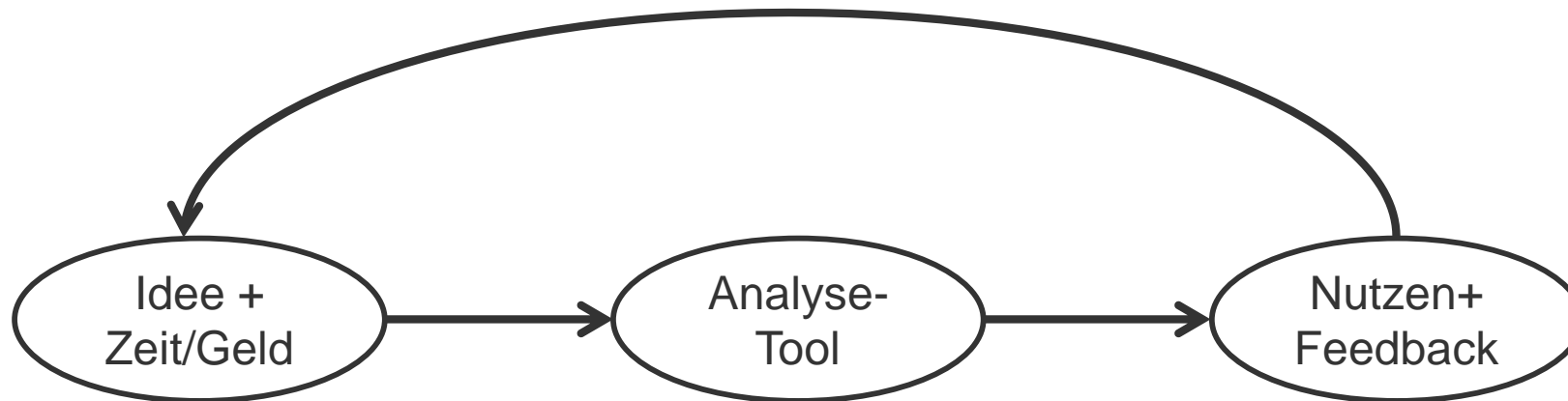
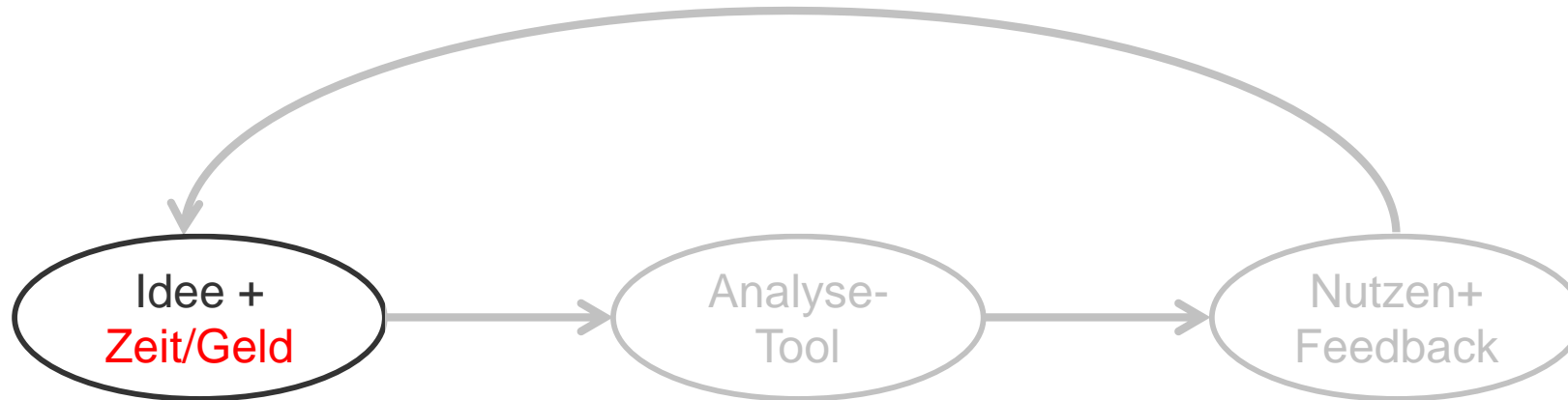
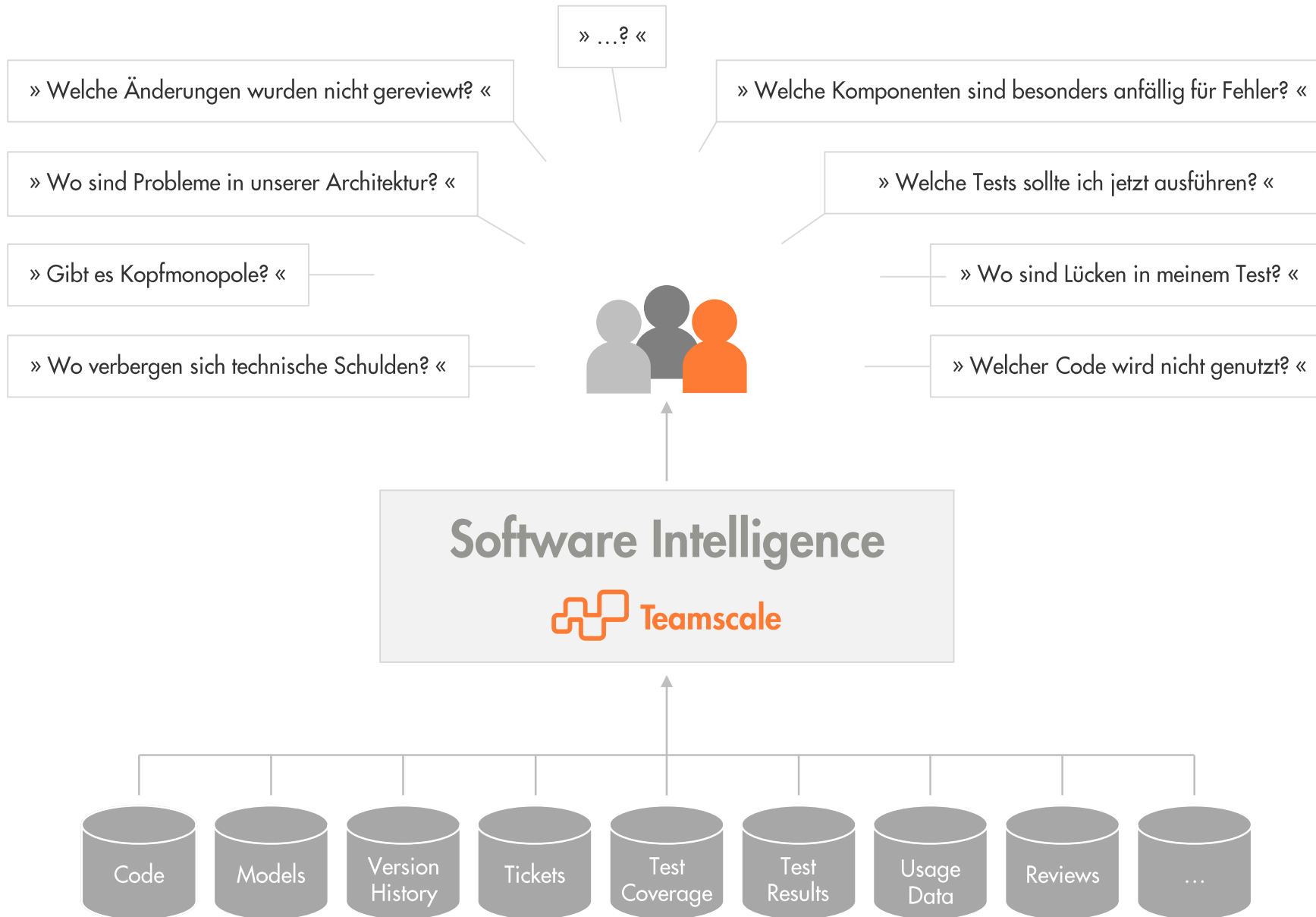


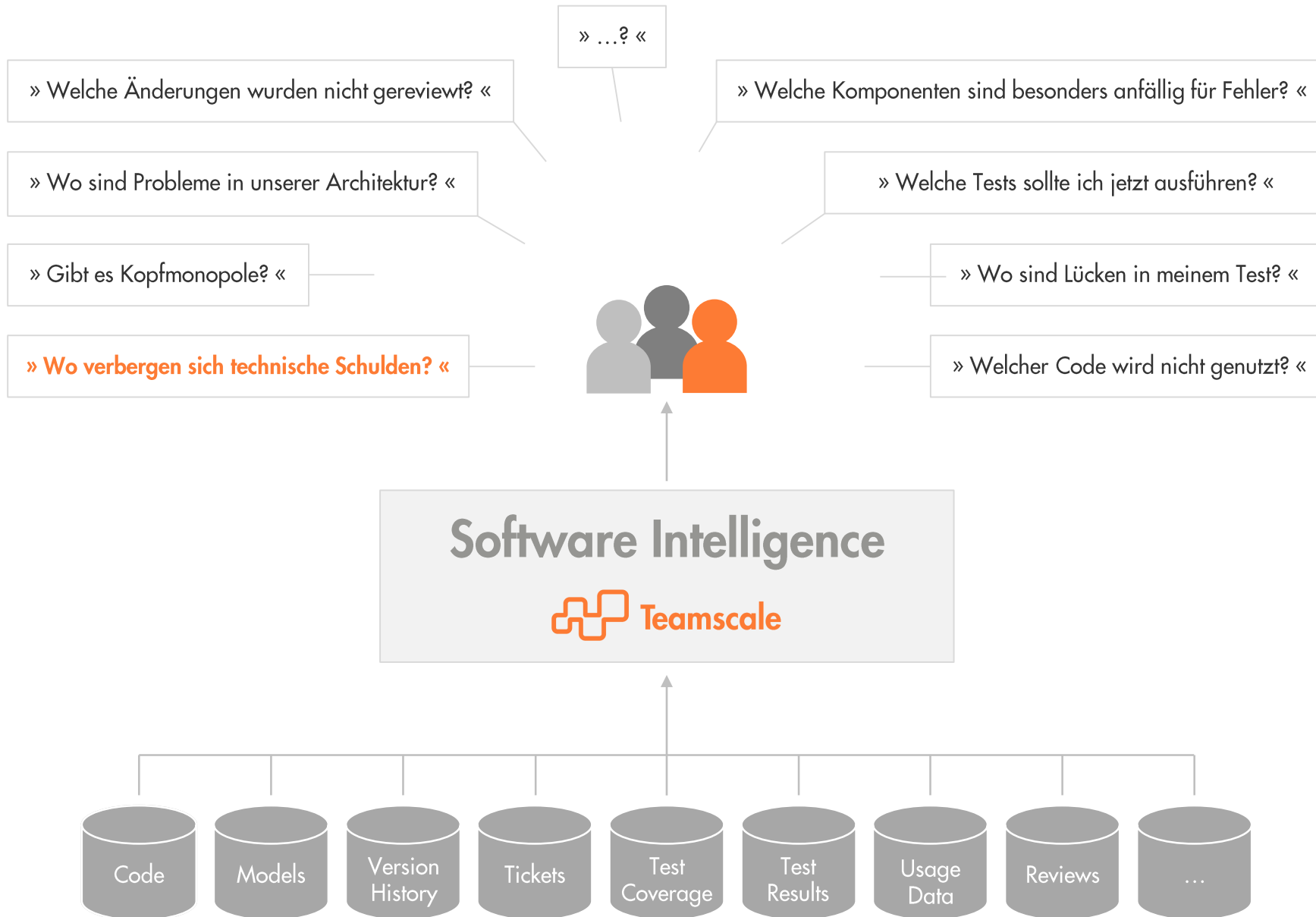
Meine Lehren aus 10 Jahren Forschung & Lehre als Säulen unseres Firmenwachstums









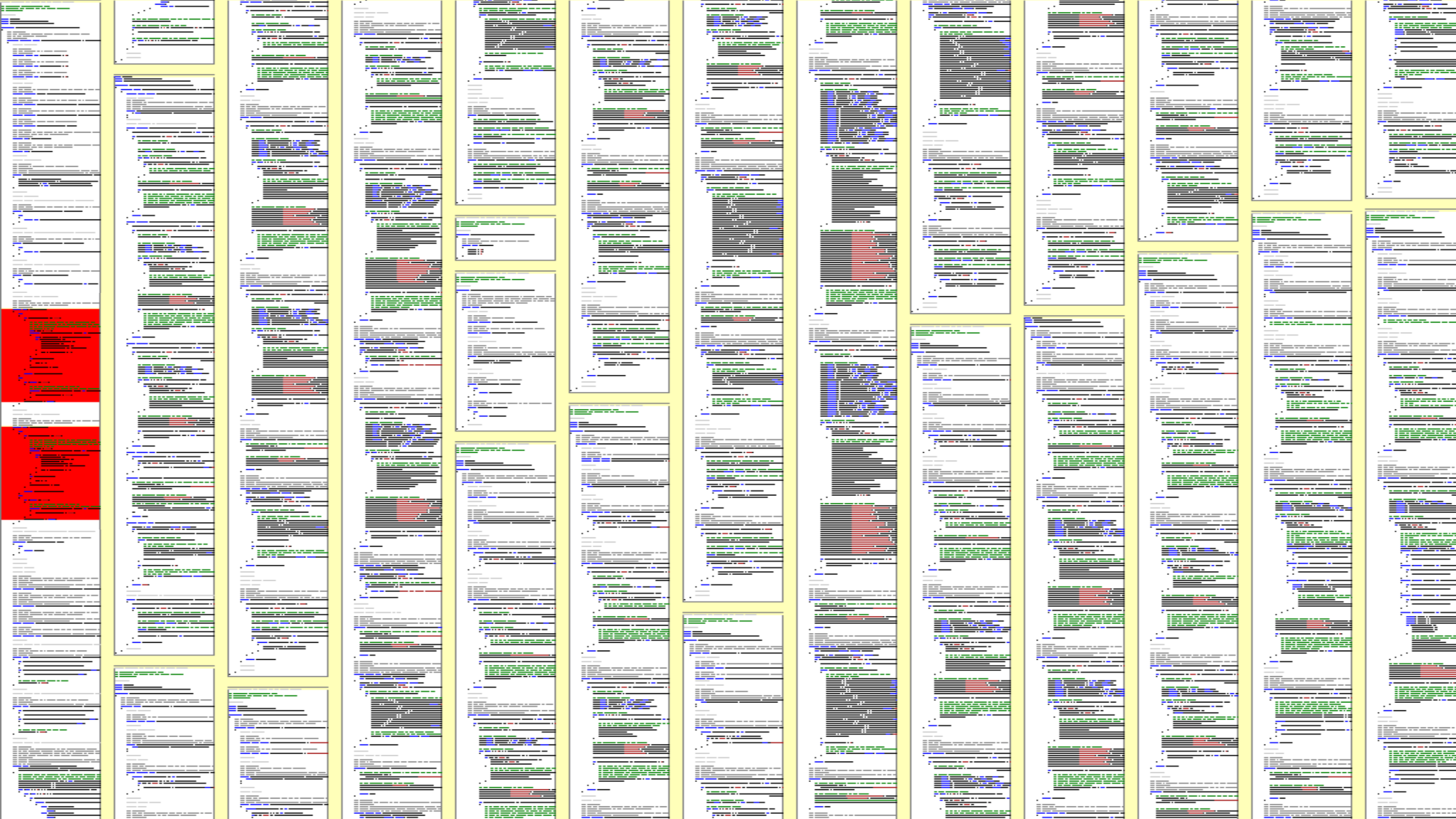


```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());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







Study













Munich Re  

- Over 100 bugs in productive Software



- 52% of all unintentional differences buggy

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

-  Dashboard
-  Activity
-  Findings
-  Metrics
-  Tests
-  Issues
-  Tasks
-  Architecture
-  Delta
-  Projects
-  System
-  Admin

jenkins/test/src/main/java/org/jvnet/hudson/test/HudsonTestCase.java

(revision 12d96a56...)

```

if (lhs==null && rhs==null) return;
if (lhs==null) fail("lhs is null while rhs="+rhs);
if (rhs==null) fail("rhs is null while lhs="+lhs);

Constructor<?> lc = findDataBoundConstructor(lhs.getClass());
Constructor<?> rc = findDataBoundConstructor(rhs.getClass());
assertEquals("Data bound constructor mismatch. Different type?",lc,rc);

List<String> primitiveProperties = new ArrayList<String>();

String[] names = ClassDescriptor.loadParameterNames(lc);
Class<?>[] types = lc.getParameterTypes();
assertEquals(names.length,types.length);
for (int i=0; i<types.length; i++) {
    Object lv = ReflectionUtils.getPublicProperty(lhs, names[i]);
    Object rv = ReflectionUtils.getPublicProperty(rhs, names[i]);

    if (Iterable.class.isAssignableFrom(types[i])) {
        Iterable lcol = (Iterable) lv;
        Iterable rcol = (Iterable) rv;
        Iterator ltr,rtr;
        for (ltr=lcol.iterator(), rtr=rcol.iterator(); ltr.hasNext() && rtr.hasNext();){
            Object litem = ltr.next();
            Object ritem = rtr.next();

            if (findDataBoundConstructor(litem.getClass())!=null) {
                assertEqualDataBoundBeans(litem,ritem);
            } else {
                assertEquals(litem,ritem);
            }
        }
        assertFalse("collection size mismatch between "+lhs+" and "+rhs, ltr.hasNext() ^
    } else
    if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructo
        // recurse into nested databound objects
        assertEqualDataBoundBeans(lv,rv);
    } else {
        primitiveProperties.add(names[i]);
    }
}

// compare shallow primitive properties
if (!primitiveProperties.isEmpty())
    assertEqualBeans(lhs,rhs,Util.join(primitiveProperties,""));

*
Makes sure that two collections are identical via {@link #assertEqualDataBoundBeans(Object,
/
public void assertEqualDataBoundBeans(List<?> lhs, List<?> rhs) throws Exception {
    assertEquals(lhs.size(), rhs.size());
}
    
```

jenkins/test/src/main/java/org/jvnet/hudson/test/JenkinsRule.java

(revision 3909f5ac...)

