3.– 6. September 2012
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Copy & Paste & Bug?

Umgang mit Redundanz in Software-Artefakten

Dr. Elmar Juergens

CQSE GmbH

http://upload.wikimedia.org/wikipedia/commons/e/e6/Karl-Theodor_Freiherr_von_und_zu_Guttenberg.jpg

Über Mich

Forschung

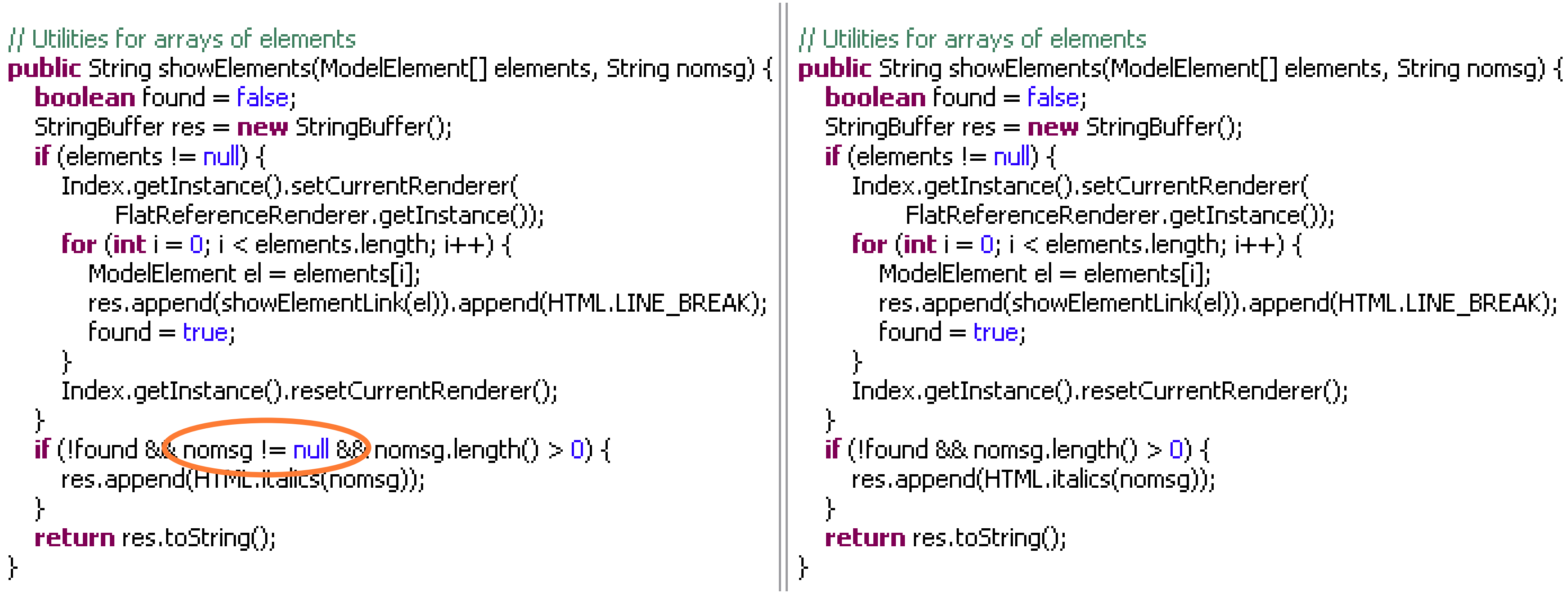
- Clone Detection
- Architekturanalyse

Beratung

- Mitgründer
- Qualitäts-Bewertung & Qualitäts-Controlling

Entwicklung

- Continuous Quality Assessment Toolkit ConQAT
- >300 kLOC, Apache Lizenz, >20.000 Downloads



```java
// 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();
}
```

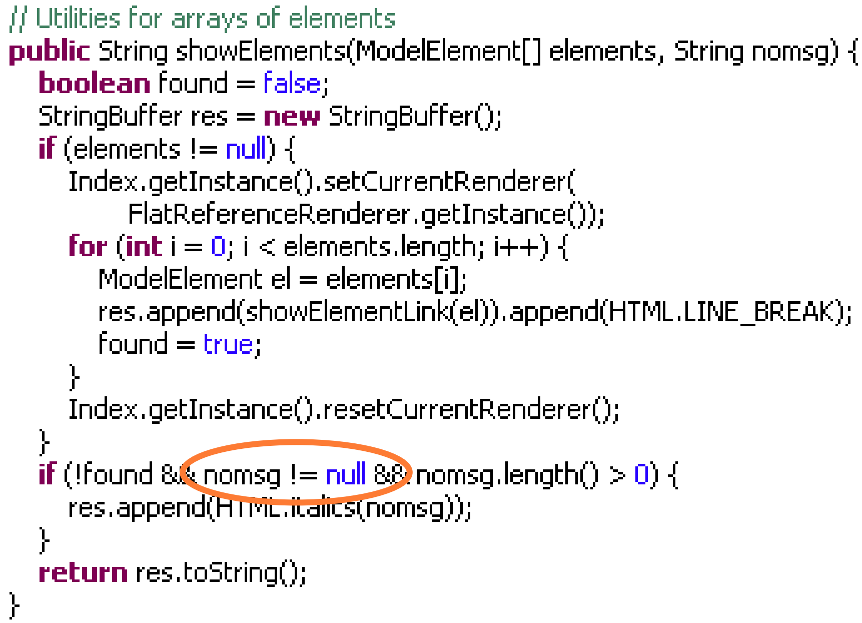
```java
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
                FlatReferenceRenderer.getInstance());
        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();
}
```

#1 Code Smell (1999)

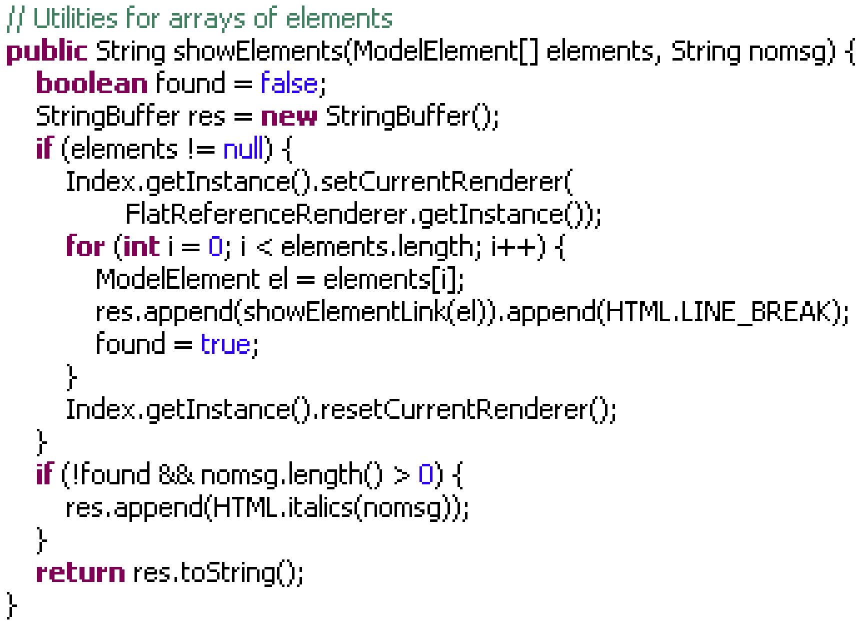
http://www.flickr.com/photos/pragdave/173640462/

1975

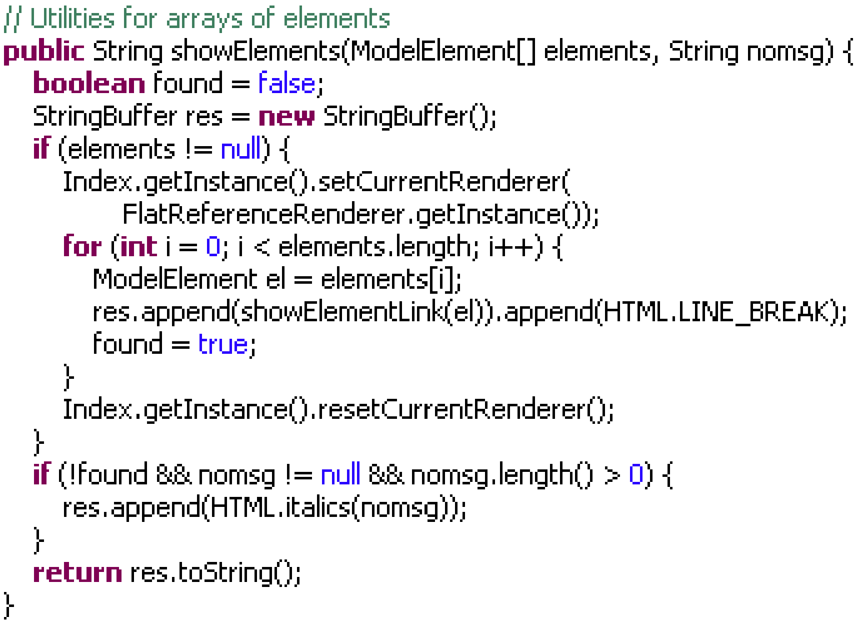
http://de.wikipedia.org/w/index.php?title=Datei:Fred_Brooks.jpg



