

# Clone Detection & Management für Architektinnen und Architekten

Dr. Elmar Jürgens



1975

A man with a beard and a striped jacket is smiling and looking towards the camera. He is holding a small, flat object in his hands. He is wearing a black strap across his chest and a lanyard around his neck. The background is a plain wall.

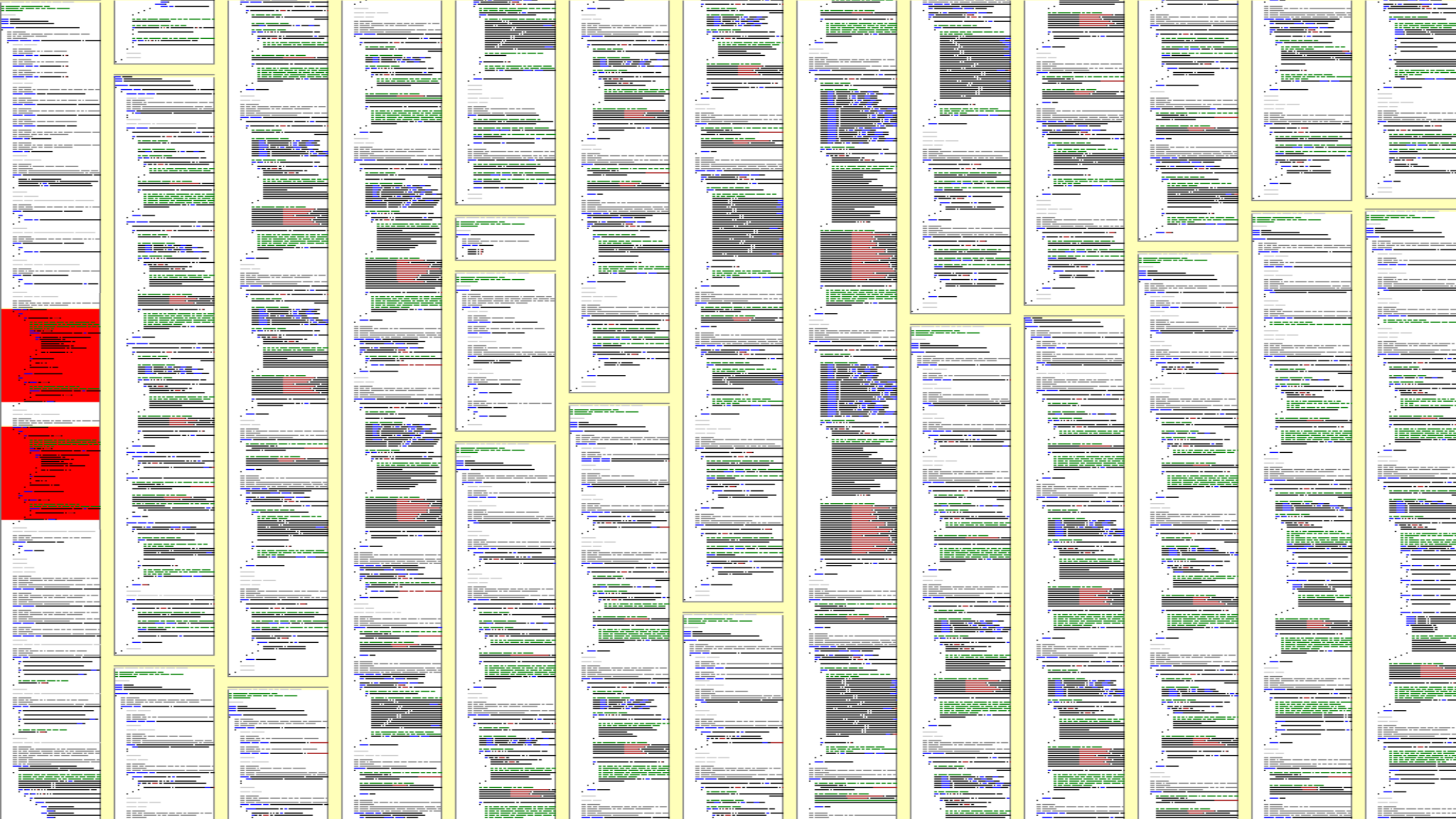
# #1 Code Smell (1999)

