

Copy & Paste & Bug?

Die überraschende Wahrheit über Klone in Software-Artefakten

Über Mich

Forschung

- Clone Detection
- Architekturanalyse

Beratung

- Mitgründer
- Qualitäts-Bewertung & Qualitäts-Controlling

Entwicklung

- Continuous Quality Assessment Toolkit ConQAT
- >400 kLOC, Apache Lizenz, >25.000 Downloads
- Kommerzielle Erweiterung: Teamscale



```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

A man with a beard and a striped jacket, smiling and holding a small object. He is wearing a lanyard with a badge around his neck. The background is a plain wall.

#1 Code Smell (1999)

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

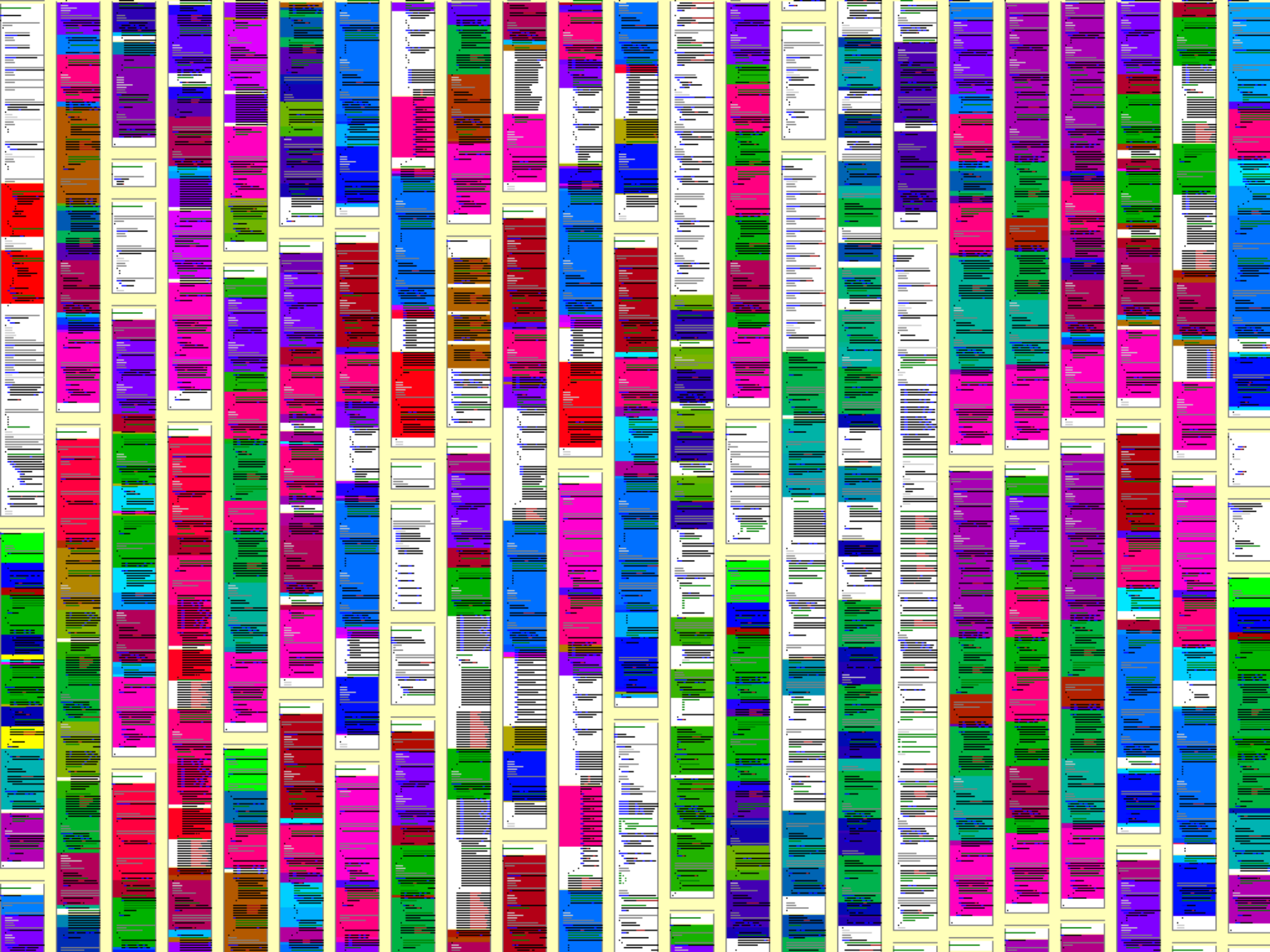
```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```









Studie

Munich Re

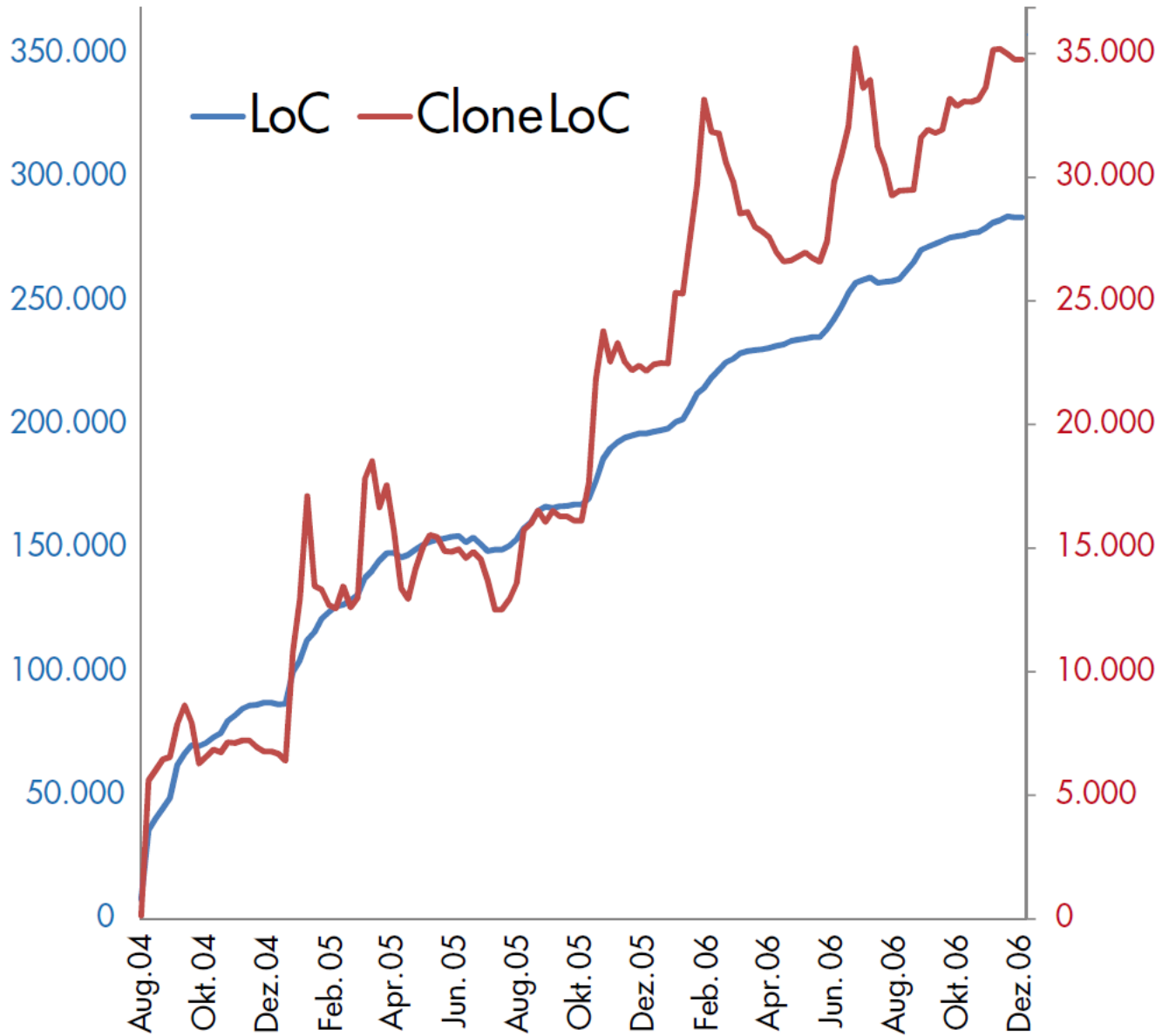


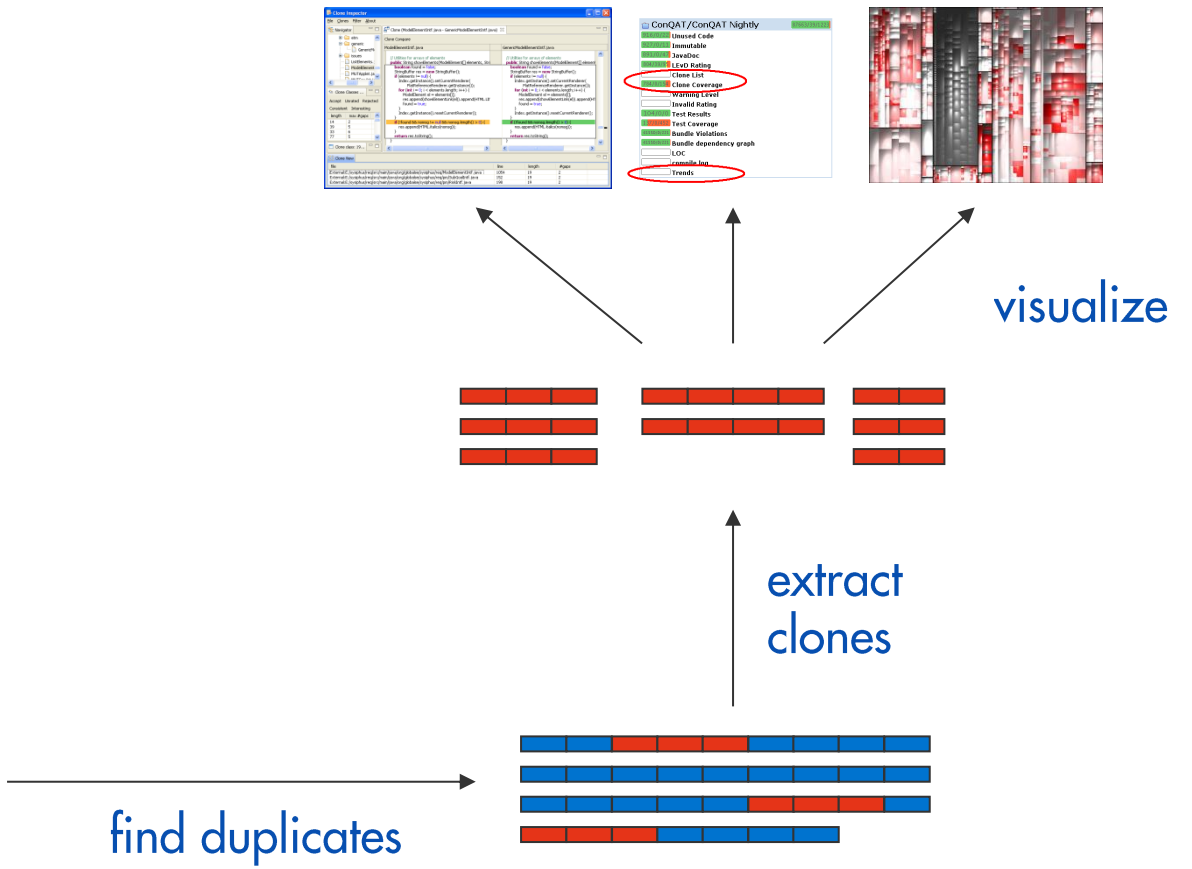
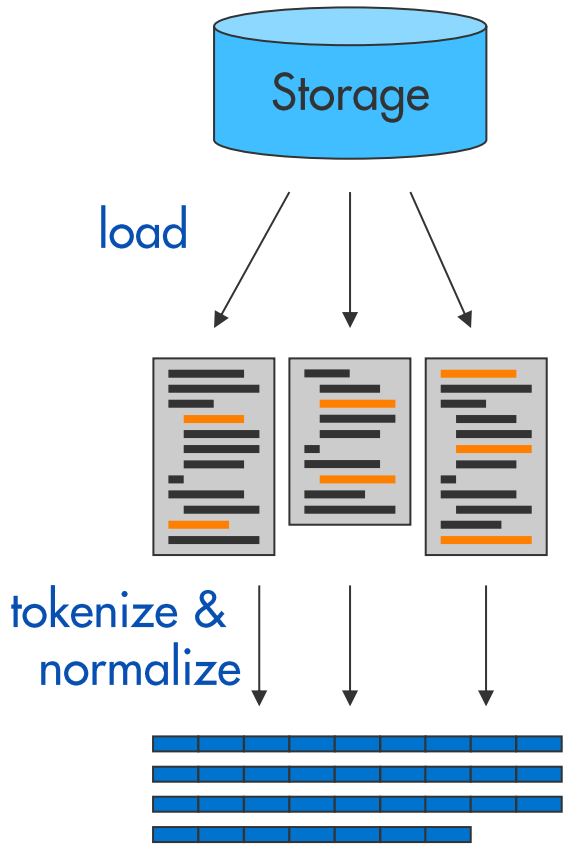
- Über 100 Fehler in produktiver Software



