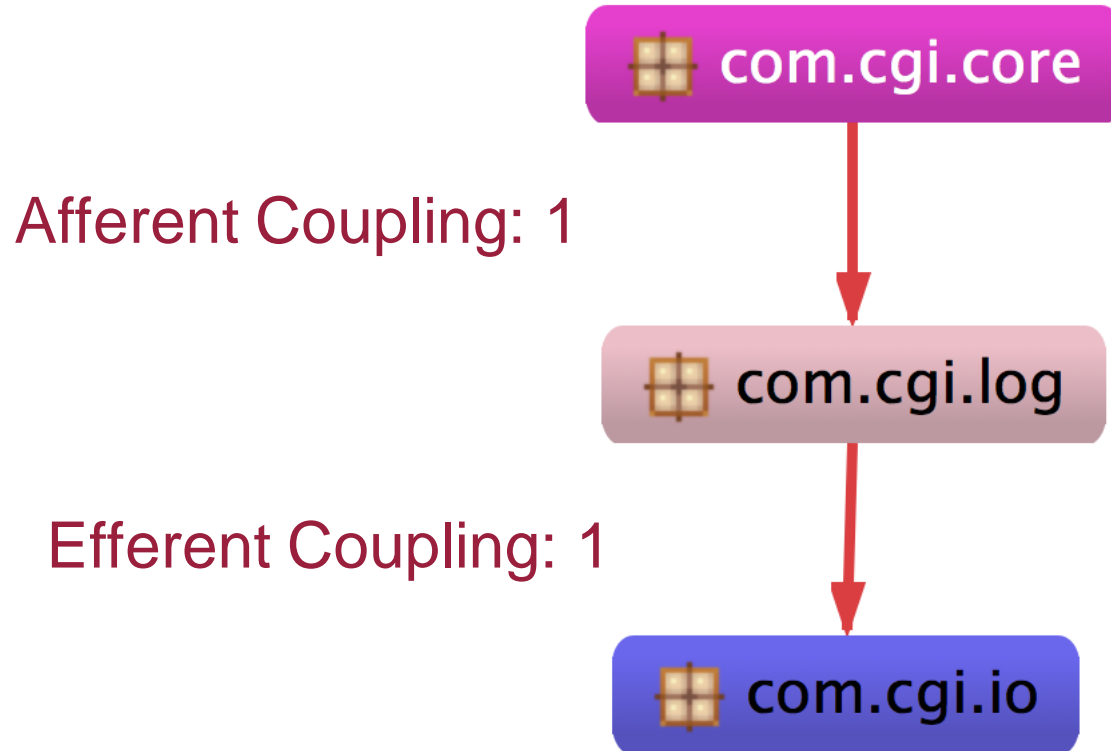
```
1 public class Robot
2 {
3     private int x;
4
5     private int y;
6
7     public void move(int dx, int dy)
8     {
9         x += dx;
10        y += dy;
11    }
12 }
```


McCabe's Cyclomatic Complexity

```
public double calcBonus(  
    double totalPrice,  
    boolean premium) {  
  
    double discount = 0.0;  
    if (totalPrice > 100) {  
        discount = 0.05;  
        if (premium) {  
            discount += 0.1;  
        }  
    }  
    return discount;  
}
```