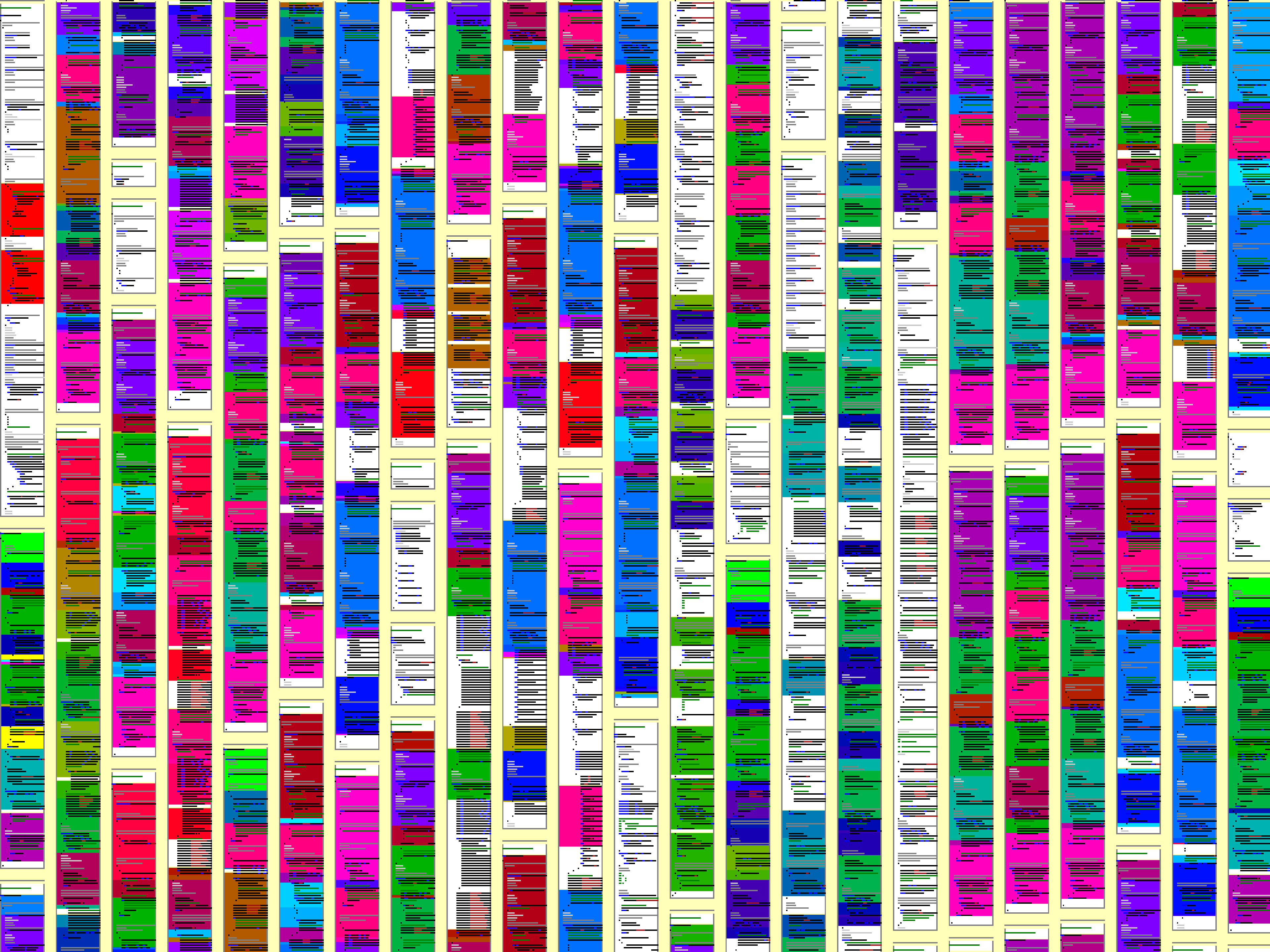
```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
   boolean found = false;
   StringBuffer res = new StringBuffer();
   if (elements != null) {
      Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
      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();
}
```

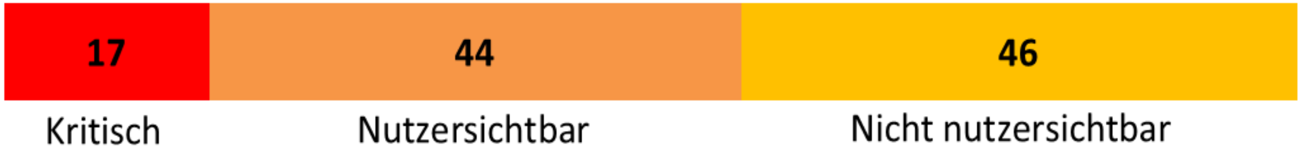


```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
                FlatReferenceRenderer.getInstance());
        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();
}
```

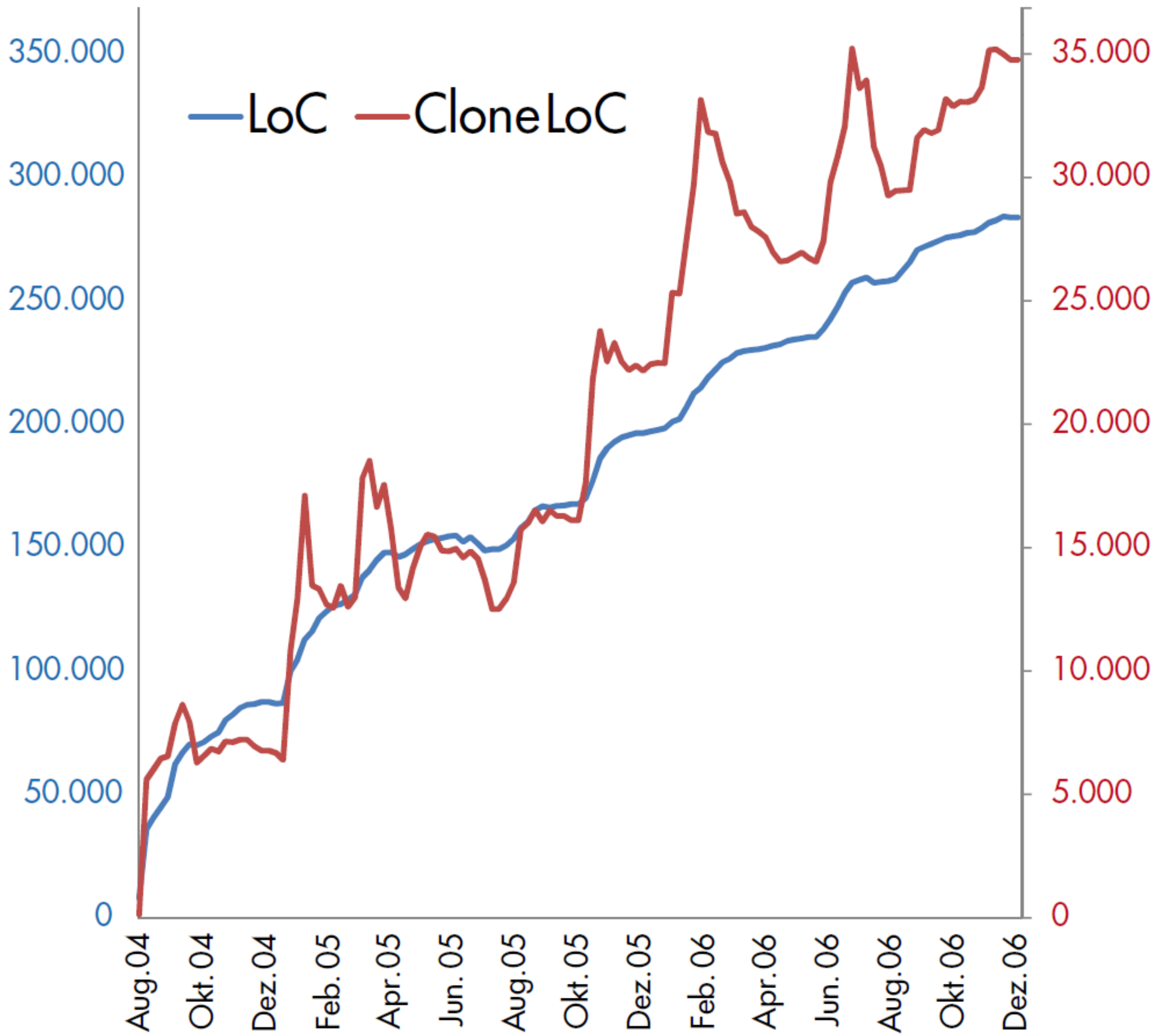


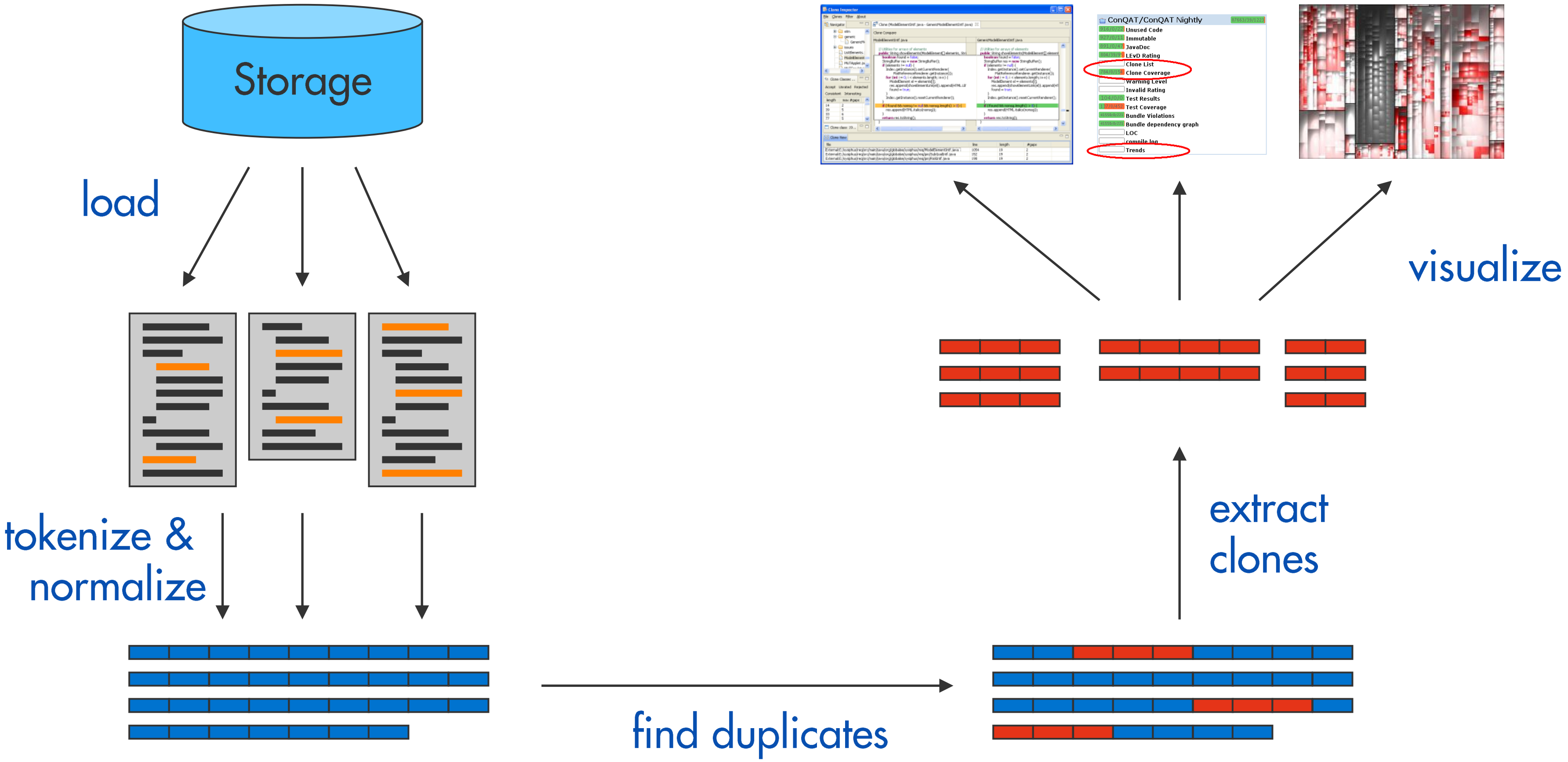
Studie

- Über 100 Fehler in produktiver Software



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


LoC
CloneLoC
350.000
300.000
250.000
200.000
150.000
100.000
50.000
0
35.000
30.000
25.000
20.000
15.000
10.000
5.000
0
Aug.04
Okt.04
Dez.04
Feb.05
Apr.05
Jun.05
Aug.05
Okt.05
Dez.05
Feb.06
Apr.06
Jun.06
Aug.06
Okt.06
Dez.06