- 52% aller ungewollten Unterschiede fehlerhaft

Juergens, Deissenboeck et al: *Do Code Clones Matter?* ICSE 2009





Normalisierung

```
String readFileUtf8(File file) {  
    FileInputStream in = new FileInputStream(file);  
    byte[] buffer = new byte[file.length()];  
    in.read(buffer); in.close();  
    return new String(buffer, „UTF-8“);  
}
```

```
String readFileUtf16(File file) {  
    FileInputStream in = new FileInputStream(file);  
    byte[] buffer = new byte[file.length()];  
    in.read(buffer); in.close();  
    return new String(buffer, „UTF-16“);  
}
```

```
id0 id1(id2 id3) {  
    id0 id2 = new id0(id4);  
    id0[] id1 = new id0[id2.id3()];  
    id0.id1(id2); id0.id3();  
    return new id0(id1, lit0);  
}
```

```
id0 id1(id2 id3) {  
    id0 id2 = new id0(id4);  
    id0[] id1 = new id0[id2.id3()];  
    id0.id1(id2); id0.id3();  
    return new id0(id1, lit0);  
}
```

Normalisierung

```
String readFileUtf8(File file) {  
    FileInputStream in = new FileInputStream(file);  
    byte[] buffer = new byte[file.length()];  
    in.read(buffer); in.close();  
    return new String(buffer, „UTF-8“);  
}
```

```
String readFileUtf16(File file) {  
    FileInputStream in = new FileInputStream(file);  
    byte[] buffer = new byte[file.length()];  
    in.read(buffer); in.close();  
    return new String(buffer, „UTF-16“);  
}
```

```
id0 id1(id2 id3) {  
    id0 id2 = new id0(id4);  
    id0[] id1 = new id0[id2.id3()];  
    id0.id1(id2); id0.id3();  
    return new id0(id1, lit0);  
}
```

```
id0 id1(id2 id3) {  
    id0 id2 = new id0(id4);  
    id0[] id1 = new id0[id2.id3()];  
    id0.id1(id2); id0.id3();  
    return new id0(id1, lit0);  
}
```

```
JabRef/java/net/sf/jabref/Util.java
if (fileParts.length > 1) {
    for (int i = 0; i < fileParts.length - 1; i++) {
        String dirToProcess = fileParts[i];
    }
}

JabRef/java/net/sf/jabref/external/RegExpFileSearch.java
if (fileParts.length > 1) {
    for (int i = 0; i < fileParts.length - 1; i++) {
        String dirToProcess = fileParts[i];
        findBrackets(dirToProcess, entry,
            ("^.:$")) { // Windows Drive Le
                (dirToProcess + "/");
            }
        ".") { // Stay in current dire
            }
        "..") {
            (directory
        }
    }
}

```

Clone Coverage Trend



```
fail();
} catch (AssertionFail
}
}

Focus on CloneClass: 12
Select this clone instance: DoublePrecisionAssertTest.java(13-30)

```

ABAP, ADA, C, C#, C++, Cobol, Java, JavaScript, Matlab, PHP, PL/I, PL/SQL, Python, T-SQL, VB, VB.NET, XML, ...



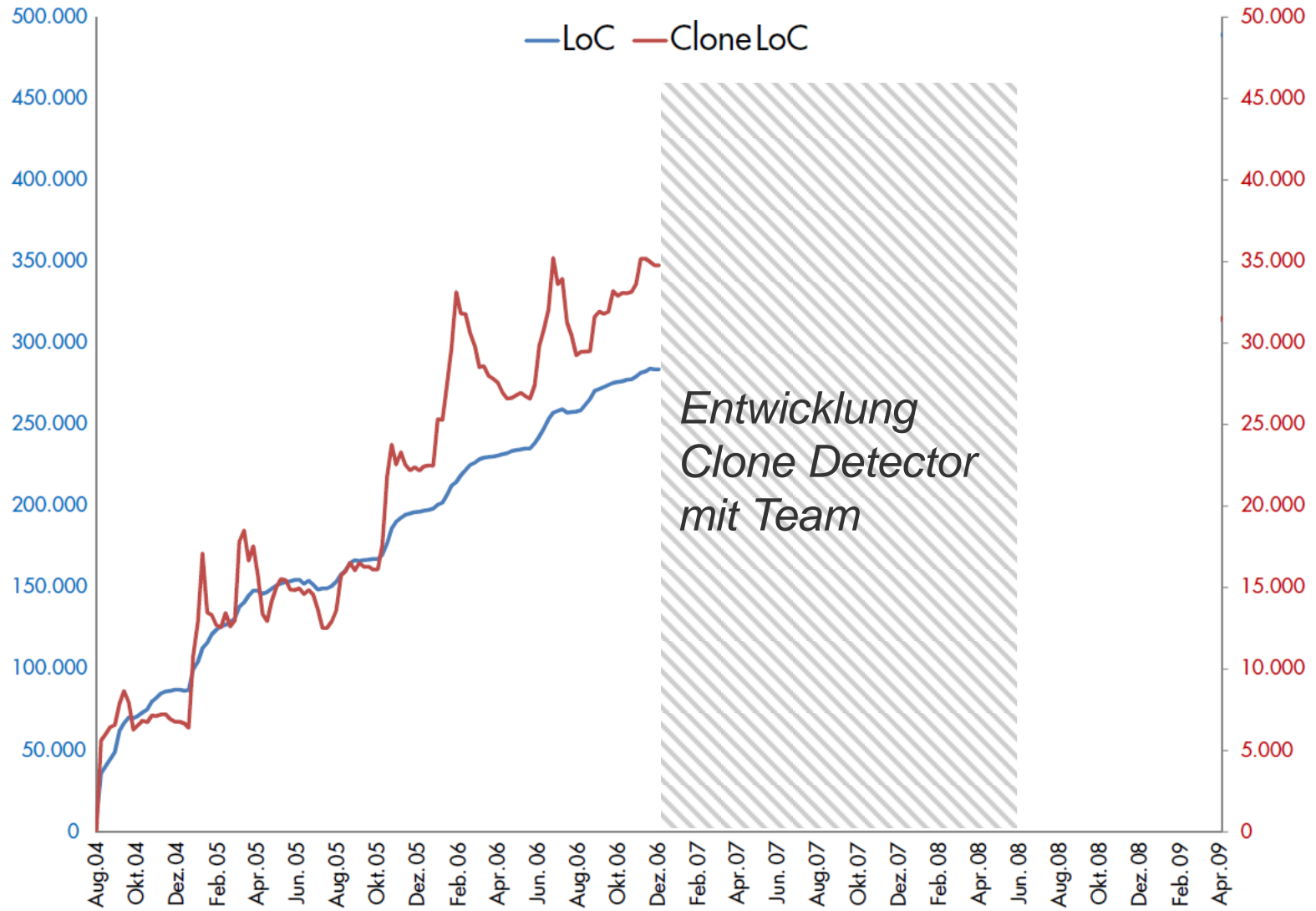
Download

ConQAT Complete

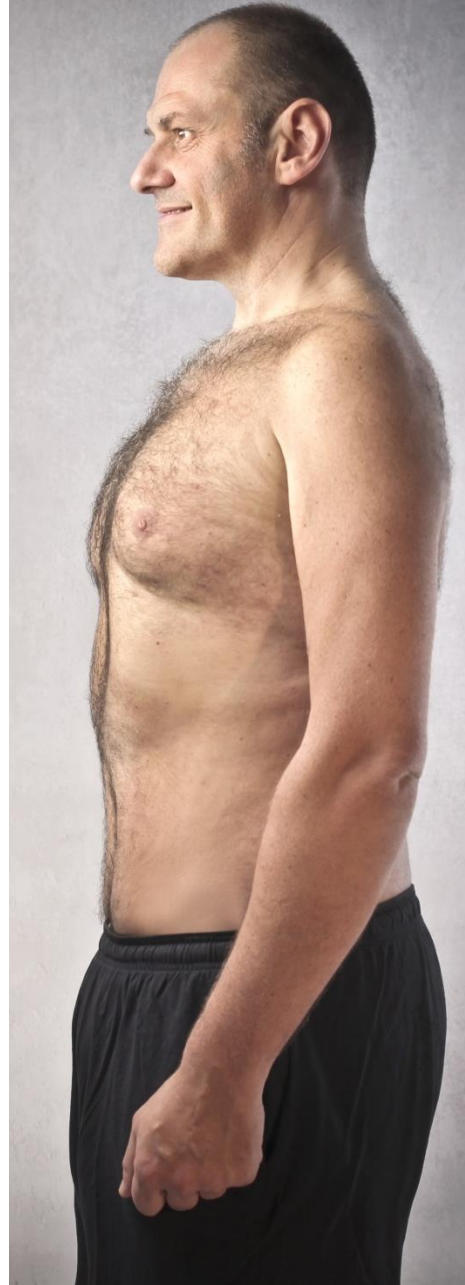
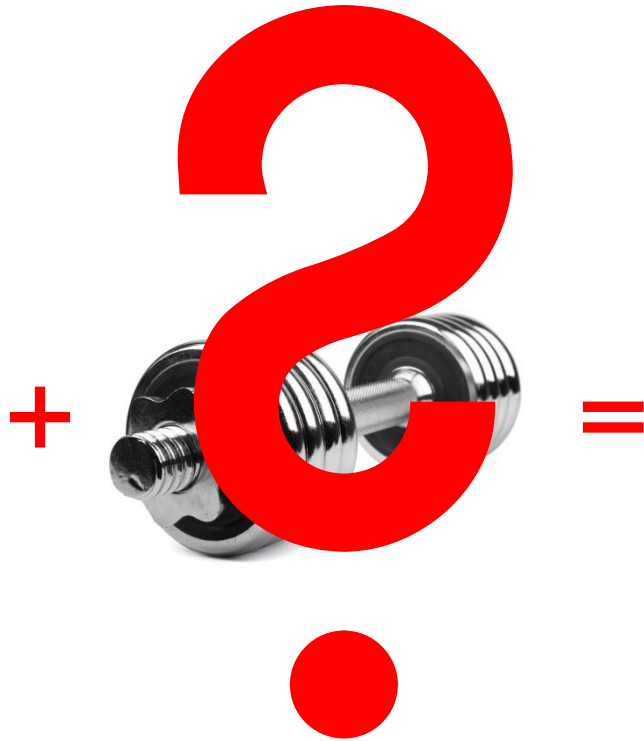
ConQAT Complete is a complete Eclipse IDE bundles with the ConQAT plug-ins and analysis engine. It enables you to build ConQAT blocks and processors. If you use one of the operating systems listed below, this is probably the right package for you.