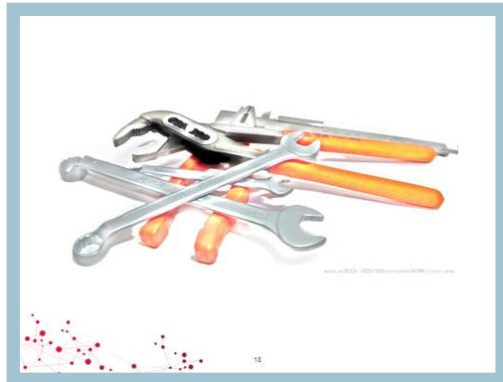
Efferent / Afferent Coupling



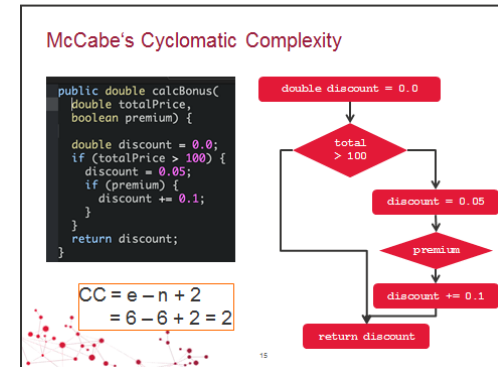
Agenda



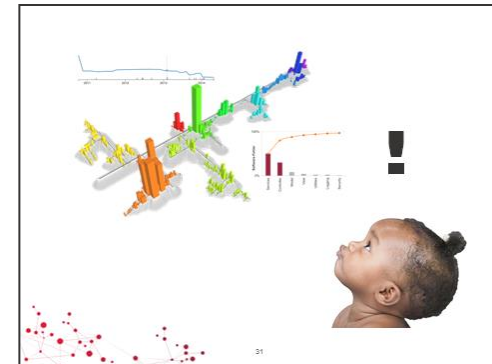
Einführung



Tool



Metriken



Visualisierung

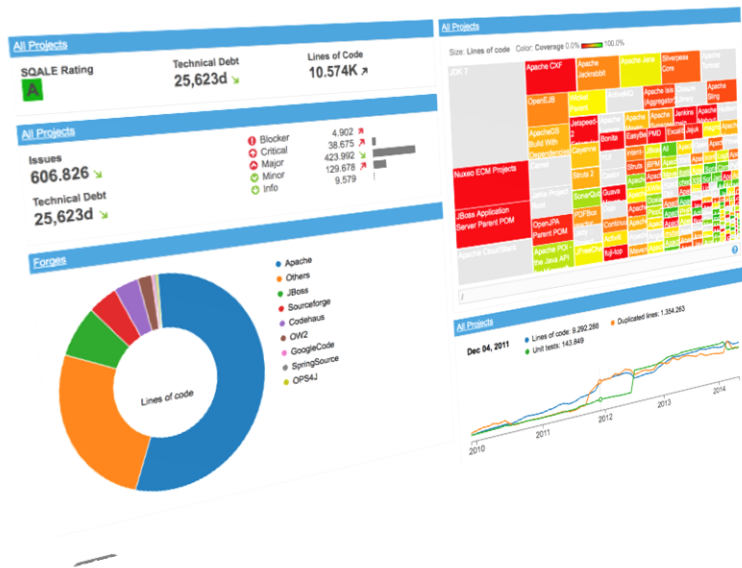
Demo Sonarqube



woozie2010::025/365/projct365@flickr.com



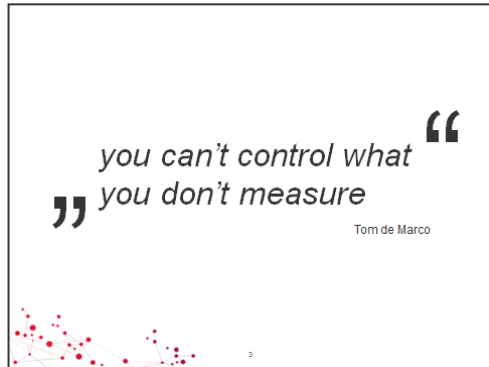
Sonarqube



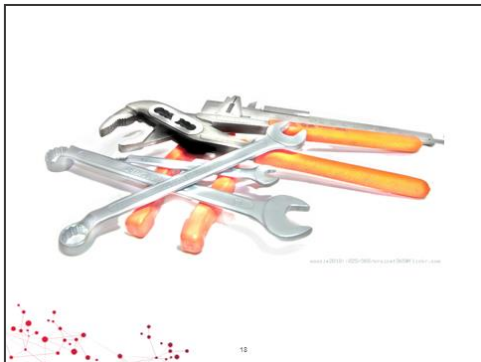
- Website: <http://www.sonarqube.org>
- Integration in
 - Ant
 - Maven
 - Jenkins / Hudson



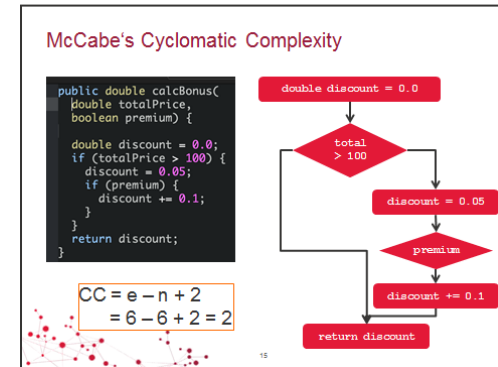
Agenda



Einführung



Tool



Metriken



Visualisierung

Coverage Duplications Issues LCOM4 Metrics Source

Complexity
 Complexity: 122
 Complexity /class: 122.0
 Complexity /file: 122.0
 Complexity /method: 2.4
 Complexity in functions: 122