```

if (lhs==null && rhs==null) return;
if (lhs==null) fail("lhs is null while rhs="+rhs);
if (rhs==null) fail("rhs is null while lhs="+lhs);

Constructor<?> lc = findDataBoundConstructor(lhs.getClass());
Constructor<?> rc = findDataBoundConstructor(rhs.getClass());
assertThat("Data bound constructor mismatch. Different type?", (Constructor)rc, is((Cons

List<String> primitiveProperties = new ArrayList<String>();

String[] names = ClassDescriptor.loadParameterNames(lc);
Class<?>[] types = lc.getParameterTypes();
assertThat(types.length, is(names.length));
for (int i=0; i<types.length; i++) {
    Object lv = ReflectionUtils.getPublicProperty(lhs, names[i]);
    Object rv = ReflectionUtils.getPublicProperty(rhs, names[i]);

    if (Iterable.class.isAssignableFrom(types[i])) {
        Iterable lcol = (Iterable) lv;
        Iterable rcol = (Iterable) rv;
        Iterator ltr,rtr;
        for (ltr=lcol.iterator(), rtr=rcol.iterator(); ltr.hasNext() && rtr.hasNext();){
            Object litem = ltr.next();
            Object ritem = rtr.next();

            if (findDataBoundConstructor(litem.getClass())!=null) {
                assertEqualDataBoundBeans(litem,ritem);
            } else {
                assertThat(ritem, is(litem));
            }
        }
        assertThat("collection size mismatch between " + lhs + " and " + rhs, ltr.hasNext()
            is(false));
    } else
    if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructo
        // recurse into nested databound objects
        assertEqualDataBoundBeans(lv,rv);
    } else {
        primitiveProperties.add(names[i]);
    }
}

// compare shallow primitive properties
if (!primitiveProperties.isEmpty())
    assertEqualBeans(lhs,rhs,Util.join(primitiveProperties,""));

*
Makes sure that two collections are identical via {@link #assertEqualDataBoundBeans(Object,
/
public void assertEqualDataBoundBeans(List<?> lhs, List<?> rhs) throws Exception {
    assertEquals(lhs.size(), rhs.size());
}
    
```



Xcode



```

BusinessRuleTaskXMLConverter.java  DefinitionsParser.java
model.setTargetNamespace(xtr.getAttributeValue(null, TARGET_NAMESPACE_ATTRIBUTE));
for (int i = 0; i < xtr.getNamespaceCount(); i++) {
    String prefix = xtr.getNamespacePrefix(i);
    if (StringUtils.isEmpty(prefix)) {
        model.addNamespace(prefix, xtr.getNamespaceURI(i));
    }
}

for (int i = 0; i < xtr.getAttributeCount(); i++) {
    ExtensionAttribute extensionAttribute = new ExtensionAttribute();
    extensionAttribute.setName(xtr.getAttributeLocalName(i));
    extensionAttribute.setValue(xtr.getAttributeValue(i));
    if (StringUtils.isEmpty(xtr.getAttributeNamespace(i))) {
        extensionAttribute.setNamespace(xtr.getAttributeNamespace(i));
    }
    if (StringUtils.isEmpty(xtr.getAttributePrefix(i))) {
        extensionAttribute.setNamespacePrefix(xtr.getAttributePrefix(i));
    }
    if (!BpmnXMLUtil.isBlacklisted(extensionAttribute, defaultAttributes)) {
        model.addDefinitionsAttribute(extensionAttribute);
    }
}

```

Clone with 2 instances of length 10

Problems @ Javadoc Declaration Findings Orphans View Properties

4 items

Description	Group	Category	Resource	Location
Clone with 2 instances of length 10	Code Duplica...	Cloning	DefinitionsPa...	47-60
Interface comment missing	Documentati...	Comment co...	DefinitionsPa...	29
Interface comment missing	Documentati...	Comment co...	DefinitionsPa...	38
Name violates naming convention: defaultAttributes. Should be	Naming	Java naming ...	DefinitionsPa...	31



Eclipse



Visual Studio



IntelliJ



Netbeans





- Dashboard
- Activity
- Findings
- Metrics
- Tests
- Issues
- Tasks
- Architecture
- Delta
- Projects
- System
- Admin

jenkins/test/src/main/java/org/jvnet/hudson/test/HudsonTestCase.java

(revision 12d96a56...)

```

if (lhs==null && rhs==null) return;
if (lhs==null) fail("lhs is null while rhs="+rhs);
if (rhs==null) fail("rhs is null while lhs="+lhs);

Constructor<?> lc = findDataBoundConstructor(lhs.getClass());
Constructor<?> rc = findDataBoundConstructor(rhs.getClass());
assertEquals("Data bound constructor mismatch. Different type?",lc,rc);

List<String> primitiveProperties = new ArrayList<String>();

String[] names = ClassDescriptor.loadParameterNames(lc);
Class<?>[] types = lc.getParameterTypes();
assertEquals(names.length,types.length);
for (int i=0; i<types.length; i++) {
    Object lv = ReflectionUtils.getPublicProperty(lhs, names[i]);
    Object rv = ReflectionUtils.getPublicProperty(rhs, names[i]);

    if (Iterable.class.isAssignableFrom(types[i])) {
        Iterable lcol = (Iterable) lv;
        Iterable rcol = (Iterable) rv;
        Iterator ltr,rtr;
        for (ltr=lcol.iterator(), rtr=rcol.iterator(); ltr.hasNext() && rtr.hasNext();){
            Object litem = ltr.next();
            Object ritem = rtr.next();

            if (findDataBoundConstructor(litem.getClass())!=null) {
                assertEqualsDataBoundBeans(litem,ritem);
            } else {
                assertEquals(litem,ritem);
            }
        }
        assertFalse("collection size mismatch between "+lhs+" and "+rhs, ltr.hasNext() ^
    } else
    if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructo
        // recurse into nested databound objects
        assertEqualsDataBoundBeans(lv,rv);
    } else {
        primitiveProperties.add(names[i]);
    }
}

// compare shallow primitive properties
if (!primitiveProperties.isEmpty())
    assertEqualsBeans(lhs,rhs,Util.join(primitiveProperties,""));

*
Makes sure that two collections are identical via {@link #assertEqualDataBoundBeans(Object
/
public void assertEqualsDataBoundBeans(List<?> lhs, List<?> rhs) throws Exception {
    assertEquals(lhs.size(), rhs.size());
}
    
```

jenkins/test/src/main/java/org/jvnet/hudson/test/JenkinsRule.java

(revision 3909f5ac...)

```

if (lhs==null && rhs==null) return;
if (lhs==null) fail("lhs is null while rhs="+rhs);
if (rhs==null) fail("rhs is null while lhs="+lhs);

Constructor<?> lc = findDataBoundConstructor(lhs.getClass());
Constructor<?> rc = findDataBoundConstructor(rhs.getClass());
assertThat("Data bound constructor mismatch. Different type?", (Constructor)rc, is((Cons

List<String> primitiveProperties = new ArrayList<String>();

String[] names = ClassDescriptor.loadParameterNames(lc);
Class<?>[] types = lc.getParameterTypes();
assertThat(types.length, is(names.length));
for (int i=0; i<types.length; i++) {
    Object lv = ReflectionUtils.getPublicProperty(lhs, names[i]);
    Object rv = ReflectionUtils.getPublicProperty(rhs, names[i]);

    if (lv != null && rv != null && Iterable.class.isAssignableFrom(types[i])) {
        Iterable lcol = (Iterable) lv;
        Iterable rcol = (Iterable) rv;
        Iterator ltr,rtr;
        for (ltr=lcol.iterator(), rtr=rcol.iterator(); ltr.hasNext() && rtr.hasNext();){
            Object litem = ltr.next();
            Object ritem = rtr.next();

            if (findDataBoundConstructor(litem.getClass())!=null) {
                assertEqualsDataBoundBeans(litem,ritem);
            } else {
                assertThat(ritem, is(litem));
            }
        }
        assertThat("collection size mismatch between " + lhs + " and " + rhs, ltr.hasNext()
            is(false));
    } else
    if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructo
        // recurse into nested databound objects
        assertEqualsDataBoundBeans(lv,rv);
    } else {
        primitiveProperties.add(names[i]);
    }
}

// compare shallow primitive properties
if (!primitiveProperties.isEmpty())
    assertEqualsBeans(lhs,rhs,Util.join(primitiveProperties,""));

*
Makes sure that two collections are identical via {@link #assertEqualDataBoundBeans(Object
/
public void assertEqualsDataBoundBeans(List<?> lhs, List<?> rhs) throws Exception {
    assertEquals(lhs.size(), rhs.size());
}
    