- [Download ConQAT 2013.10 \(Windows 32-bit\)](#)
- [Download ConQAT 2013.10 \(Windows 64-bit\)](#)
- [Download ConQAT 2013.10 \(Mac OS X Cocoa, 32-bit\)](#)
- [Download ConQAT 2013.10 \(Mac OS X Cocoa, x86/64-bit\)](#)
- [Download ConQAT 2013.10 \(Linux GTK, x86/32-bit\)](#)
- [Download ConQAT 2013.10 \(Linux GTK, x86/64-bit\)](#)

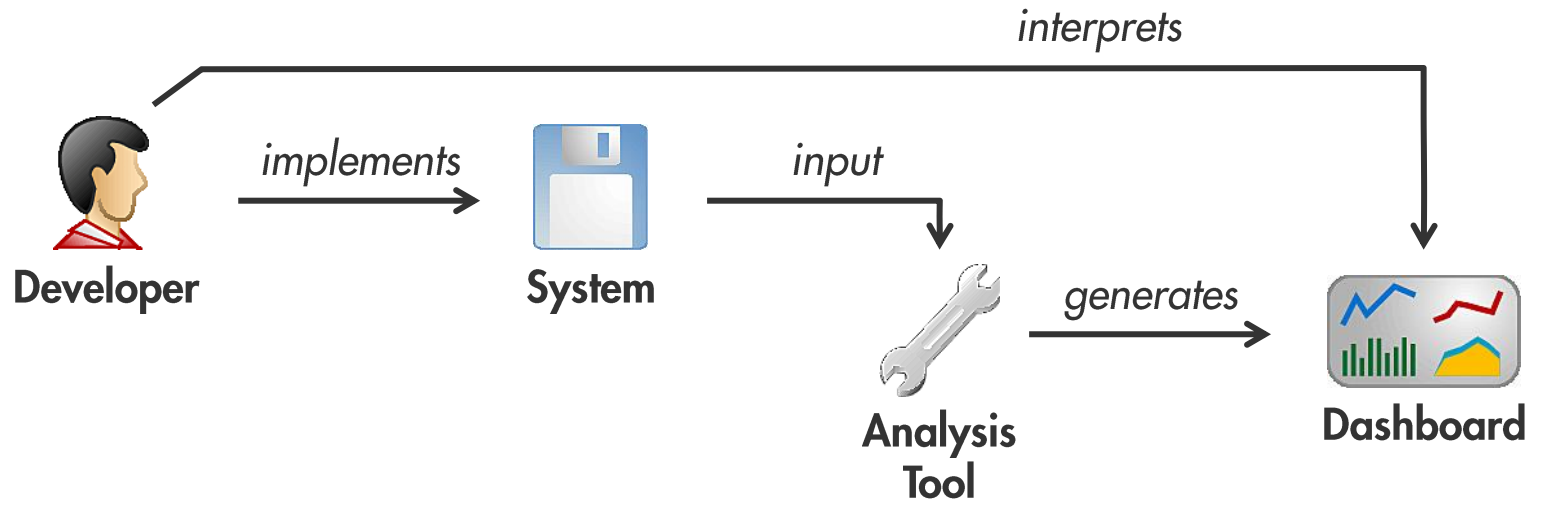
ConQAT Engine



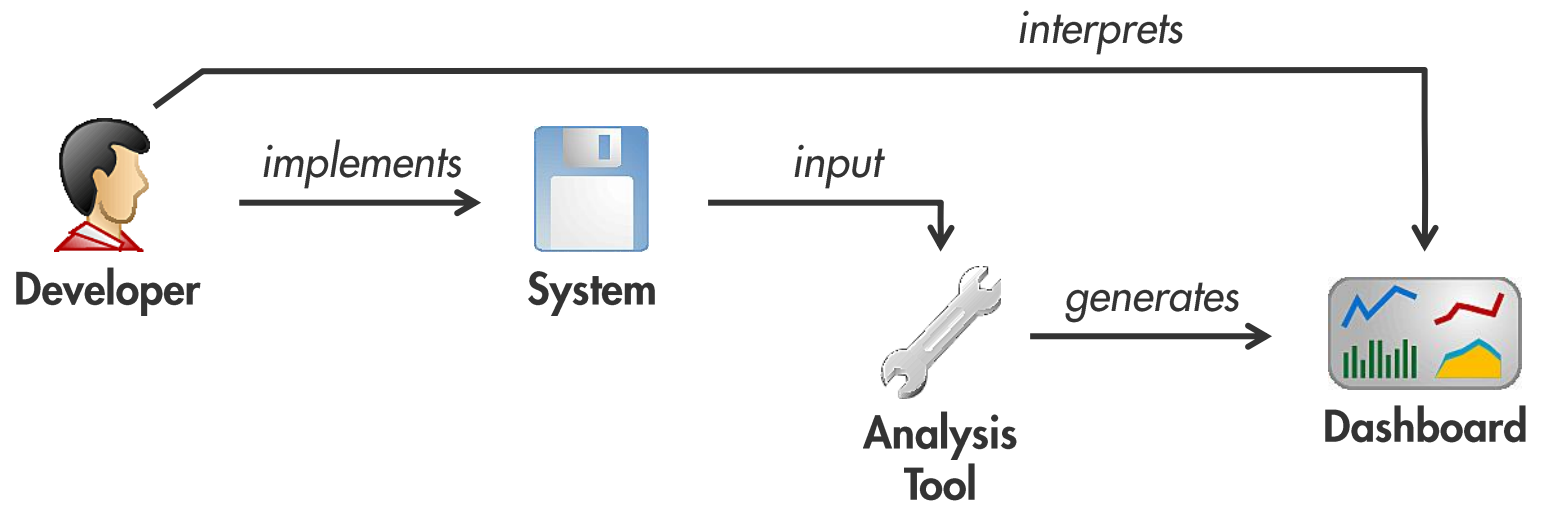








Use a Code Analysis Tool



- False positives
 - Information overflow
- => Analyseergebnisse werden ignoriert









Best Practice: Diskriminierung (von Code)

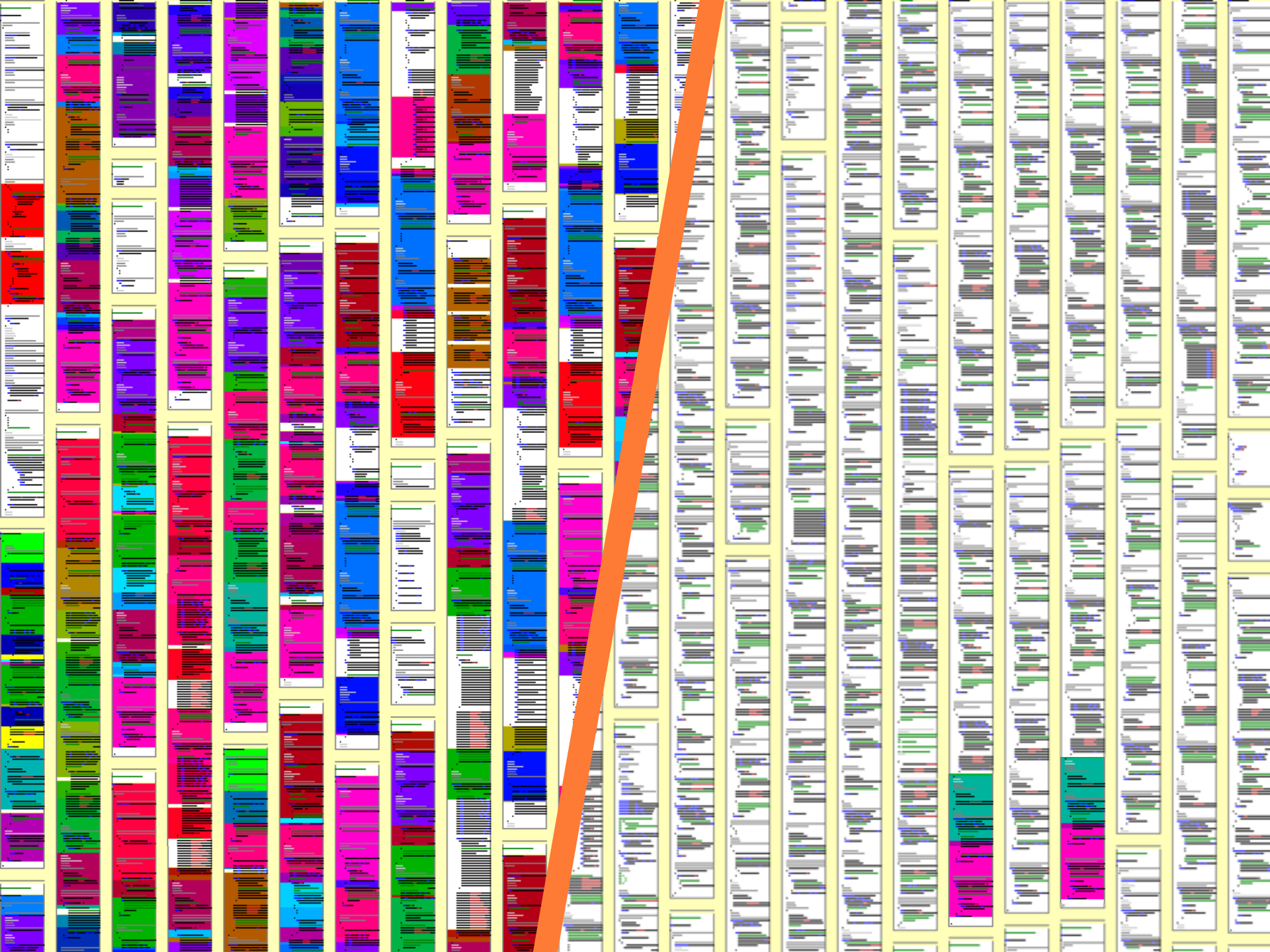
Art der Wartung

- Manuell
- Generator
- Überhaupt nicht (Wegwerf-Prototyp)