Storage

load

tokenize &
normalize

find duplicates

extract
clones

visualize

ConQAT/ConQAT Nightly 87663/39/1223

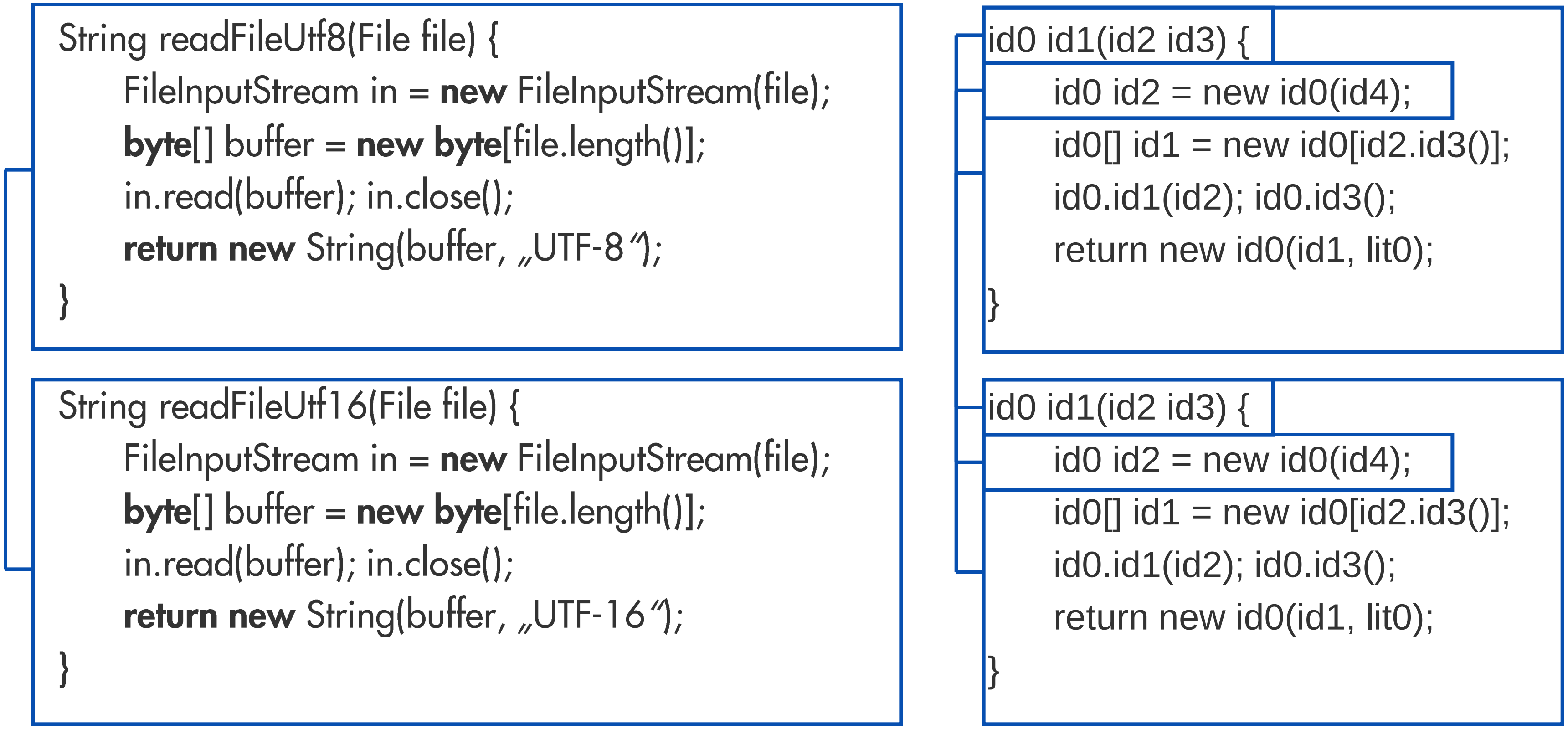
916/0/22	Unused Code
927/0/11	Immutable
891/0/47	JavaDoc
804/39/95	LEvD Rating
	Clone List
784/0/154	Clone Coverage
	Warning Level
	Invalid Rating
104/0/0	Test Results
137/0/452	Test Coverage
4155D/0/221	Bundle Violations
4155D/0/221	Bundle dependency graph
	LOC
	compile log
	Trends

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

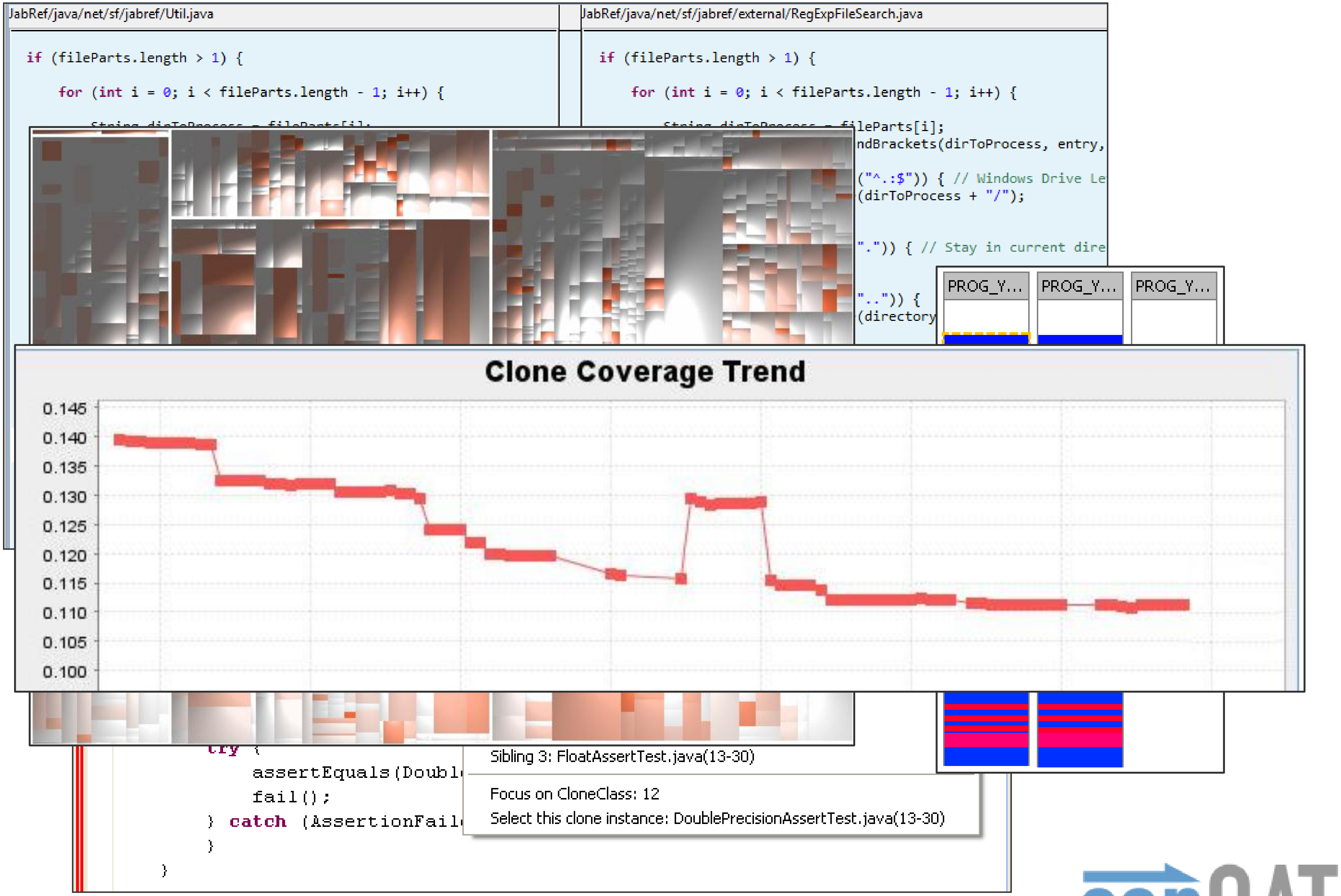


```
String readFileUtf8(File file) {
	FileInputStream in = new FileInputStream(file);
	byte[] buffer = new byte[file.length()];
	in.read(buffer); in.close();
	return new String(buffer, „UTF-8“);
}
```

```
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];
ndBrackets(dirToProcess, entry,

("^.:$")) { // Windows Drive Le
(dirToProcess + "/");


".")) { // Stay in current dire


"..")) {
(directory
```

```
try {
    assertEquals(Doubl
    fail();
} catch (AssertionFail
}
}
```

Sibling 3: FloatAssertTest.java(13-30)

Focus on CloneClass: 12

Select this clone instance: DoublePrecisionAssertTest.java(13-30)

conQAT

Best Practice: Diskriminierung (von Code)