Design
 Affherent couplings: 2
 Depth in Tree: 1

Design Classes and Methods Complexity: 50.0%
 Design Coupling Between Objects: 0.0%
 Design Depth of Inheritance Tree: 100.0%
 Design Lack of Cohesion of Methods: 100.0%
 Design Response for Class: 0.0%
 Efferent couplings: 36
 LCOM4: 1.0
 Number of Children: 0
 Response for Class: 114

Documentation
 Comment lines: 535
 Comments (%): 62.1%
 comment_lines_data: 1=1;2=0;3=0;4=0;5=0;6=0;7=

Cyclomatic complexity ?
 Complexity average by class -
 Complexity average by file -
 Complexity average by method -
 Cyclomatic complexity in methods -

Afferent couplings
 Depth in Inheritance Tree
 NOM from classes with function_complexity gt value (2.5 default) and class_complexity gt value (12 default)
 CBO from classes with ce.gt.value (5 default)
 DIT from classes with dit.gt.value (5 default)
 LCOM4 from classes with lcom4.gt.vale (50 default)
 RFC from classes with rfc.gt.vale (50 default)
 Efferent couplings
 Lack of Cohesion of Methods
 Number of Children
 Response for Class

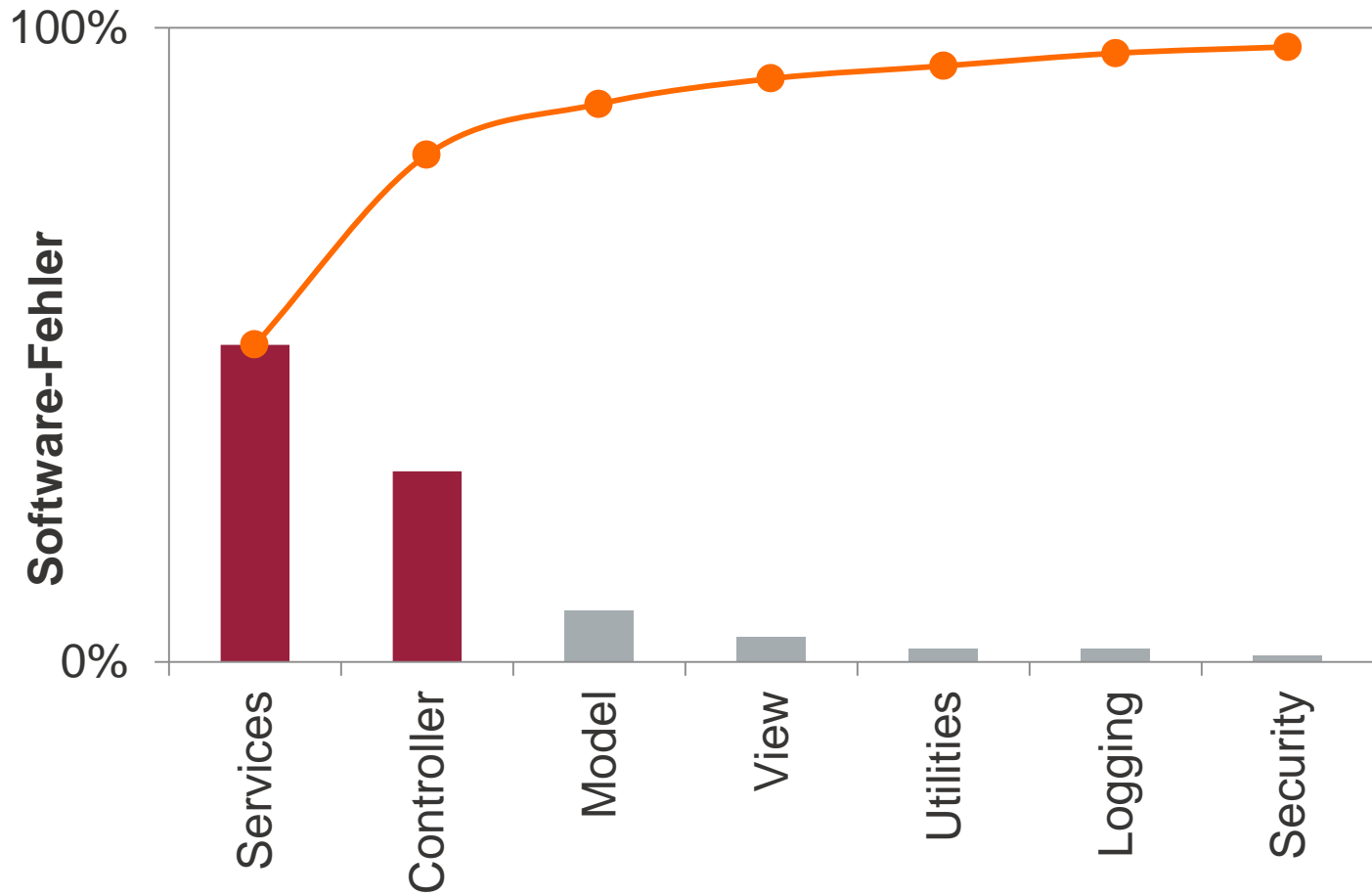
Number of comment lines
 Comments balanced by ncloc + comment lines