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

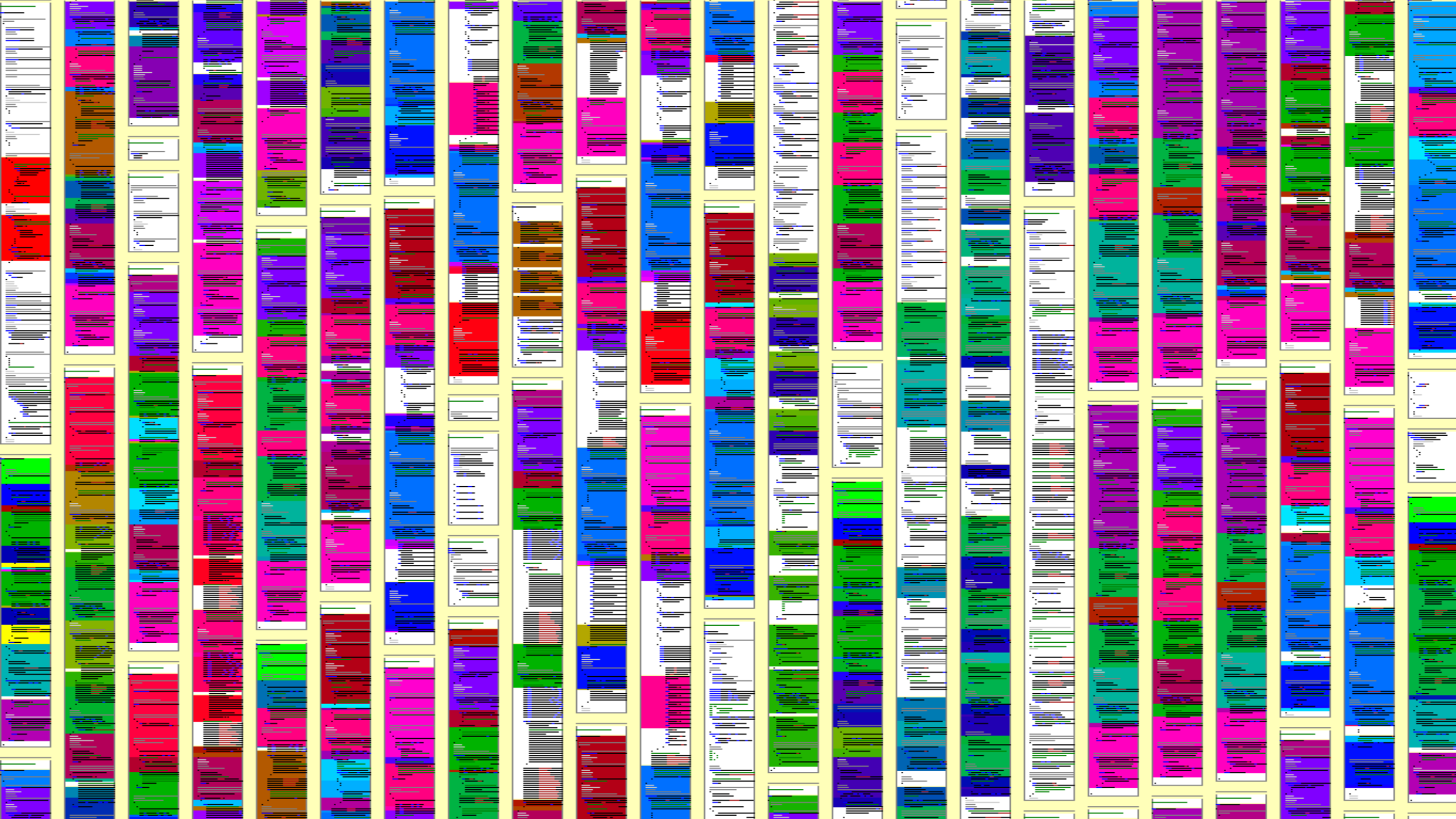
```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());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

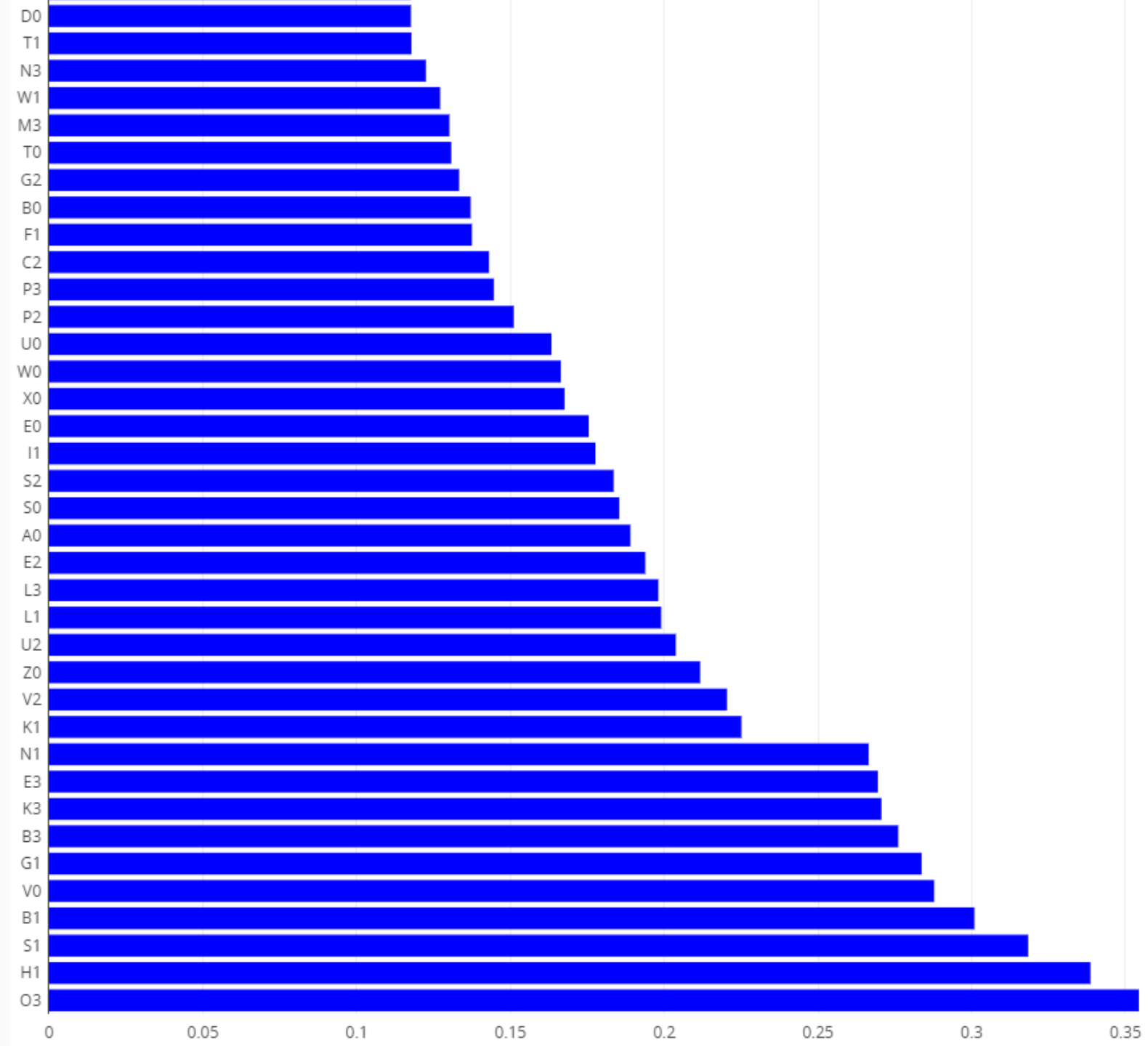










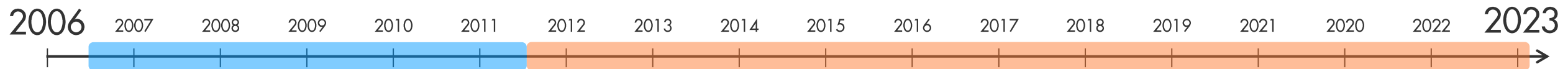




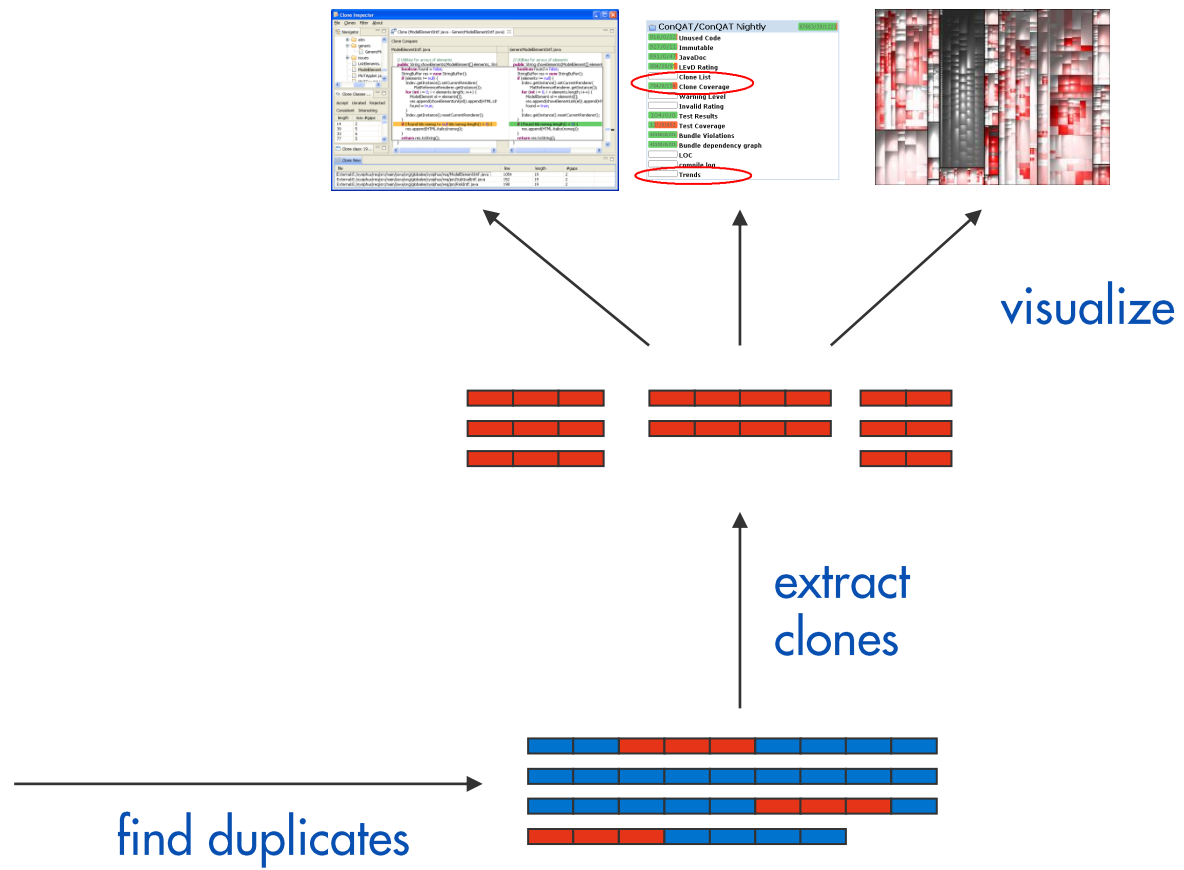
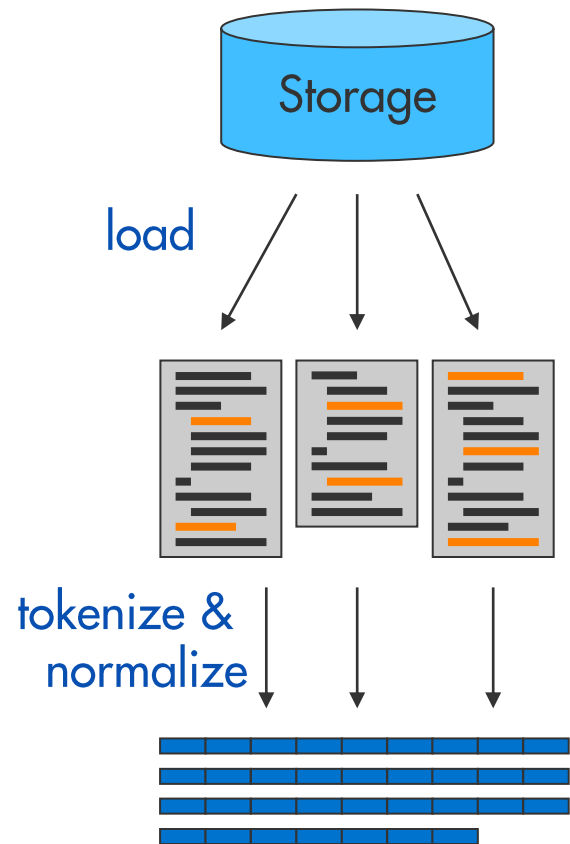
TUM

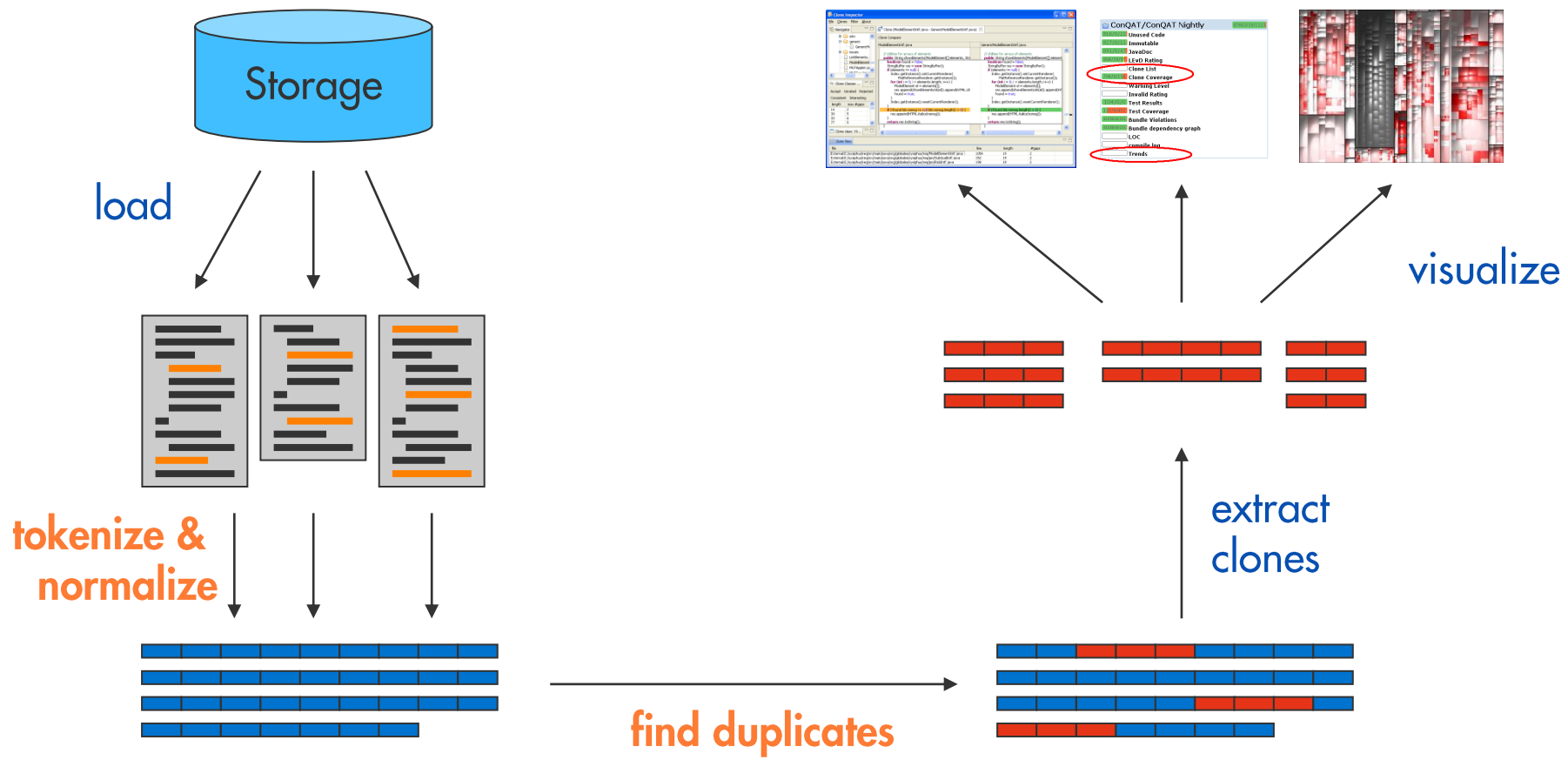


CQSE



# Wie funktioniert Clone Detection?





# 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);  
}
```

```

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.getAttributeNamespaces(i));
    if (StringUtils.isEmpty(xtr.getAttributePrefix(i)))
        extensionAttribute.setNamespacePrefix(xtr.getAttributePrefixes(i));
    if (!BpmnXMLUtil.isBlacklisted(extensionAttribute, defaultAttributes))
        model.addDefinitionsAttribute(extensionAttribute);
}
}
}