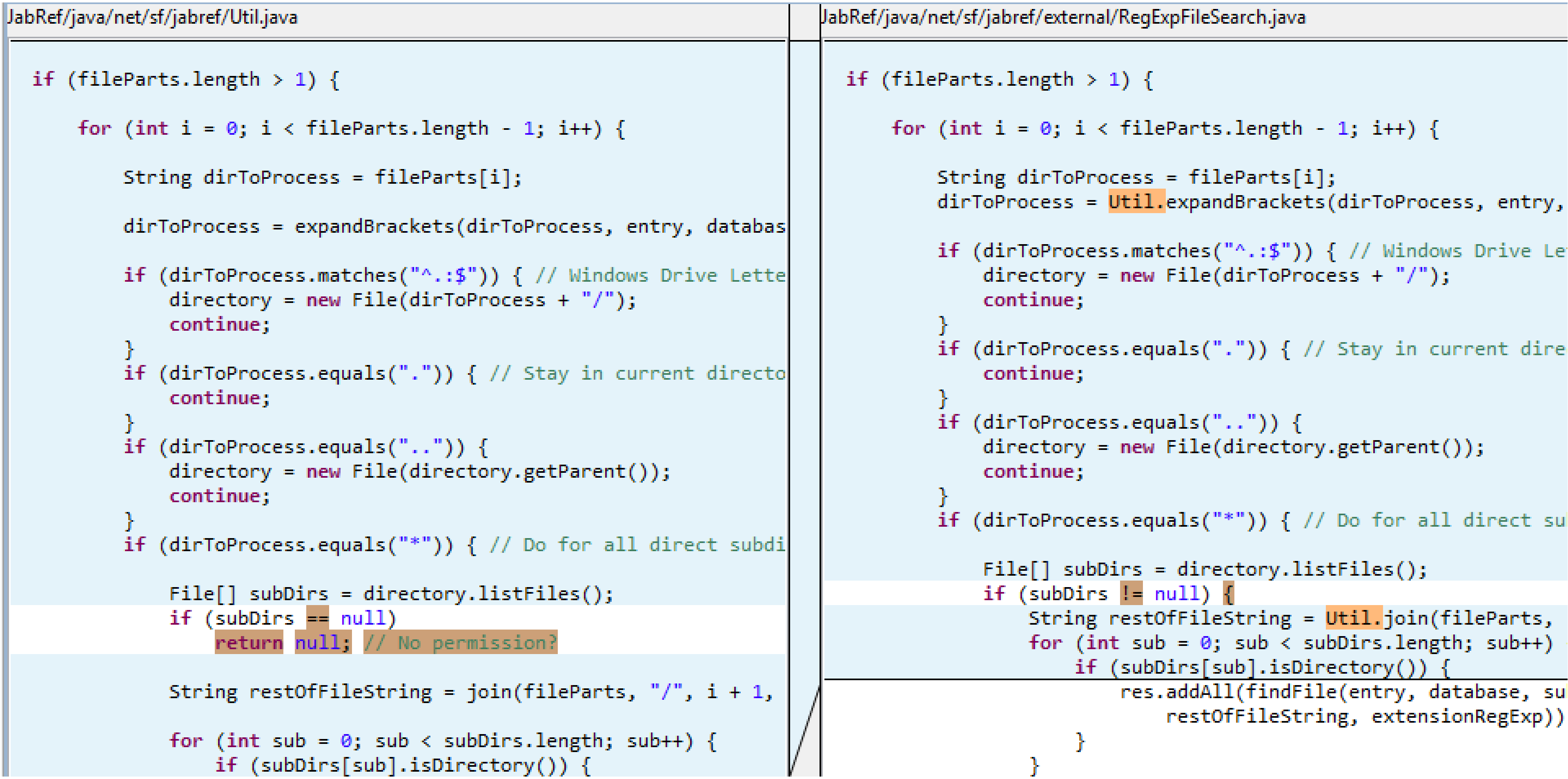
Art der Wartung

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

Aufgabe im Projekt

- Teil der Anwendung
- Test
- Hilfswerkzeug

Mit manuell gewartetem Applikationscode beginnen

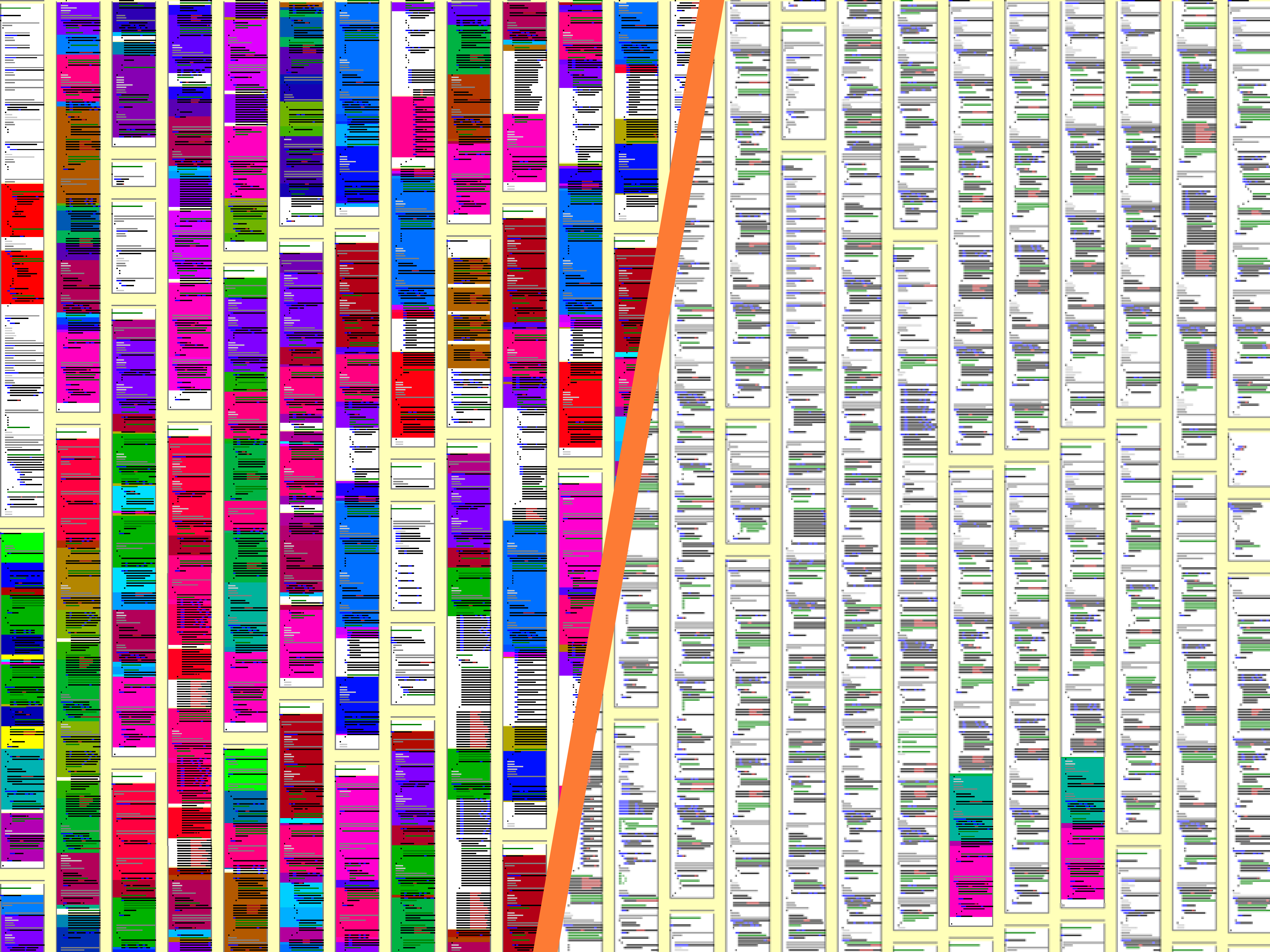


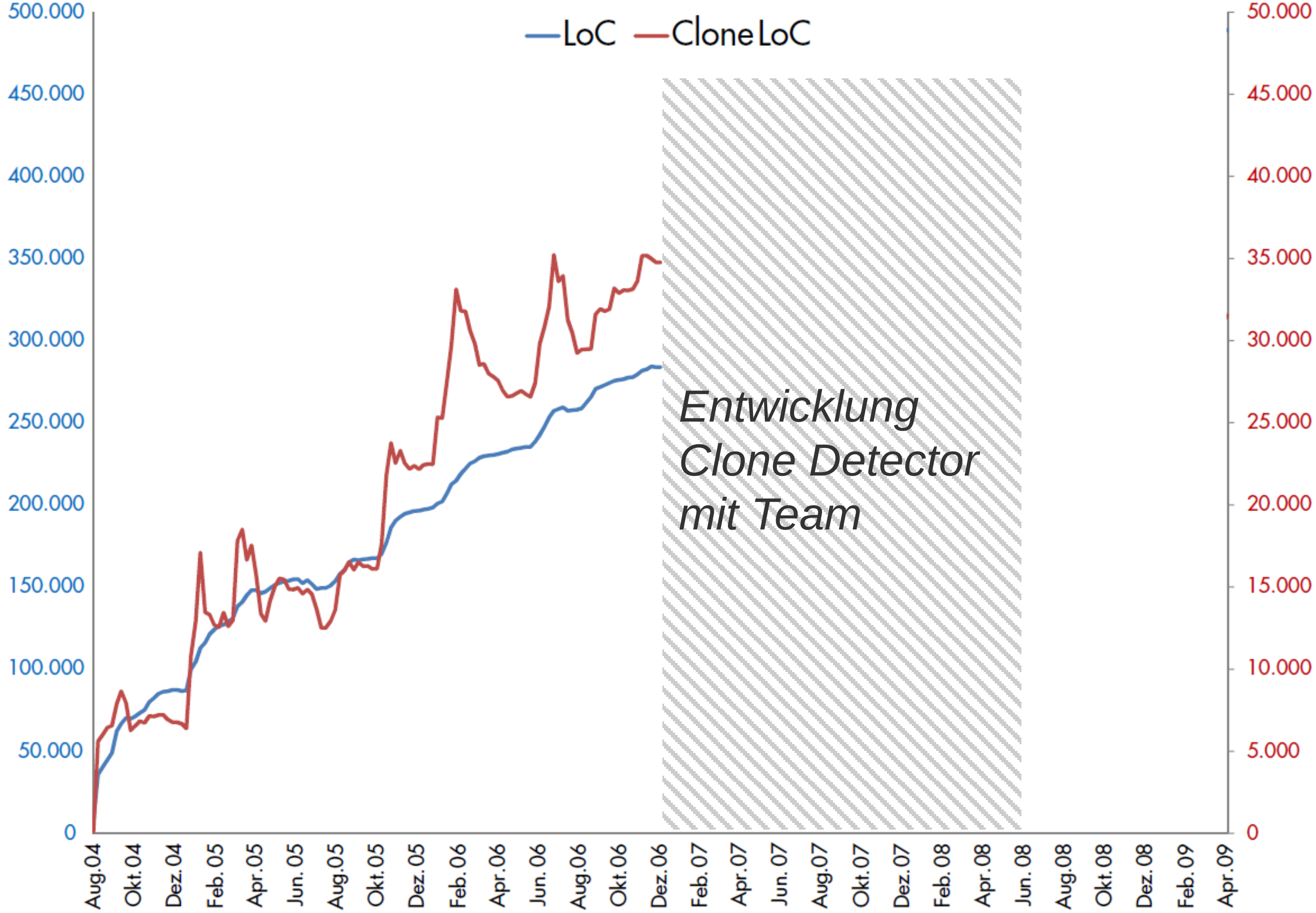
JabRef/java/net/sf/jabref/Util.java

```java
if (fileParts.length > 1) {

    for (int i = 0; i < fileParts.length - 1; i++) {

        String dirToProcess = fileParts[i];

        dirToProcess = expandBrackets(dirToProcess, entry, databas

        if (dirToProcess.matches("^.:$")) { // Windows Drive Lette
            directory = new File(dirToProcess + "/");
            continue;
        }
        if (dirToProcess.equals(".")) { // Stay in current directo
            continue;
        }
        if (dirToProcess.equals("..")) {
            directory = new File(directory.getParent());
            continue;
        }
        if (dirToProcess.equals("*")) { // Do for all direct subdi

            File[] subDirs = directory.listFiles();
            if (subDirs == null)
                return null; // No permission?

            String restOfFileString = join(fileParts, "/", i + 1,

            for (int sub = 0; sub < subDirs.length; sub++) {
                if (subDirs[sub].isDirectory()) {
```