Wo anfangen?

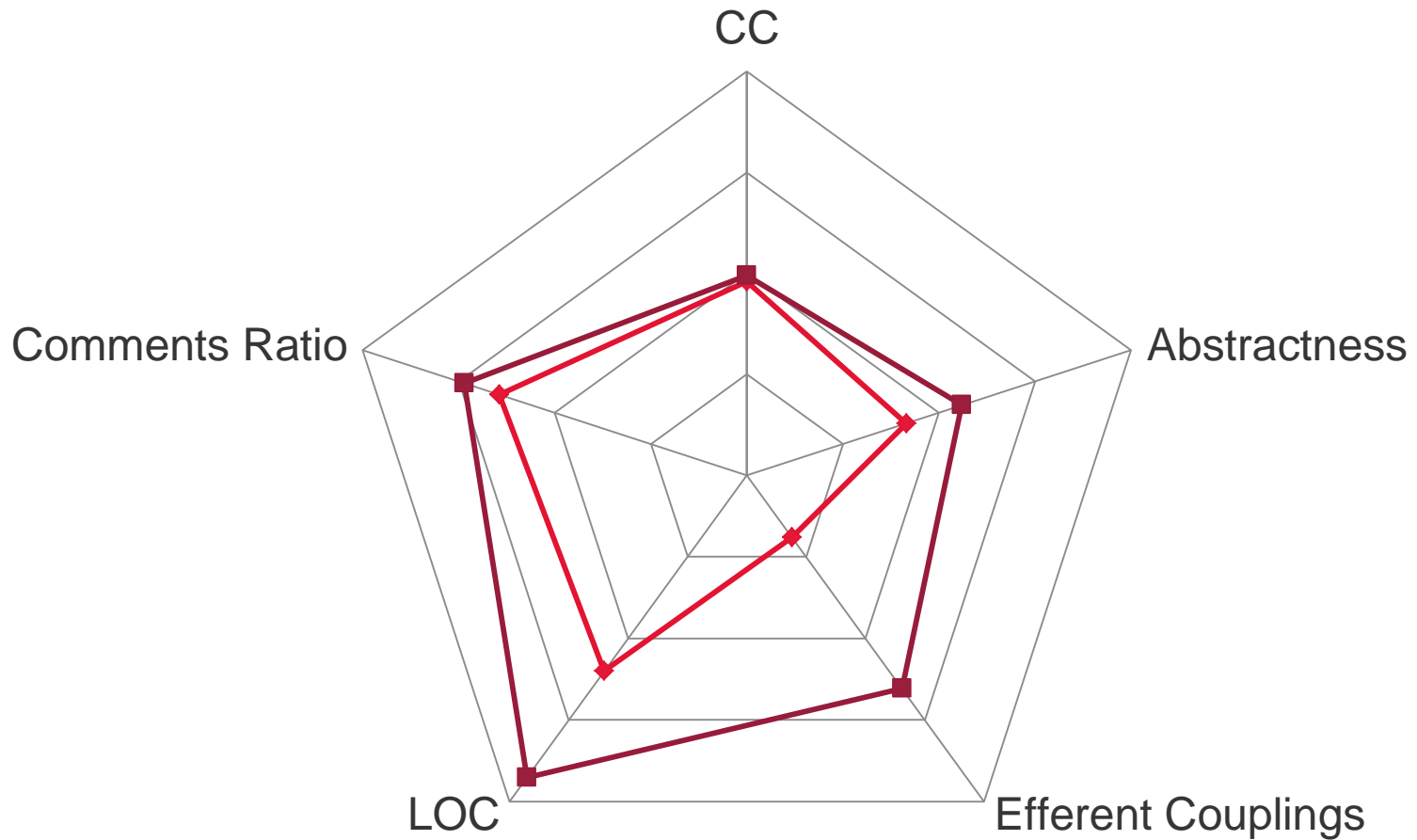


Pareto Diagramm





Kiviati Diagramm

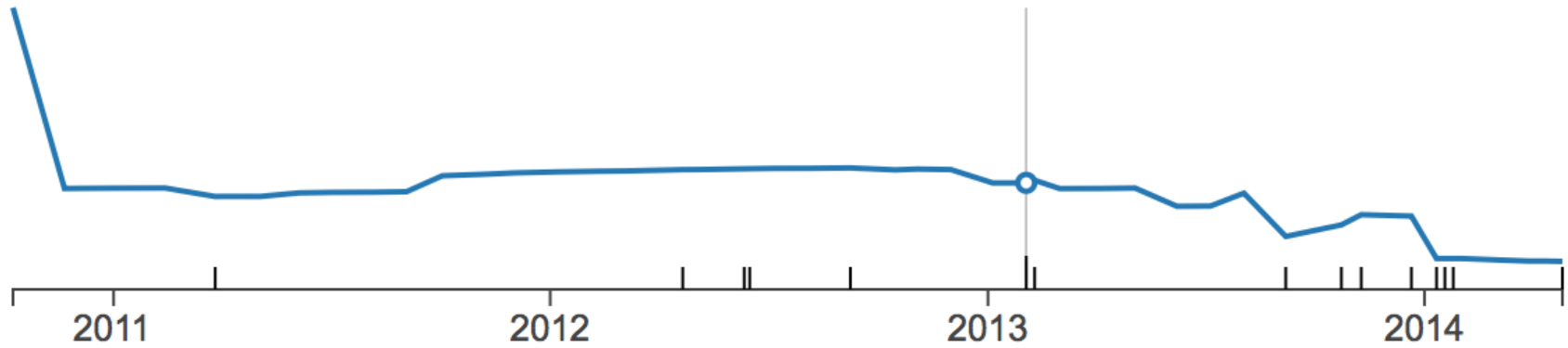


Sind wir auf dem richtigen Weg?







Timeline

Feb 02, 2013 ● Technical Debt: 1,197d
5.8-SNAPSHOT



Sparklines

18. Jan 2014
2.11.2-SNAPSHOT

Duplicated lines (%)	10,6%	
Duplicated lines	48.890	
Duplicated blocks	3.384	
Duplicated files	366	