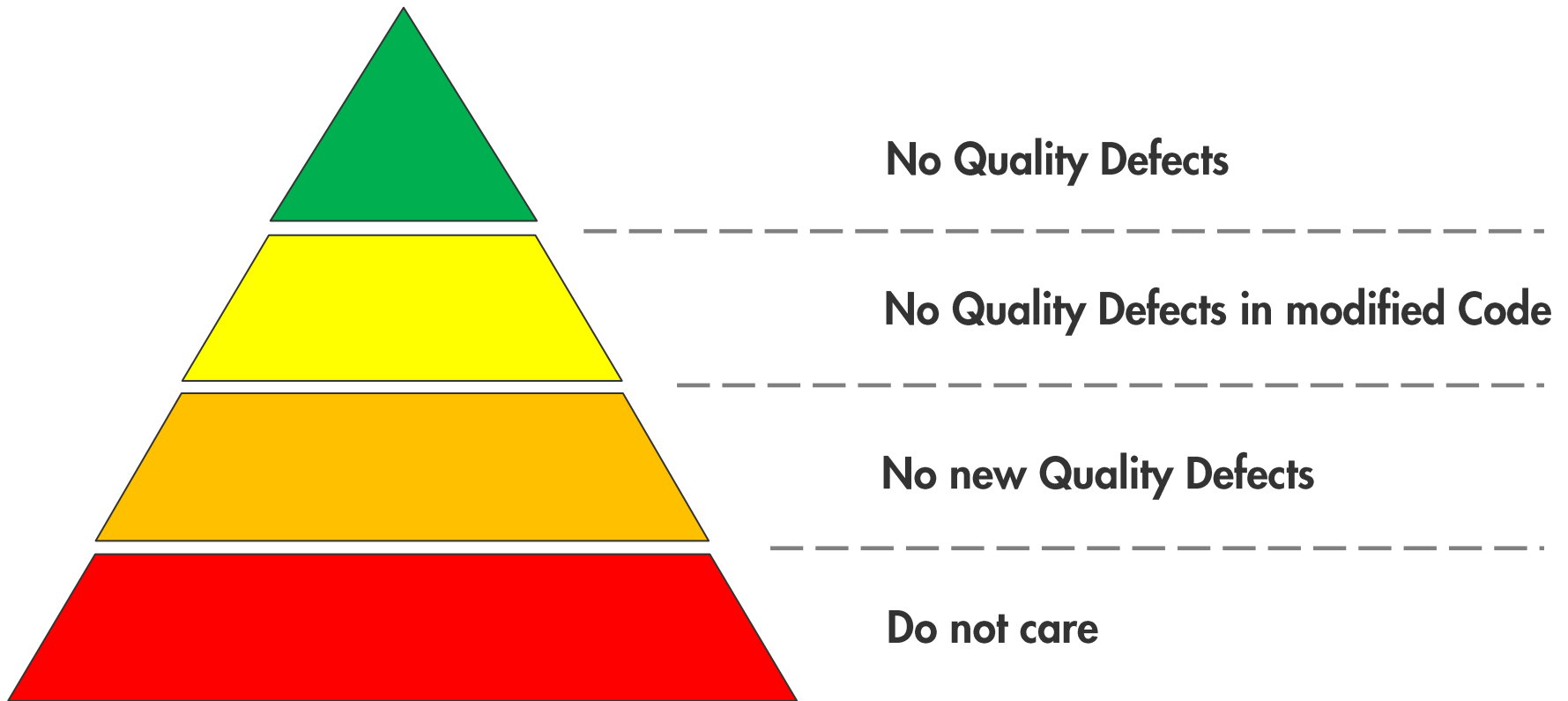
Aufgabe im Projekt

- Teil der Anwendung
- Test
- Hilfswerkzeug

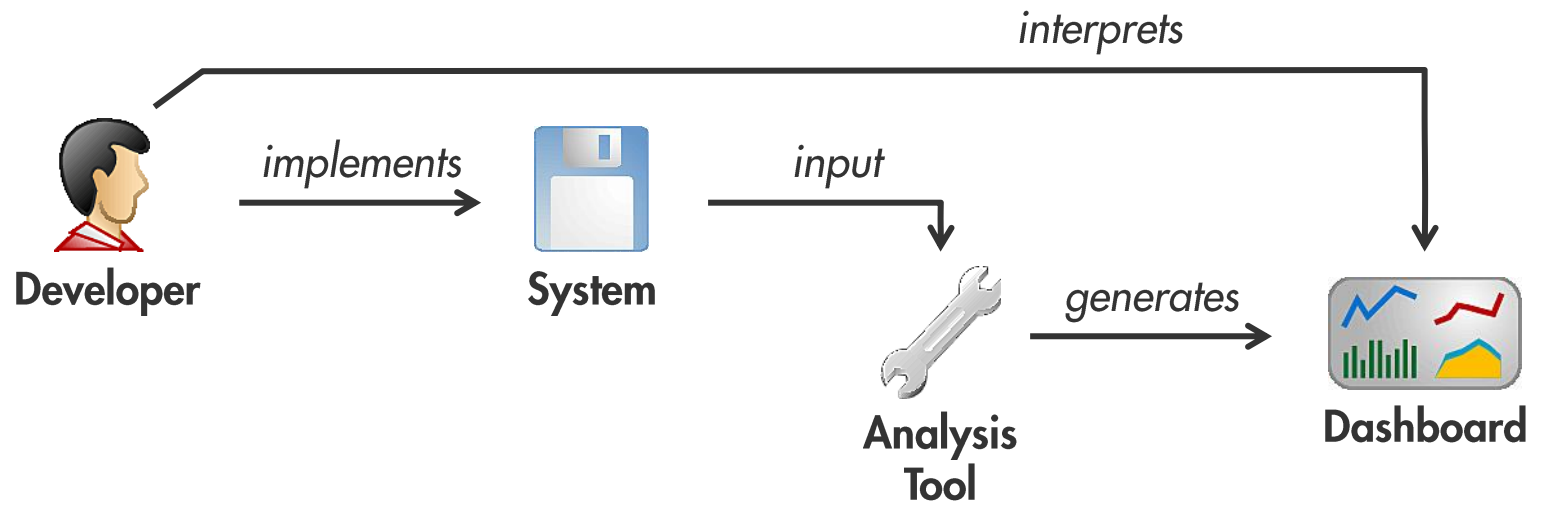
Mit manuell gewartetem Applikationscode beginnen