JabRef/java/net/sf/jabref/external/RegExpFileSearch.java

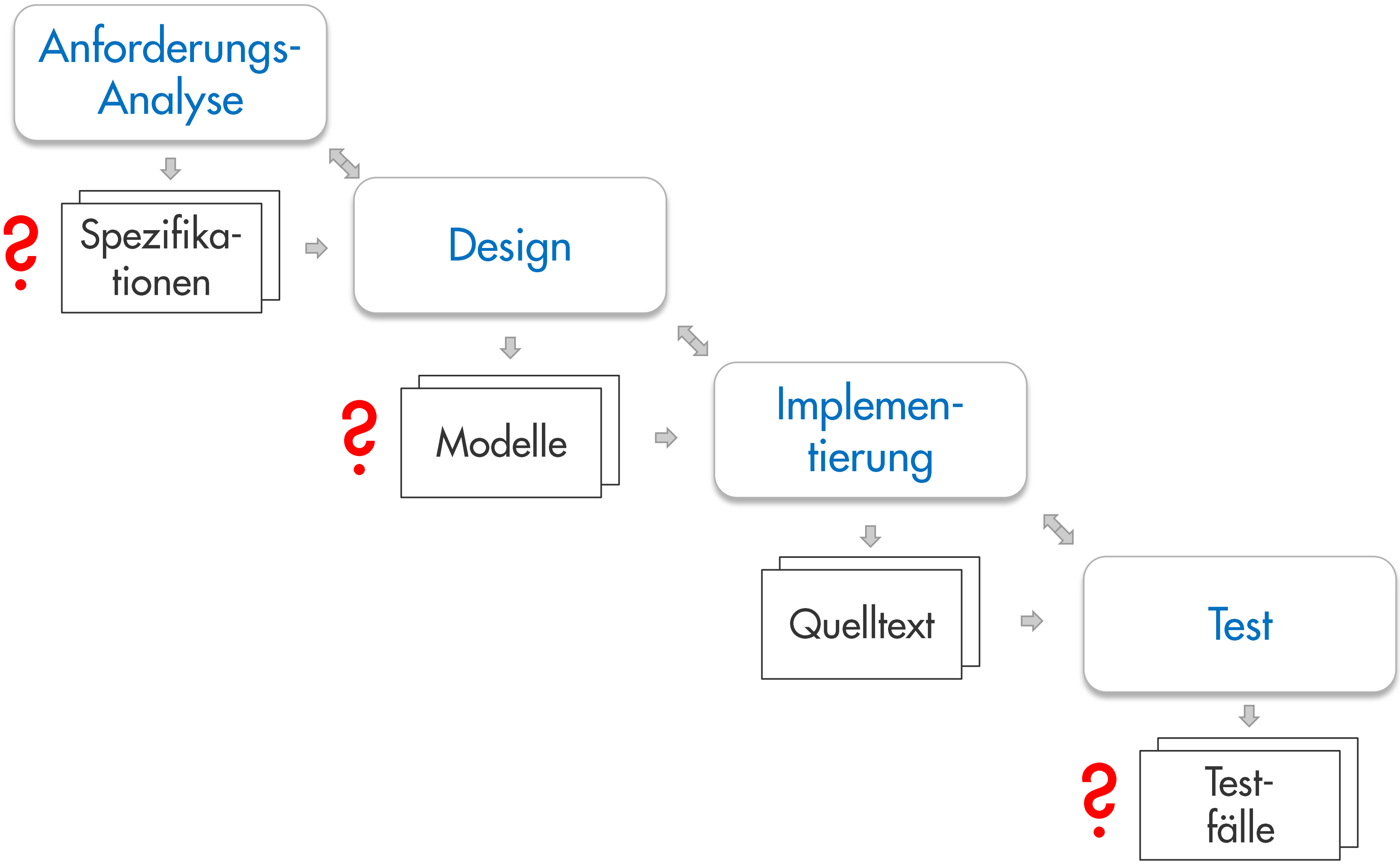
```java
if (fileParts.length > 1) {

    for (int i = 0; i < fileParts.length - 1; i++) {

        String dirToProcess = fileParts[i];
        dirToProcess = Util.expandBrackets(dirToProcess, entry,

        if (dirToProcess.matches("^.:$")) { // Windows Drive Le
            directory = new File(dirToProcess + "/");
            continue;
        }
        if (dirToProcess.equals(".")) { // Stay in current dire
            continue;
        }
        if (dirToProcess.equals("..")) {
            directory = new File(directory.getParent());
            continue;
        }
        if (dirToProcess.equals("*")) { // Do for all direct su

            File[] subDirs = directory.listFiles();
            if (subDirs != null) {
                String restOfFileString = Util.join(fileParts,
                for (int sub = 0; sub < subDirs.length; sub++)
                    if (subDirs[sub].isDirectory()) {
                        res.addAll(findFile(entry, database, su
                            restOfFileString, extensionRegExp))
                    }
                }
```





Entwicklung Clone Detector mit Team

Juergens: *Why and How to Control Cloning in Software Artifacts*, Dissertation 2011