```




Fix a NPE while trying to compare two data bound beans with null Iterable field.

by [Vincent Latombe](#) in revision [3909f5ac](#) in branch [Vlatombe/jenkinsrule_compare_null_field_list](#) (repository1)

⌵ Affected files (2)

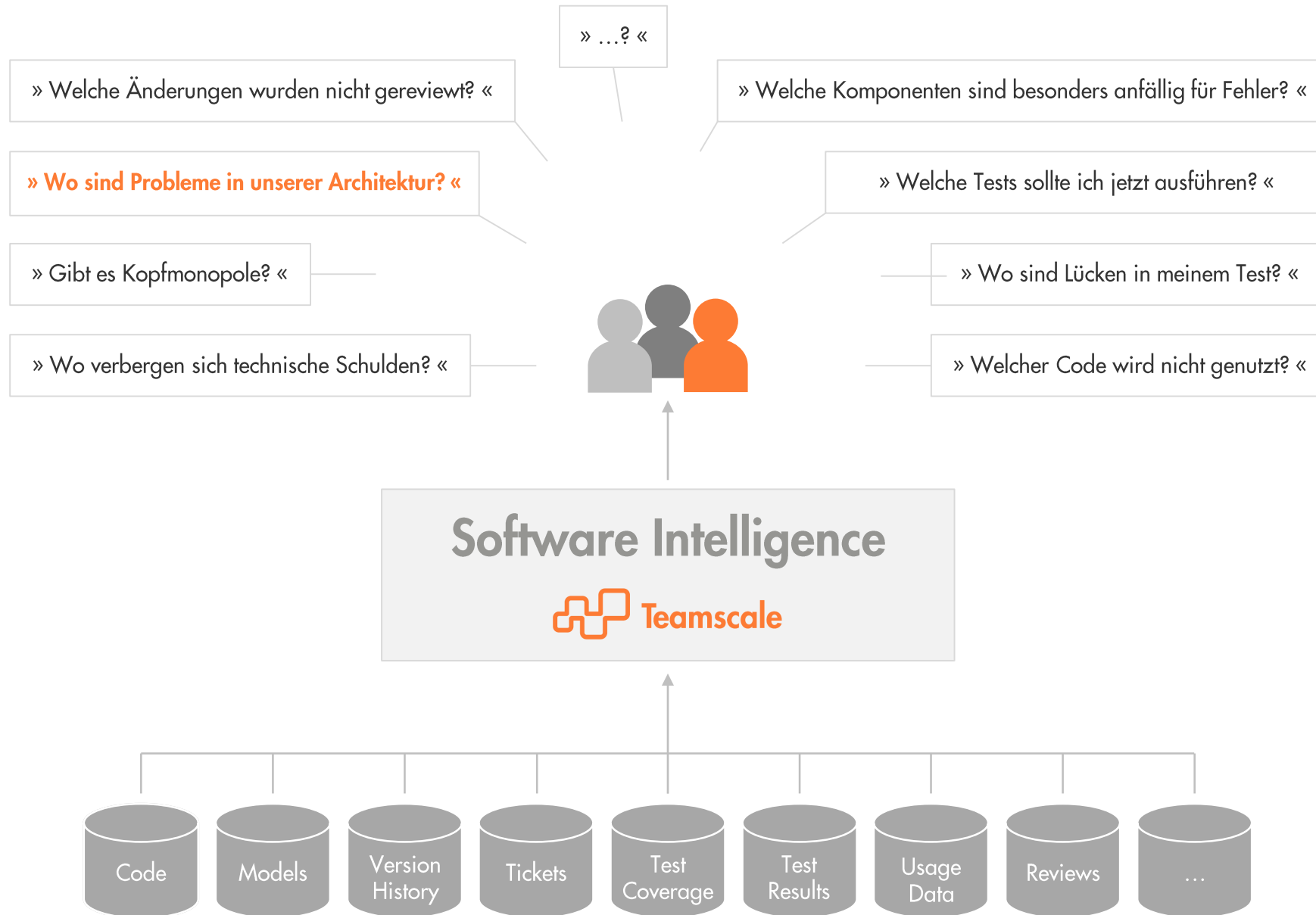
1 Alerts

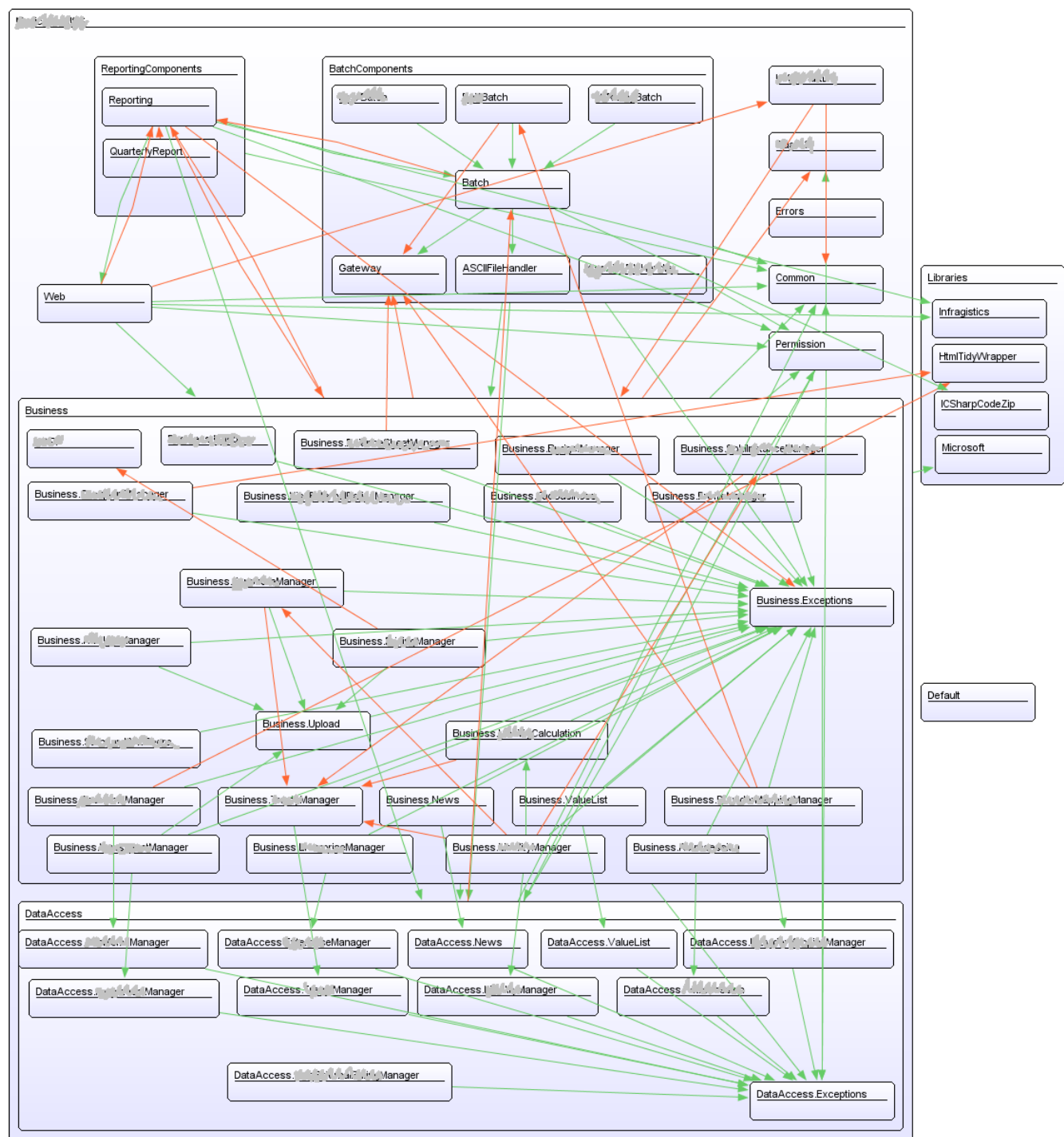
Message

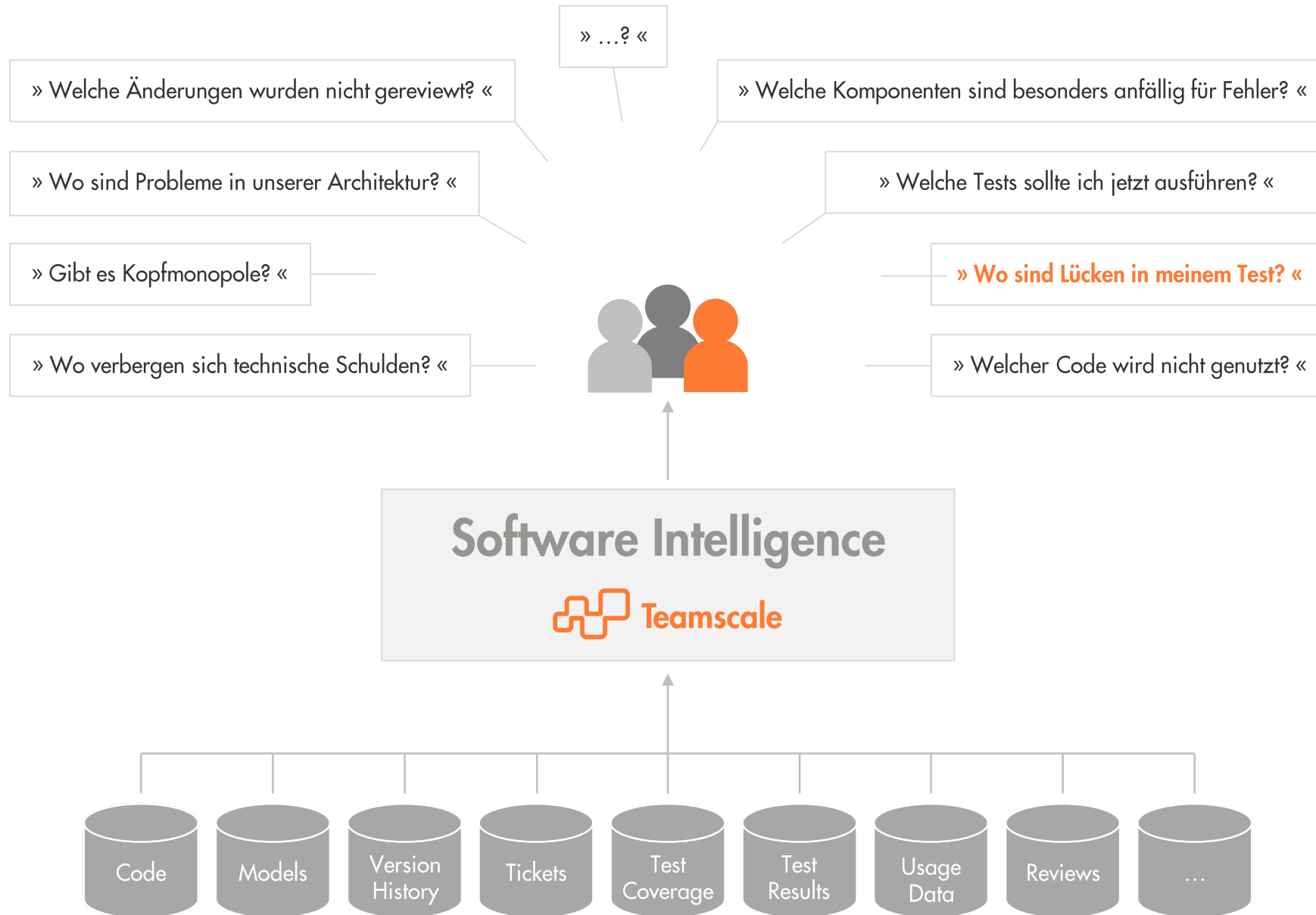
Found potential inconsistent clone change in JenkinsRule.java

Context

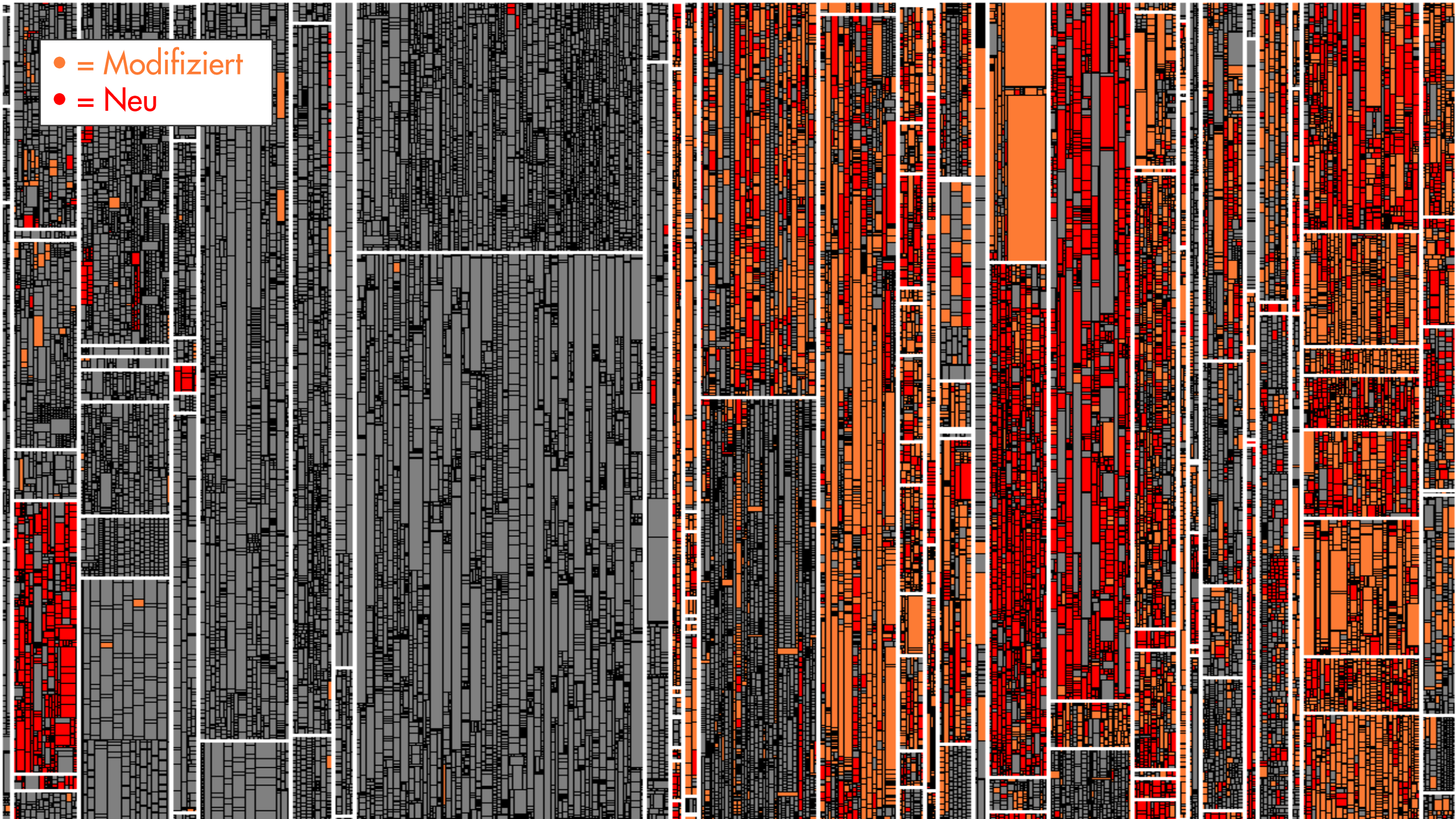
[\[Broken clone\]](#) [\[Old clone finding\]](#) [\[Code change\]](#)





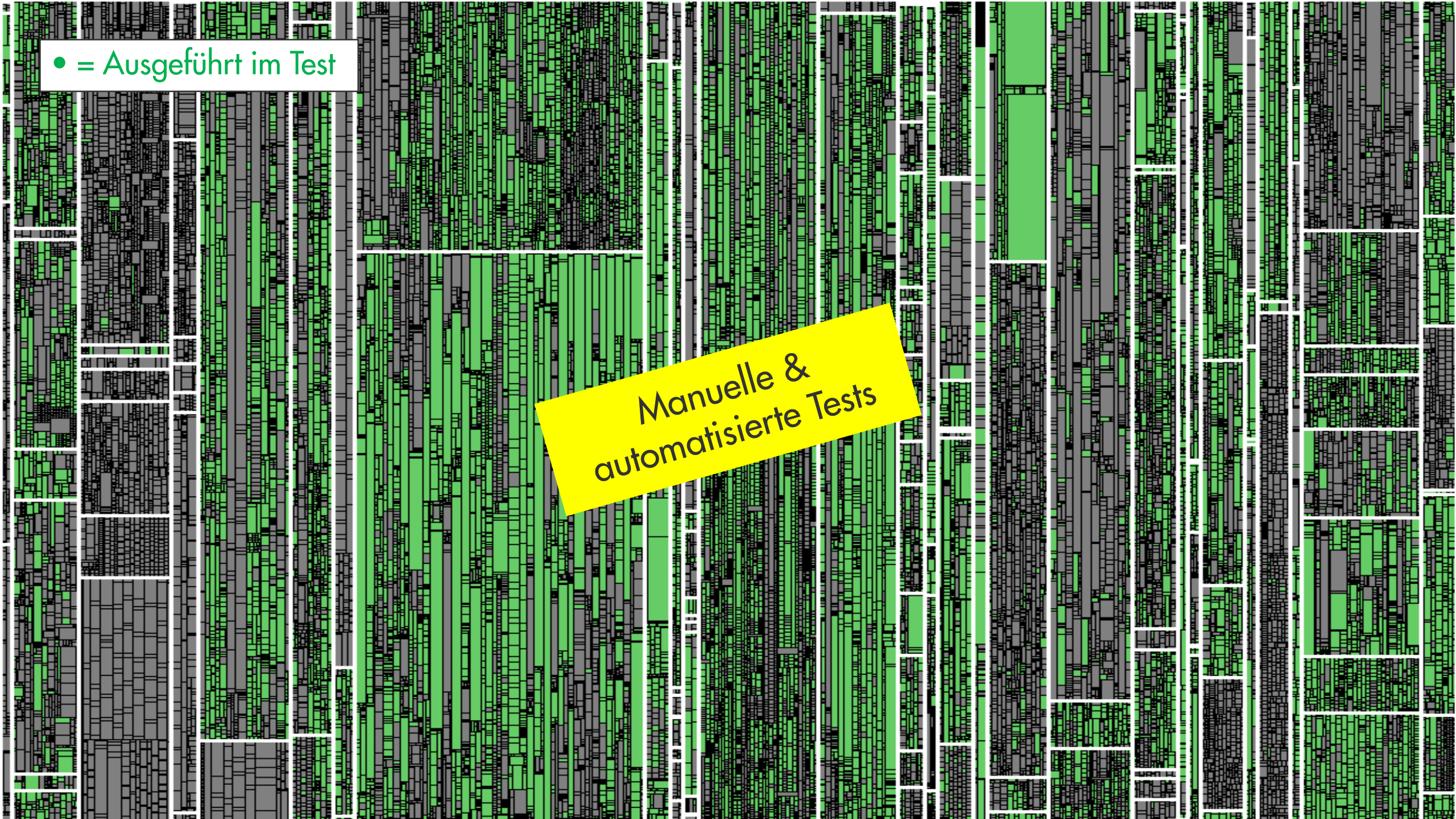


- = Modifiziert
- = Neu



● = Ausgeführt im Test

Manuelle &
automatisierte Tests

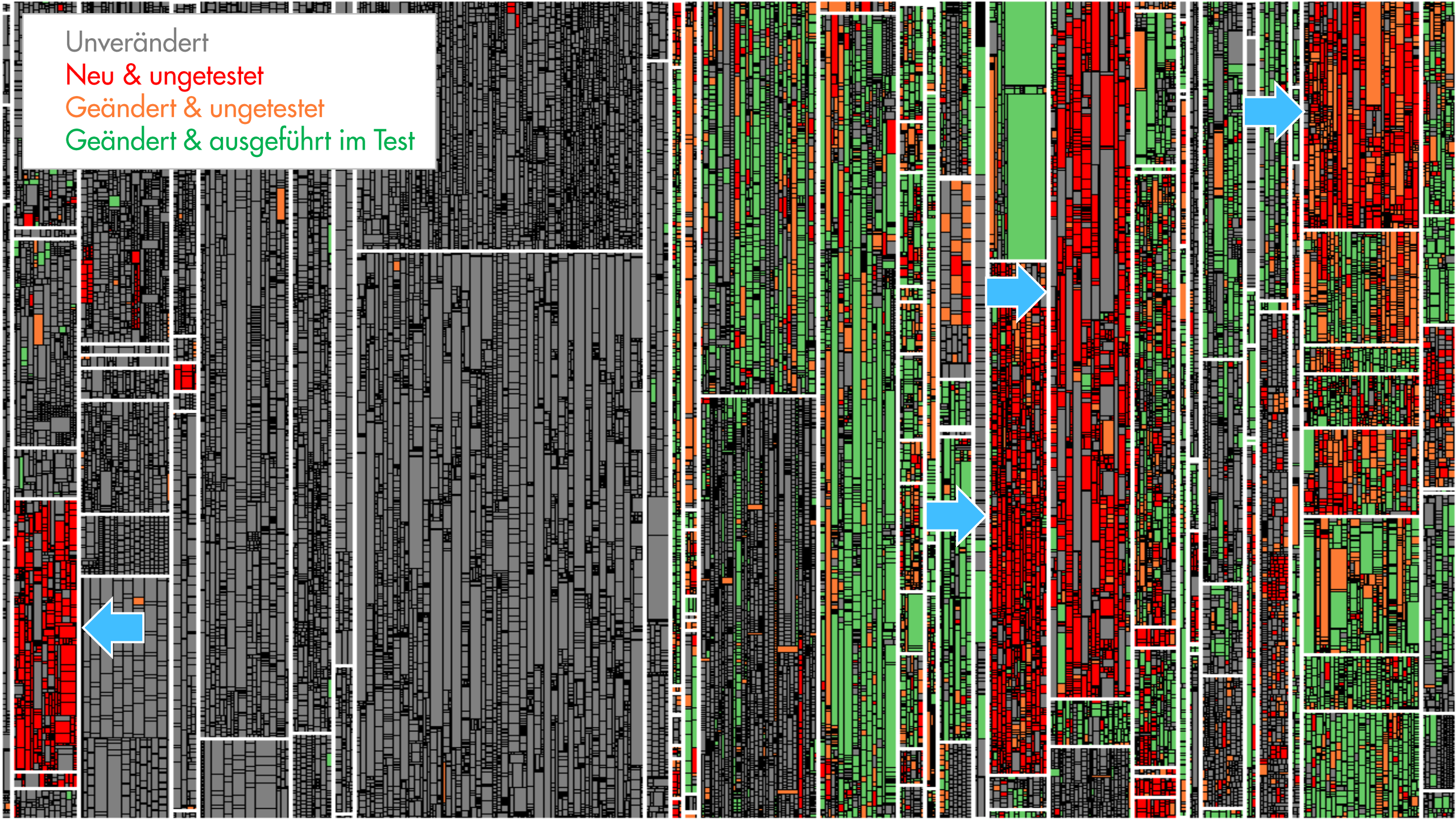



















Unverändert





Neu & ungetestet

Geändert & ungetestet


Geändert & ausgeführt im Test




Issue # ▼	Subject	Done		Test Gap
TS-10549	Undo/Redo for web-based architecture editor	Done		0% 
TS-10784	Fix long method finding in TaintAnalysisRunner	Done		0% 
TS-10923	Implement metric 'Nesting Depth' for Simulink	Done		29% 
TS-11364	External findings are not registered during first upload	Done		14% 
TS-11942	Manual test coverage upload during development	Done		43% 
TS-12050	Tool for transferring findings blacklists and tasks	Done		50% 
TS-12262	Cannot set or alter alias without reanalysis	Done		0% 
TS-13151	Fetch parent relationship of TFS work items	Done		0% 

Issue # ▾	Subject		Test Gap
TS-14421	Get rid of TestGapSynchronizer block	Done 	0% 
TS-14733	Remove Dataflow blocks	Done 	22% 

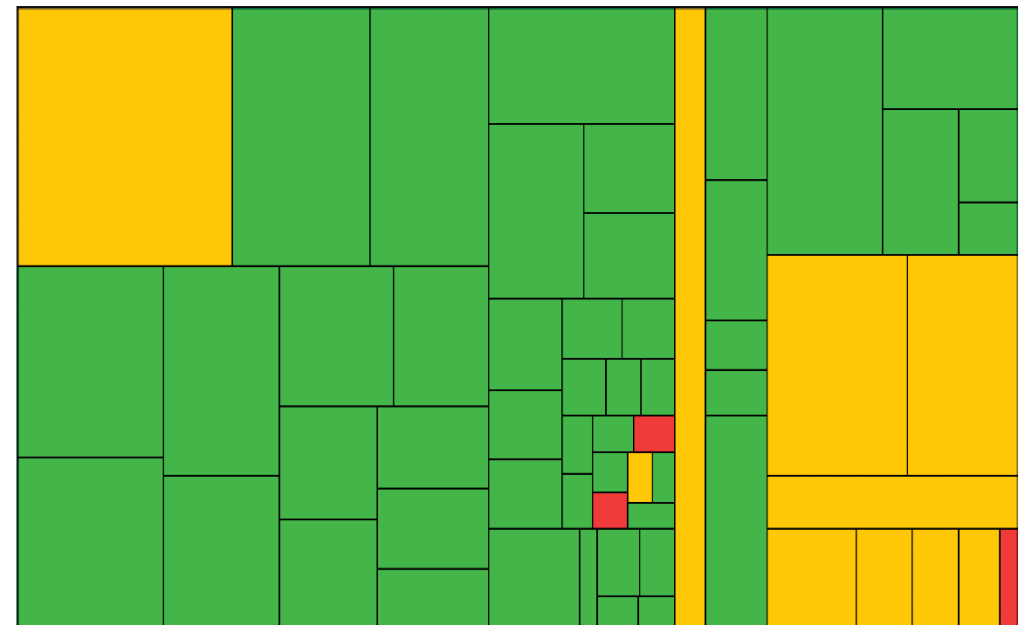
Done Issue TS-14733 - Remove Dataflow blocks

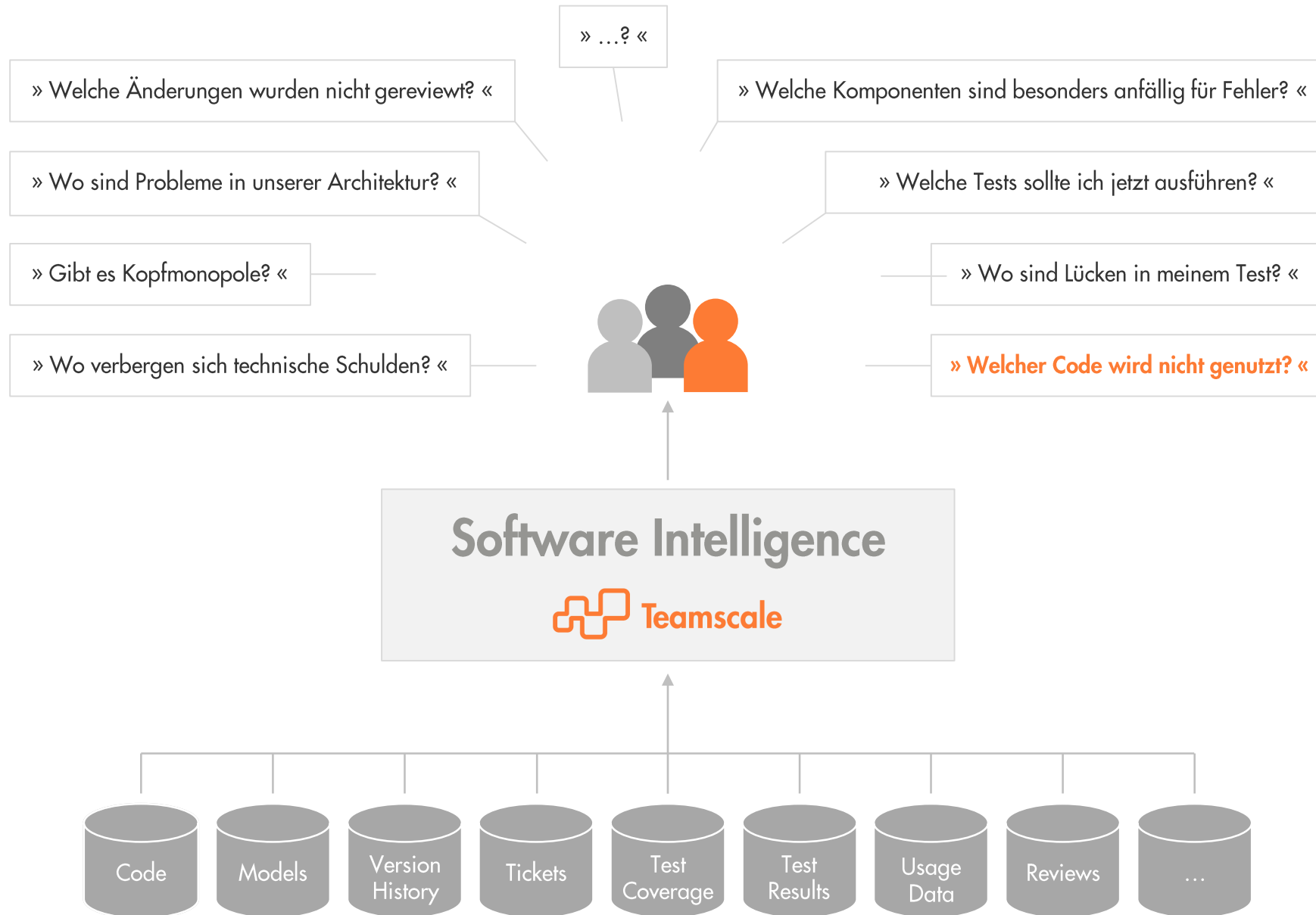
Creator:  (on Apr 06 2018 19:44) Last update: Aug 24 2018 09:32

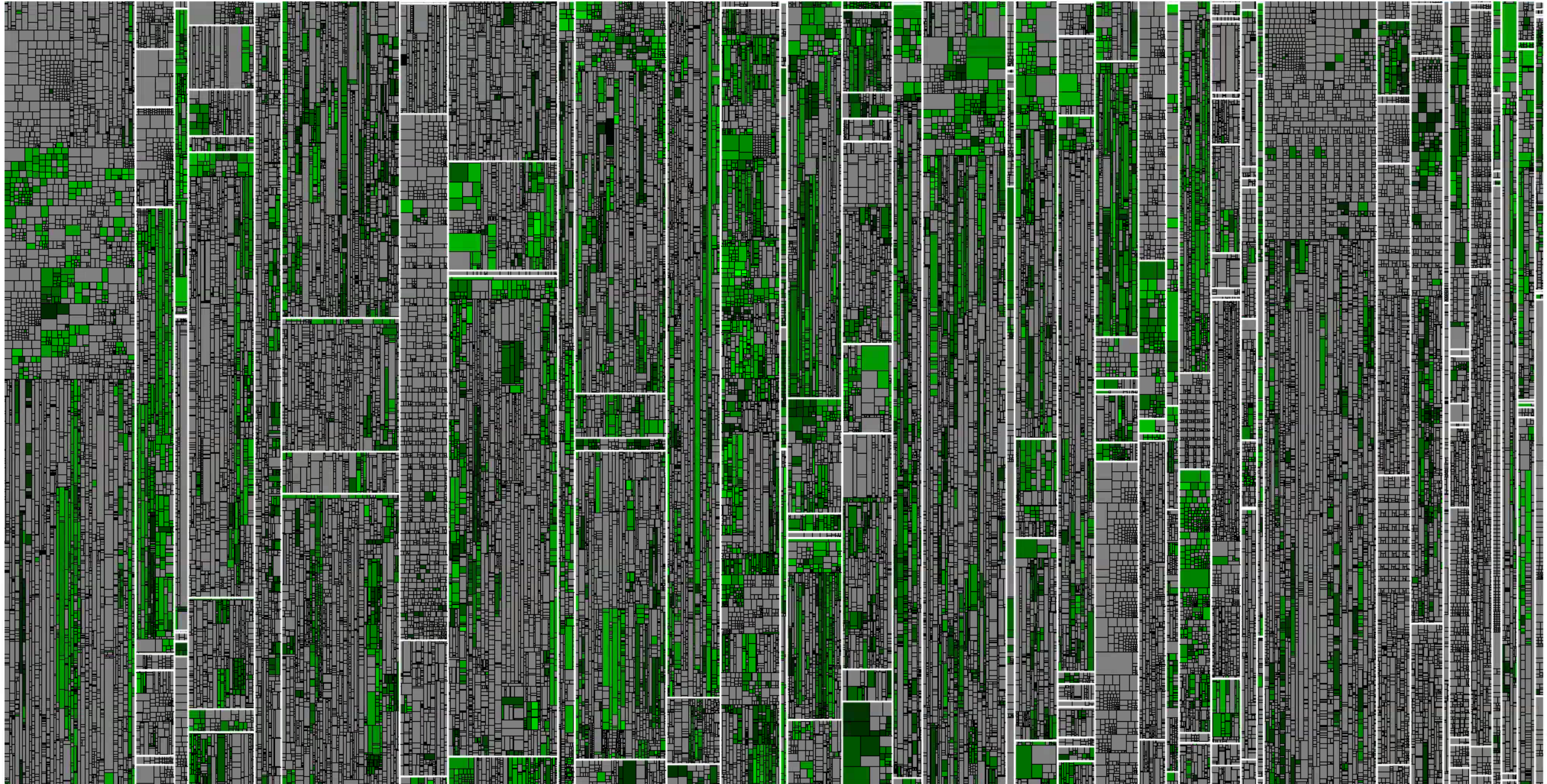
Assignee: 

Project	Type	Priority	Resolution	Fix Version
TS	Maintenance	Normal	Green	Teamscale 4.5
Component	Labels	Affected Version	Customer	Customer Issue
Backend	Performance			
Epic Name	Freshdesk URL	Merge Request		
		https://git.cqse.eu/cqse/teamscale/3621		

Aug 15 2018 12:37–Now | Test Gap: 22%

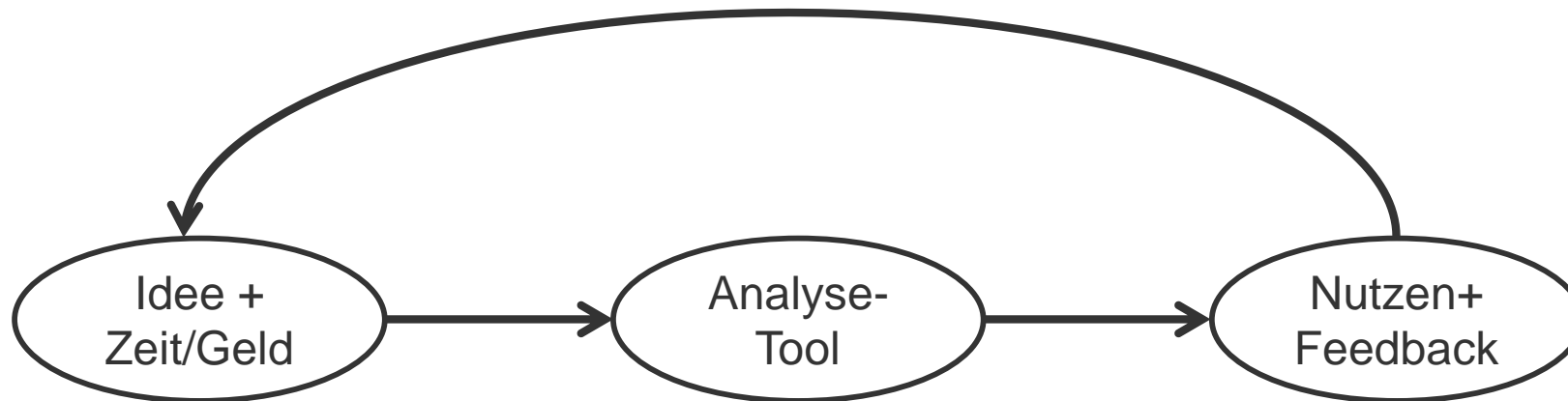


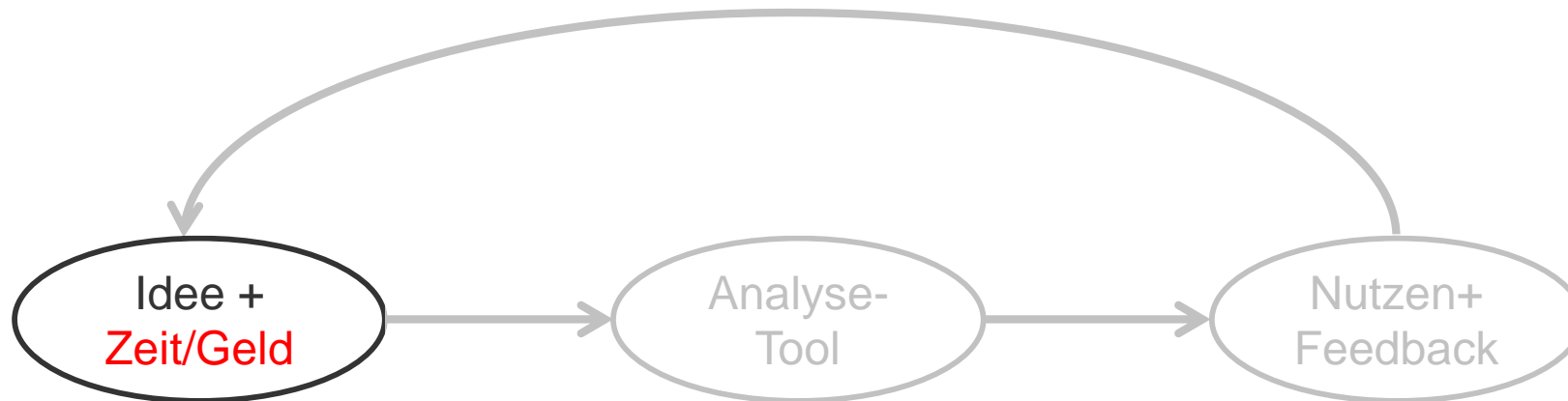




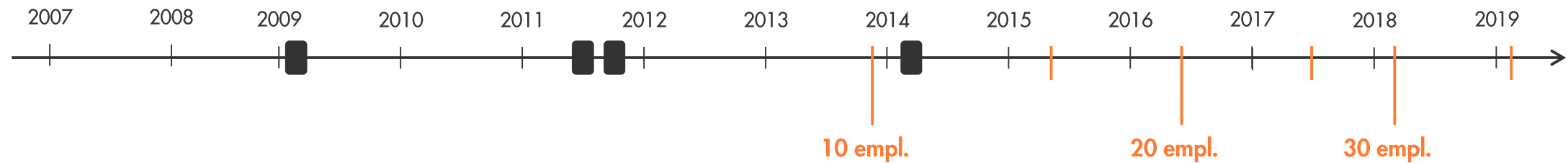


Frei für Forschung und Lehre

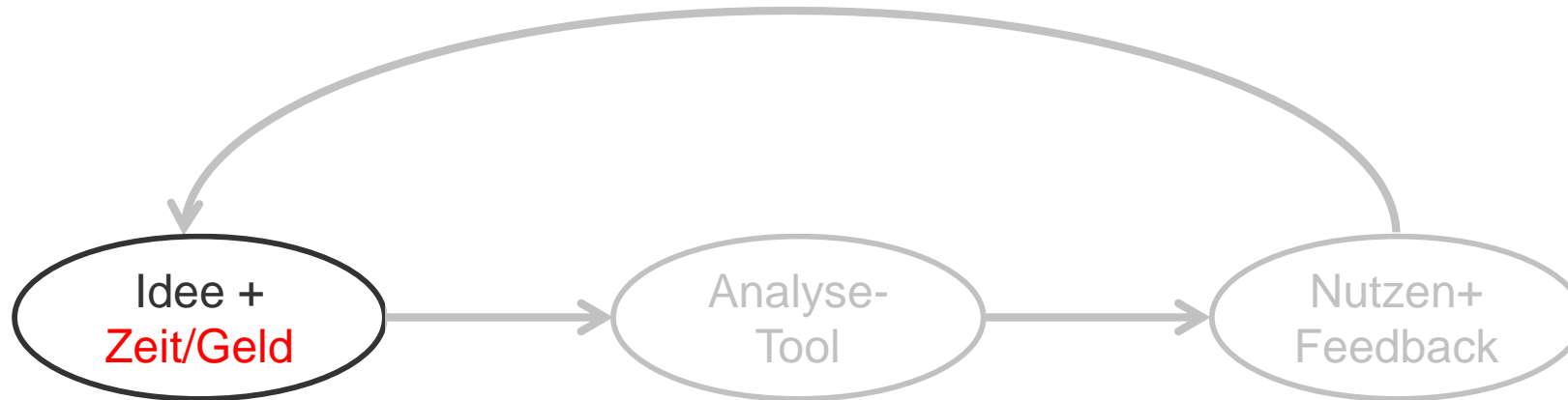




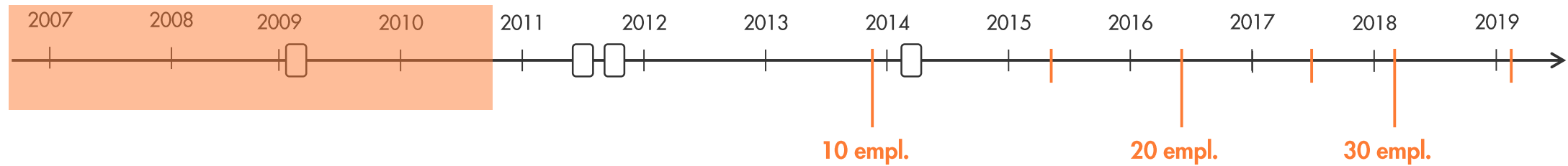
CQSE History



- Jan 2009: Gründung
- Feb 2009: Erste Beratungskunden
- Mid 2011: Promotionen abgeschlossen, Focus auf CQSE
- Sept 2011: Erstes Büro, erster Mitarbeiter
- Jan 2014: Release Teamscale 1.0



CQSE History



- Team: 4 Gründer
- Alle in Promotion
- CQSE nur nebenbei



