

Copy & Paste & Bug?

Die überraschende Wahrheit über Klone in Software-Artefakten

Dr. Elmar Jürgens



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



```

// 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();
}

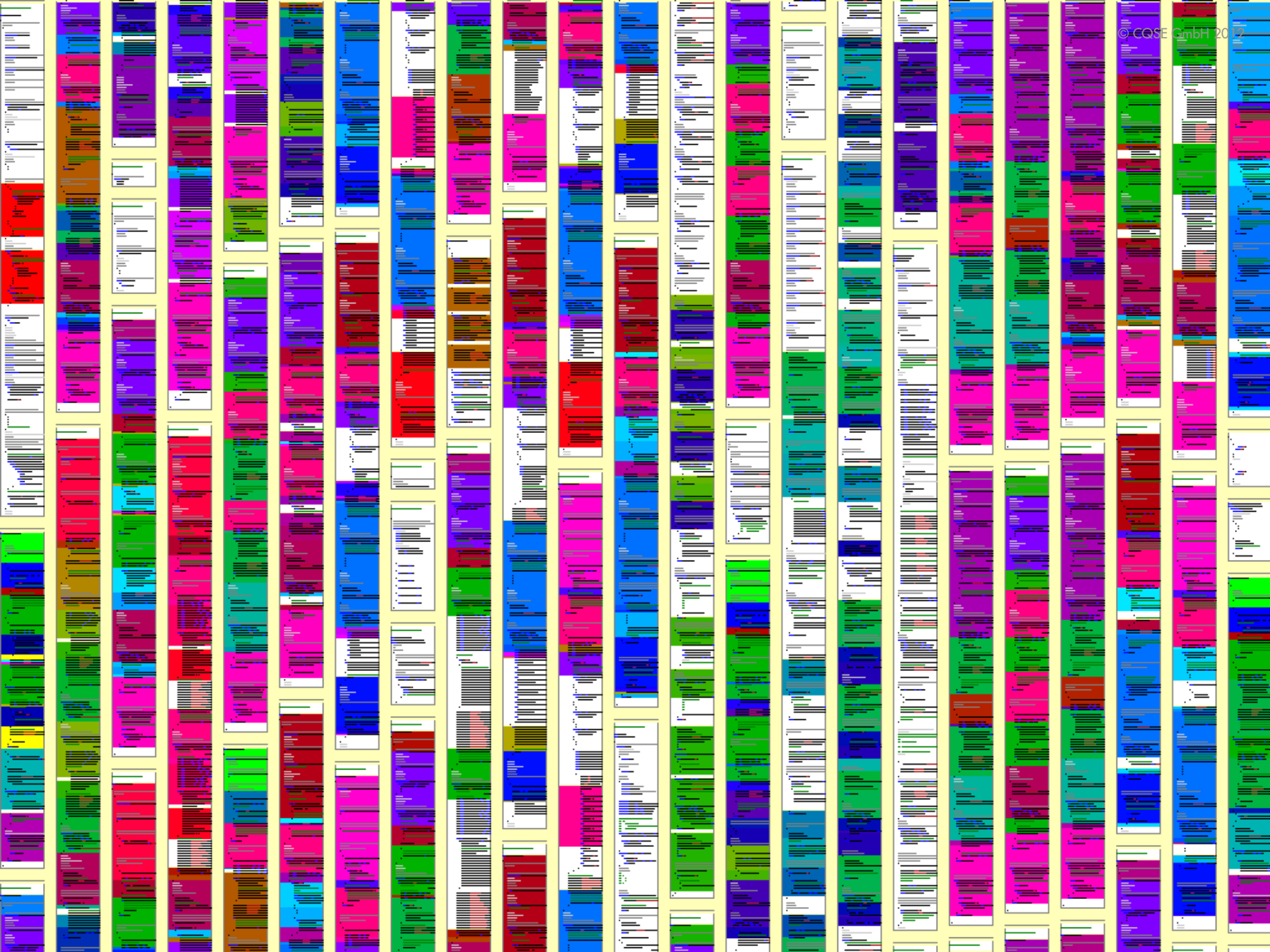
```

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









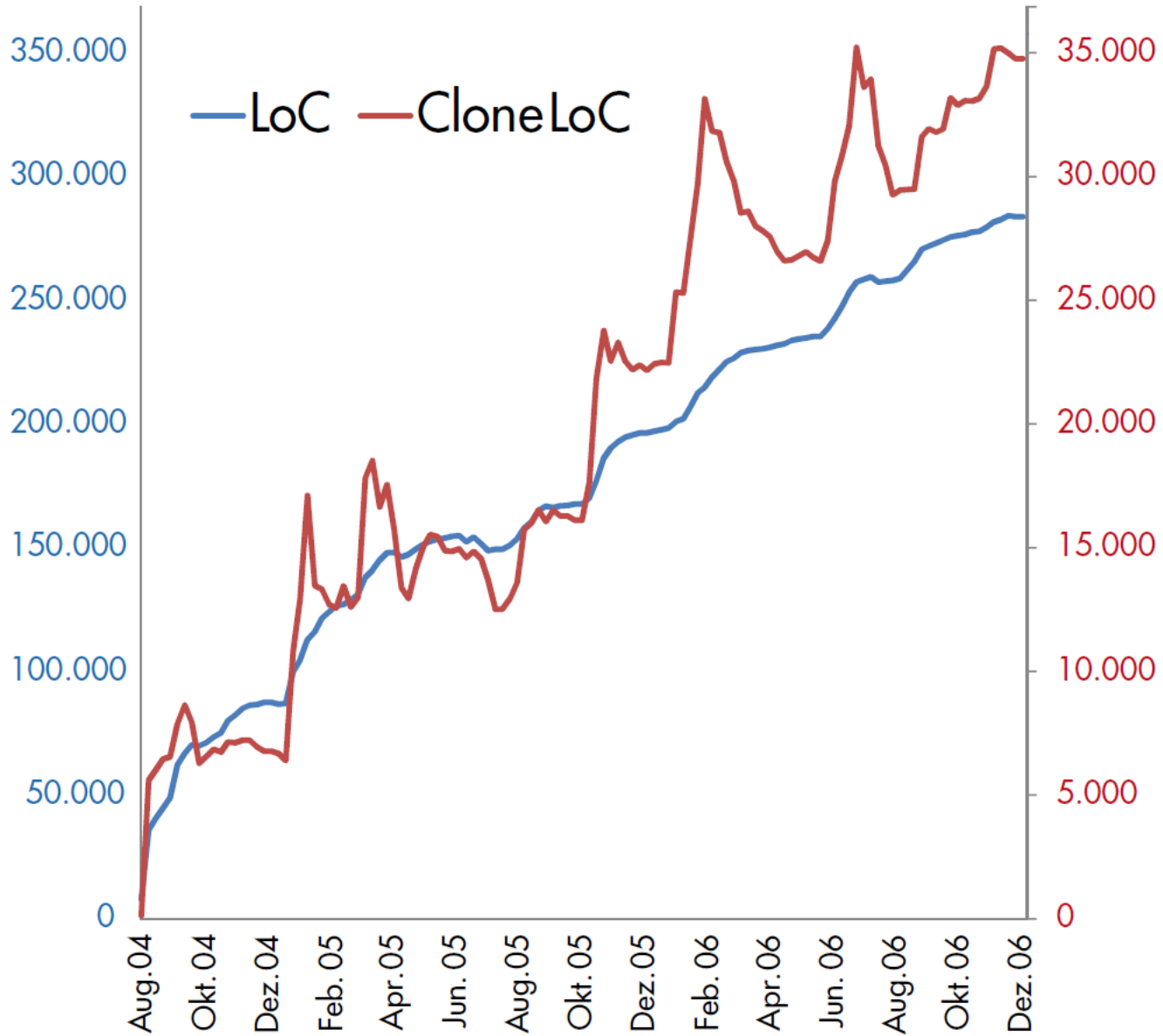
Studie

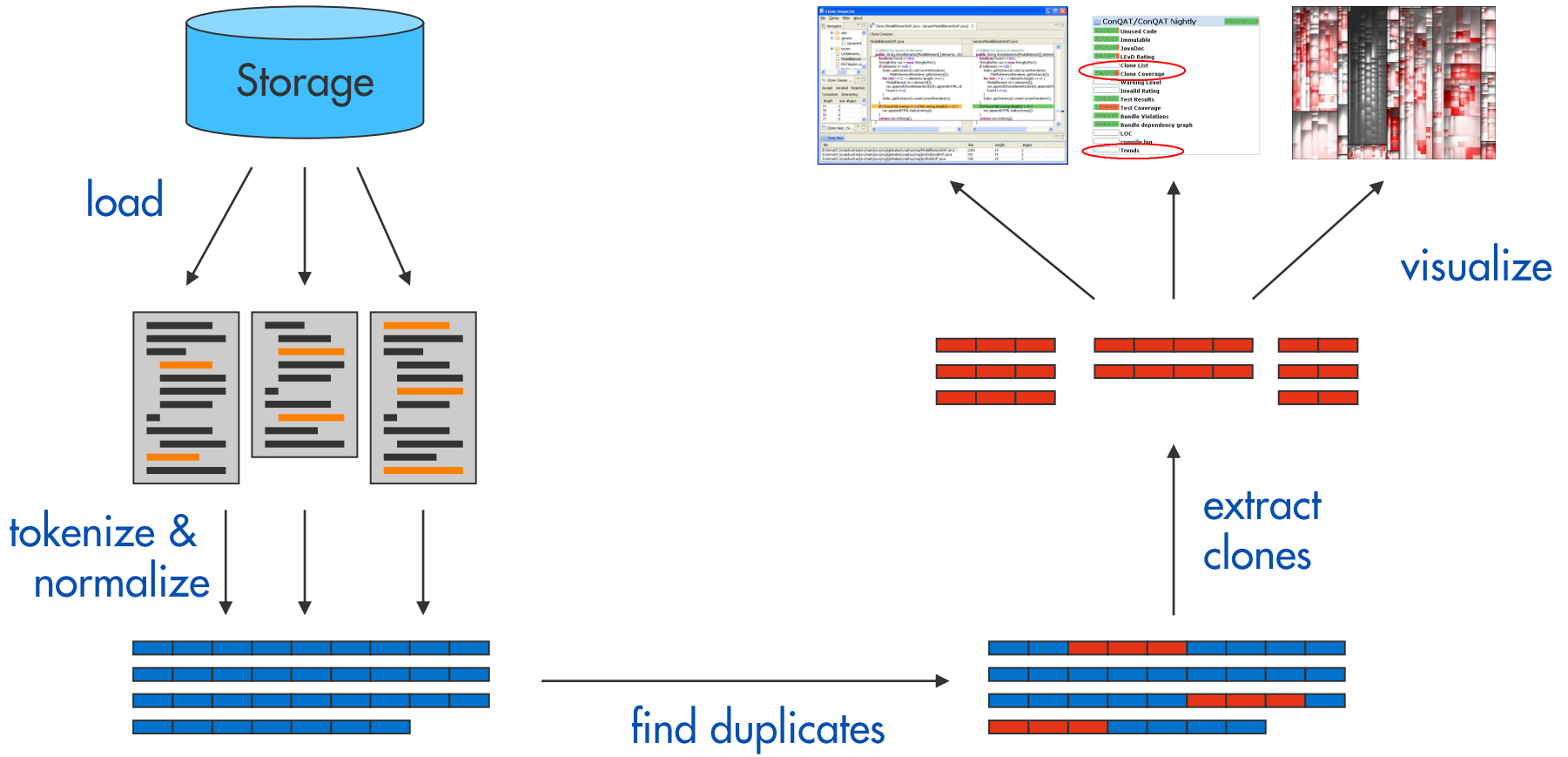


- Über 100 Fehler in produktiver Software



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);
}
```

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

Clone Coverage Trend



<pre>try { assertEquals(Double fail(); } catch (AssertionFail) }</pre>	<p>Sibling 3: FloatAssertTest.java(13-30)</p> <p>Focus on CloneClass: 12</p> <p>Select this clone instance: DoublePrecisionAssertTest.java(13-30)</p>
---	---

This image displays a highly complex and dense grid of small, multi-colored rectangular panels. Each panel contains a unique, intricate pattern of lines, dots, and shapes, resembling a highly detailed data visualization or a complex code structure. The panels are arranged in a regular grid, with each panel having a thin yellow border. The overall appearance is that of a large-scale data matrix or a complex code structure, possibly representing a large dataset or a complex system's state. The patterns within the panels are diverse, with some showing more regular, grid-like structures and others appearing more chaotic or abstract. The colors used in the patterns include black, white, red, green, blue, and purple, creating a vibrant and multi-colored display. The overall layout is a dense, multi-column and multi-row grid of these small, complex panels.