Anforderungs-Analyse

? Spezifikationen

Design

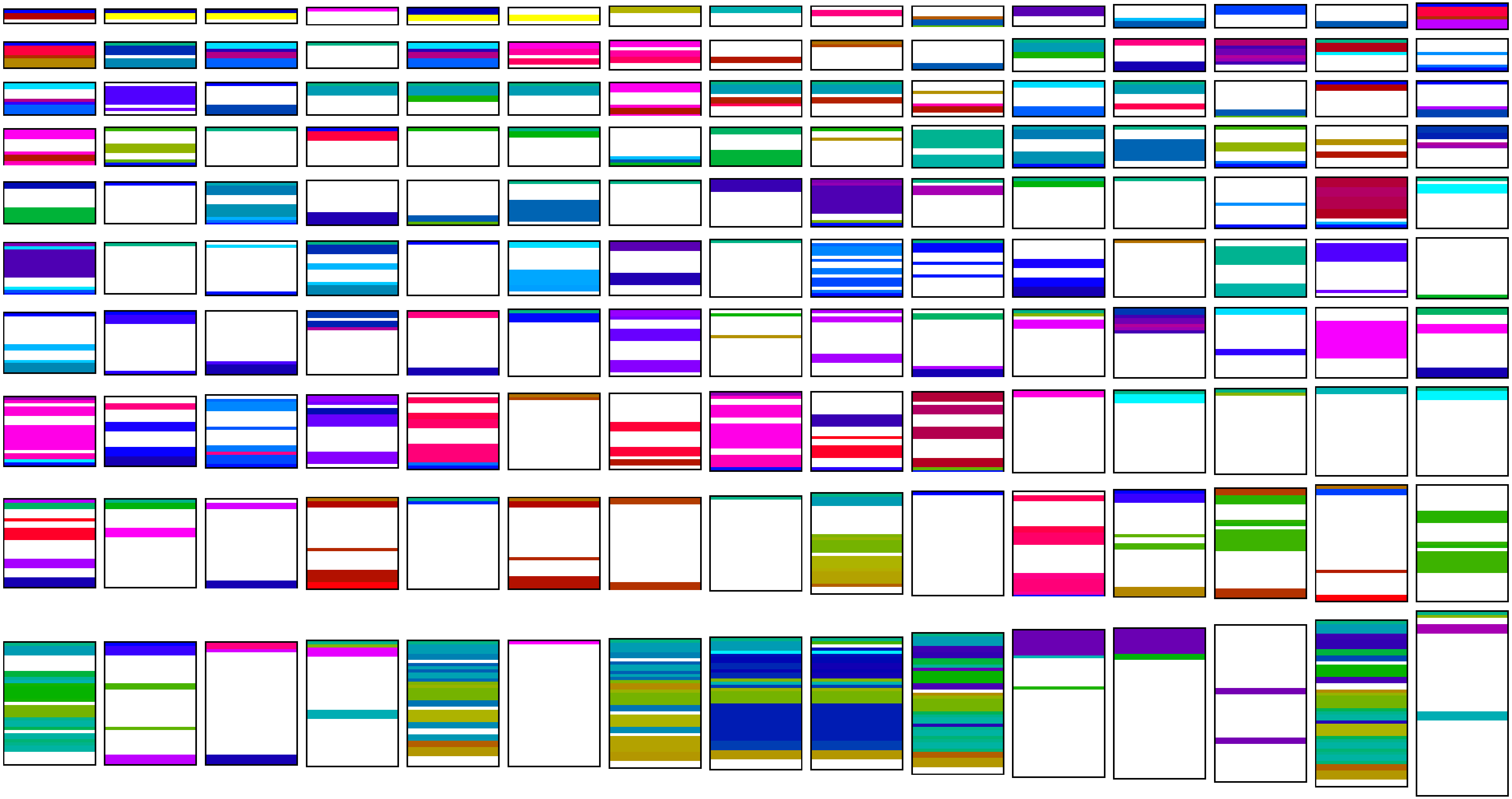
? Modelle

Implementierung

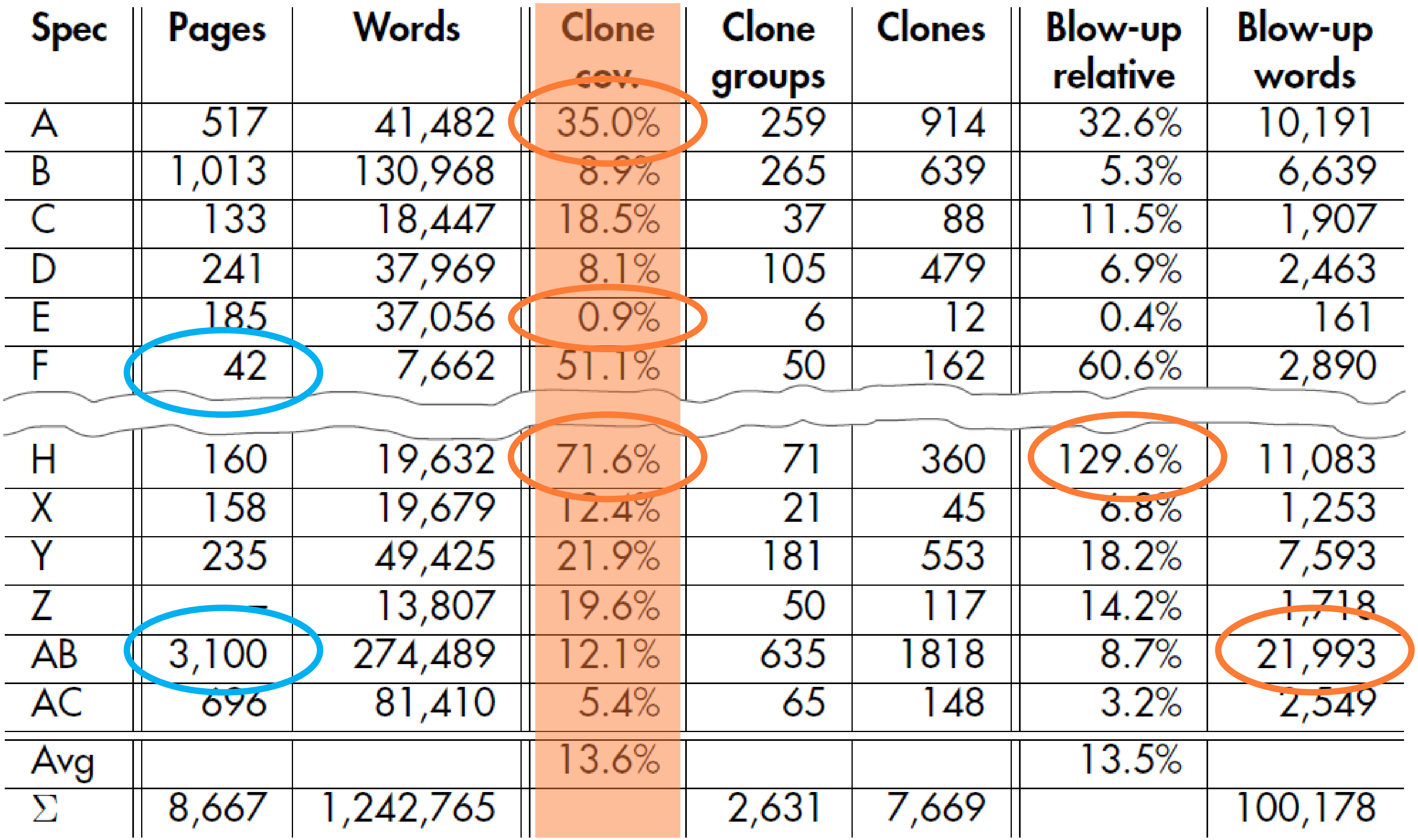
Quelltext

Test

? Testfälle

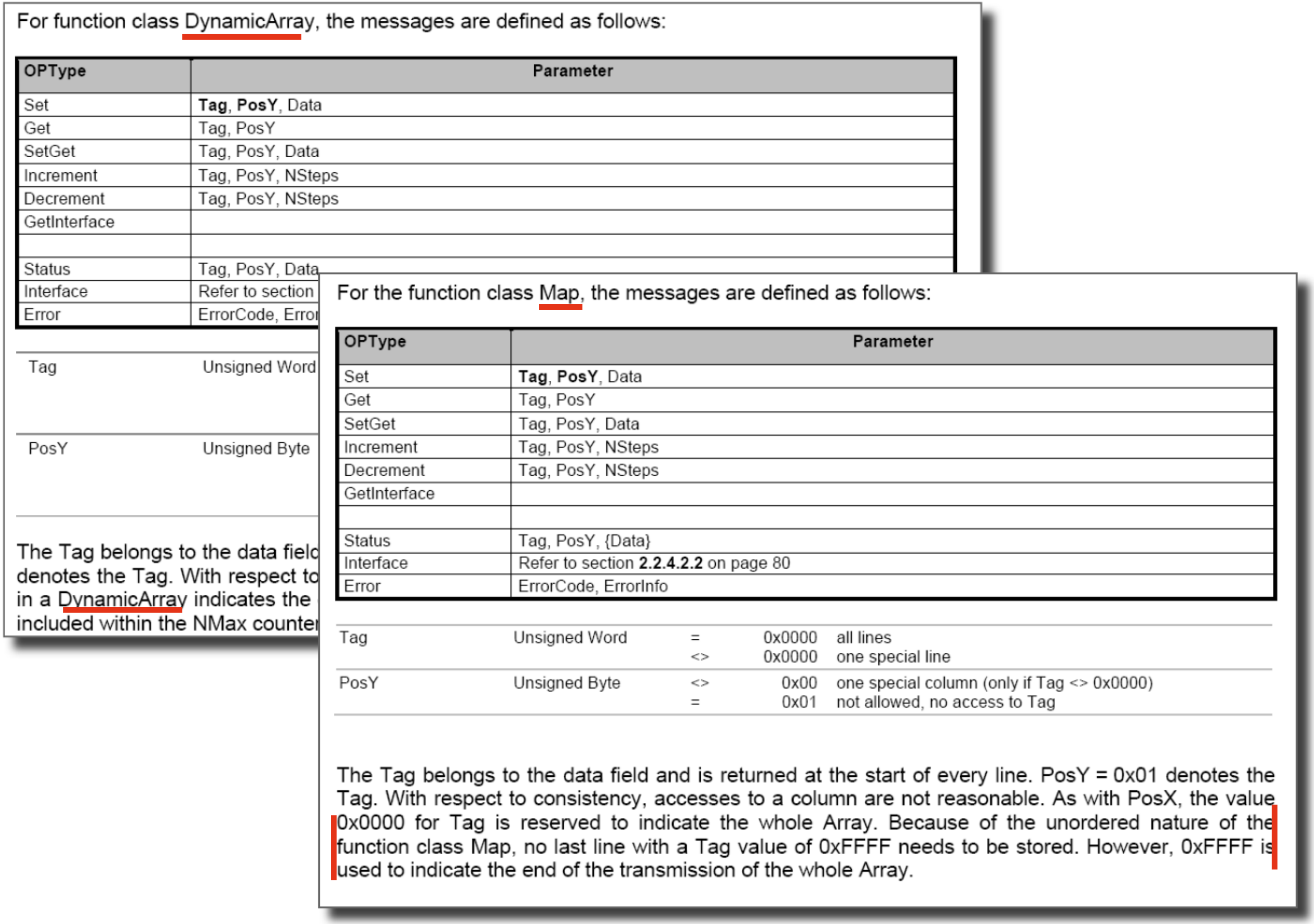


176 use cases in total, 150 contain cloning



Spec	Pages	Words	Clone cov.	Clone groups	Clones	Blow-up relative	Blow-up words
A	517	41,482	35.0%	259	914	32.6%	10,191
B	1,013	130,968	8.9%	265	639	5.3%	6,639
C	133	18,447	18.5%	37	88	11.5%	1,907
D	241	37,969	8.1%	105	479	6.9%	2,463
E	185	37,056	0.9%	6	12	0.4%	161
F	42	7,662	51.1%	50	162	60.6%	2,890
H	160	19,632	71.6%	71	360	129.6%	11,083
X	158	19,679	12.4%	21	45	6.8%	1,253
Y	235	49,425	21.9%	181	553	18.2%	7,593
Z	–	13,807	19.6%	50	117	14.2%	1,718
AB	3,100	274,489	12.1%	635	1818	8.7%	21,993
AC	696	81,410	5.4%	65	148	3.2%	2,549
Avg			13.6%			13.5%	
Σ	8,667	1,242,765		2,631	7,669		100,178

Juergens et al: *Can Clone Detection Support QA of Requirement Specs?*, ICSE 2010



For function class DynamicArray, the messages are defined as follows:

OPType	Parameter
Set	**Tag**, **PosY**, Data
Get	Tag, PosY
SetGet	Tag, PosY, Data
Increment	Tag, PosY, NSteps
Decrement	Tag, PosY, NSteps
GetInterface	
Status	Tag, PosY, Data
Interface	Refer to section
Error	ErrorCode, Error