```

33 * Creates a new job by reading stdin as a configuration XML file.
34 *
35 * @author Kohsuke Kawaguchi
36 */
37 @Extension
38 public class CreateJobCommand extends CLICommand {
39     @Override
40     public String getShortDescription() {
41         return Messages.CreateJobCommand_ShortDescription();
42     }
43
44     @Argument(metaVar="NAME",usage="Name of the job to create",required=true)
45     public String name;
46
47     protected int run() throws Exception {
48         Jenkins h = Jenkins.getInstance();
49
50         if (h.getItemByFullName(name)!=null) {
51             stderr.println("Job '"+name+"' already exists");
52             return -1;
53         }
54
55         ModifiableTopLevelItemGroup ig = h;
56         int i = name.lastIndexOf('/');
57         if (i > 0) {
58             String group = name.substring(0, i);
59             Item item = h.getItemByFullName(group);
60             if (item == null) {
61                 throw new IllegalArgumentException("Unknown ItemGroup " + group);
62             }
63
64             if (item instanceof ModifiableTopLevelItemGroup) {
65                 ig = (ModifiableTopLevelItemGroup) item;
66             } else {
67                 throw new IllegalArgumentException("Can't create job from CLI in " + group);
68             }
69             name = name.substring(i + 1);
70         }
71
72         Jenkins.checkGoodName(name);
73         ig.createProjectFromXML(name, stdin);
74         return 0;
75     }
76 }
77
78

```

```

36 *
37 * @author Kohsuke Kawaguchi
38 */
39 @Extension
40 public class CopyJobCommand extends CLICommand {
41     @Override
42     public String getShortDescription() {
43         return Messages.CopyJobCommand_ShortDescription();
44     }
45
46     @Argument(metaVar="SRC",usage="Name of the job to copy",required=true)
47     public TopLevelItem src;
48
49     @Argument(metaVar="DST",usage="Name of the new job to be created.",index=1,required=true)
50     public String dst;
51
52     protected int run() throws Exception {
53         Jenkins jenkins = Jenkins.getInstance();
54
55         if (jenkins.getItemByFullName(dst)!=null) {
56             stderr.println("Job '"+dst+"' already exists");
57             return -1;
58         }
59
60         ModifiableTopLevelItemGroup ig = jenkins;
61         int i = dst.lastIndexOf('/');
62         if (i > 0) {
63             String group = dst.substring(0, i);
64             Item item = jenkins.getItemByFullName(group);
65             if (item == null) {
66                 throw new IllegalArgumentException("Unknown ItemGroup " + group);
67             }
68
69             if (item instanceof ModifiableTopLevelItemGroup) {
70                 ig = (ModifiableTopLevelItemGroup) item;
71             } else {
72                 throw new IllegalArgumentException("Can't create job from CLI in " + group);
73             }
74             dst = dst.substring(i + 1);
75         }
76
77         ig.copy(src,dst).save();
78         return 0;
79     }
80 }
81