Best Practice: Angemessenes Qualitäts-Ziel

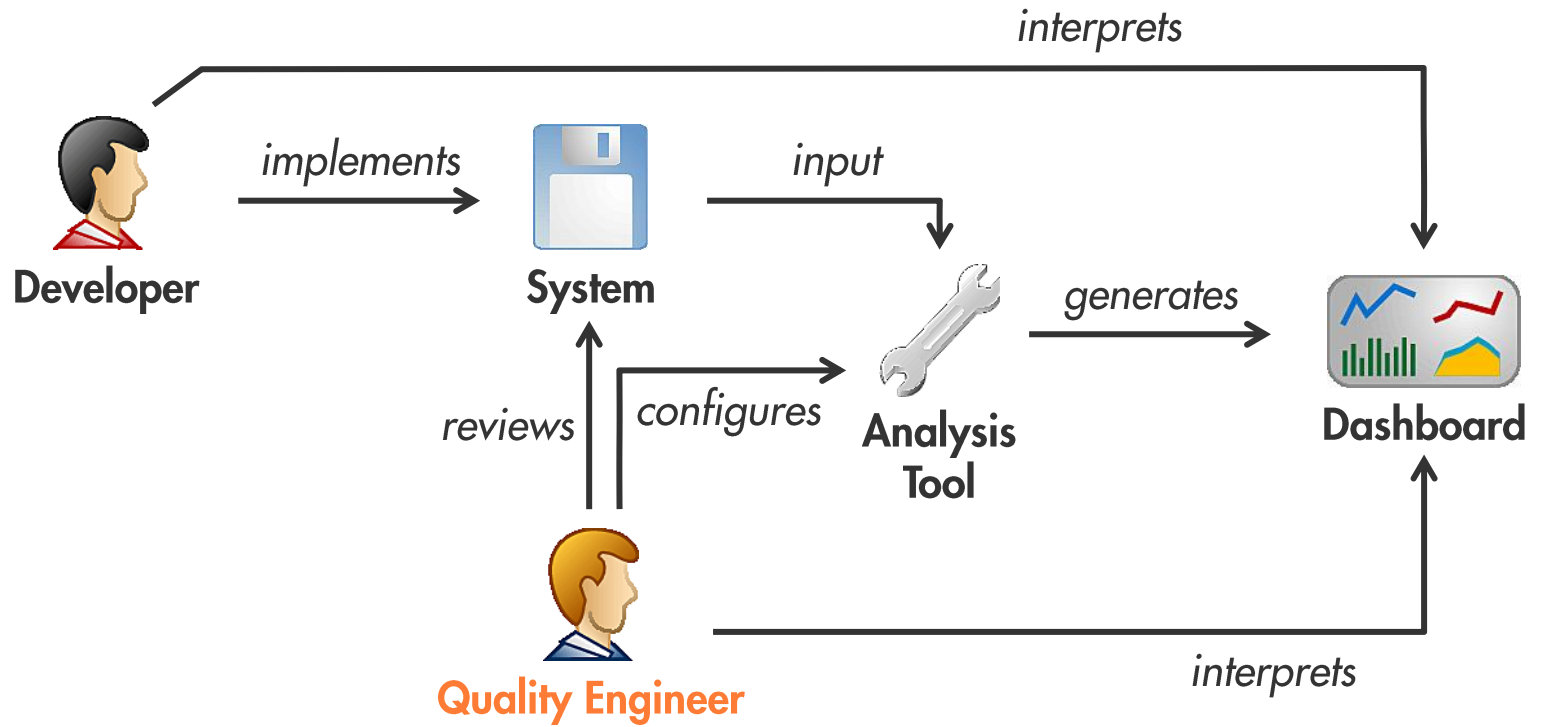


Z.B. ConQAT, FxCop, Teamscale

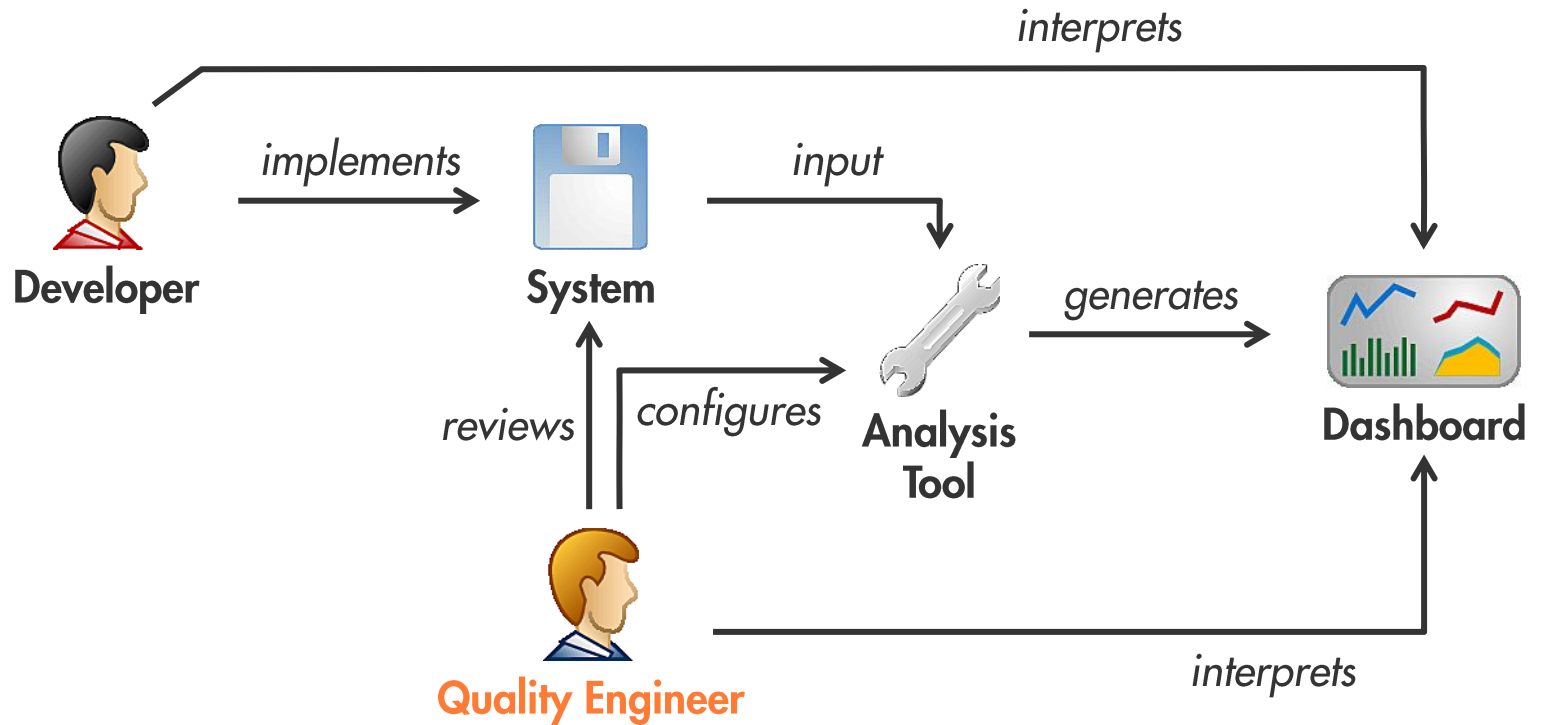


- False positives
 - Information overflow
- => Analyseergebnisse werden ignoriert



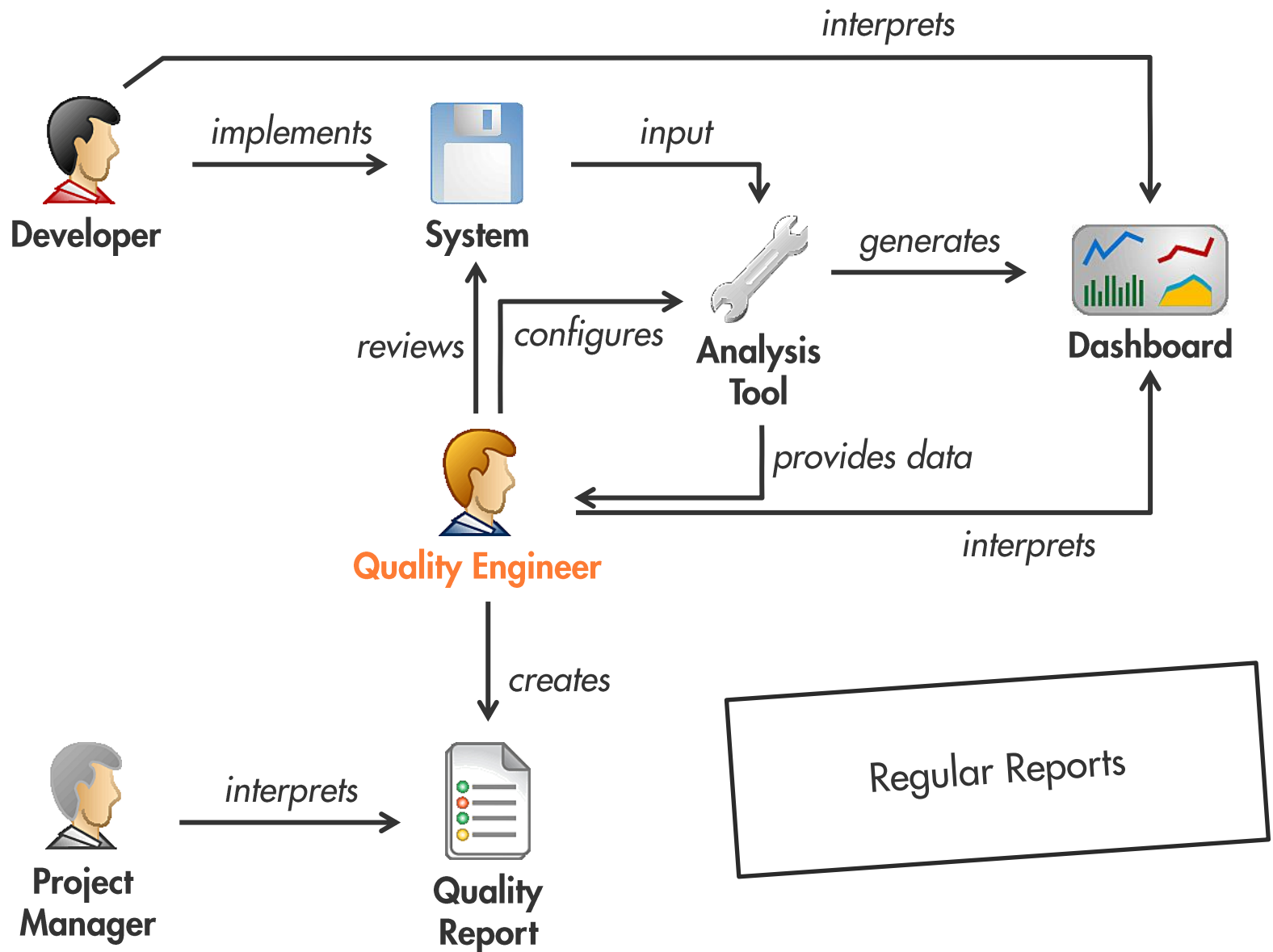


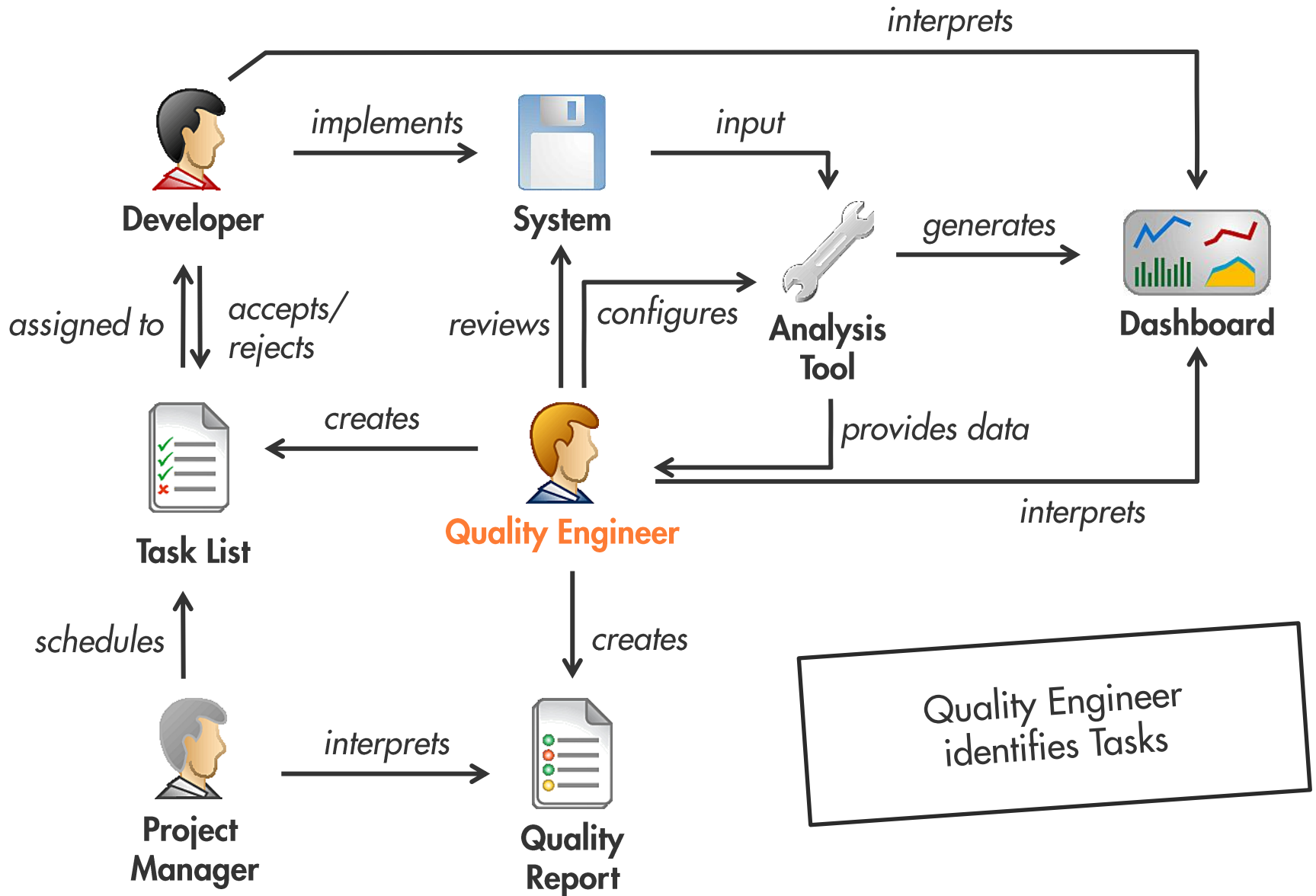
Someone to take Care of Tool & Quality Goals



- Awareness of Management?
 - Features first
- => Keine Verbesserung





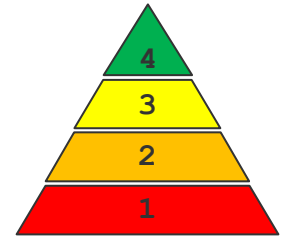


Best Practice: Quality Reports

Quality Indicator	Full Assessment		Delta Assessment	
Stability of Regular Team Builds	Build is stable.	✓	At the time of the last report, there was no automated build at all.	➡
Architecture Conformance	There are zero violations.	✓	Architecture specification was completed and is now fully	➡
Stability of U Tests	<div style="border: 1px solid black; padding: 5px;"> <p>1.7 Duplicated Code ✗ ➡</p> <p>TQE Target: Clone coverage of less than 10%.</p> <div style="text-align: center;"> <p>Clone Coverage</p> <p>0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 %</p> </div> </div>			
Code Coverage				
Compiler Warnings				
Coding Guidelines Violations				
Duplicated Code	With a clone coverage of 12.2% the threshold of 10% is slightly violated.	✗	Clone coverage did not change significantly.	➡
File Size	The threshold regarding files > 400 LOC is violated.	✗	Use of partial classes improves the metric values but does not improve code quality	➡

Best Practice: TQE Tasks

- Bei jedem Report identifiziert
- Passend zu Quality Goal des Projekts
- Projekt entscheidet *ob* und *wann*



form routine.