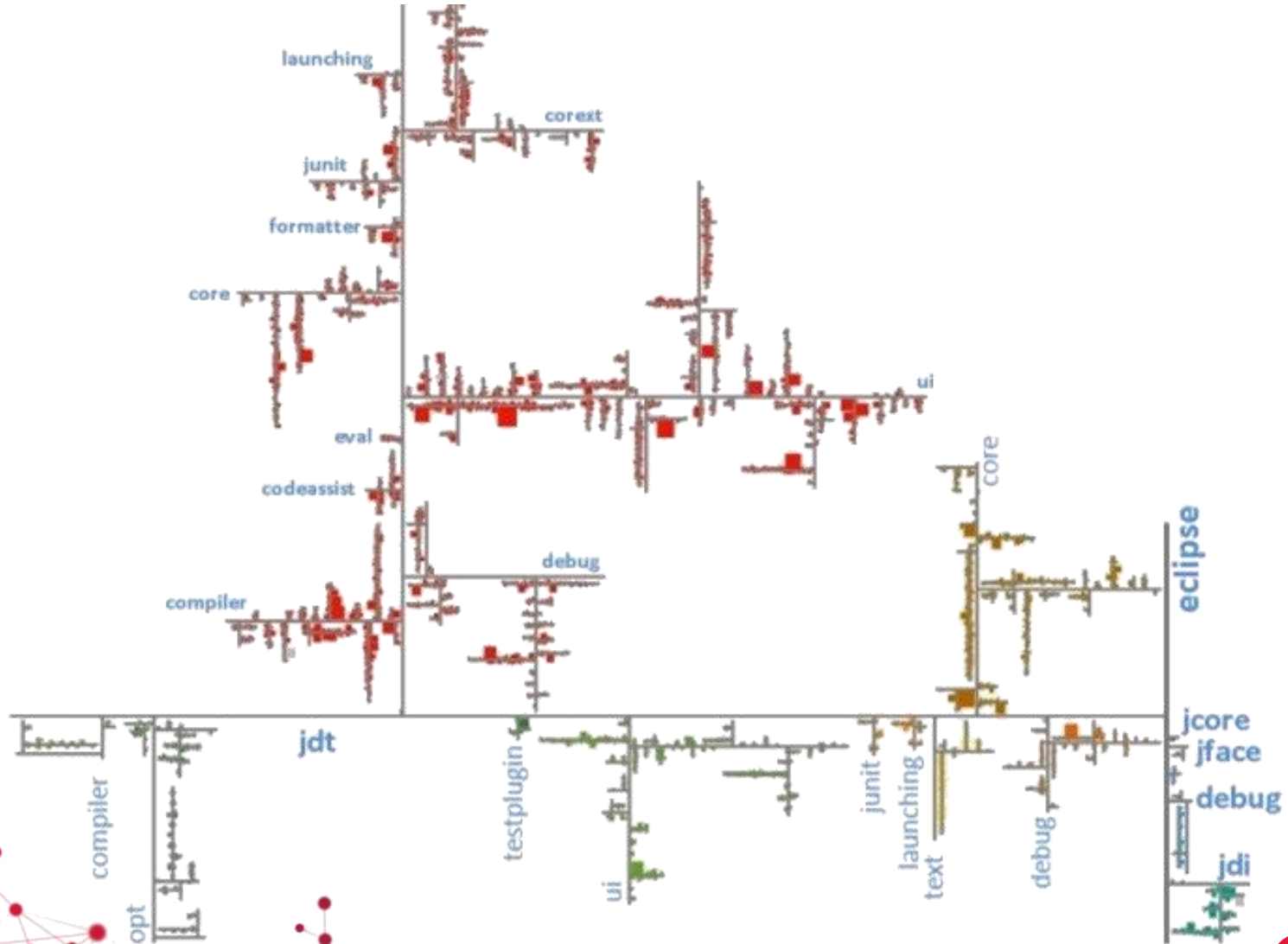


Vogelperspektive