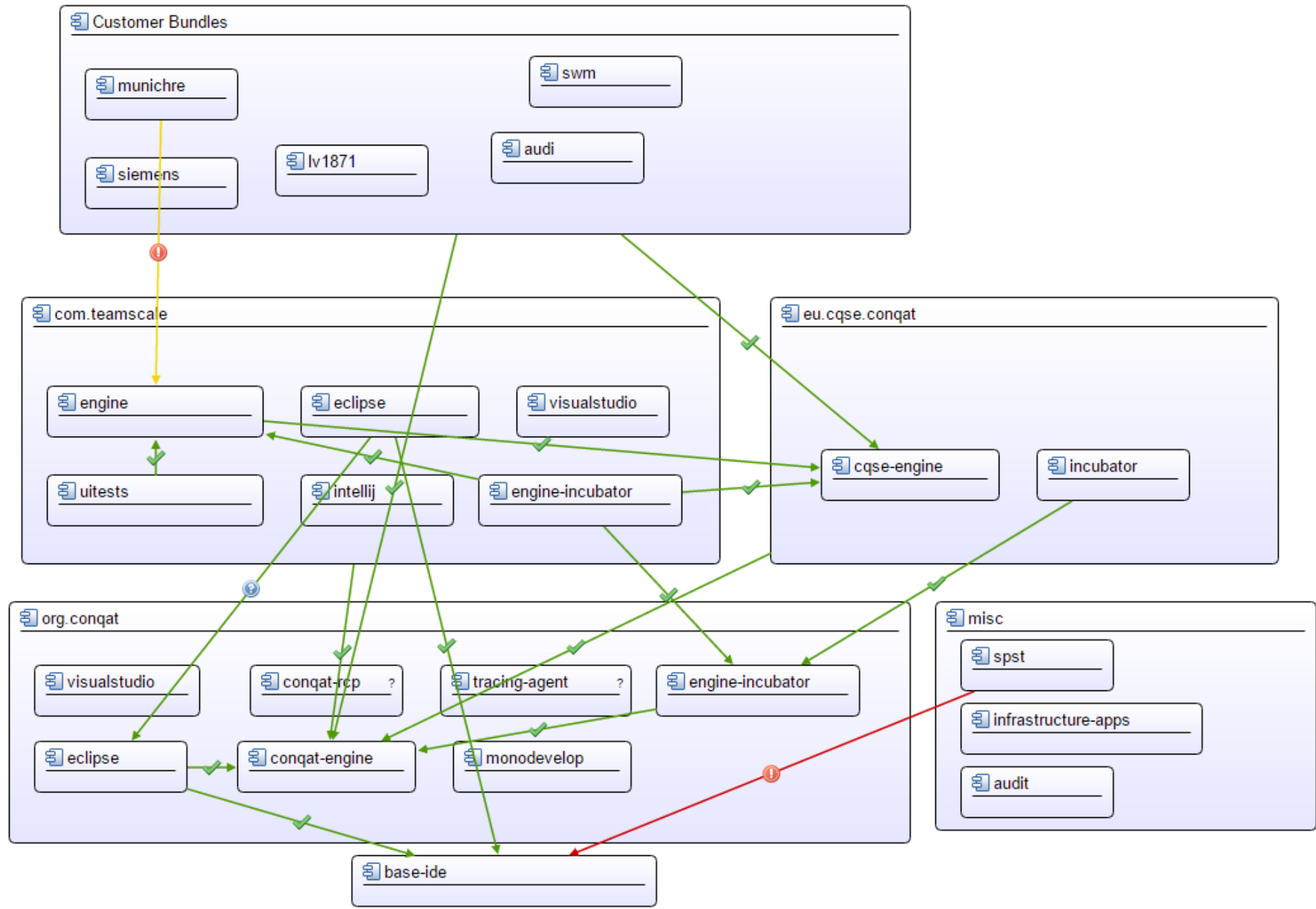
```

TreeAdministrationQuarterly.cs

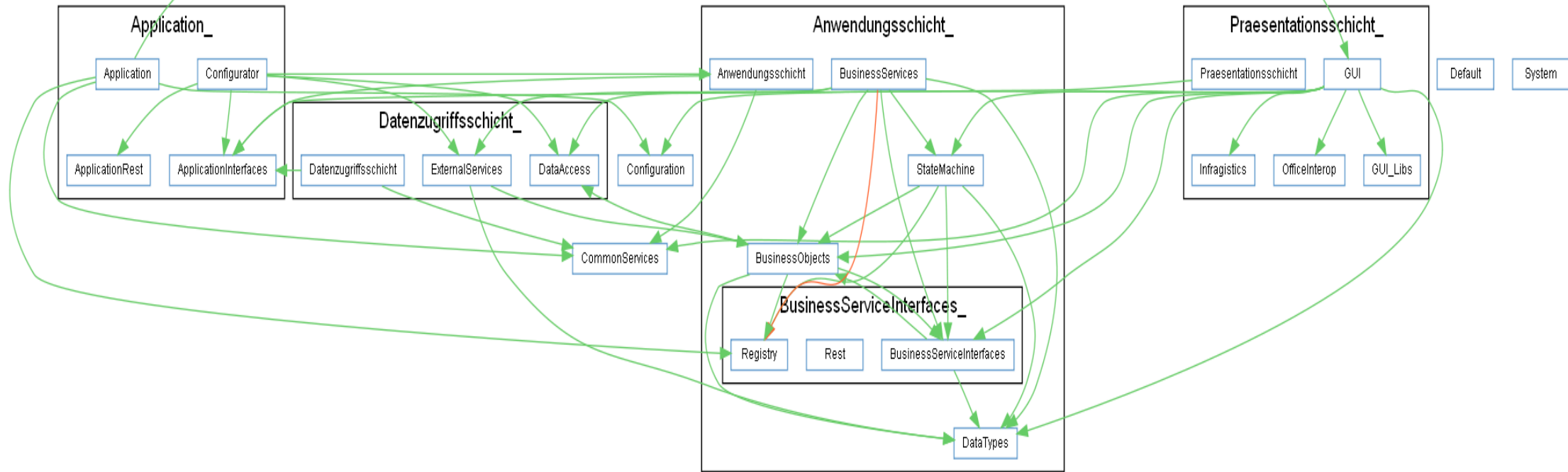
```
93  /// </summary>
94  protected override bool DoLazyLoading(
95      UltraTreeNode node)
96  {
97      // Some types of segments always have the
98      // expand icon
99      if ( node.Tag is GaSegment ||
100         node.Tag is MainSegment ||
101         node.Tag is GeneralMainSegment ||
102         node.Tag is Branch )
103         return true;
104     // All others only if there are child nodes
105     return false;
106 }
107
108 /// <summary>
109 /// Structure segments return only loss or
110 /// premium segments according to
111 /// the mode.
112 /// </summary>
113 protected override IList GetChildSegments(
114     ISegment parent, ref bool alreadySorted )
115 {
116     if ( parent is StructureSegment )
117     {
118         // Show either premium or loss beyond
119         // structure segment
120         StructureSegment strucSeg = (StructureSegment
121             )parent;
122         if ( (TypeOfSgmtEnum) GetFilterValue( typeof(
123             TypeOfSgmtEnum) ) == TypeOfSgmtEnum.Loss
124             )
125             return strucSeg.GetLossProcessingSegments();
126         else
127             return strucSeg.GetPremiumProcessingSegments
128             ();
129     }
130 }
```

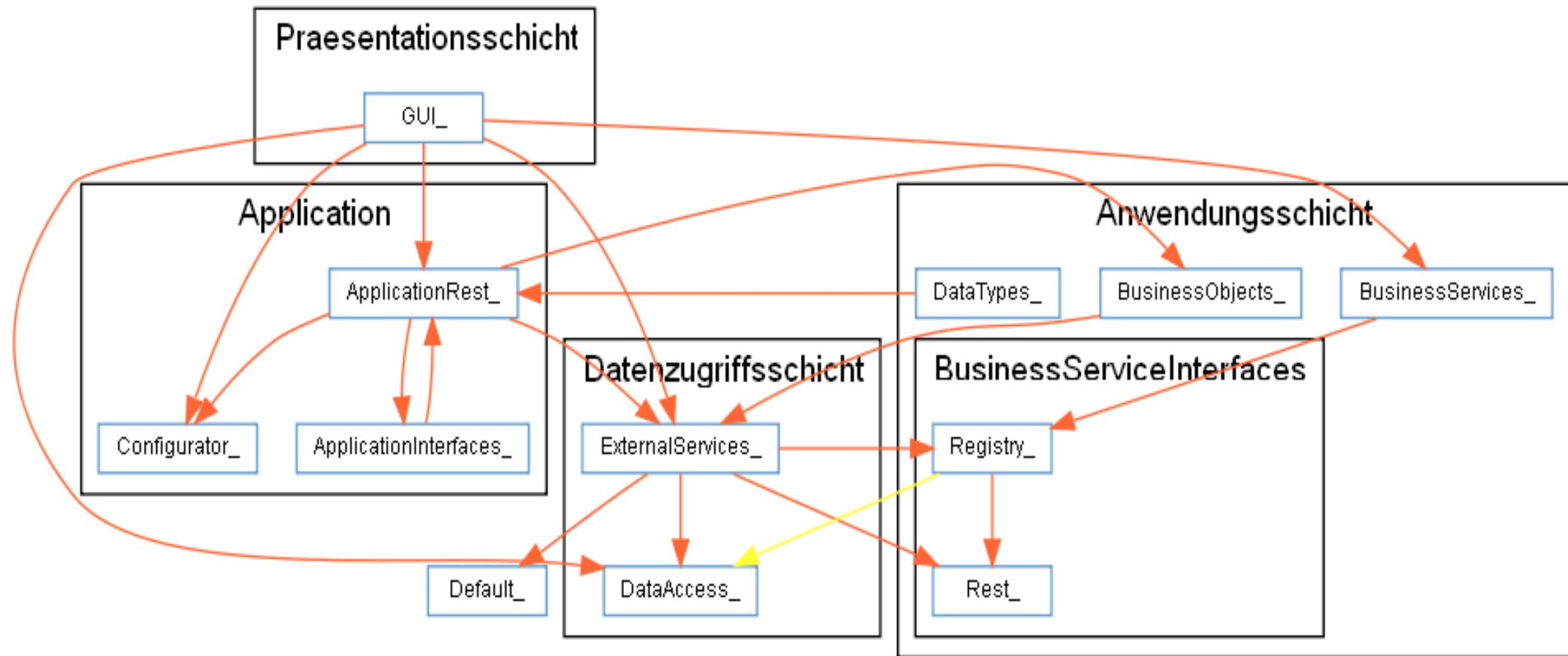
TreeAdministrationYearly.cs

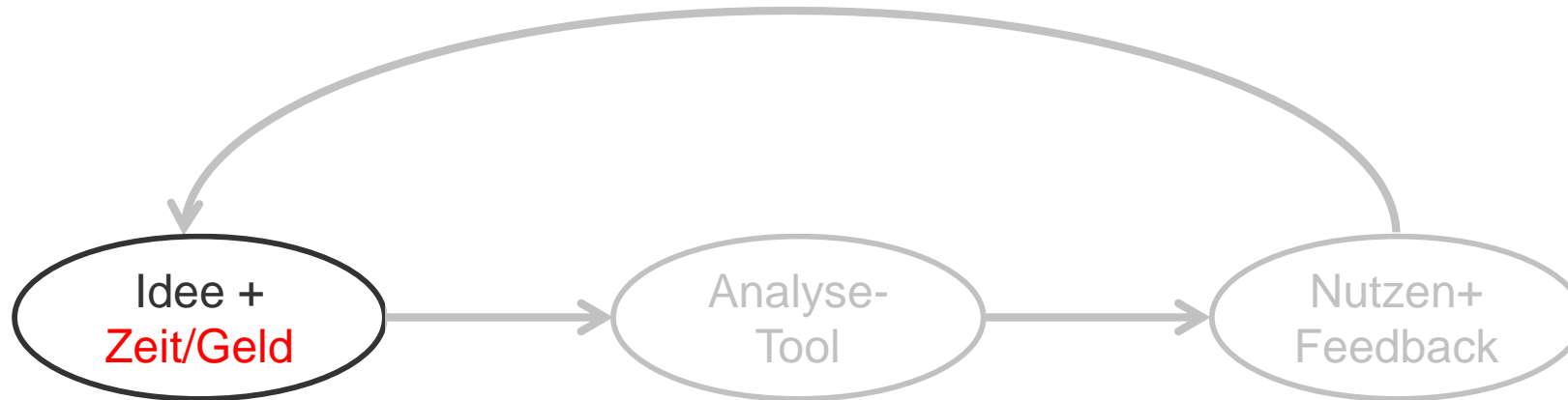
```
106  /// </summary>
107  protected override bool DoLazyLoading(
108      UltraTreeNode node)
109  {
110      // Some types of segments always have the
111      // expand icon
112      if ( node.Tag is GaSegment ||
113         node.Tag is MainSegment ||
114         node.Tag is GeneralMainSegment ||
115         node.Tag is Branch )
116         return true;
117     // All others only if there are child nodes
118     return false;
119 }
120
121 /// <summary>
122 /// Structure segments return only loss or
123 /// premium segments according to
124 /// the mode.
125 /// </summary>
126 protected override IList GetChildSegments(
127     ISegment parent, ref bool alreadySorted )
128 {
129     if ( parent is StructureSegment )
130     {
131         // Show either premium or loss beyond
132         // structure segment
133         StructureSegment strucSeg = (StructureSegment
134             )parent;
135         if ( (TypeOfSgmtEnum) GetFilterValue( typeof(
136             TypeOfSgmtEnum) ) == TypeOfSgmtEnum.Loss
137             )
138             return strucSeg.GetLossProcessingSegments();
139         else
140             return strucSeg.GetPremiumProcessingSegments
141             ();
142     }
143 }
```



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <conqat-architecture xmlns="http://conqat.cs.tum.edu/ns/architecture" policy="der
  exclude="">
3   <component name="Customer Bundles">
4     <component name="Customer Bundles/munichre">
5       <code-mapping type="include" regex="engine/com.munichre.*" />
6     </component>
7     <component name="Customer Bundles/siemens">
8       <code-mapping type="include" regex="engine/com.siemens.*" />
9     </component>
10    <component name="Customer Bundles/audi">
11      <code-mapping type="include" regex="engine/de.audi.*" />
12    </component>
13    <component name="Customer Bundles/lv1871">
14      <code-mapping type="include" regex="engine/de.lv1871.*" />
15    </component>
16    <component name="Customer Bundles/swm">
17      <code-mapping type="include" regex="engine/de.swm.*" />
18    </component>
19  </component>
20  <component name="base-ide" dim="240,40">
21    <code-mapping type="include" regex="ide/plugins/org\.conqat\.ide\.commons.*"
22  </component>
23  <component name="base-libs" dim="240,40">
24    <code-mapping type="include" regex="lib/.*" />
25  </component>
26  <component name="com.teamscale">
27    <component name="com.teamscale/uitests" dim="170,40" pos="20,140">
28      <code-mapping type="include" regex="testing/.*" />
29      <code-mapping type="include" regex="engine/com\.teamscale\.test/.*" />
```







Blockpraktikum

- Weiterentwicklung der Werkzeuge im Praktikum
- Von Anfang an: Code Open Source (Apache-Lizenz)
- Jedes Praktikum neue Aufgaben.

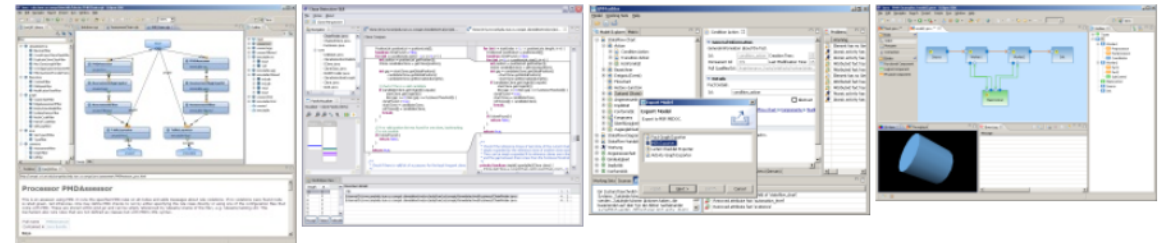


Blockpraktikum *Eclipse Rich Client Platform Application Development*

Thematische Einordnung

Eclipse ist ein Open-Source-Framework zur Entwicklung von Rich-Client-Applikationen. Insbesondere als Basis für die Entwicklung von Werkzeugen im Bereich Software-Engineering hat es sich mittlerweile als de-facto Standard etabliert, da man mit verhältnismäßig geringem Aufwand sehr mächtige und einfach integrierbare Werkzeuge erstellen kann. Andererseits stellt die Arbeit mit einem Framework, das inzwischen auf über 17 Millionen Code-Zeilen angewachsen ist, eine ganz eigene Herausforderung dar.

Dieses Praktikum gibt einen Einblick in das Thema der Werkzeugentwicklung mit dem Eclipse-Framework. Neben dem Kennenlernen der zentralen Konzepte und Plugins von Eclipse wird vor allem praktische Erfahrung mit Softwareentwicklung, insbesondere im Umgang mit sehr großen Systemen, vermittelt.



Screenshots von am Lehrstuhl Prof. Broy auf Basis von Eclipse entwickelten Software-Engineering-Werkzeugen

Zielsetzung und Durchführung

Das Praktikum gliedert sich in zwei Teile. Im ersten Teil des Praktikums werden an Hand von Übungsaufgaben grundlegende Techniken und Werkzeuge des Software-Engineering eingeübt. Außerdem werden einzelne Elemente von Eclipse wie Editoren und Views, Decorators und Markers, aber auch das für die Entwicklung grafischer modellgetriebener Editoren zentrale GEF (Graphical Editing Framework) Gegenstand dieser Aufgaben sein. Hierbei wird vermittelt, wie sich mit den Mitteln von Eclipse selbst, aber auch mit anderen Techniken, die Orientierung in großen Systemen erleichtern lässt.

Im zweiten Teil des Praktikums wird das Gelernte konkret an der Entwicklung eines größeren Werkzeugs erprobt. Neben einer tieferen Einarbeitung in Eclipse steht hier die praktische Erfahrung bei der Mitarbeit in einem größeren Entwicklungsprojekt im Vordergrund. Geplant ist hier die Erweiterung des grafischer Editors *cq.edit* für das am Lehrstuhl entwickelte Werkzeug *ConQAT*. Sowohl *ConQAT* als auch *cq.edit* sind Ergebnisse aus früheren Praktika.

Woche 1

- Arbeitsblatt mit Aufgaben
- Umgang mit IDE, Debugger, Versionskontrolle, CI
- Einführung Code Reviews
- Einstieg in Code-Basis

Eclipse Rich Client Platform and Android Application Development
Wintersemester 2011/2012
Prof. Dr. Dr. h.c. M. Broy
V. Bauer, L. Heinemann, B. Hauptmann, B. Hummel, M. Junker, E. Jürgens
05.03.2012

Übungsblatt 1: Grundlagen

Aufgabe 1: Projektinfrastruktur

Ein wesentlicher Schritt in jeder Projektentwicklung ist das Kennenlernen der verwendeten

Entwicklungsumgebung

Wir benutzen Eclipse in der Version *Eclipse Classic* von der Eclipse Foundation. Über das Menü *Help* → *Install New Software* → *Available Software Sites* → *Add Site* → *Next* → *OK* → *Next* → *Finish* werden diese Features bei Ihnen nicht automatisch installiert.

Versionsverwaltung

Als Versionsverwaltung setzen wir *Subclipse* ein. Um die TortoiseSVN zu installieren, finden Sie die Dokumentationsseite der TortoiseSVN unter <http://www.tortoisersvn.org/>. Seine URL auch mit einem Webbrowser öffnen.

Für Eclipse gibt es das *Subclipse* Plugin. Sie es von der entsprechenden Website herunterladen. Für das Praktikum ist ein Subclipse-Plugin erforderlich.