The image displays a 10x10 grid of 100 small panels. Each panel is a vertical rectangle with a distinct background color and a dense, repeating pattern of horizontal lines. The colors of the panels vary across the grid, including shades of blue, green, yellow, red, purple, and cyan. The lines within each panel are thin and closely spaced, creating a textured, barcode-like appearance. The overall layout is highly structured and repetitive, suggesting a data visualization or a complex pattern generation process.





Best Practice: Diskriminierung (von Code)

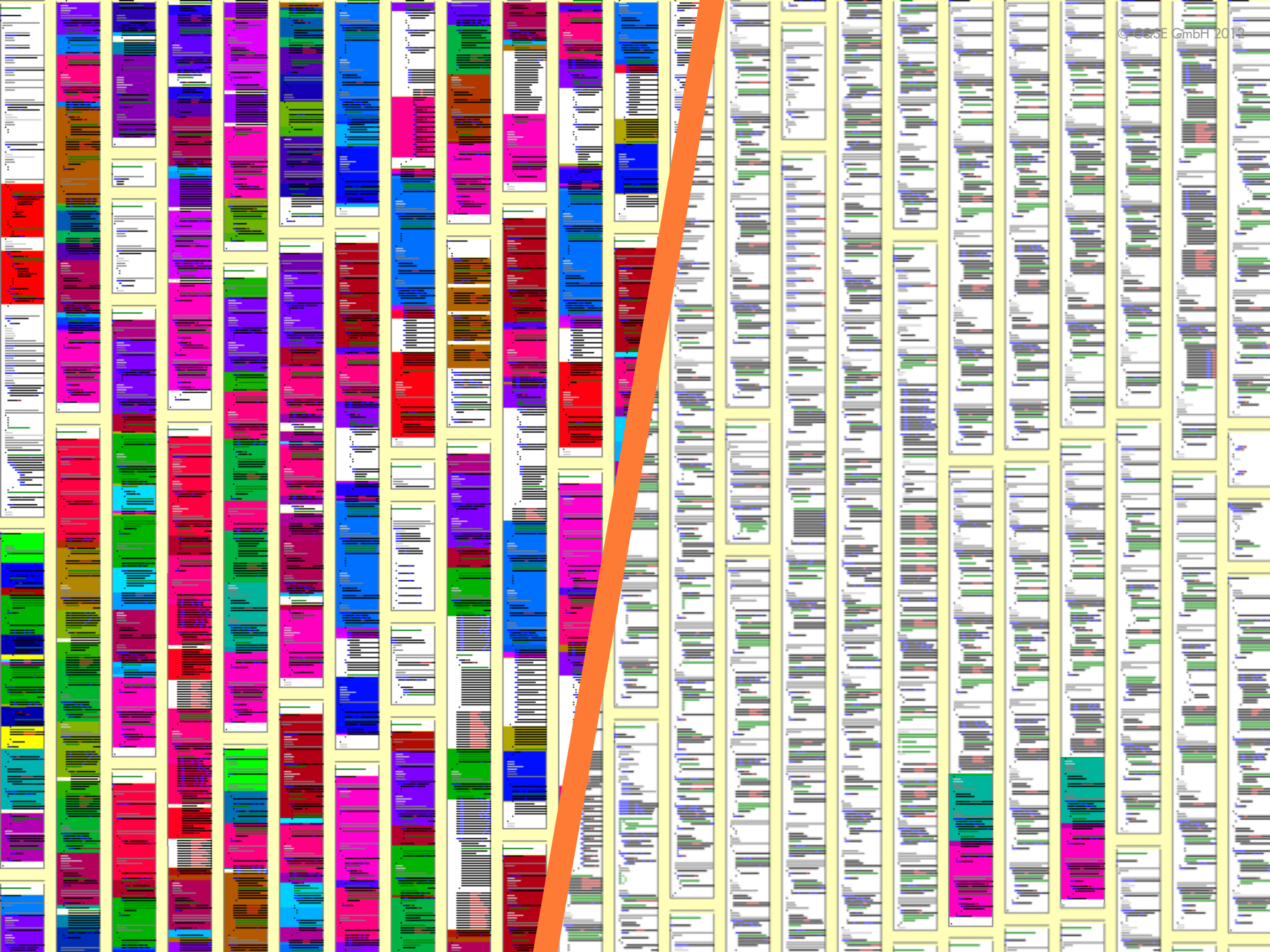