```

Clone with 2 instances of length 10

Problems @ Javadoc Declaration Findings Orphans View

4 items

Description	Group
Clone with 2 instances of length 10	Code Duplication
Interface comment missing	Documentation
Interface comment missing	Documentation
Name violates naming convention: defaultAttributes. Should be	Naming

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() ^ rtr.hasNext());
} else
if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructor(types[i])!=null))
// 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 #assertEqualsDataBoundBeans(Object, 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

```

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(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 {
                assertEquals(litem,ritem);
            }
        }
    }
    assertFalse("collection size mismatch between "+lhs+" and "+rhs, ltr.hasNext() ^ rtr.hasNext());
} else
if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructor(types[i])!=null))
// 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 #assertEqualsDataBoundBeans(Object, Object)}
/
public void assertEqualsDataBoundBeans(List<?> lhs, List<?> rhs) throws Exception {
assertEquals(lhs.size(), rhs.size());
}

```



**Sind Clones ein Problem  
für die Weiterentwicklung?**

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

- 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 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());
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());
}
    
```

```

/* Process the input string received prior to the
newline. */
do
{
    /* Pass the string to FreeRTOS+CLI. */
    xMoreDataToFollow = FreeRTOS_CLIProcessCommand( cInputString, cO

    /* Send the output generated by the command's
implementation. */
    sendto( xSocket, cOutputString, strlen( cOutputString ), 0, ( S

} while( xMoreDataToFollow != pdFALSE ); /* Until the command does n

/* All the strings generated by the command processing
have been sent. Clear the input string ready to receive
the next command. */
cInputIndex = 0;
memset( cInputString, 0x00, cmdMAX_INPUT_SIZE );

/* Transmit a spacer, just to make the command console
easier to read. */
sendto( xSocket, "\r\n", strlen( "\r\n" ), 0, ( SOCKADDR * ) &xClie
}
else
{
    if( cInChar == '\r' )
    {
        /* Ignore the character. Newlines are used to
detect the end of the input string. */
    }
    else if( cInChar == '\b' )
    {
        /* Backspace was pressed. Erase the last character
in the string - if any. */
        if( cInputIndex > 0 )
        {
            cInputIndex--;
            cInputString[ cInputIndex ] = '\0';
        }
    }
    else
    {

```

```

/* Process the input string received prior to the
newline. */
do
{
    /* Pass the string to FreeRTOS+CLI. */
    xMoreDataToFollow = FreeRTOS_CLIProcessCommand( cInputString,

    /* Send the output generated by the command's
implementation. */
    sendto( xSocket, cOutputString, strlen( cOutputString ), 0,

} while( xMoreDataToFollow != pdFALSE ); /* Until the command doe

/* All the strings generated by the command processing
have been sent. Clear the input string ready to receive
the next command. */
cInputIndex = 0;
memset( cInputString, 0x00, cmdMAX_INPUT_SIZE );

/* Transmit a spacer, just to make the command console
easier to read. */
sendto( xSocket, "\r\n", strlen( "\r\n" ), 0, ( SOCKADDR * ) &xCl

}
else
{
    if( cInChar == '\r' )
    {
        /* Ignore the character. Newlines are used to
detect the end of the input string. */
    }
    else if( ( cInChar == '\b' ) || ( cInChar == cmdASCII_DEL ) )
    {
        /* Backspace was pressed. Erase the last character
in the string - if any. */
        if( cInputIndex > 0 )
        {
            cInputIndex--;
            cInputString[ cInputIndex ] = '\0';
        }
    }
    else
    {

```

```

if i_kart-life_value < 0.
  write '50'          to it_down-newbs.
  i_kart-life_value = i_kart-life_value * -1.
  write i_kart-life_value to it_down-wrbtr
                        currency i_kost_ges-currency.

endif.
it_down-xblnr      = w_xblnr.
it_down-zuonr     = w_zuonr.
it_down-sgtxt     = w_sgtxt.
it_down-kostl     = i_kost_ges-fi_life_cost_center.
it_down-zzvber    = '10'.
it_down-bukrs     = i_kost_ges-booking.
*
if it_down-bukrs ne '0001'.
  write 'SA'        to it_down-blart.
  clear it_down-zzvber.
endif.
*
replace all occurrences of '.' in it_down-wrbtr with ''.
append it_down.
endloop.
*
describe table it_down lines l_it_down.
it_down-blkz = 'E'.
modify it_down index l_it_down transporting blkz.
endloop.
*
*--- Non-Life Quartal
loop at i_kost_ges where period_type = 'q' and
                        non_life_value ne 0.
                        and fehler ne 'X'.
*
  if i_kost_ges-fibu_type = 'Y'.
    d_fibu_type = text-k01.
  .

```

```

if i_kart-non_life_value < 0.
  write '50'          to it_down-newbs.
  i_kart-non_life_value = i_kart-non_life_value * -1.
  write i_kart-non_life_value to it_down-wrbtr
                        currency i_kost_ges-currency.

endif.
*
it_down-xblnr      = w_xblnr.
it_down-zuonr     = w_zuonr.
it_down-sgtxt     = w_sgtxt.
it_down-kostl     = i_kost_ges-fi_non_life_cost_center.
it_down-zzvber    = '10'.
it_down-bukrs     = i_kost_ges-booking.
*
if it_down-bukrs ne '0001'.
  write 'SA'        to it_down-blart.
  clear it_down-zzvber.
endif.
*
append it_down.
*
endloop.
describe table it_down lines l_it_down.
it_down-blkz = 'E'.
modify it_down index l_it_down transporting blkz.
endloop.
*
*--- Monatlich Health
loop at i_kost_ges where period_type = 'm' and
                        health_value ne 0.
                        fehler ne 'X'.
*
  if i_kost_ges-fibu_type = 'Y'.
    d_fibu_type = text-k01.
  .

```

larger/source/.../PriceElementUtil.plsql (revision default:1... )

```

223     AND NVL(t.valid_from, Database_SYS.Get_First_C
224     AND NVL(t.valid_to, Database_SYS.Get_Last_Cale
225
226 BEGIN
227     OPEN get_price_interval;
228     FETCH get_price_interval INTO valid_from_, valid_ur
229     CLOSE get_price_interval;
230     IF valid_from_ > Site_API.Get_Site_Date(contract_)
231         valid_date_ := valid_from_;
232     ELSE
233         valid_date_ := LEAST(NVL(valid_until_, TRUNC(Sit
234     END IF;
235
236     -- only calculate unit base (False) but no metal su
237     temp1_ :=0;
238
239     OPEN get_pl_currency_code;
240     FETCH get_pl_currency_code INTO pl_currency_code_;
241     CLOSE get_pl_currency_code;
242
243     FOR rec_ IN get_all_conn_unit_base_false LOOP

```

larger/source/.../PriceElementUtil.plsql (revision default:1... )