99902 [*Duplicated Code*] Remove duplicated code in function modules

(from I. 169),

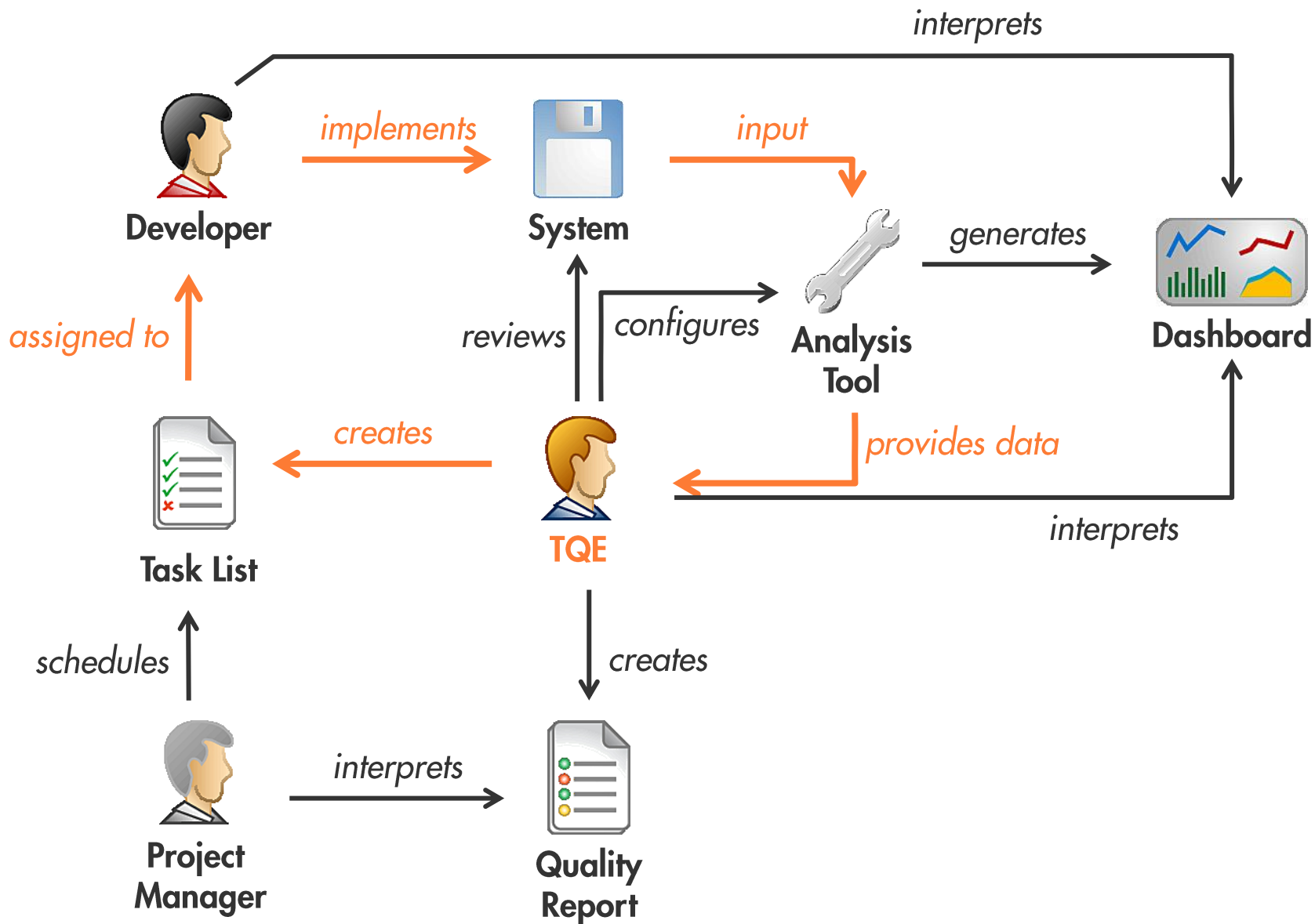
(from I. 274) and (from II. 279, 577 and 1037) which recently were equally modified in 5 clone instances. Each duplicate is 91 lines long and equal unless two literals.

log parts.

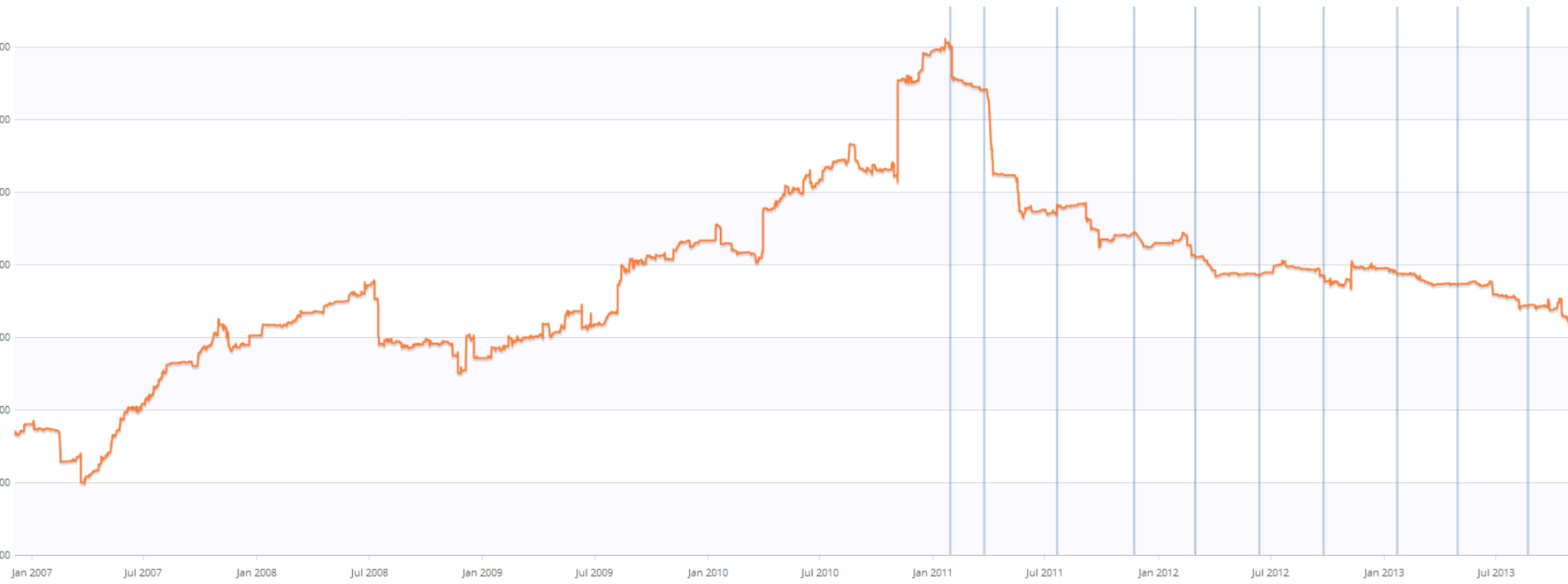
99914 [*Nesting Depth*] Restructure function module

to reduce the deep nesting which was added. E.g. by extracting code within loops to helper function modules.