Art der Wartung

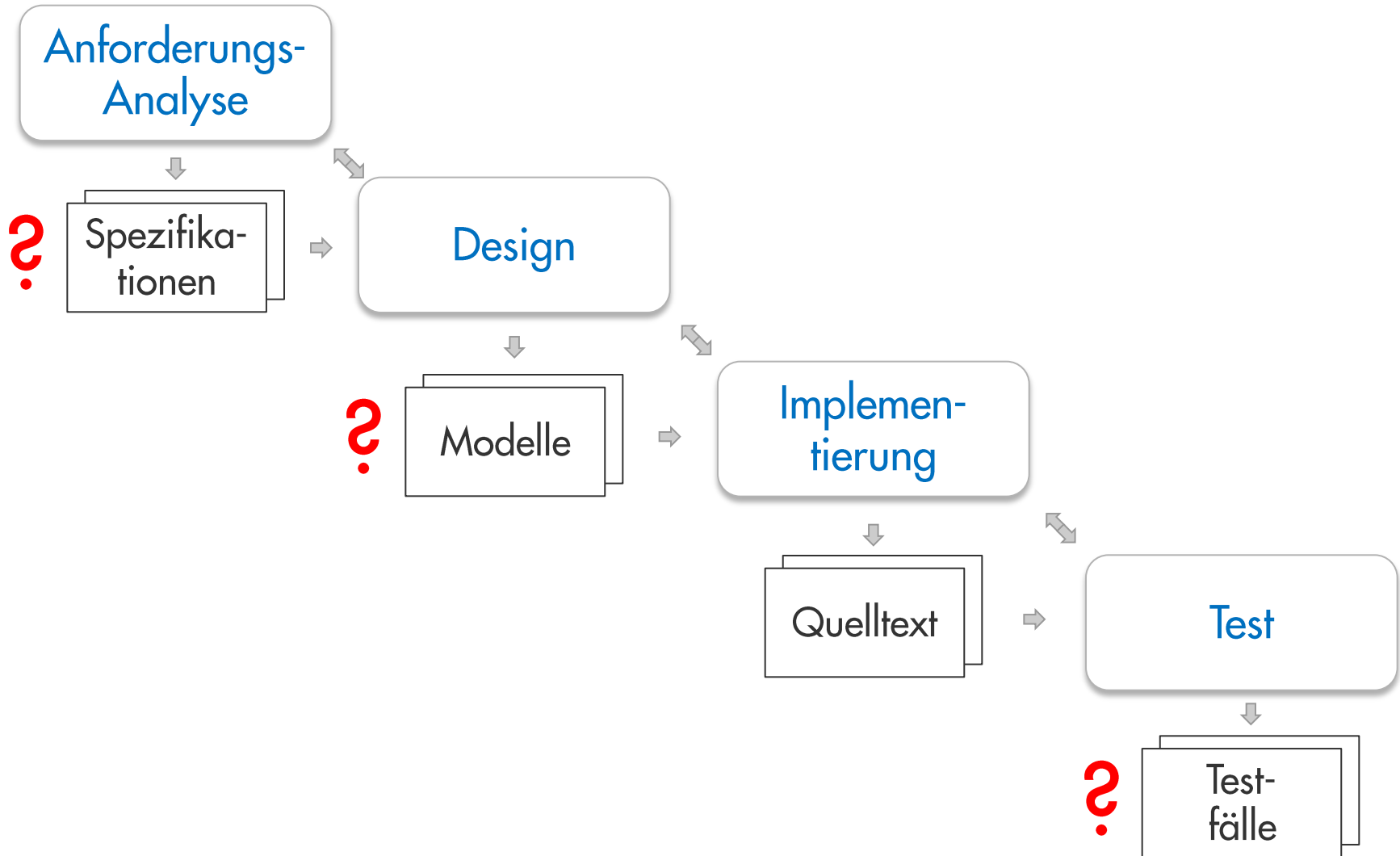
- Manuell
- Generator
- Überhaupt nicht (Wegwerf-Prototyp)

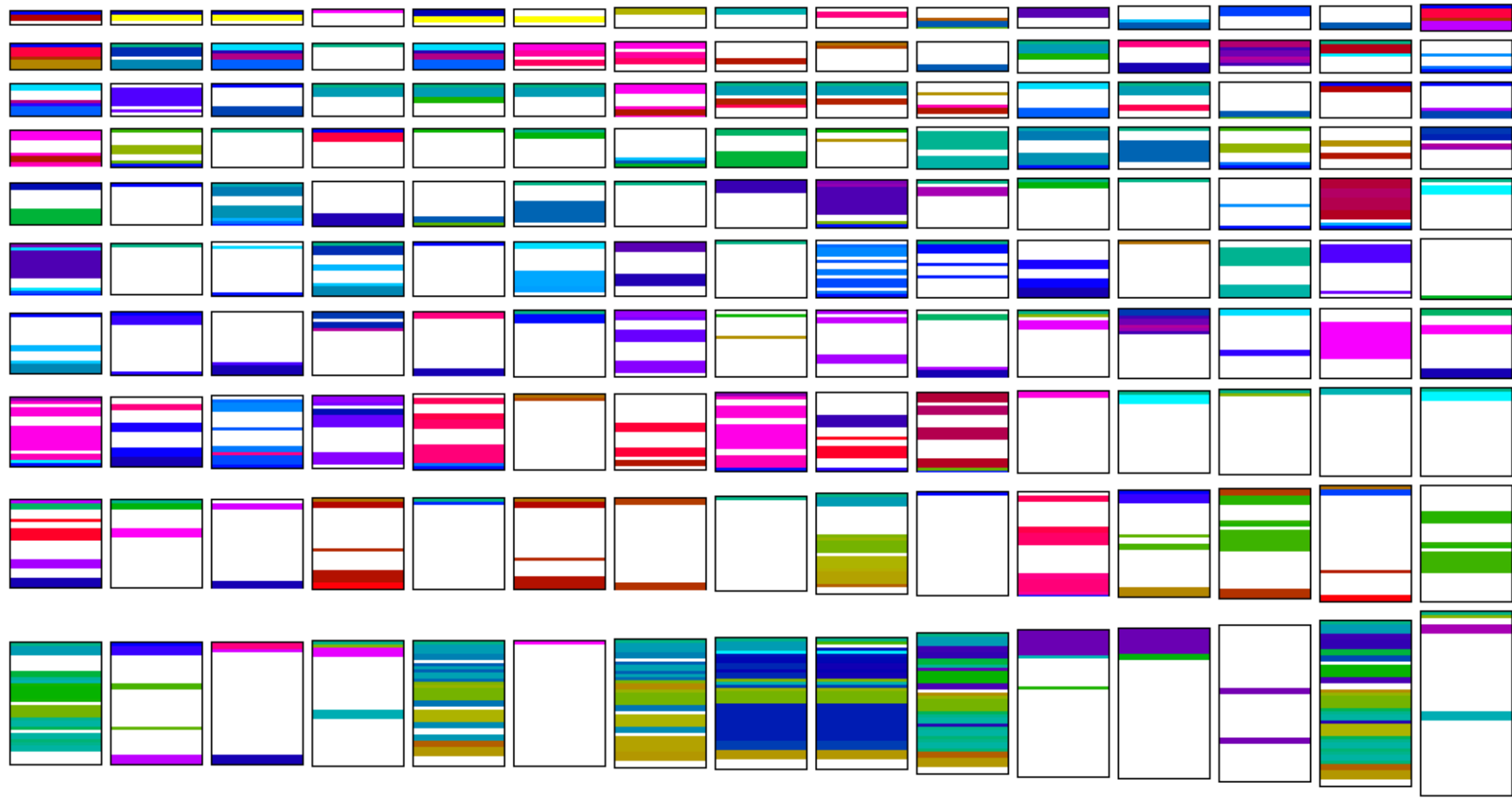
Aufgabe im Projekt

- Teil der Anwendung
- Test
- Hilfswerkzeug

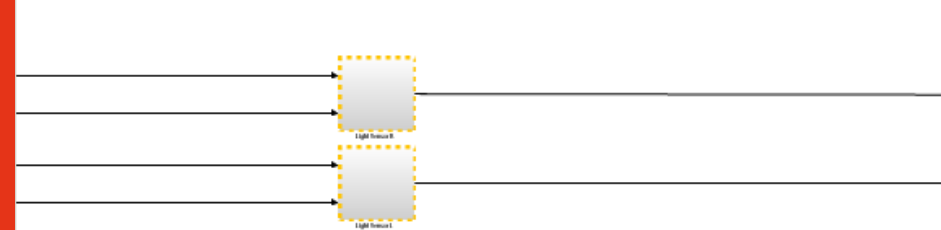
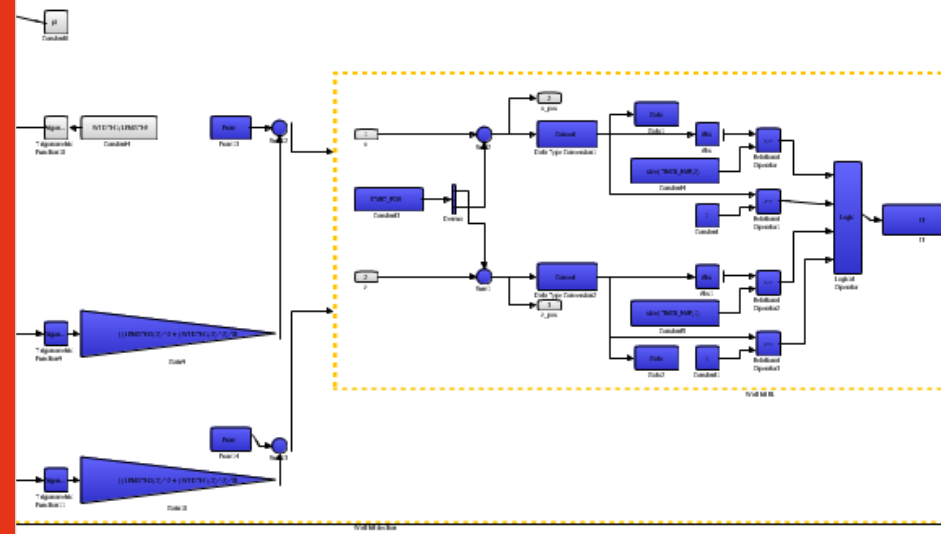