Tag	Unsigned Word
PosY	Unsigned Byte

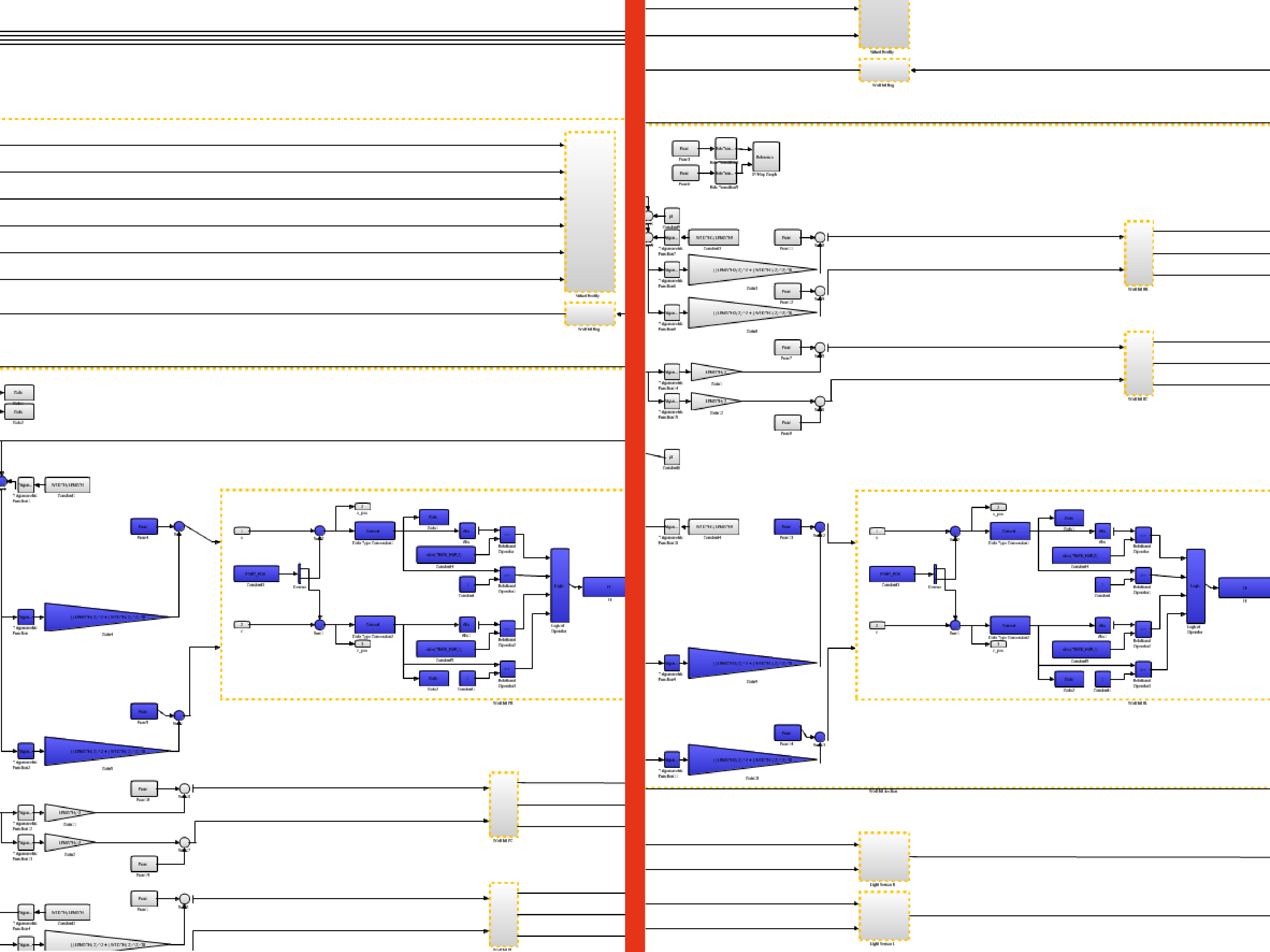
The Tag belongs to the data field
denotes the Tag. With respect to
in a DynamicArray indicates the
included within the NMax counter

For the function class Map, the messages are defined as follows:

OPType	Parameter
Set	**Tag**, **PosY**, Data
Get	Tag, PosY
SetGet	Tag, PosY, Data
Increment	Tag, PosY, NSteps
Decrement	Tag, PosY, NSteps
GetInterface	
Status	Tag, PosY, {Data}
Interface	Refer to section **2.2.4.2.2** on page 80
Error	ErrorCode, ErrorInfo

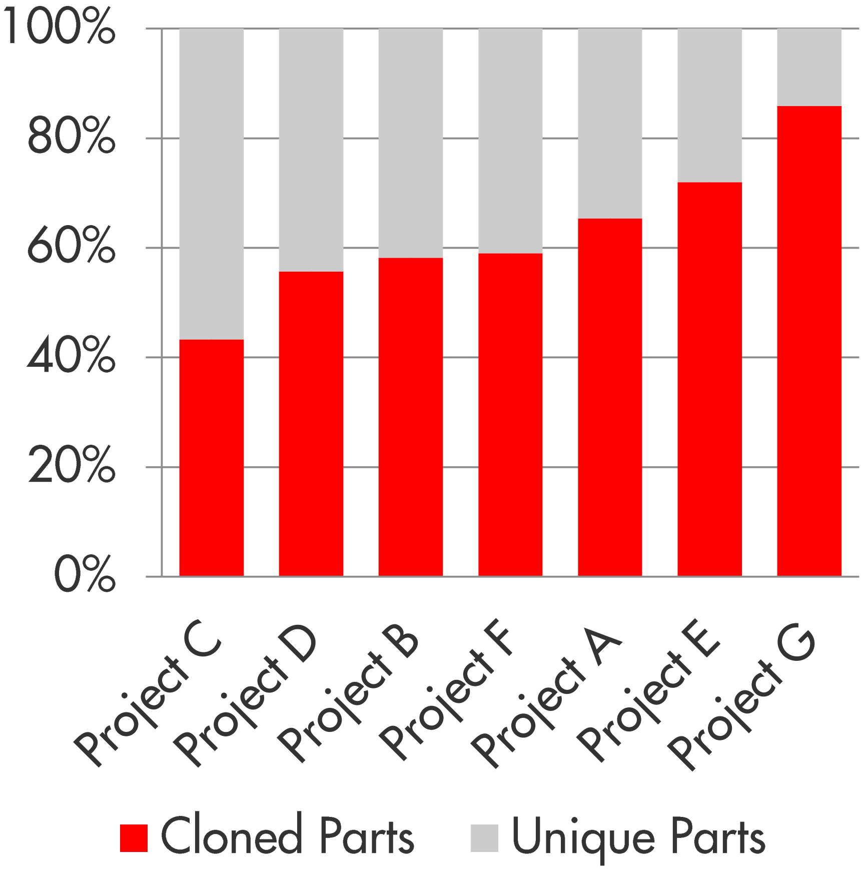
Tag	Unsigned Word	=	0x0000	all lines
		<>	0x0000	one special line
PosY	Unsigned Byte	<>	0x00	one special column (only if Tag <> 0x0000)
		=	0x01	not allowed, no access to Tag

The Tag belongs to the data field and is returned at the start of every line. PosY = 0x01 denotes the Tag. With respect to consistency, accesses to a column are not reasonable. As with PosX, the value 0x0000 for Tag is reserved to indicate the whole Array. Because of the unordered nature of the function class Map, no last line with a Tag value of 0xFFFF needs to be stored. However, 0xFFFF is used to indicate the end of the transmission of the whole Array.

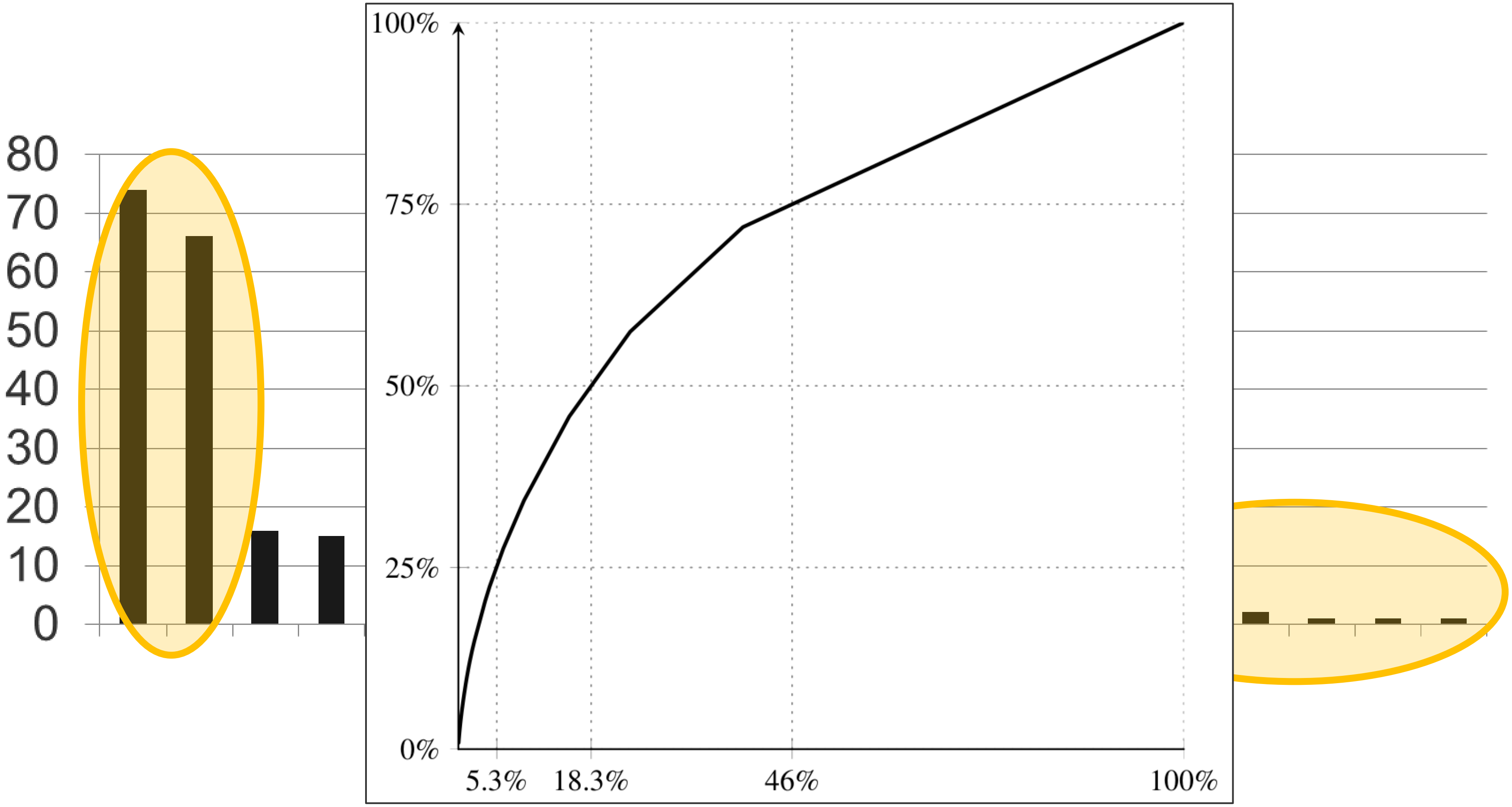

START_POS
Demux
Data Type Conversion1
Data Type Conversion2
x_pos
y_pos
abs
Relational Operator
Logic
Logical Operator
Constant
WIDTH/LENGTH
((LENGTH/2)^2 + (WIDTH/2)^2)^0.5
LENGTH/2
pi