99915 [*Nesting Depth*] Restructure function module







100k

4.7%

no change

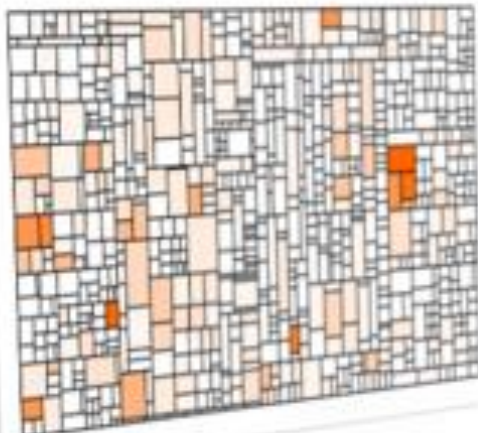


488

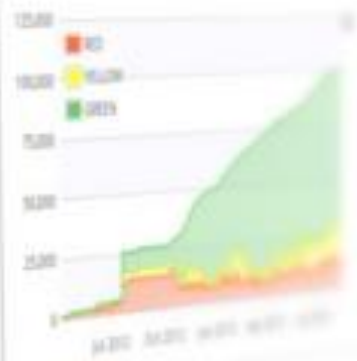
30 Day Trend for teamscale



Findings Treemap for teamscale



Review Rating Trend for teamscale

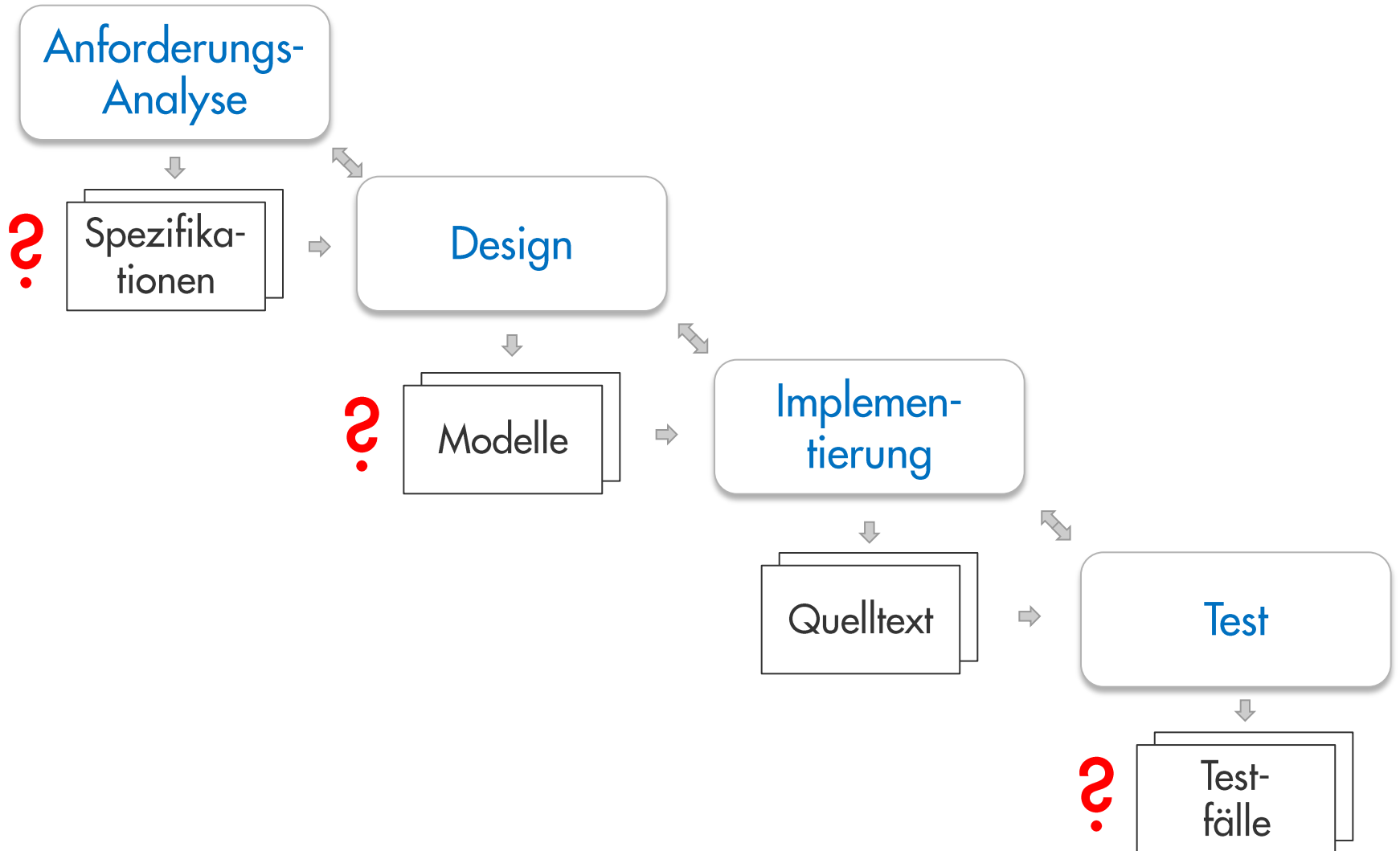


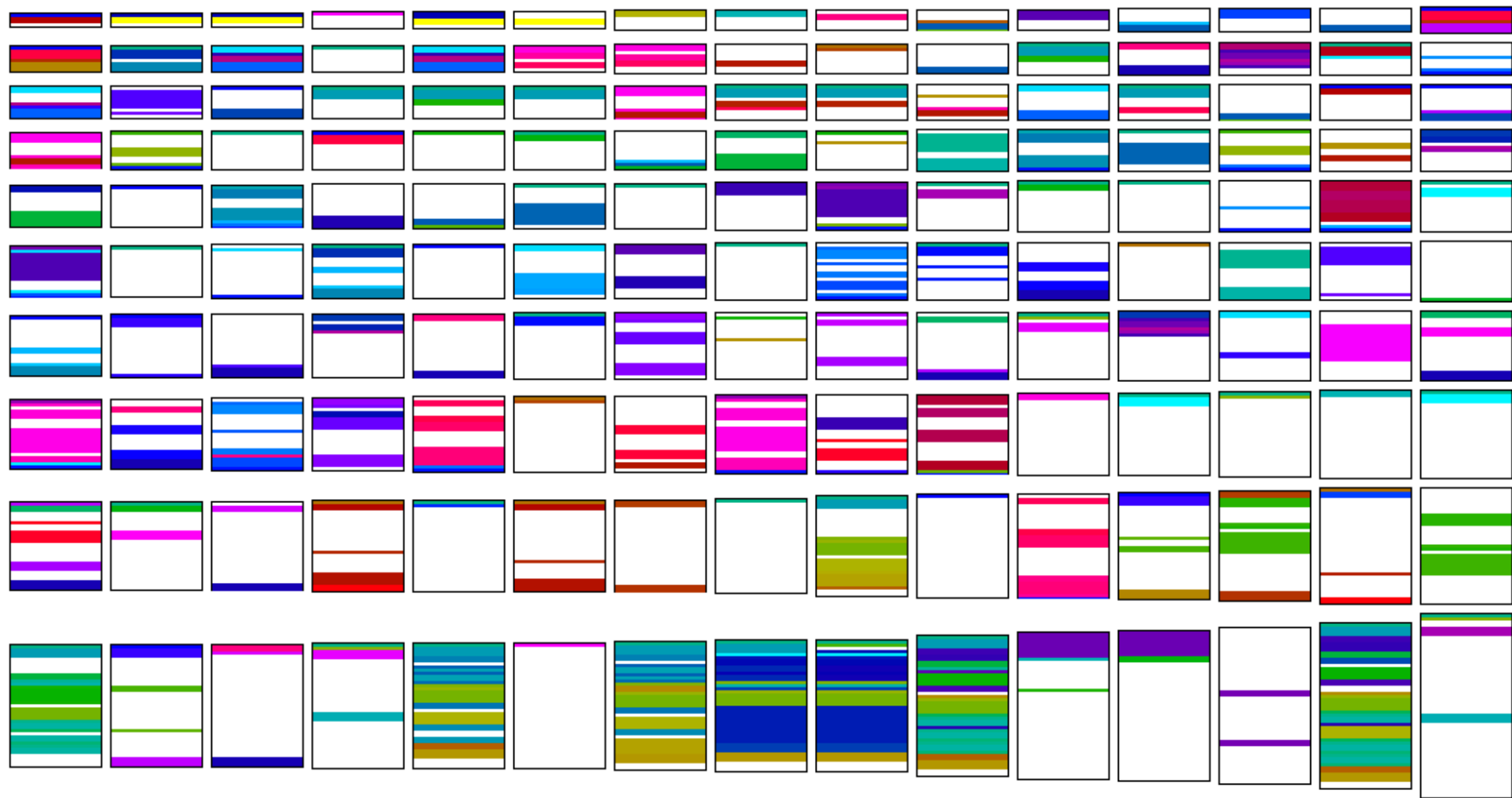
Findings distribution for teamscale

Findings Summary for teamscale

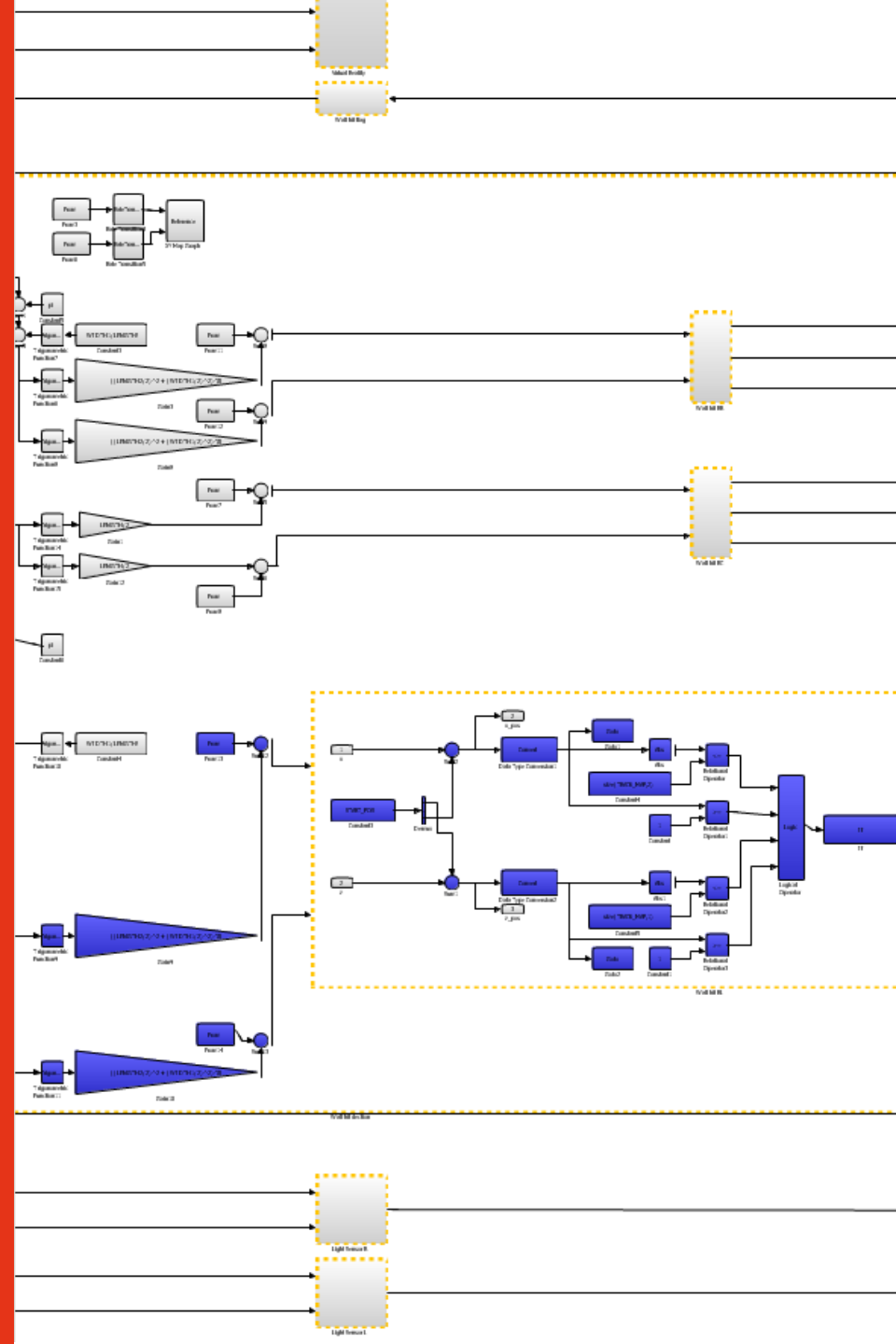
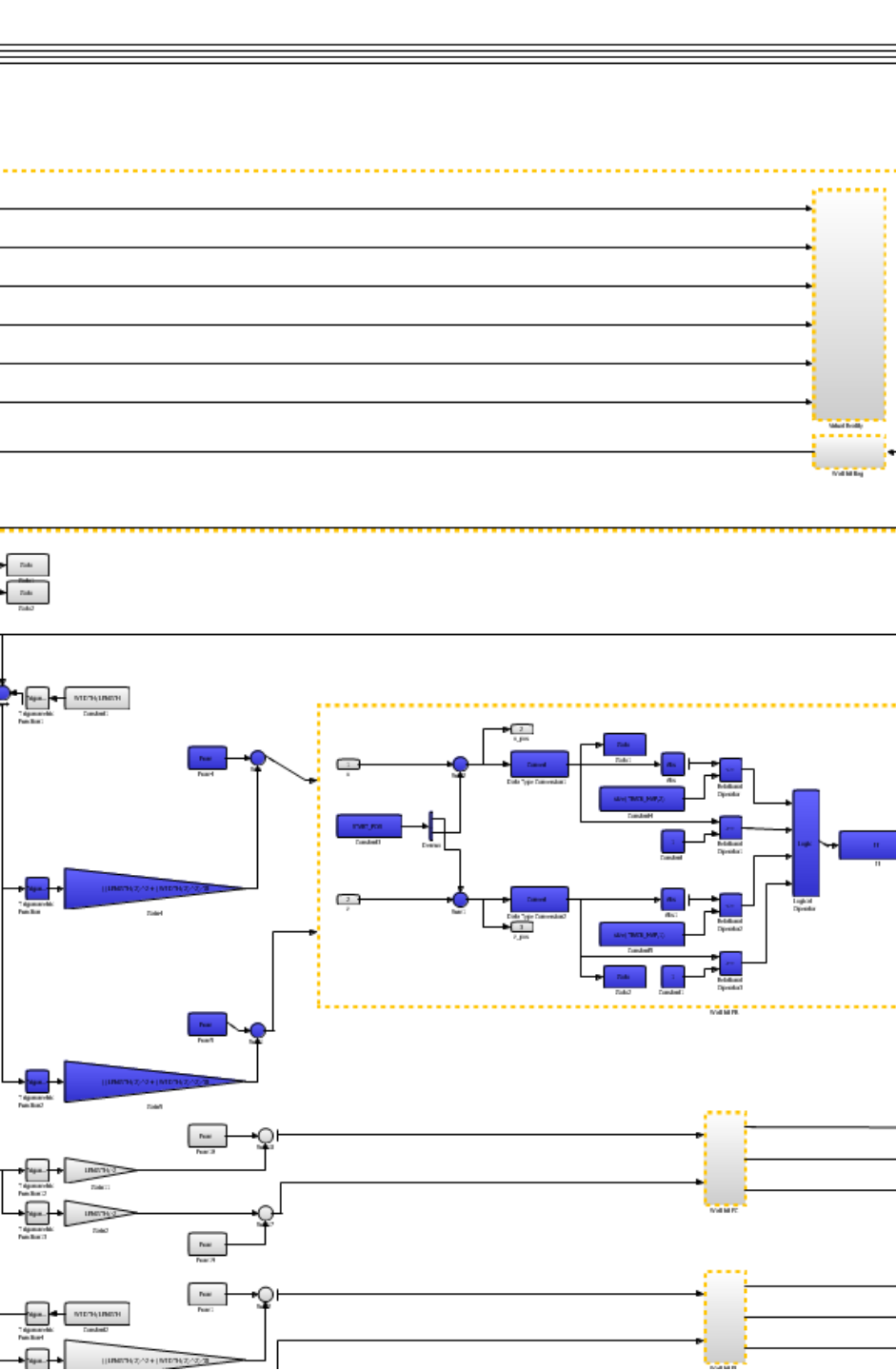
Core Metrics for teamscale