Das Verzeichnis *infrastructure* statt der installierten⁴ benutzen.

¹<http://www.eclipse.org/downloads>

²<http://www4.in.tum.de/~ccsm/eclipse>

³http://subclipse.tigris.org/update_1.0.0

⁴Unter Windows 7 ist das %APPDATA%

Checkliste zum Code-Review

1. Die Java Coding Conventions (<http://java.sun.com/javase/6/docs/technotes/guides/coding/>)
 - a) Namen von Interfaces fangen (fast) mit *I* an
 - b) Namen von Enumerations fangen (fast) mit *E* an
 - c) Namen von abstrakten Basisklassen fangen (fast) mit *A* an
2. Es darf keine Warnings geben (bzgl. der Coding Conventions)
3. Kommentare:
 - a) Alle Klassen, Attribute und Methodendeklarationen haben Kommentare (Parameter und @returns), auch wenn sie offensichtlich sind
 - b) Kein auskommentierter Code
 - c) Keine offenen TODOs, FIXMEs, o.ä.
 - d) Jedes package muss eine (zumindest) kurze Dokumentation reinkommen.
4. Visibilities: Sind die Visibilities gut gewählt?
5. Kein *unused code*
6. Existierende Bibliotheksfunktionen benutzen
7. Clone (redundanter oder duplizierter Code vermeiden)
8. Verständlichkeit: sind die Bezeichner gut gewählt und verständlich?
9. Sind die Methoden nicht zu lang (ca. 100 Zeilen)?
10. Sind die Klassen nicht zu lang (ca. 1000 Zeilen)?
11. Werden ternäre Ausdrücke benutzt? (Sparen Sie sich das für die nächsten Semester)
12. Werden innere oder anonyme Klassen übermäßig eingesetzt bzw. missbraucht? (bei GUI-Code manchmal kaum zu vermeiden)
13. Öffentliche Methoden, die Collections zurückgeben, sollten im Normalfall eine UnmodifiableCollection oder eine der Subklassen zurückgeben.
14. Throws Exception und catch Exception sind tabu, da zu allgemein.
15. Unveränderliche Felder müssen mit final ausgezeichnet sein.

Eclipse Rich Client Platform and Android Application Development
Wintersemester 2011/2012
Prof. Dr. Dr. h.c. M. Broy
V. Bauer, L. Heinemann, B. Hauptmann, B. Hummel, M. Junker, E. Jürgens
07.03.2012



Übungsblatt 3: Eclipse Views, Testautomatisierung

Dieses Aufgabenblatt umfasst zwei Teile: Der erste Teil behandelt die Programmierung von Views und Editoren in Eclipse. Der zweite Teil befasst sich mit der Erstellung von Tests und Testautomatisierung von Eclipse-Anwendungen.

Als Ausgangspunkt für alle folgenden Aufgaben laden Sie sich bitte `org.conqat.ercpad.assignment03.zip` aus dem ERCPAD4 SVN unter <https://svn.broy.in.tum.de/ercpad4/assignments/blatt03>. Importieren Sie die Projekte in Ihren Workspace und benennen Sie sie nach folgendem Schema:

```
edu.tum.cs.assignment03_team<Team-#>  
edu.tum.cs.assignment03_manualtest_team<Team-#>  
edu.tum.cs.assignment03_automatedtest_team<Team-#>
```

Importieren Sie das Projekt mit dem neuen Namen in das Unterverzeichnis `submissions/assignment03` unseres Subversion-Repositories.

Teil 1: Eclipse Views & Editors

Ziel dieses Teils ist die Entwicklung eines einfachen Editors für ConQAT XML Konfigurationsdateien. Das gegebene Plugin bietet bereits einen XML Editor mit einfachem Syntax-Highlighting, der in Abb. ?? dargestellt ist. Der Editor arbeitet auf `.conqat` Dateien.

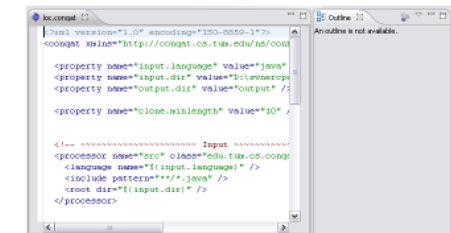


Abbildung 1: Editor and empty outline

Um den Umgang mit dem Editor zu vereinfachen, bietet das Plugin außerdem bereits einen einfachen XML Parser, der ein einfaches Modell für ConQAT Konfigurationsdateien, das für die Aufgaben ausreichend ist, erstellt. Parser und Modell liegen in den Packages

Woche 2 und 3

- Arbeit in Team von 2-3 Personen
- Arbeit an echten Requirements, die Werkzeug erweitern

Requirement »Eclipse: Display Clone Pair View«	
Summary	Display <i>Clone Pair View</i> for a pair of clones.
Developers	
Reviewers	
Owner	Benjamin
Description	
To understand clones, a developer needs to understand their commonalities and differences. We have implemented a <i>Clone Pair View</i> that supports this. Developers should be able to open it for all clone relationships that are indicated via the clone bars.	
Preconditions	
<ol style="list-style-type: none">1. User has a project in the workspace for which the server mapping is configured.2. Clone information for the project is available on the server.3. A file that contains clones is opened in the editor, and its clone bars are visible.	
Success Scenario	
<ol style="list-style-type: none">1. User opens the context menu on a clone bar.2. Context menu displays entries to open the pair view for each pair of clones that covers the line for which the bar was clicked.3. User selects a menu entry.4. Eclipse opens the clone pair view for the selected clone pair.	
Notes	
<ul style="list-style-type: none">■ Similar functionality is already implemented in ConQAT. See which code you can reuse from it.	
Change Requests	

Tagesablauf

Woche 1

9:00-9:30 Uhr

- Besprechung Blatt
- Vorträge Aufgabe

13:30-14:00 Uhr

- Feedback Aufgaben Vortag
- Offene Fragen

16:30-17:00 Uhr

- Nachbesprechung (optional in W1)

Woche 2 & 3

09:00-9:30 Uhr

- Statusmeeting

16:30-17:00

- Nachbesprechung

Lauffähiger Code ab Tag 1

- Betreuer bereiten Dummy-Implementierung vor, die mit Test-Daten exemplarisch anzeigt.
- Ermöglicht schnellen Feedback-Zyklus für eigene Arbeit.
- Schnelles Erfolgserlebnis.

The screenshot displays the Eclipse IDE interface for a ConQAT project. The main workspace shows a flowchart with the following components and connections:

- Input** (Start node) connects to **LOCAnalyzer** (loc-analyzer).
- LOCAnalyzer** connects to **SumAggregator** (loc-aggregator).
- SumAggregator** connects to **TableLayouter** (loc-table).
- TableLayouter** connects to **html-result** (End node).
- An **info** node also connects to **TableLayouter**.
- A **result** node is shown but not connected to any other node.

Annotations in the image:

- A**: Points to the LOCAnalyzer block.
- B**: Points to the ConQAT Libr panel on the right.
- C**: Points to the Parameters table in the Properties window.
- D**: Points to the processor description in the ConQATDoc window.

Properties - Parameters

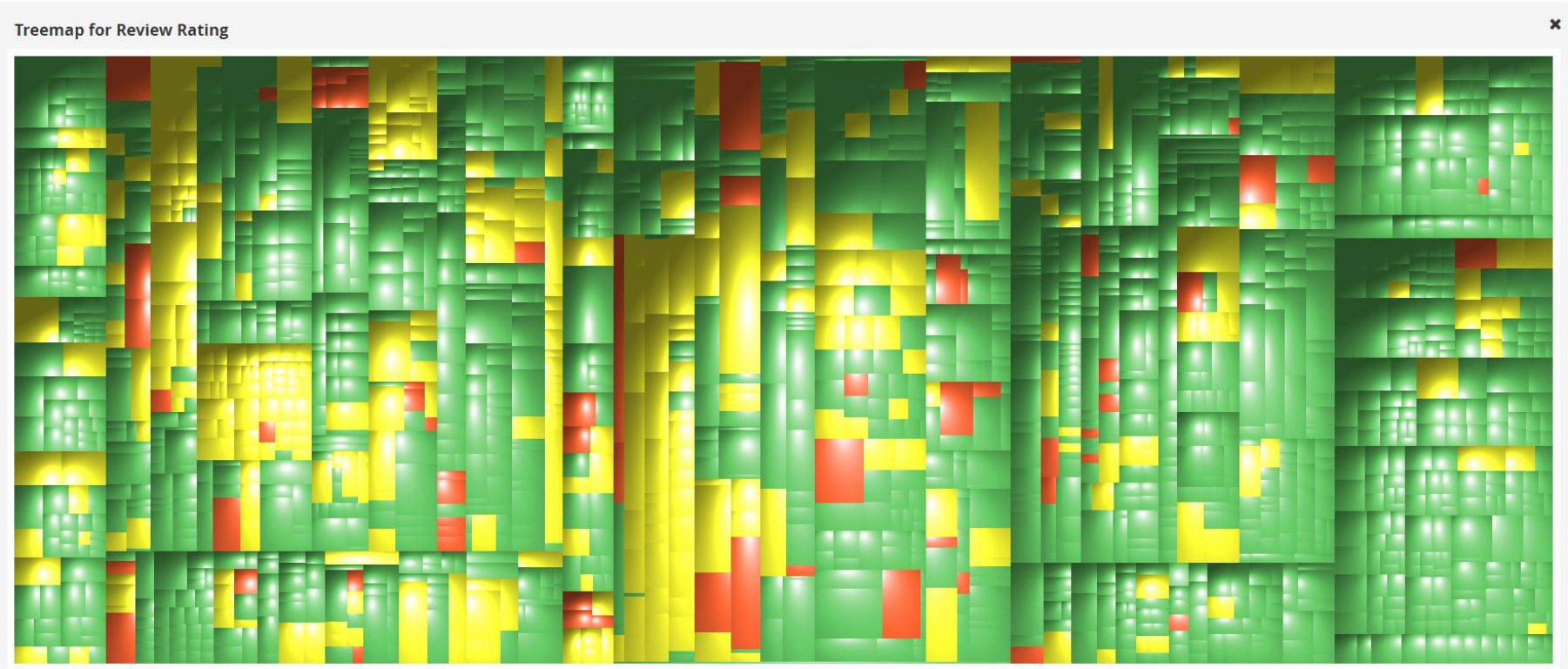
Parameter	D..	Attribute value
Input		
@ ref		@input.scope

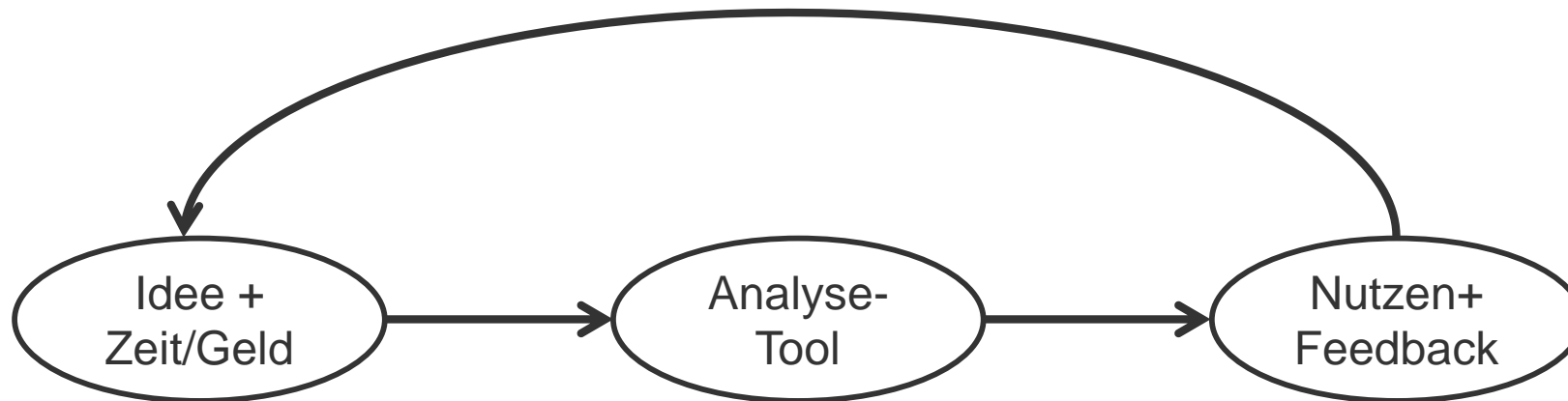
ConQATDoc - Processor LOAnalyzer

Counts lines of code. This processor benefits from distributed execution.

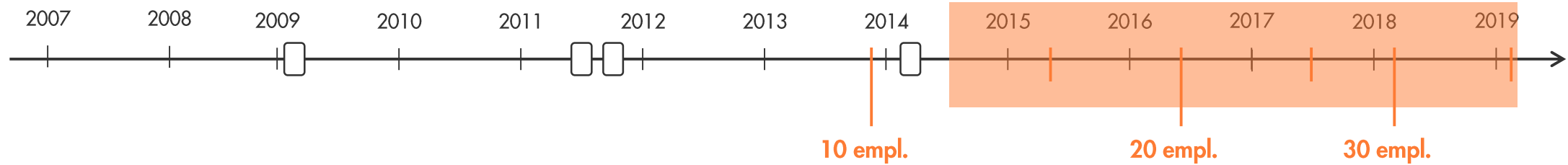
Full name: edu.tum.cs.conqat.
Contained: Filesystem Bundle in

Name	Type	Description
LoC	Integer	Lines of code

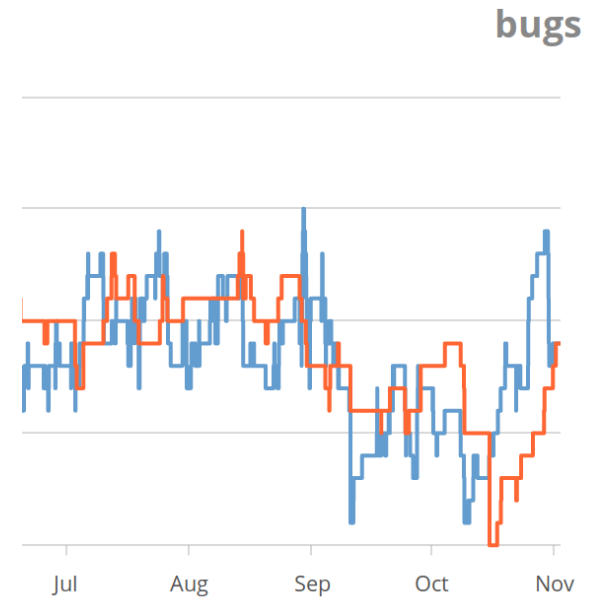
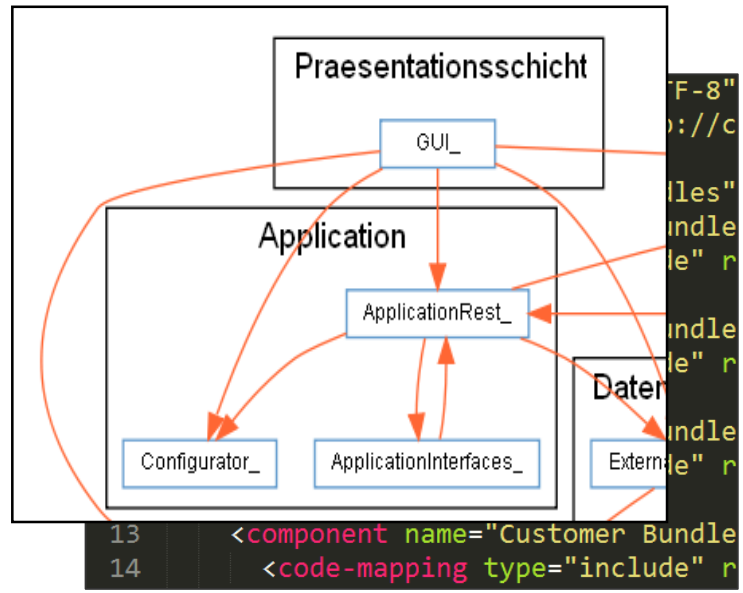