Munich RE

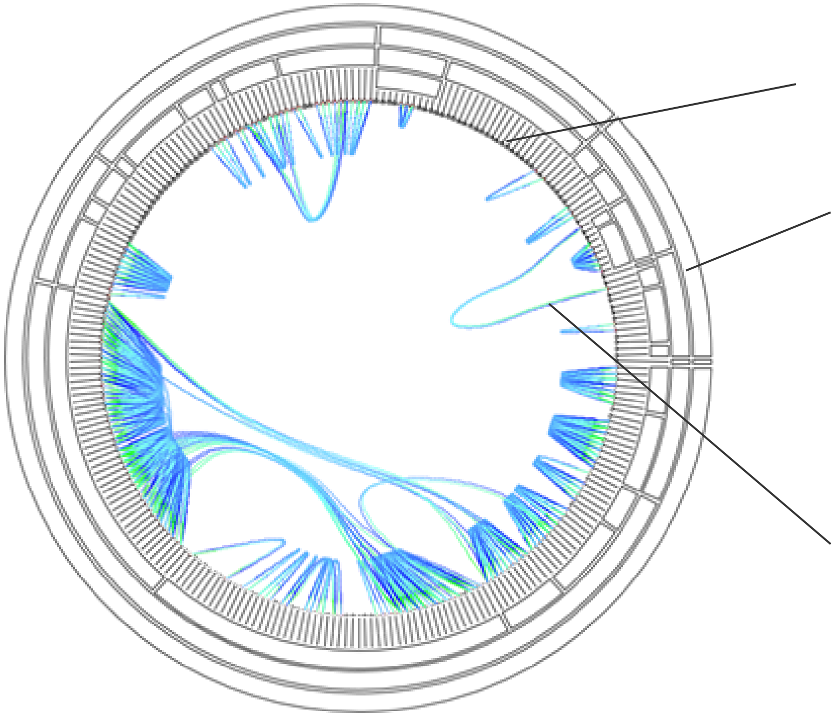
	System under Test	Test Suite
		# Tests
System A	330 kLoC	266
System B	580 kLoC	1,059
System C	150 kLoC	72
System D	430 kLoC	180
System E	760 kLoC	1,804
System F	1,400 kLoC	135
System G	160 kLoC	605



Hauptmann, Juergens et al: *Can Clone Detection Support Test Comprehension?*, ICPC 2012



Hauptmann, Juergens et al: *Can Clone Detection Support Test Comprehension?*, ICPC 2012

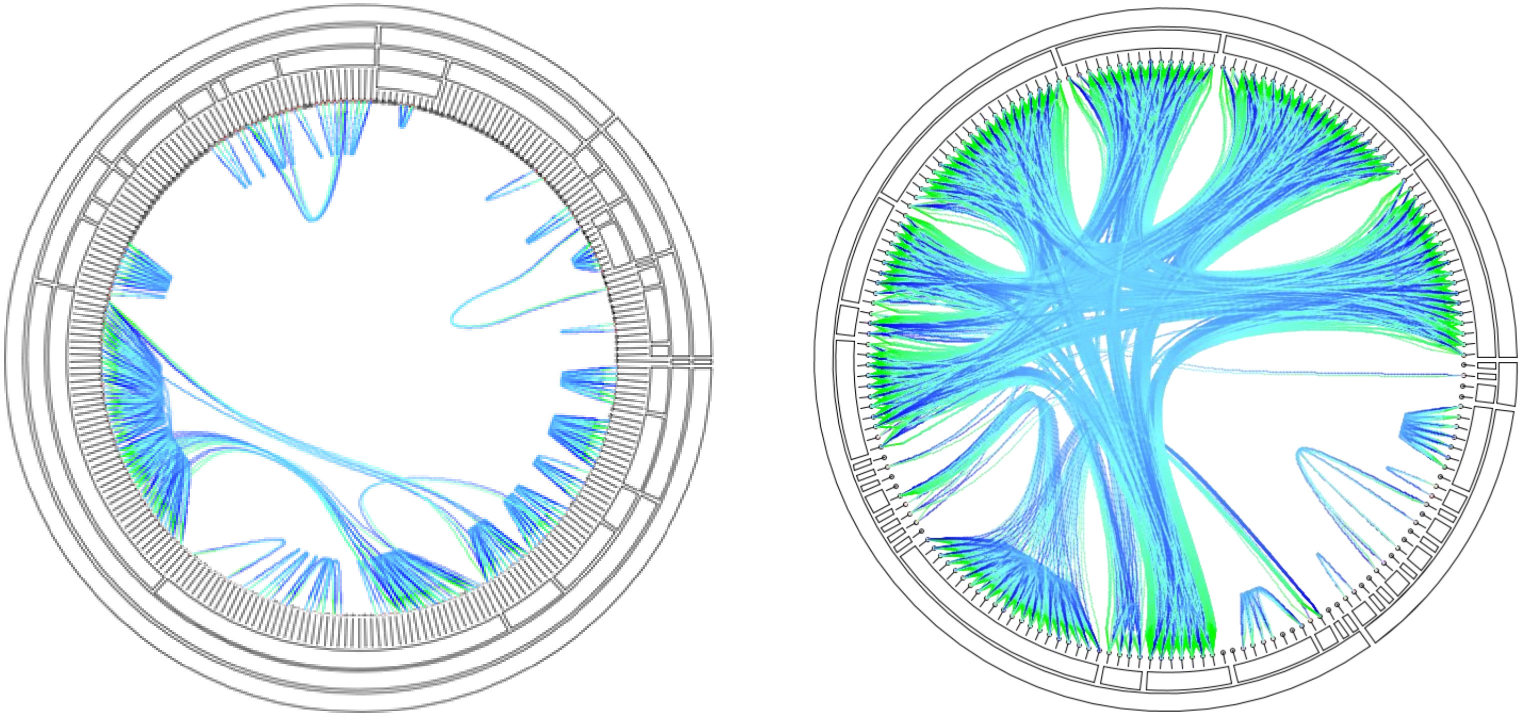


Testfallbeschreibung

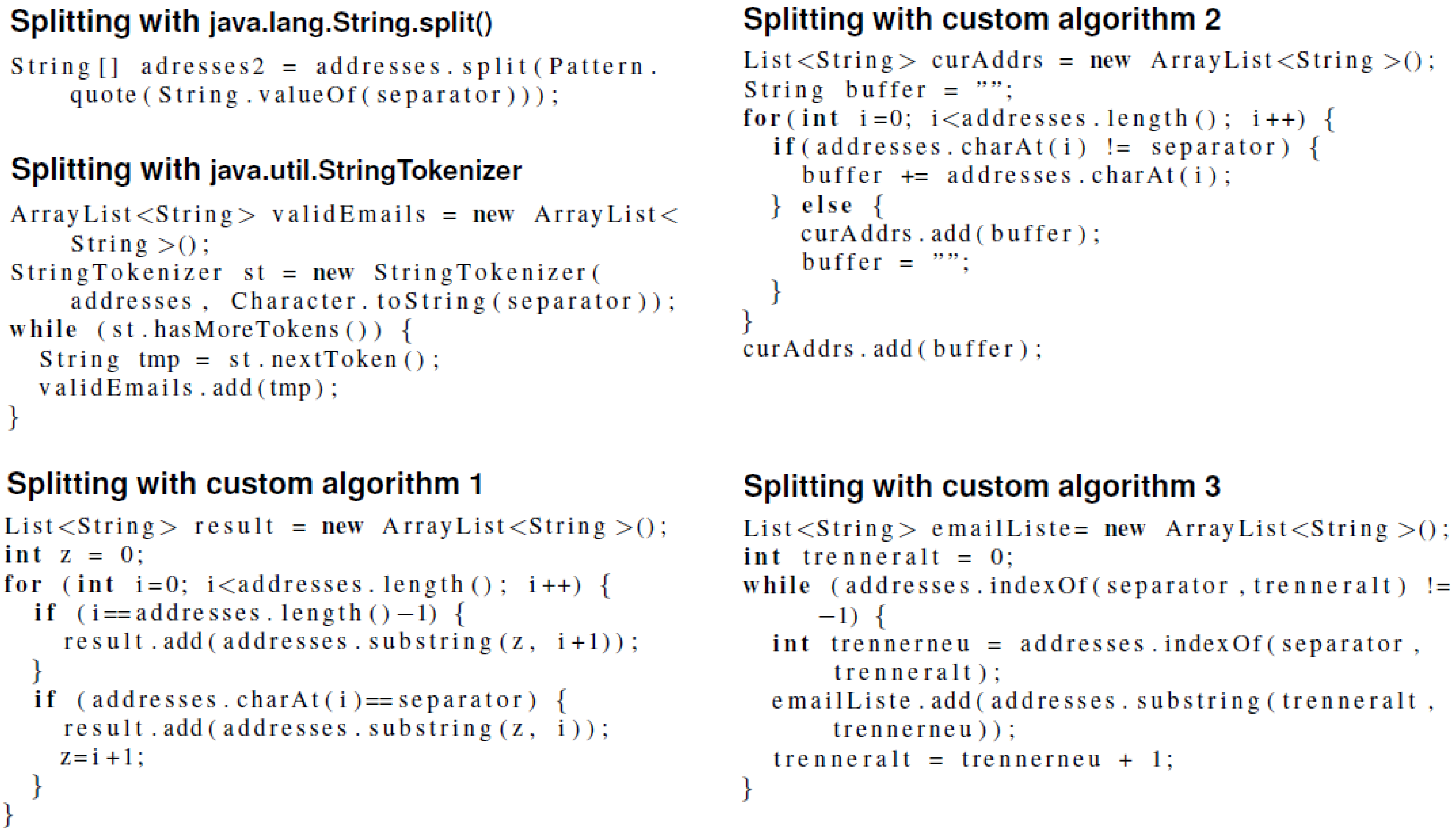
Hierarchie

Gemeinsamer Klon
zwischen zwei
Testfallbeschreibungen

Hauptmann, Juergens et al: *Can Clone Detection Support Test Comprehension?*, ICPC 2012



Hauptmann, Juergens et al: *Can Clone Detection Support Test Comprehension?*, ICPC 2012



Splitting with java.lang.String.split()

```
String[] adresses2 = 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;
}
```

Juergens et al: *Code Similarities Beyond Copy & Paste*, CSMR 2010

PASTE
PASTE
COPY
PAS
PASTE

Fazit

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

3.– 6. September 2012
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Dr. Elmar Juergens
juergens@cqse.eu