<http://www.teamscale.org>

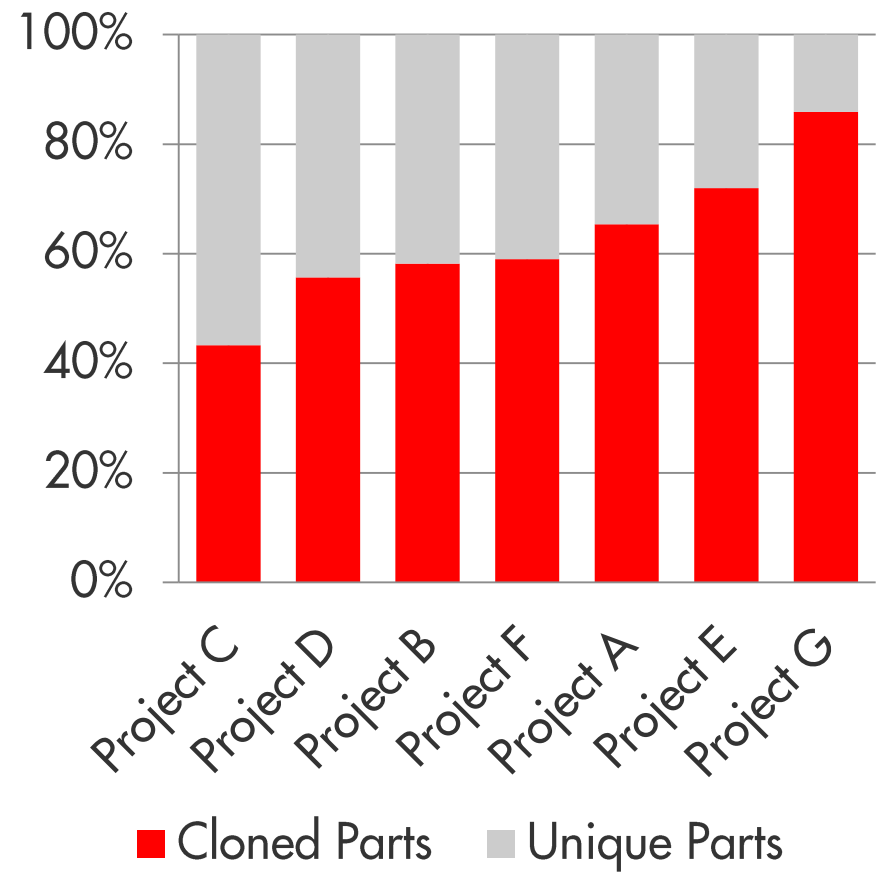


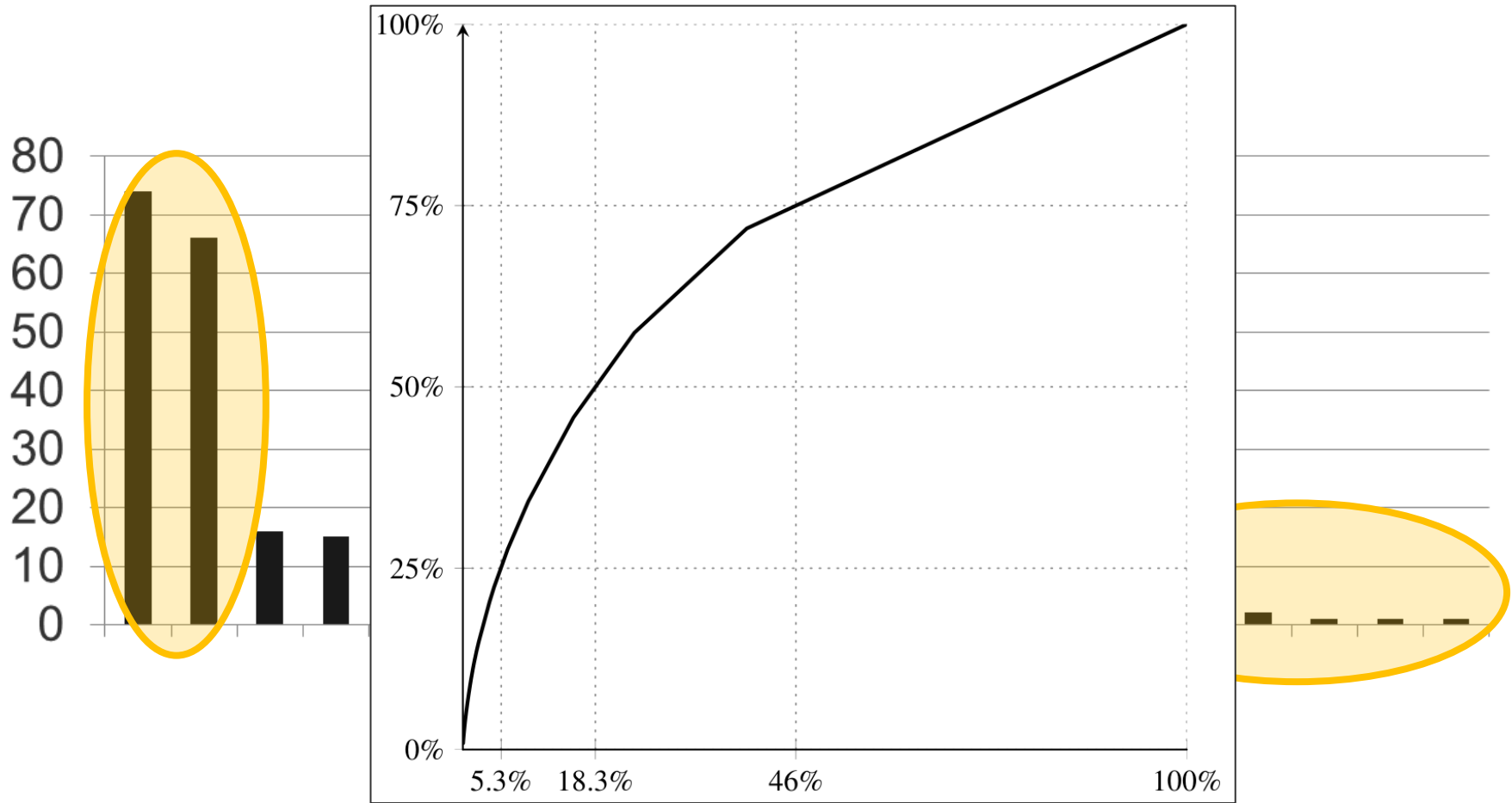


176 use cases in total, 150 contain cloning



	System under Test	Test Suite
		# Tests
System A	330 kLoC	266
System B	580 kLoC	1,059
System C	150 kLoC	72
System D	430 kLoC	180
System E	760 kLoC	1,804
System F	1,400 kLoC	135
System G	160 kLoC	605







Splitting with java.lang.String.split()

```
String[] addresses2 = addresses.split(Pattern.  
    quote(String.valueOf(separator)));
```

Splitting with java.util.StringTokenizer

```
ArrayList<String> validEmails = new ArrayList<  
    String>();  
StringTokenizer st = new StringTokenizer(  
    addresses, Character.toString(separator));  
while (st.hasMoreTokens()) {  
    String tmp = st.nextToken();  
    validEmails.add(tmp);  
}
```

Splitting with custom algorithm 1

```
List<String> result = new ArrayList<String>();  
int z = 0;  
for (int i=0; i<addresses.length(); i++) {  
    if (i==addresses.length()-1) {  
        result.add(addresses.substring(z, i+1));  
    }  
    if (addresses.charAt(i)==separator) {  
        result.add(addresses.substring(z, i));  
        z=i+1;  
    }  
}
```

Splitting with custom algorithm 2

```
List<String> curAddrs = new ArrayList<String>();  
String buffer = "";  
for (int i=0; i<addresses.length(); i++) {  
    if (addresses.charAt(i) != separator) {  
        buffer += addresses.charAt(i);  
    } else {  
        curAddrs.add(buffer);  
        buffer = "";  
    }  
}  
curAddrs.add(buffer);
```

Splitting with custom algorithm 3

```
List<String> emailListe= new ArrayList<String>();  
int trenneralt = 0;  
while (addresses.indexOf(separator, trenneralt) !=  
    -1) {  
    int trennerneu = addresses.indexOf(separator,  
        trenneralt);  
    emailListe.add(addresses.substring(trenneralt,  
        trennerneu));  
    trenneralt = trennerneu + 1;  
}
```

PASTE

PASTE

COPY



PASTE

PAS

Fazit

Klone sind ein Problem für die Weiterentwicklung. In allen Softwareartefakten.

Kontakt

Dr. Elmar Jürgens · juergens@cqse.eu · +49 179 675 3863

[@ElmarJuergens](#)

www.cqse.eu/en/blog

CQSE GmbH, Lichtenbergstraße 8,
85748 Garching bei München