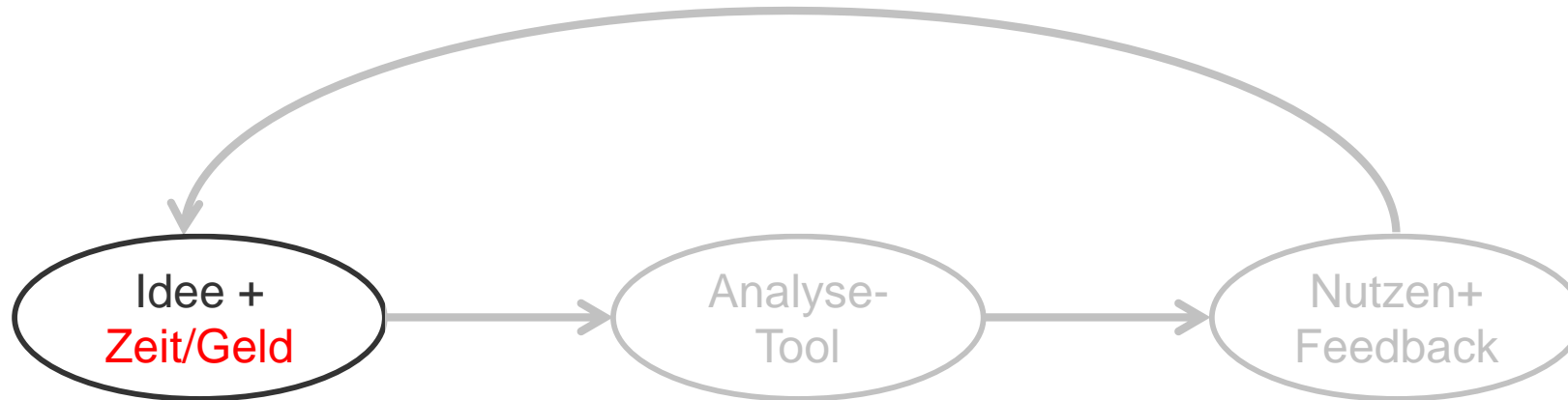


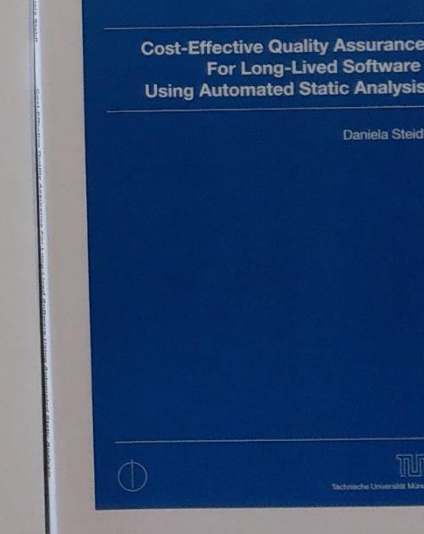
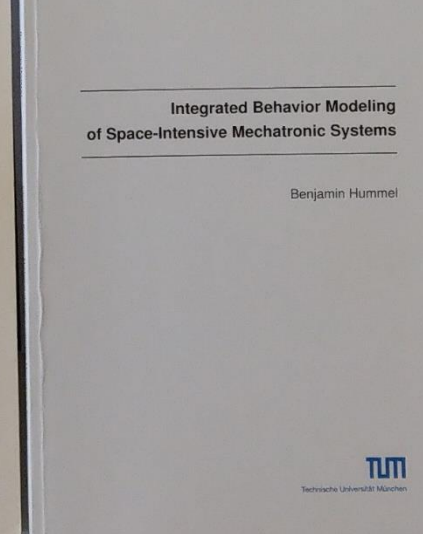
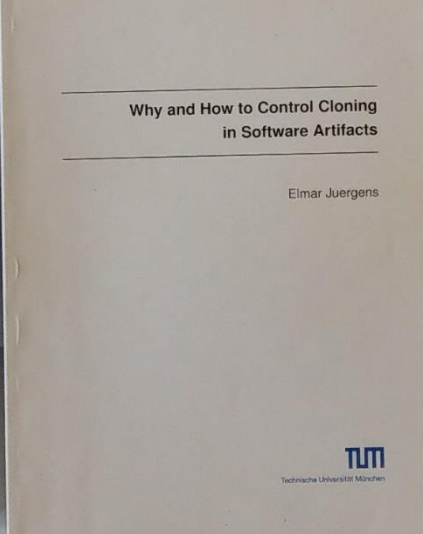
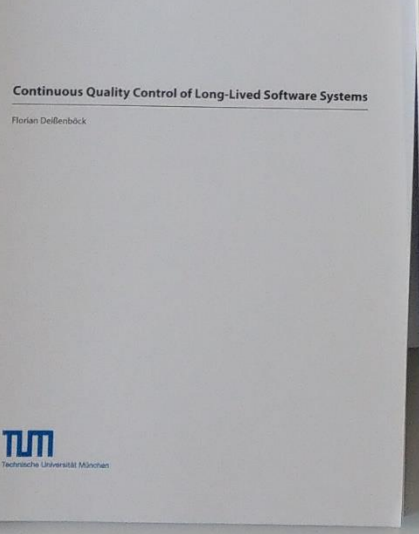
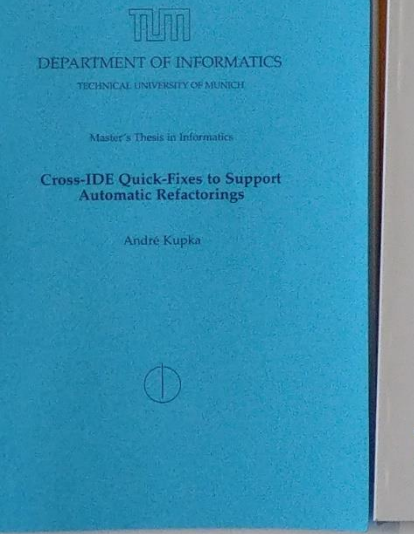
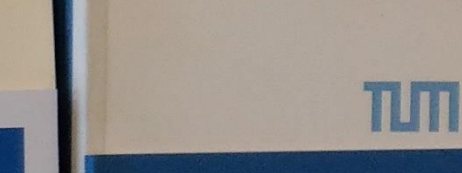
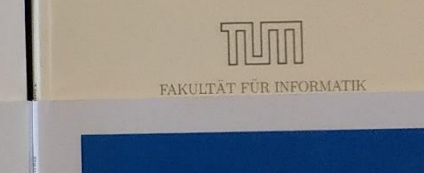
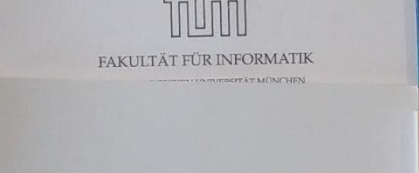
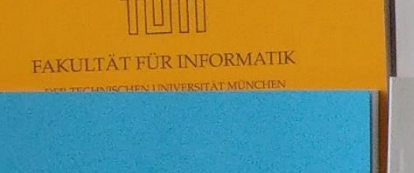
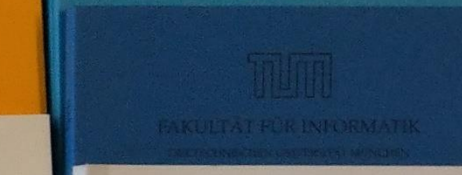
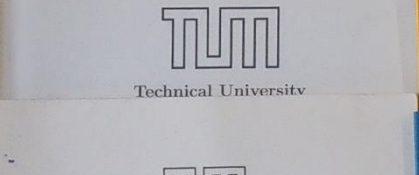
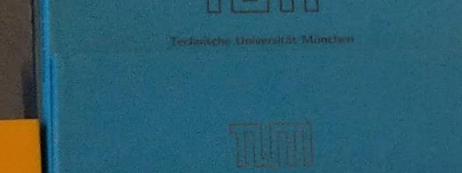
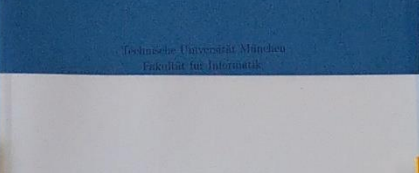
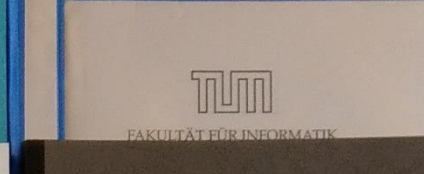
CQSE History



- Teamscale bei vielen Kunden im Einsatz
- Firma ausgelastet







Bachelor- & Masterarbeiten

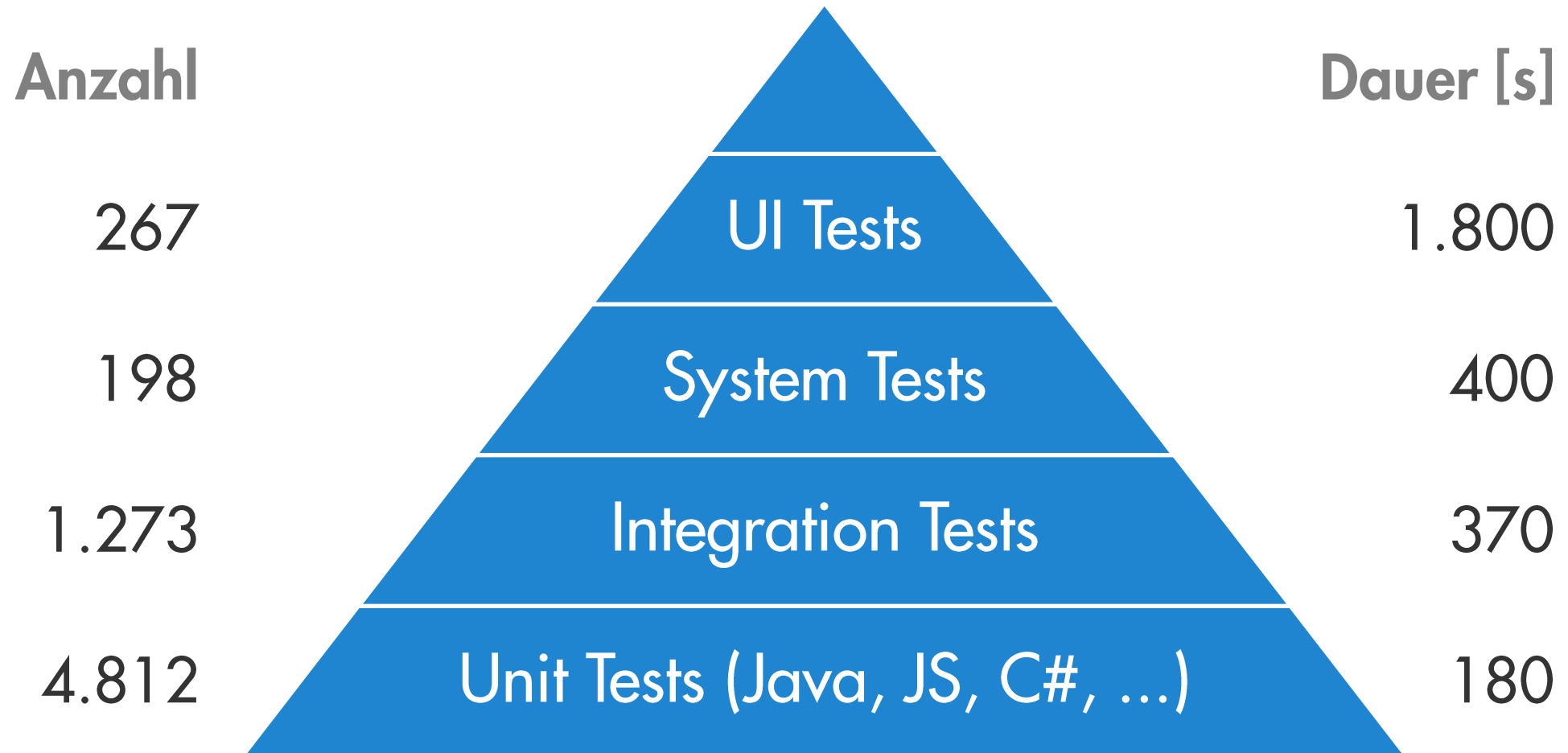
- Testballon für neue Analyse-Ideen
- Ergebnis ist Einschätzung zu Analyse, Studie usw. **KEIN CODE.**
- 10-20 pro Jahr
- Analyse-Prototyp und empirische Studie
- Eignungstest und 1 Monat (bezahlte) Einarbeitung vor Start

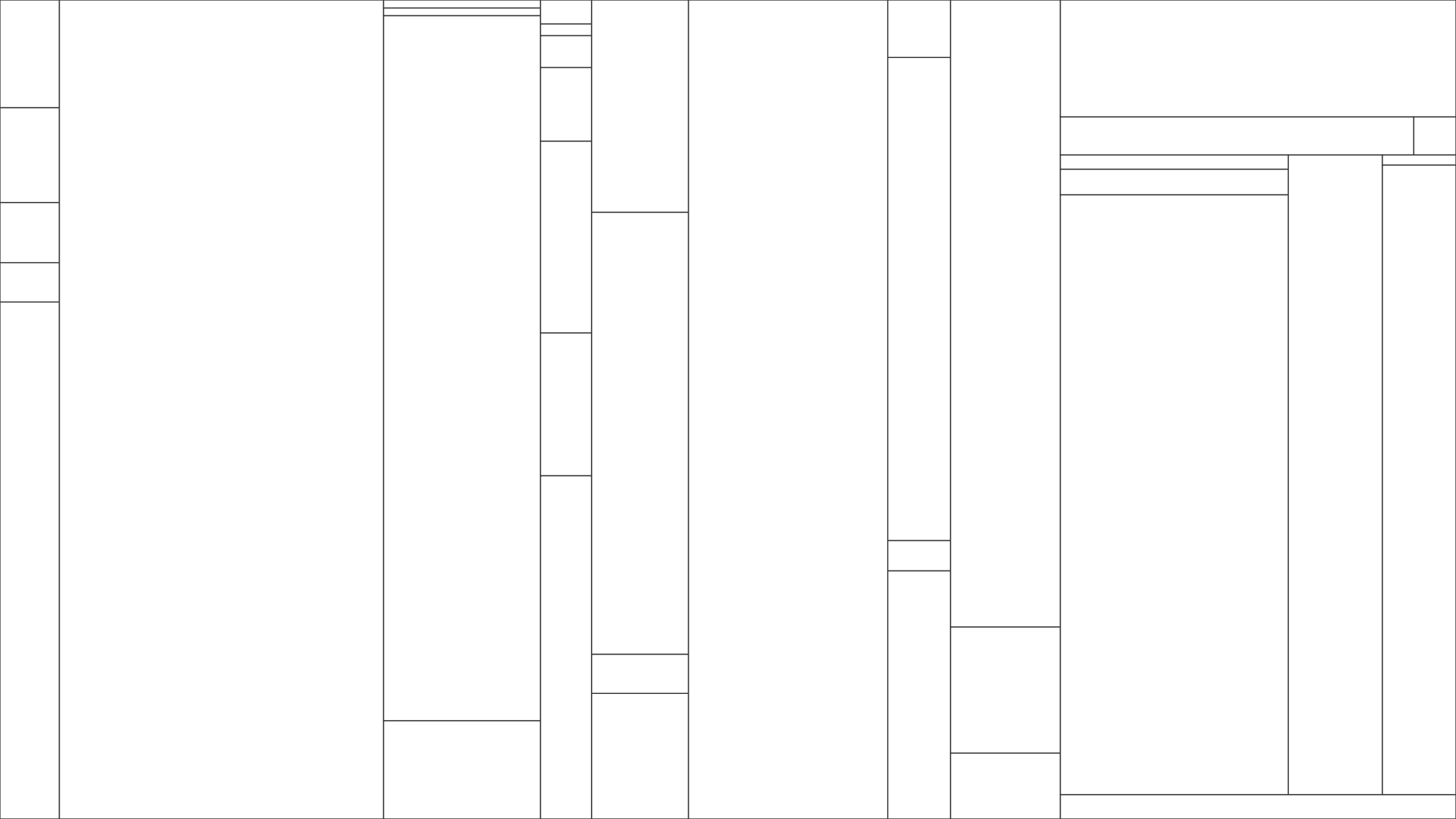
THESIS GUIDE

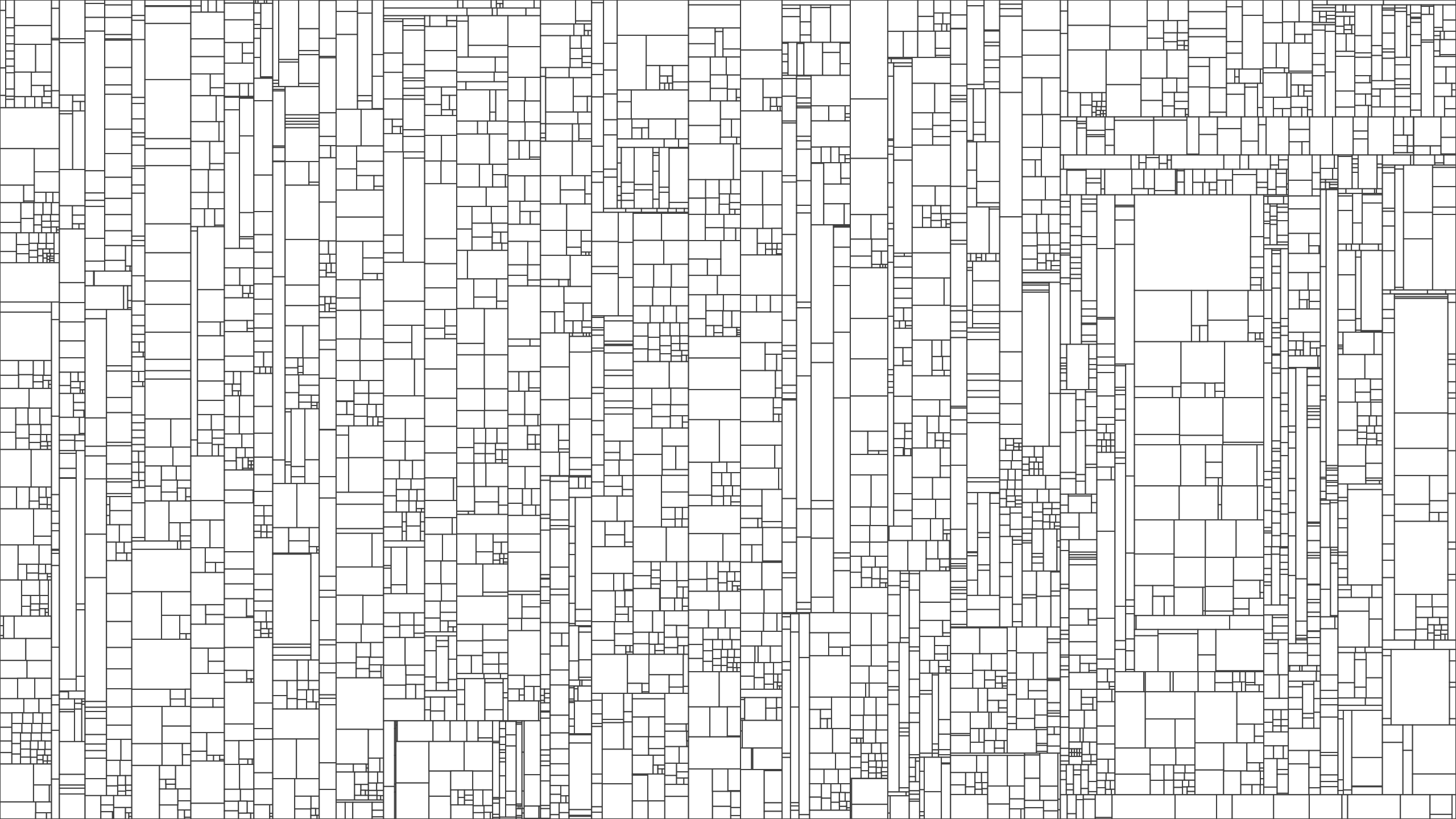
START HERE PREFACE CONTENT CONTRIBUTE ABOUT ME