```

346 BEGIN
347     OPEN get_price_interval;
348     FETCH get_price_interval INTO valid_from_, valid_ur
349     CLOSE get_price_interval;
350     IF valid_from_ > Site_API.Get_Site_Date(contract_)
351         valid_date_ := valid_from_;
352     ELSE
353         -- Change for Ticket #1234 (START)
354         --valid_date_ := LEAST(NVL(valid_until_, TRUNC($
355         valid_date_ := LEAST(NVL(valid_until_, Site_API.
356         -- Change for Ticket #1234 (FINISH)
357     END IF;
358
359     -- calculate only metal surcharge without unit base
360     temp3_ :=0;
361
362     OPEN get_pl_currency_code;
363     FETCH get_pl_currency_code INTO pl_currency_code_;
364     CLOSE get_pl_currency_code;
365
366     FOR rec_ IN get_all_metal_surcharge LOOP

```

```

54     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, polLocation.County)
55     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, null)
56 })
57 tierQuery.or(\orCriteria -> {
58     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, polLocation.City)
59     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, null)
60 })
61 tierQuery.or(\orCriteria -> {
62     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, polLocation.PostalCode)
63     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, null)
64 })
65
66 var results = tierQuery.select()
67 if(results.HasElements) {
68     var result = results.firstWhere( \ elt -> elt.ZipCode == polLocation.PostalCode and
69         elt.County == polLocation.County and elt.City == polLocation.City)
70     var result2 = results.firstWhere( \ elt -> elt.County == polLocation.County and elt.City == polLocation.City)
71     var result3 = results.firstWhere( \ elt -> elt.County == polLocation.County)
72
73     if(result != null) {

```