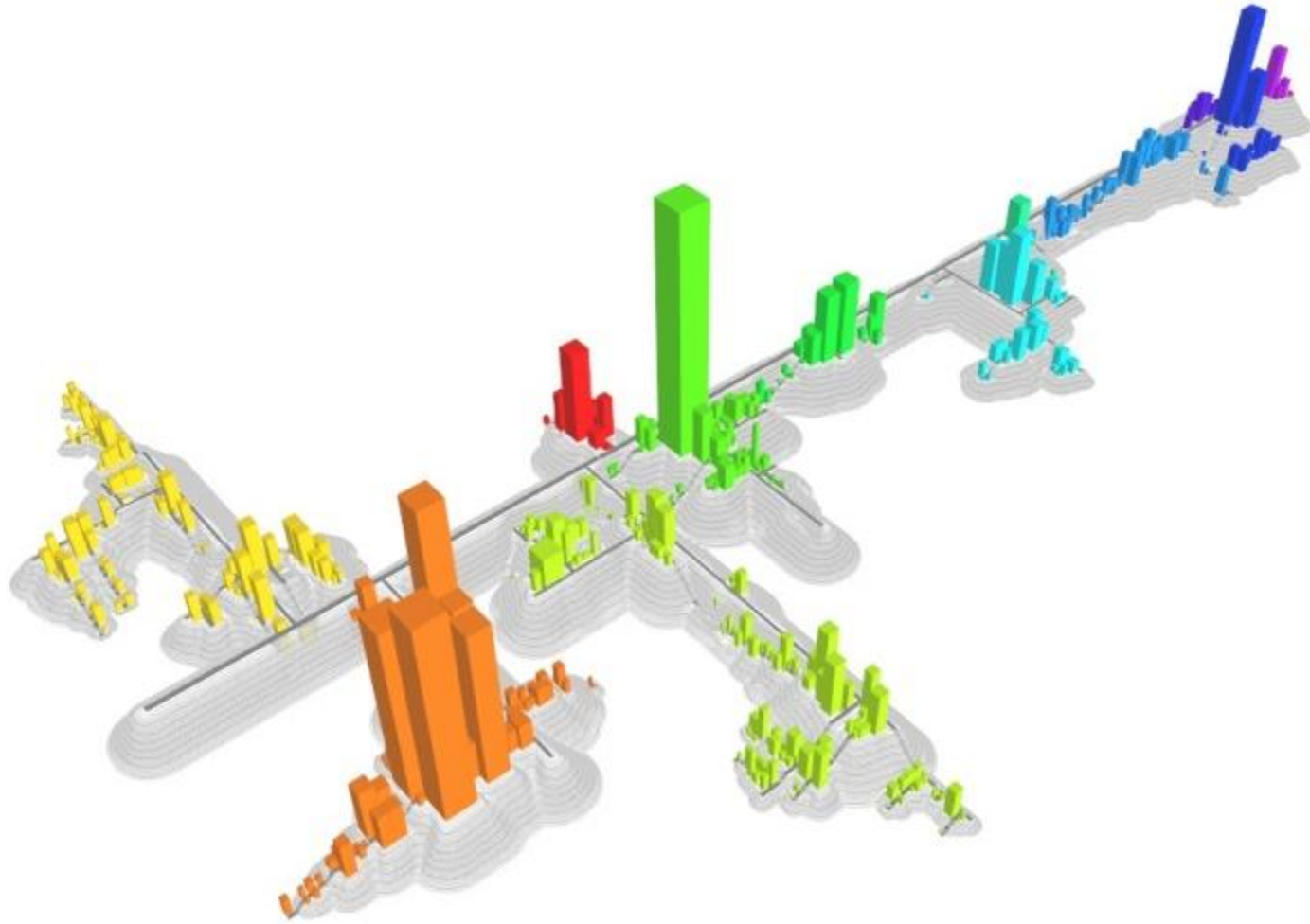


Quelle: Flickr (AcrylicArtist)