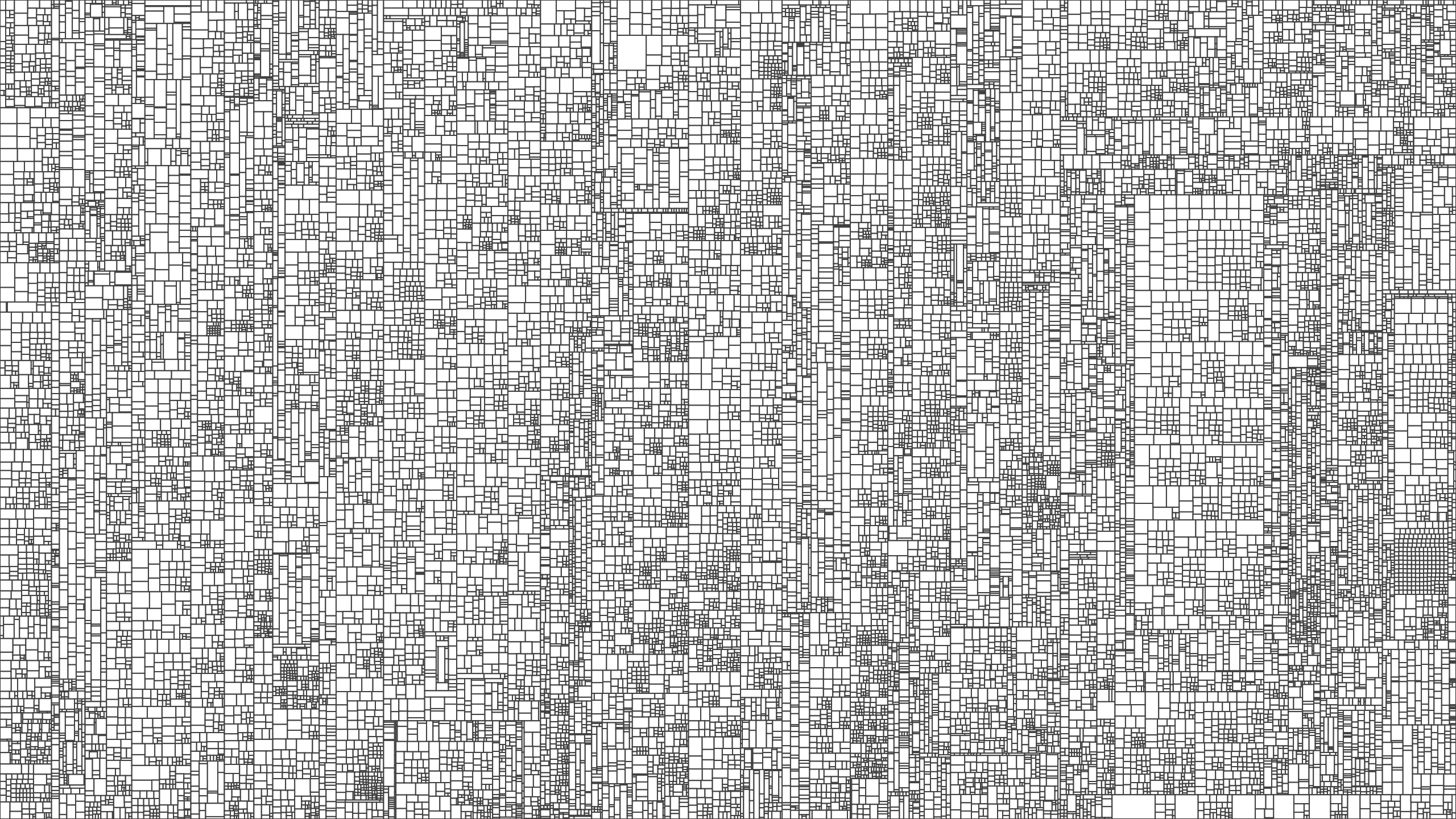


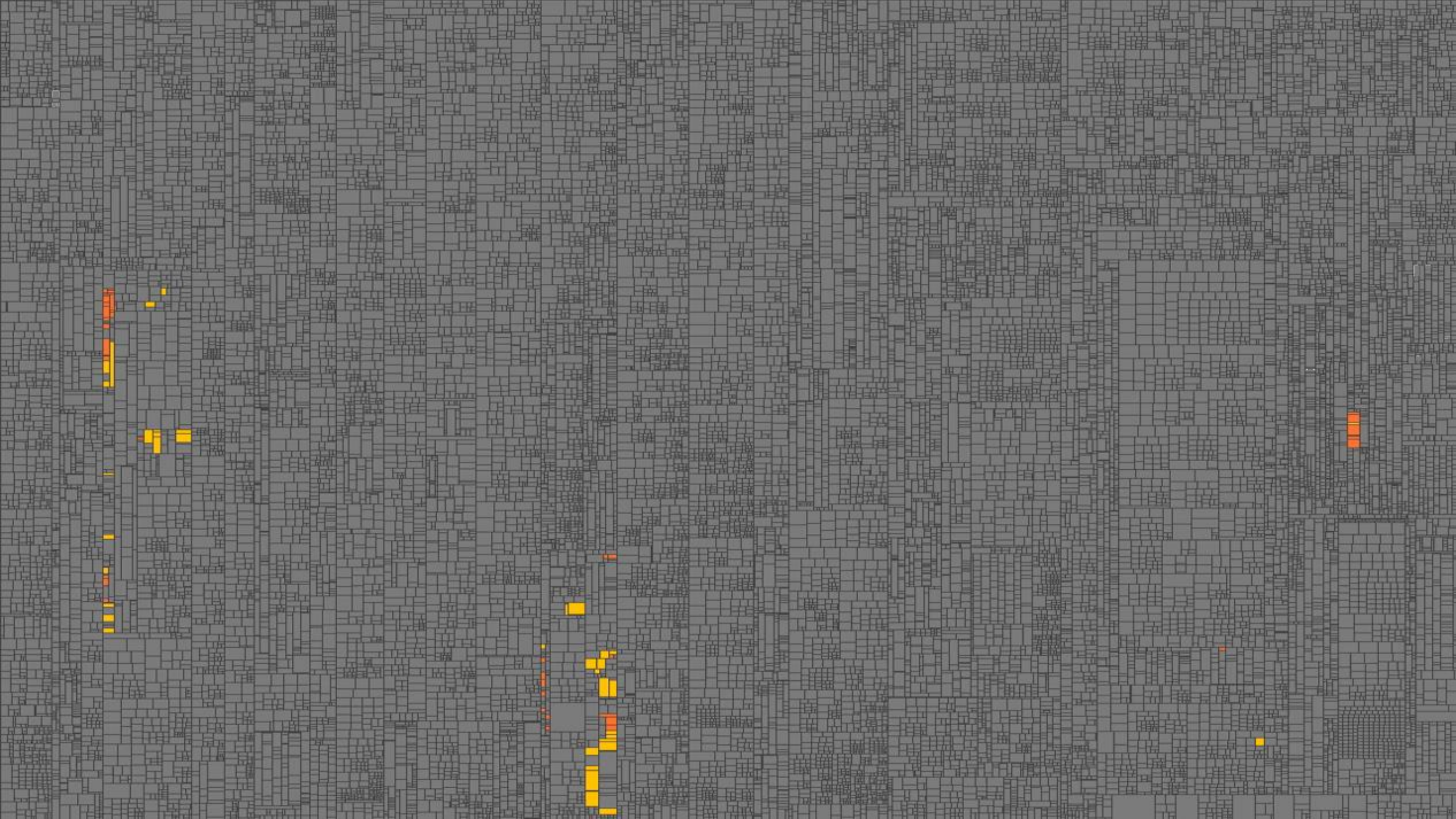
Wie wir Teamscale testen







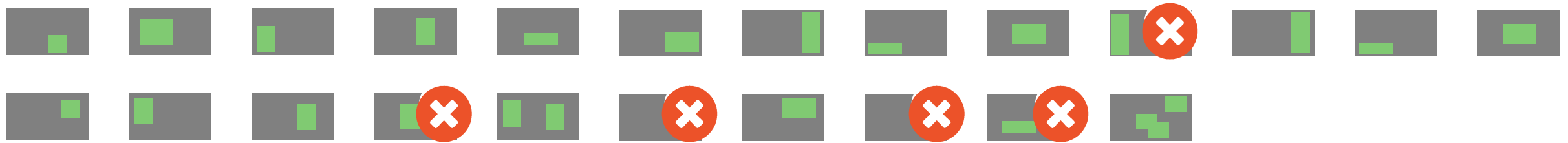




Schritt 1: Selektion betroffener Testfälle



Schritt 2: Priorisierung selektierter Testfälle

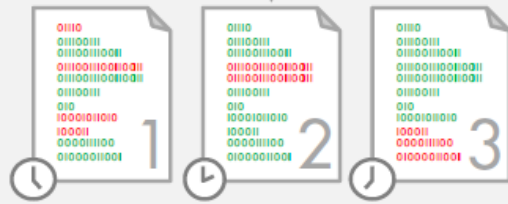




Alle Tests



Testumgebung



Testfallspezifische
Coverage und Laufzeiten



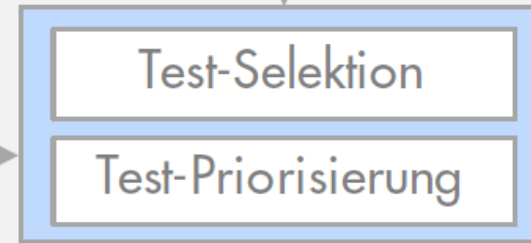
1



Versionskontrollsystem



Änderungen



Änderungsrelevante,
geordnete Teilmenge
der Tests



2

Zusammenfassung & Einordnung

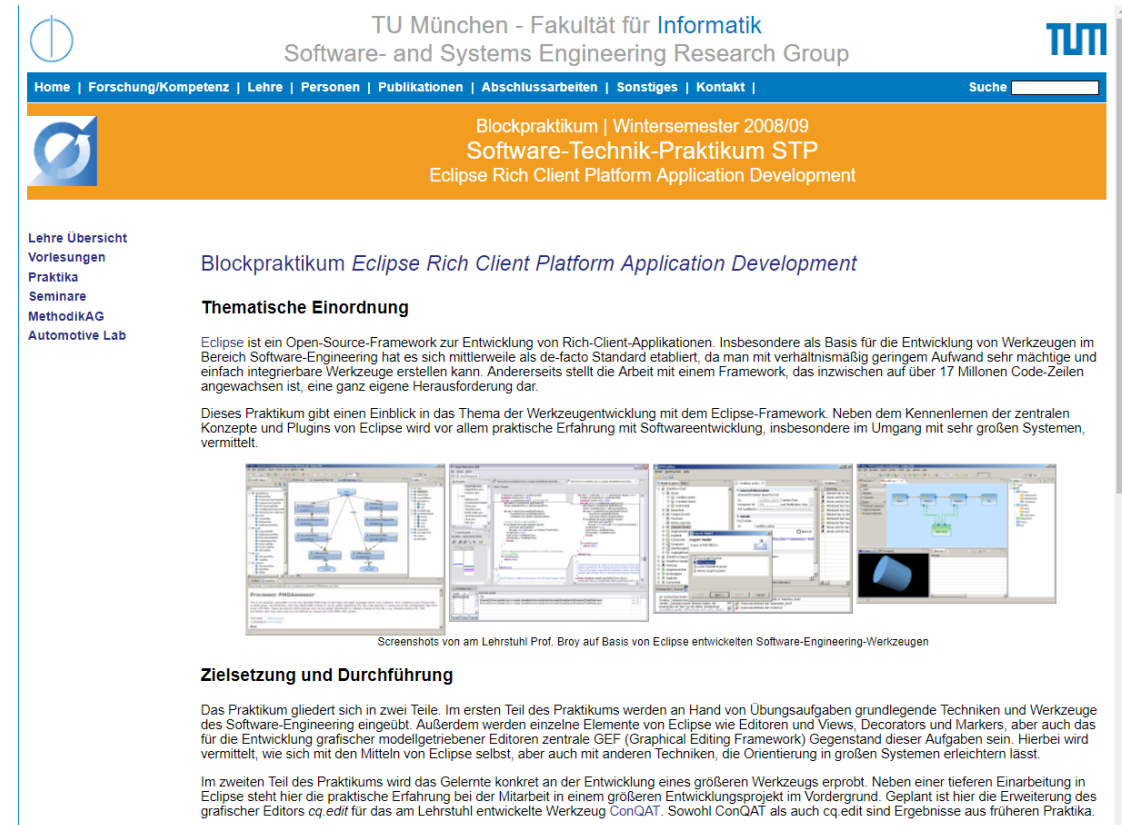
Feedback Blockpraktikum

„Gut, dass es nicht Green Field ist und dass Code nicht weggeworfen wird“

„Review-Prozess hat viel gebracht; eigene Fehler gesehen“

„hat sehr motiviert“

„Viel gelernt über Code-Qualität“



The screenshot shows the website for the Software-Technik-Praktikum STP course at TU München. The header includes the university name and faculty, along with navigation links. The main content area features a sidebar with a menu and a main section titled 'Blockpraktikum Eclipse Rich Client Platform Application Development'. The main section contains a 'Thematische Einordnung' (Thematic Placement) section, which describes the Eclipse framework and the course's focus on software engineering tools. Below the text are four screenshots of Eclipse IDE interfaces showing various tool development tasks. A caption below the screenshots reads: 'Screenshots von am Lehrstuhl Prof. Broy auf Basis von Eclipse entwickelten Software-Engineering-Werkzeugen'. The 'Zielsetzung und Durchführung' (Objectives and Execution) section follows, detailing the course structure and the development of the ConQAT tool.

TU München - Fakultät für Informatik
Software- and Systems Engineering Research Group

Home | Forschung/Kompetenz | Lehre | Personen | Publikationen | Abschlussarbeiten | Sonstiges | Kontakt | Suche

Blockpraktikum | Wintersemester 2008/09
Software-Technik-Praktikum STP
Eclipse Rich Client Platform Application Development

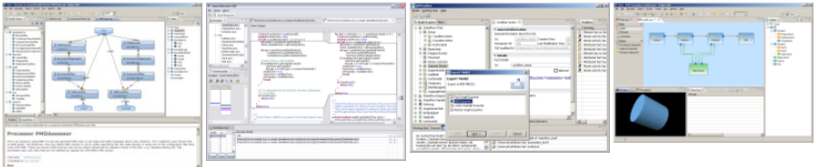
Lehre Übersicht
Vorlesungen
Praktika
Seminare
MethodikAG
Automotive Lab

Blockpraktikum *Eclipse Rich Client Platform Application Development*

Thematische Einordnung

Eclipse ist ein Open-Source-Framework zur Entwicklung von Rich-Client-Applikationen. Insbesondere als Basis für die Entwicklung von Werkzeugen im Bereich Software-Engineering hat es sich mittlerweile als de-facto Standard etabliert, da man mit verhältnismäßig geringem Aufwand sehr mächtige und einfach integrierbare Werkzeuge erstellen kann. Andererseits stellt die Arbeit mit einem Framework, das inzwischen auf über 17 Millionen Code-Zeilen angewachsen ist, eine ganz eigene Herausforderung dar.

Dieses Praktikum gibt einen Einblick in das Thema der Werkzeugentwicklung mit dem Eclipse-Framework. Neben dem Kennenlernen der zentralen Konzepte und Plugins von Eclipse wird vor allem praktische Erfahrung mit Softwareentwicklung, insbesondere im Umgang mit sehr großen Systemen, vermittelt.



Screenshots von am Lehrstuhl Prof. Broy auf Basis von Eclipse entwickelten Software-Engineering-Werkzeugen

Zielsetzung und Durchführung

Das Praktikum gliedert sich in zwei Teile. Im ersten Teil des Praktikums werden an Hand von Übungsaufgaben grundlegende Techniken und Werkzeuge des Software-Engineering eingeübt. Außerdem werden einzelne Elemente von Eclipse wie Editoren und Views, Decorators und Markers, aber auch das für die Entwicklung grafischer modellgetriebener Editoren zentrale GEF (Graphical Editing Framework) Gegenstand dieser Aufgaben sein. Hierbei wird vermittelt, wie sich mit den Mitteln von Eclipse selbst, aber auch mit anderen Techniken, die Orientierung in großen Systemen erleichtern lässt.

Im zweiten Teil des Praktikums wird das Gelernte konkret an der Entwicklung eines größeren Werkzeugs erprobt. Neben einer tieferen Einarbeitung in Eclipse steht hier die praktische Erfahrung bei der Mitarbeit in einem größeren Entwicklungsprojekt im Vordergrund. Geplant ist hier die Erweiterung des grafischer Editors *cq.edit* für das am Lehrstuhl entwickelte Werkzeug *ConQAT*. Sowohl *ConQAT* als auch *cq.edit* sind Ergebnisse aus früheren Praktika.

Timo Pawelka, Elmar Juergens:

[Is This Code Written in English? A Study of the Natural Language of Comments and Identifiers in Practice](#)
Proceedings of the 31st International Conference on Software Maintenance and Evolution (ICSME'15), 2015.

Moritz Beller, Alberto Bacchelli, Andy Zaidman, Elmar Juergens:

[Modern Code Reviews in Open-Source Projects: Which Problems Do They Find/Fix?](#)
The 11th Working Conference on Mining Software Repositories (MSR'14), 2014.

Fabian Streitl, Daniela Steidl, Elmar Jürgens:

[Dead Code Detection on Class Level.](#)
Softwaretechnik-Trends, Vol. 34, 2014.

Moritz Beller, Elmar Jürgens:

[How Strict is Your Architecture?](#)
Softwaretechnik-Trends, Vol. 32, 2012.

Daniela Steidl, Benjamin Hummel, Elmar Juergens:

[Quality Analysis of Source Code Comments](#)
Proceedings of the 21st IEEE International Conference on Program Comprehension (ICPC'13), 2013.

Martin Pöhlmann, Elmar Juergens:

[Revealing Missing Bug-Fixes in Code Clones in Large-Scale Code Bases](#)
Proceedings of the Seventh International Workshop on Software Quality and Maintainability (SQM'13), 2013.

Daniela Steidl, Benjamin Hummel, Elmar Jürgens:

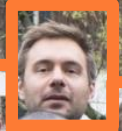
[Using Network Analysis for Recommendation of Central Software Classes.](#)
Proceedings of the 19th Working Conference on Reverse Engineering (WCRE'12), 2012.

Benjamin Hummel, Elmar Jürgens, Daniela Steidl:

[Index-based model clone detection.](#)
Proceedings of the 5th International Workshop on Software Clones (IWSC'11), 2011

Florian Deissenboeck, Benjamin Hummel, Elmar Juergens, Michael Pfaehler:

[Model Clone Detection in Practice](#)
Proceedings of the 4th International Workshop on Software Clones (IWSC'10), 2010.





Frei für Forschung und Lehre

Danke für die Aufmerksamkeit

Ich bin ganzen Tag hier & freue mich auf Diskussionen



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

CQSE GmbH
Lichtenbergstraße 8
85748 Garching bei München
www.cqse.eu