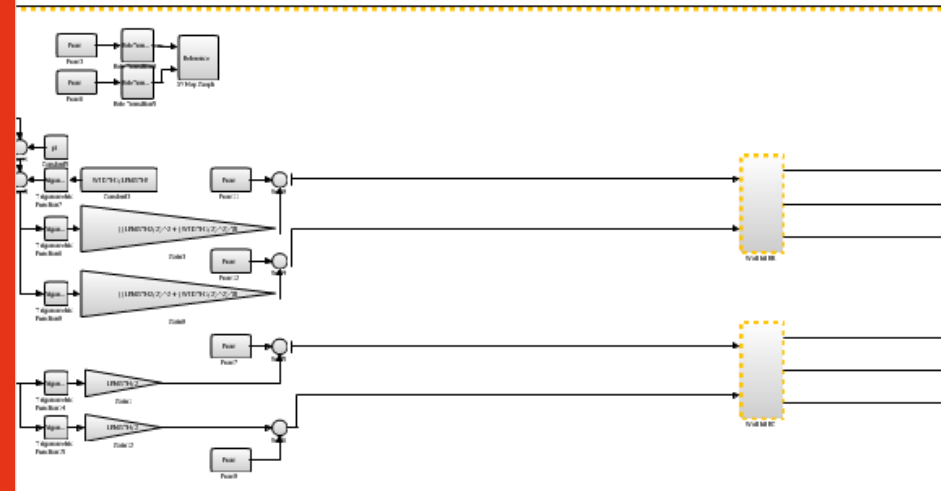
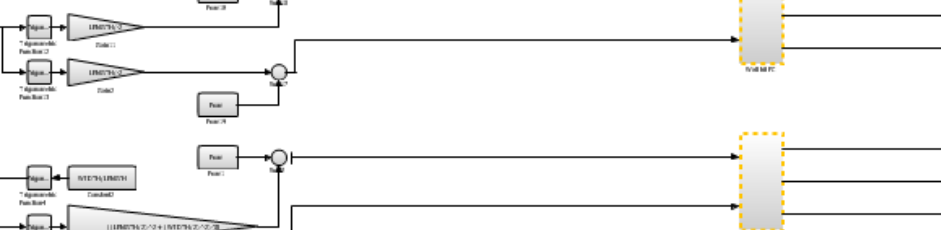
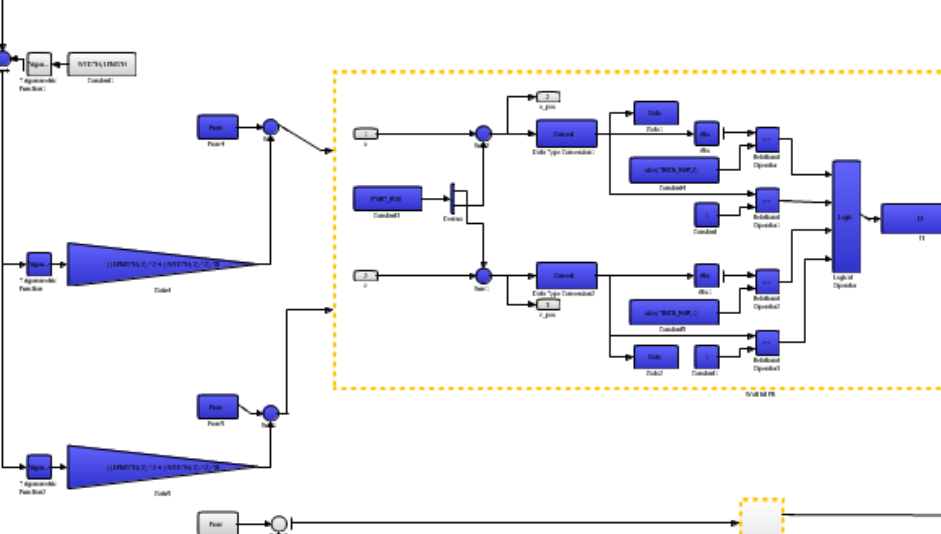
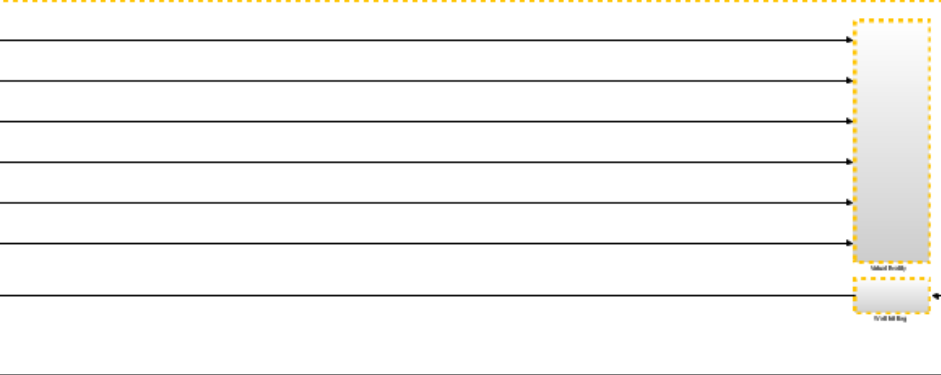
Mit manuell gewartetem Applikationscode beginnen







176 use cases in total, 150 contain cloning





PASTE

COPY

PASTE

PASTE

PASTE

Fazit

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

Kontakt

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

Ich bin heute und morgen hier und freue mich auf Diskussionen 😊

Ich spreche morgen um 17:00 in 303 über Peer-Reviews

CQSE GmbH
Lichtenbergstraße 8
85748 Garching bei München