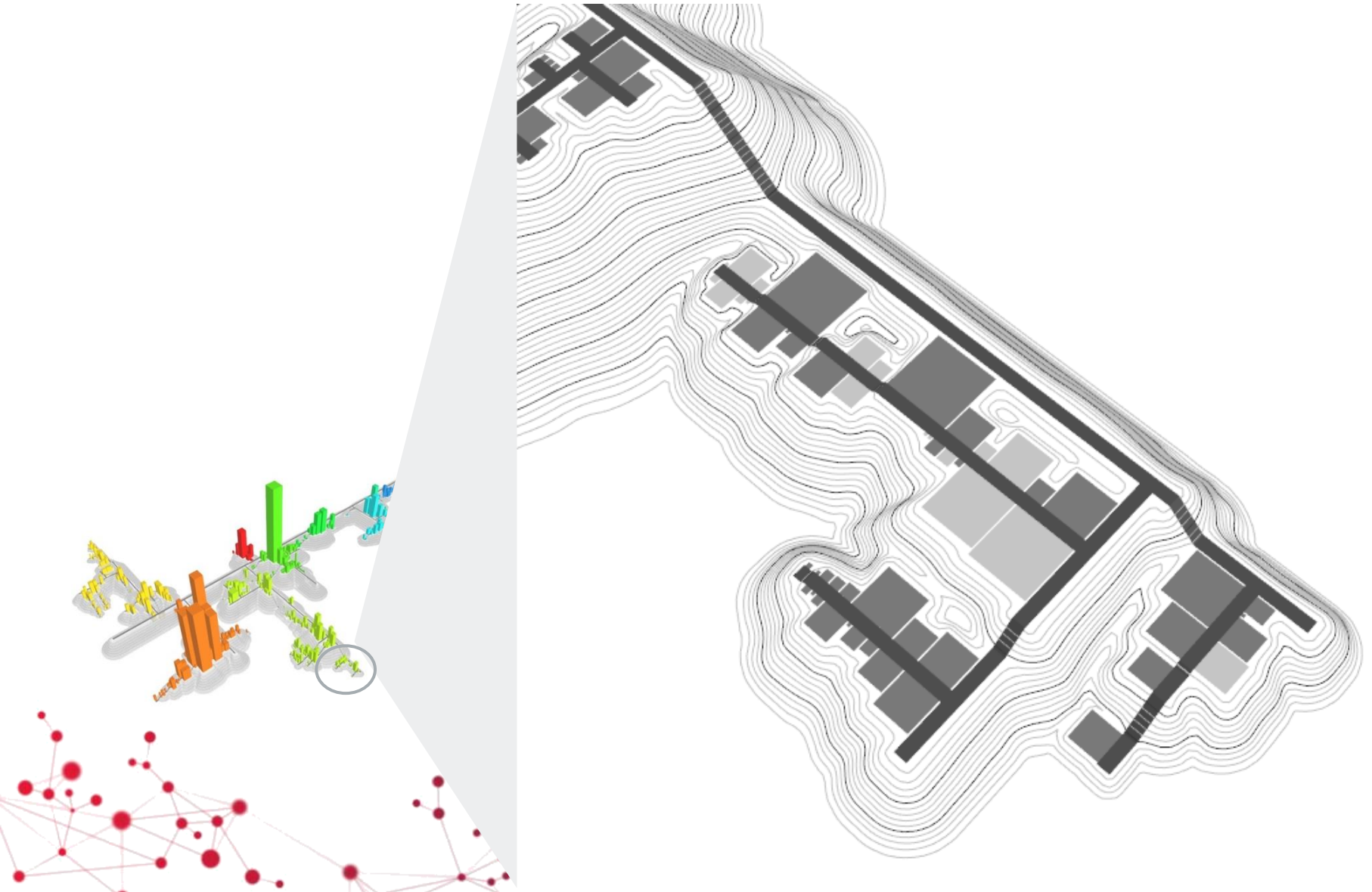
City Map

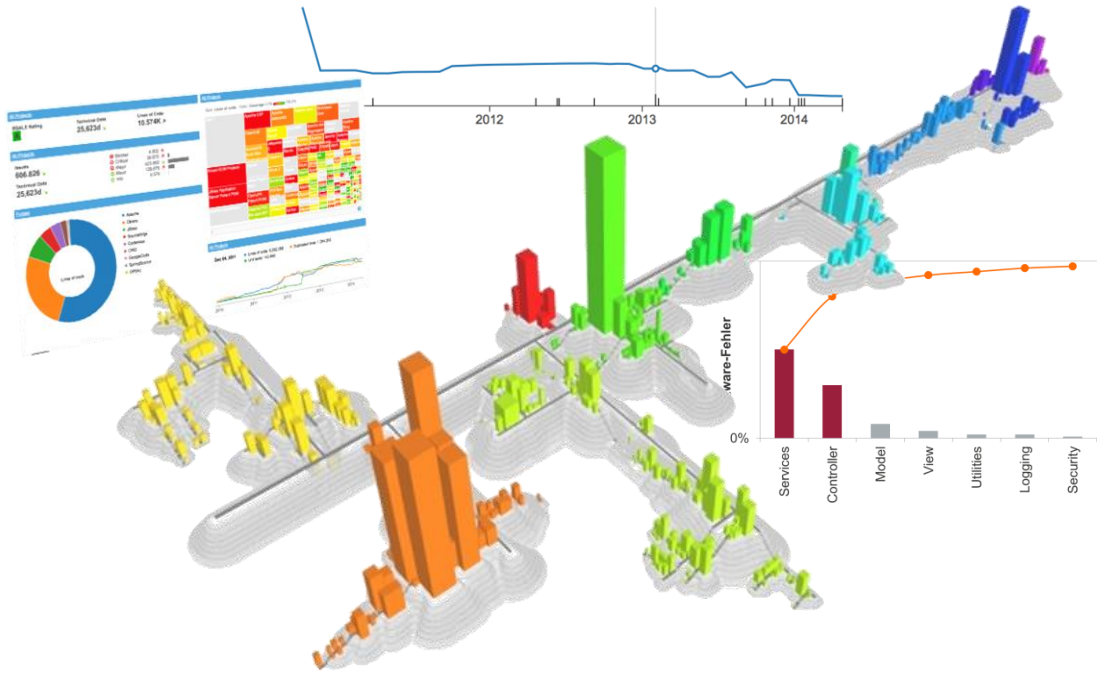


3D City



3D City - Detail





Fragen?

Tobias Getrost

tobias.getrost@cgi.com



CGI

Experience the commitment®



ENTWICKLERTAG

meet the **SPEAKER** **@speakerlounge**



1. OG DIREKT ÜBER DEM EMPFANG

Contact



Tobias Getrost

Position Consultant

Straße Ludwig-Ehrhardt-Allee 20
PLZ Ort 76131 Karlsruhe

E-Mail tobias.getrost@cgi.com