```

54     orCriteria.compareIgnoreCase(TerrorismTargetTier_Ext#County, Equals, polLocation.County)
55     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, null)
56 })
57 tierQuery.or(\orCriteria -> {
58     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, polLocation.City)
59     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, null)
60 })
61 tierQuery.or(\orCriteria -> {
62     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, polLocation.PostalCode)
63     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, null)
64 })
65
66 var results = tierQuery.select()
67 if(results.HasElements) {
68     var result = results.firstWhere( \ elt -> elt.ZipCode == polLocation.PostalCode and
69         elt.County == polLocation.County and elt.City == polLocation.City)
70     var result2 = results.firstWhere( \ elt -> elt.County == polLocation.County and elt.City == polLocation.City)
71     // - Defect 224 - Fix for defaulting target tier value irrespective of lower and upper
72     var result3 = results.firstWhere( \ elt -> elt.County.equalsIgnoreCase(polLocation.County))
73
74     if(result != null) {

```

## Study

Munich Re 

- Over 100 Bugs in produktive Software

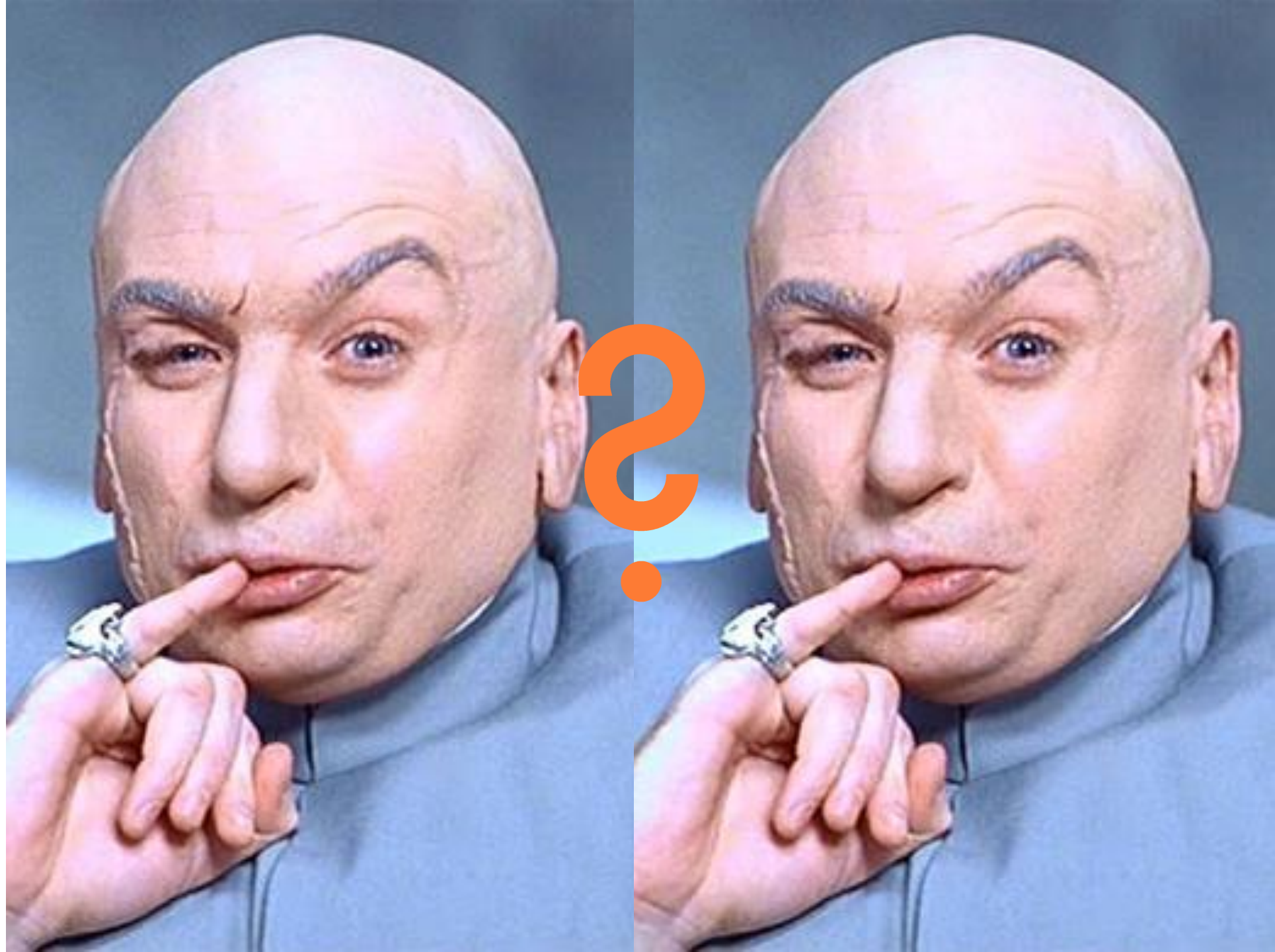


- 52% of all unintentional inconsistencies are buggy

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



Was tun?



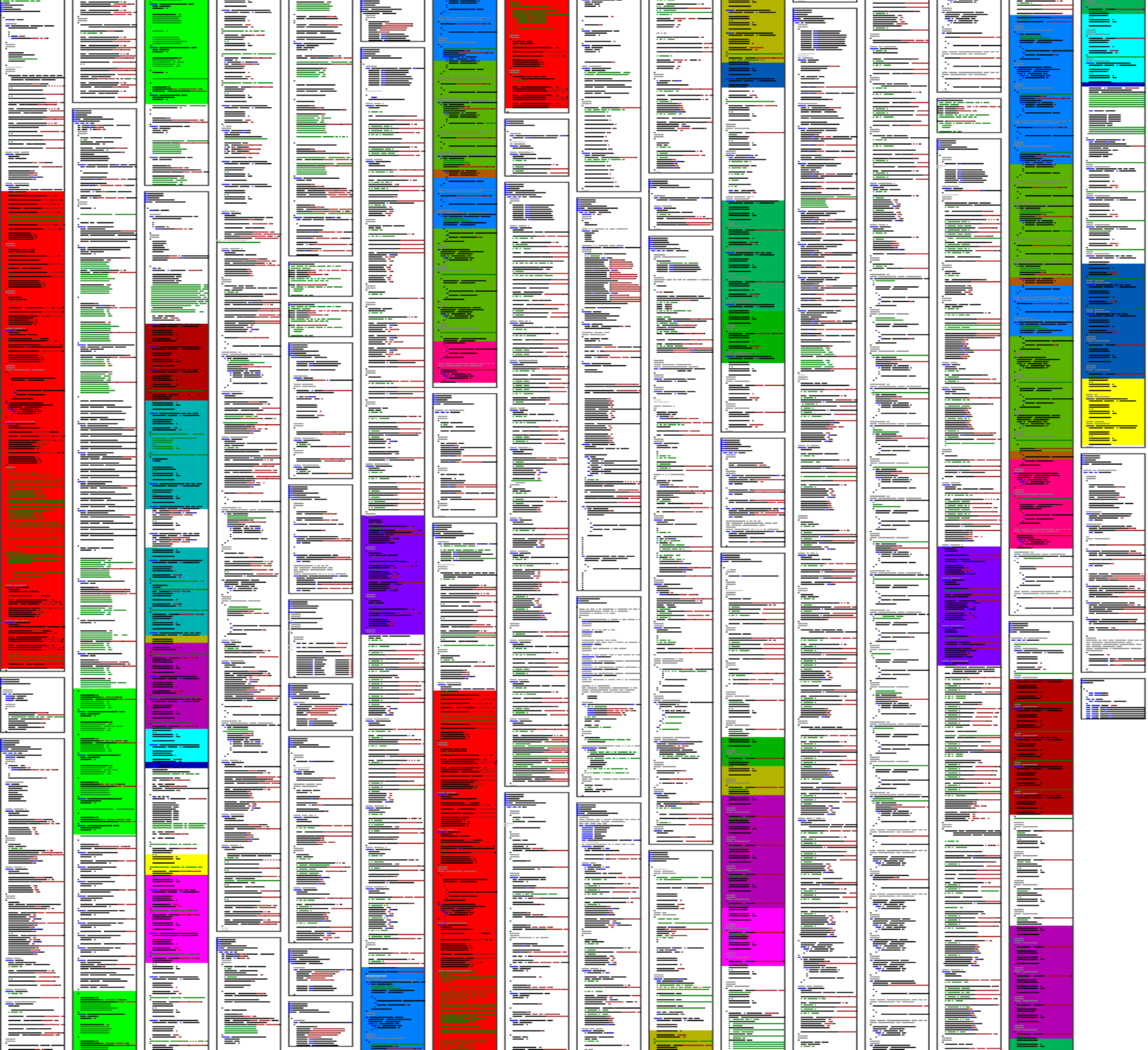




Komponente A



Komponente B



## Copy & Paste in Build-Konfigurationen (MS Build)



	Ad hoc Reuse
Build Definitions	81
Lines of Code	457.388
Clone Coverage	97,5%
Blow-Up	929%

Blow-Up: 0%



20 Lines

20 Lines

Blow-Up: 50%



20 Lines

20 Lines

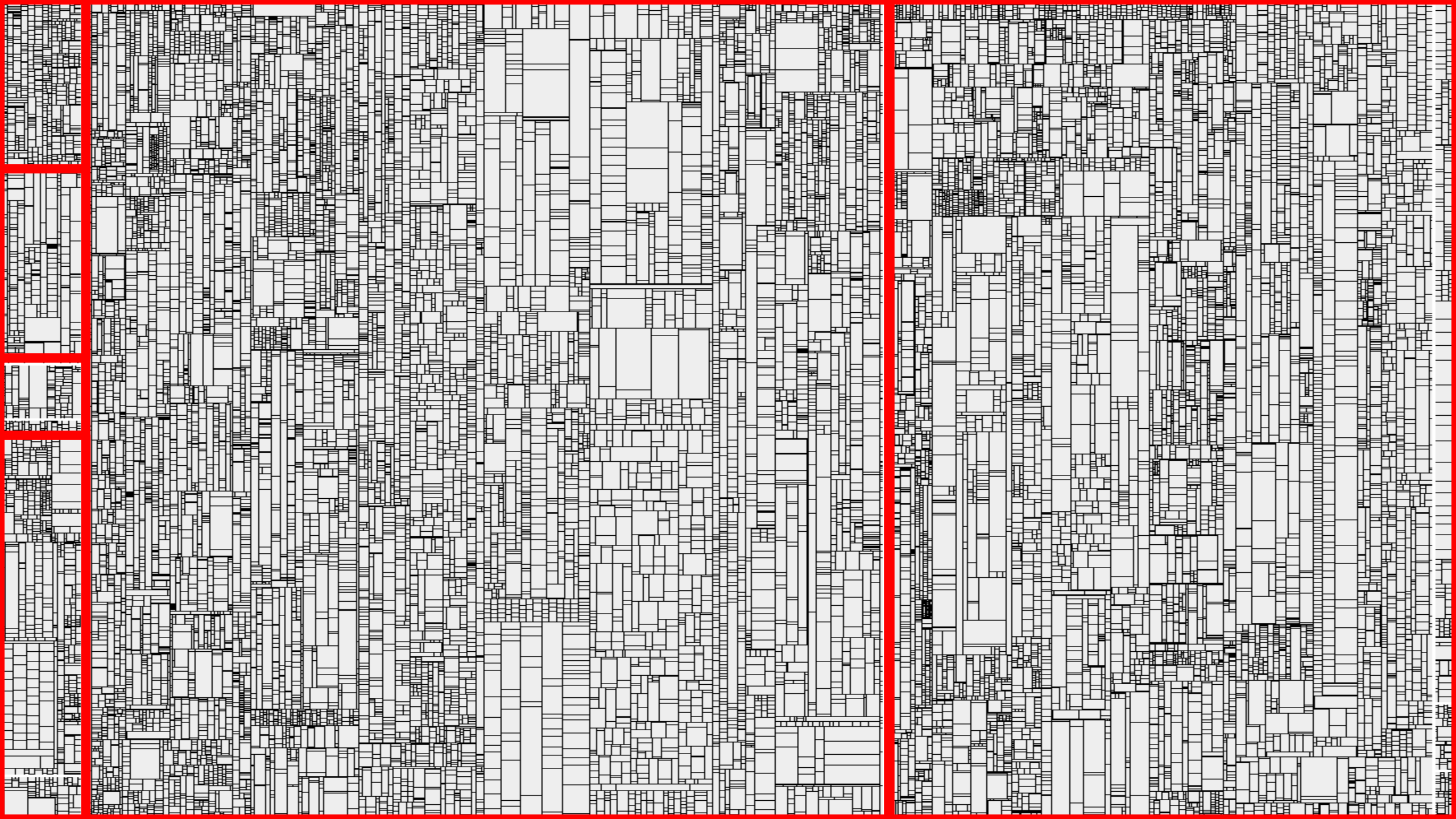
20 Lines



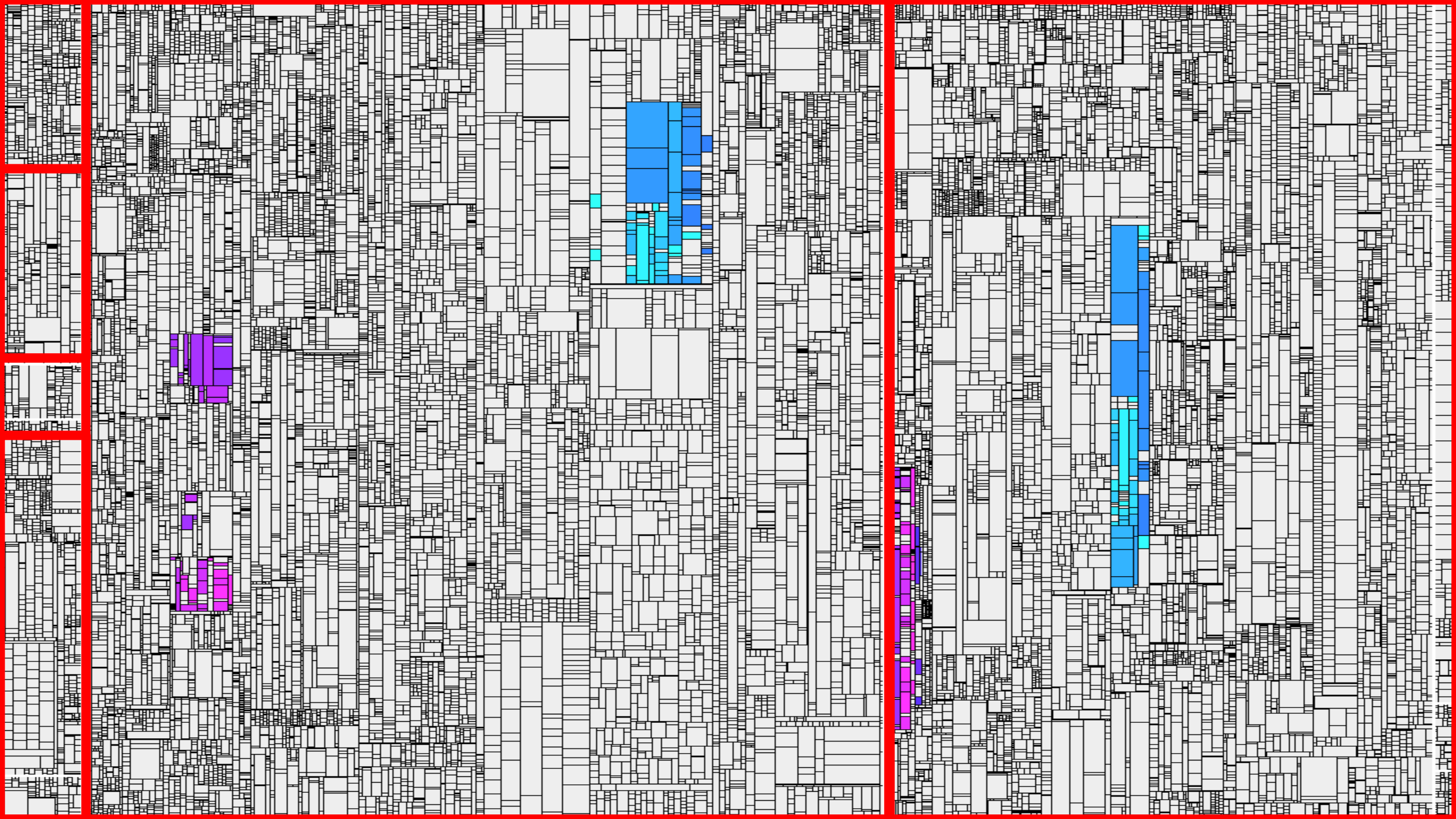
## Copy & Paste in Build-Konfigurationen (MS Build)

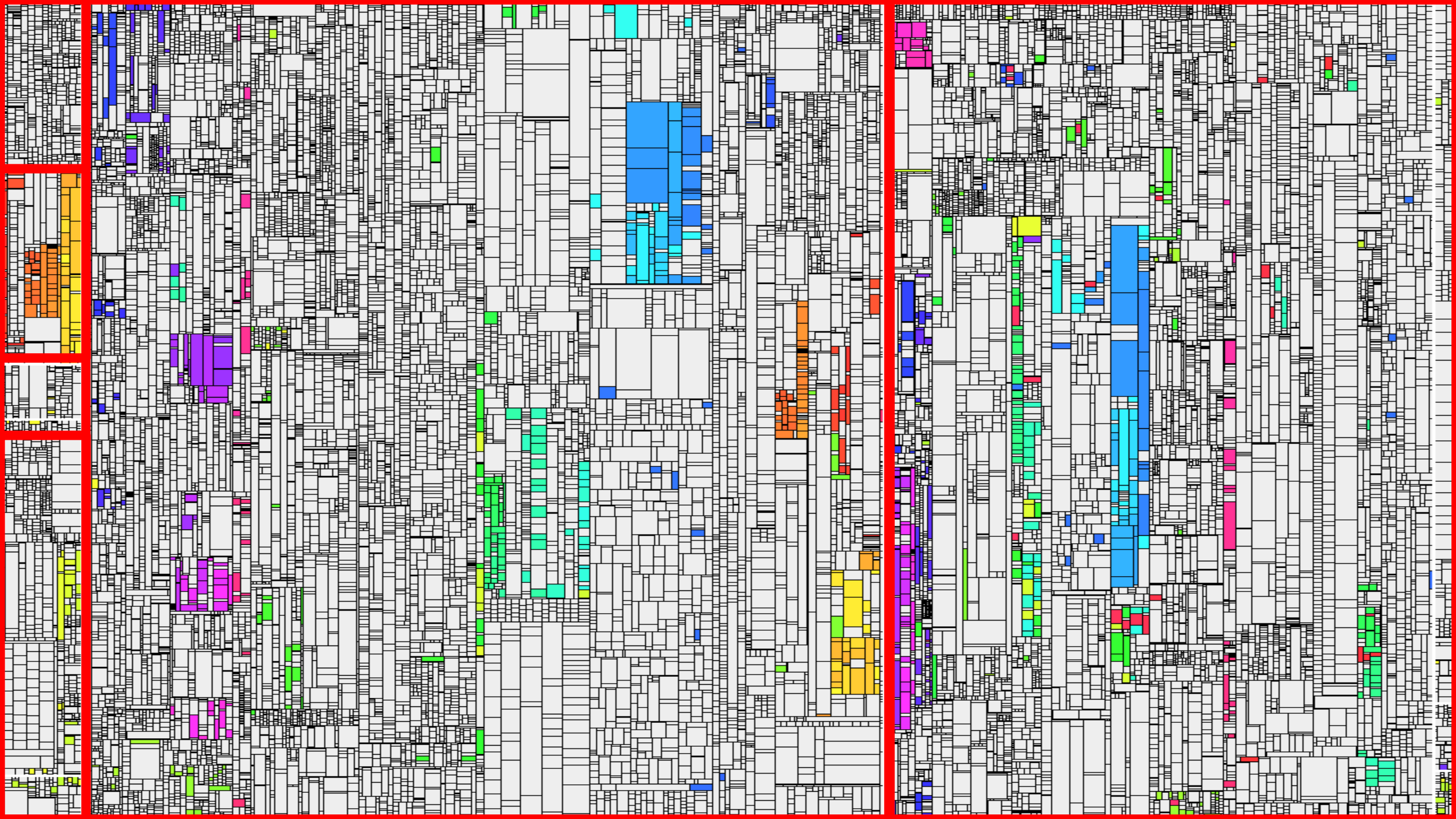


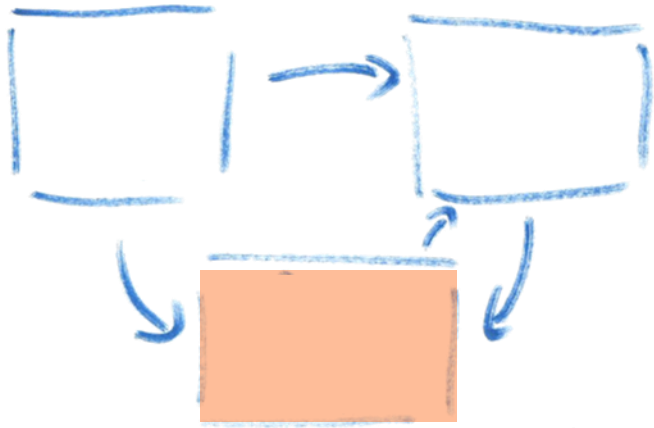
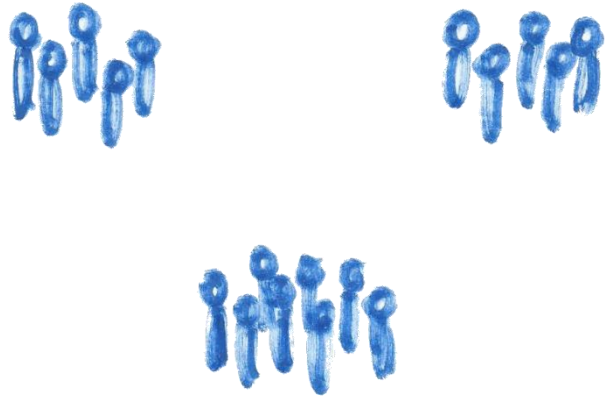
	Ad hoc Reuse
Build Definitions	81
Lines of Code	457.388
Clone Coverage	97,5%
Blow-Up	929%

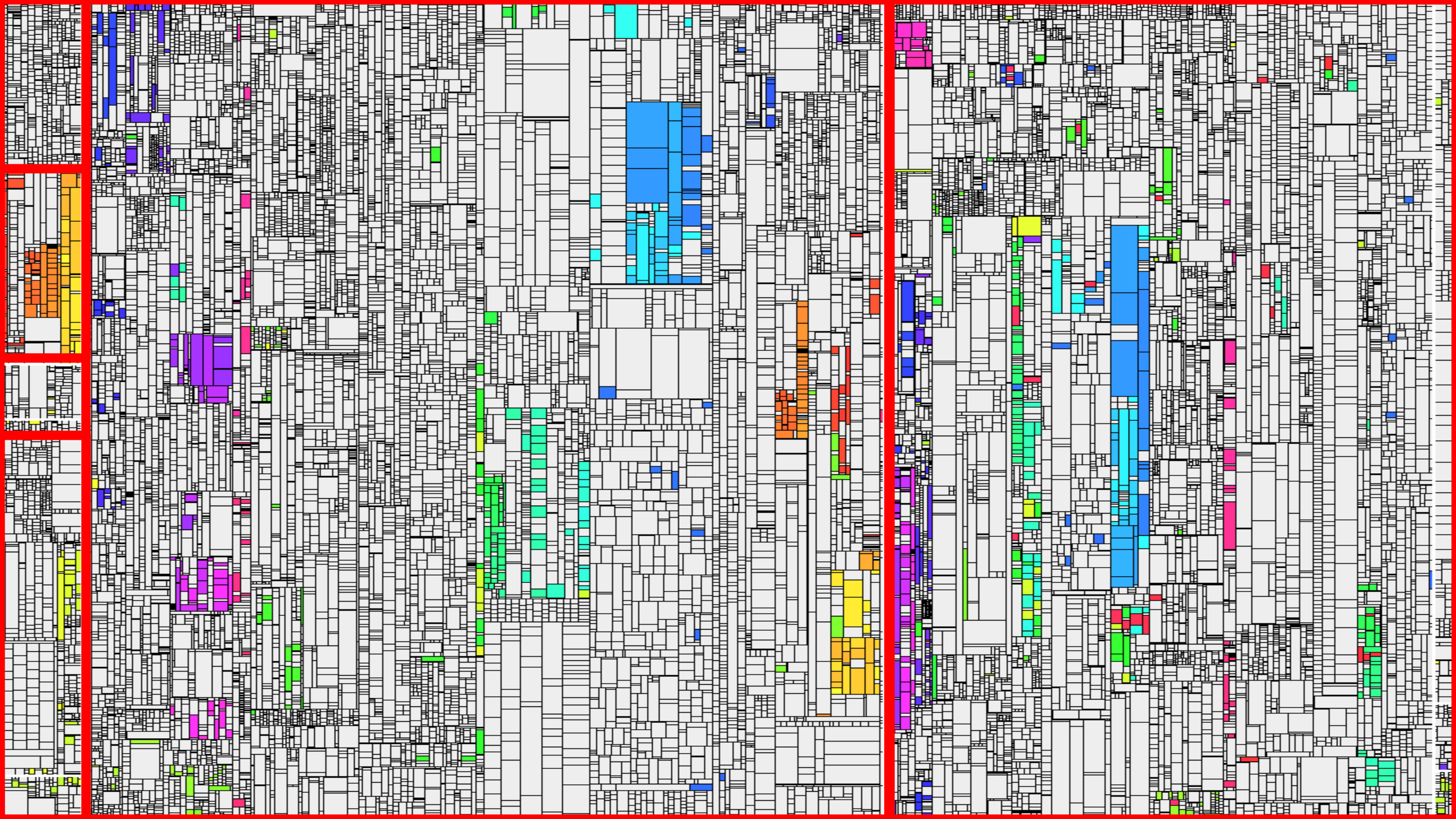












- Dashboard
- Activity
- Findings
- Metrics
- Tests
- Issues
- Tasks
- Architecture
- Delta
- Projects
- System
- Admin

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());
}
    
```





## Fixed an issue that broke the interactive console under the new class loading scheme

Dec 16 2016  
04:18

svn path=/; revision=13900

by Mark Hall as revision f4bb3289 in main (repository1)

### ^ Affected files 1

Show impacted tests

Path	Change type
<a href="#">trunk/packages/internal/RPlugin/src/main/java/weka/gui/beans/JavaGDConsolePanel.java</a>	changed

### ^ Alerts 1

Message	Context
Found potential inconsistent clone change in JavaGDConsolePanel.java	<span>Broken clone</span> <span>Old clone finding</span> <span>Code change</span>

### ^ Findings 1 51 2

# Evolution von Produktlinien

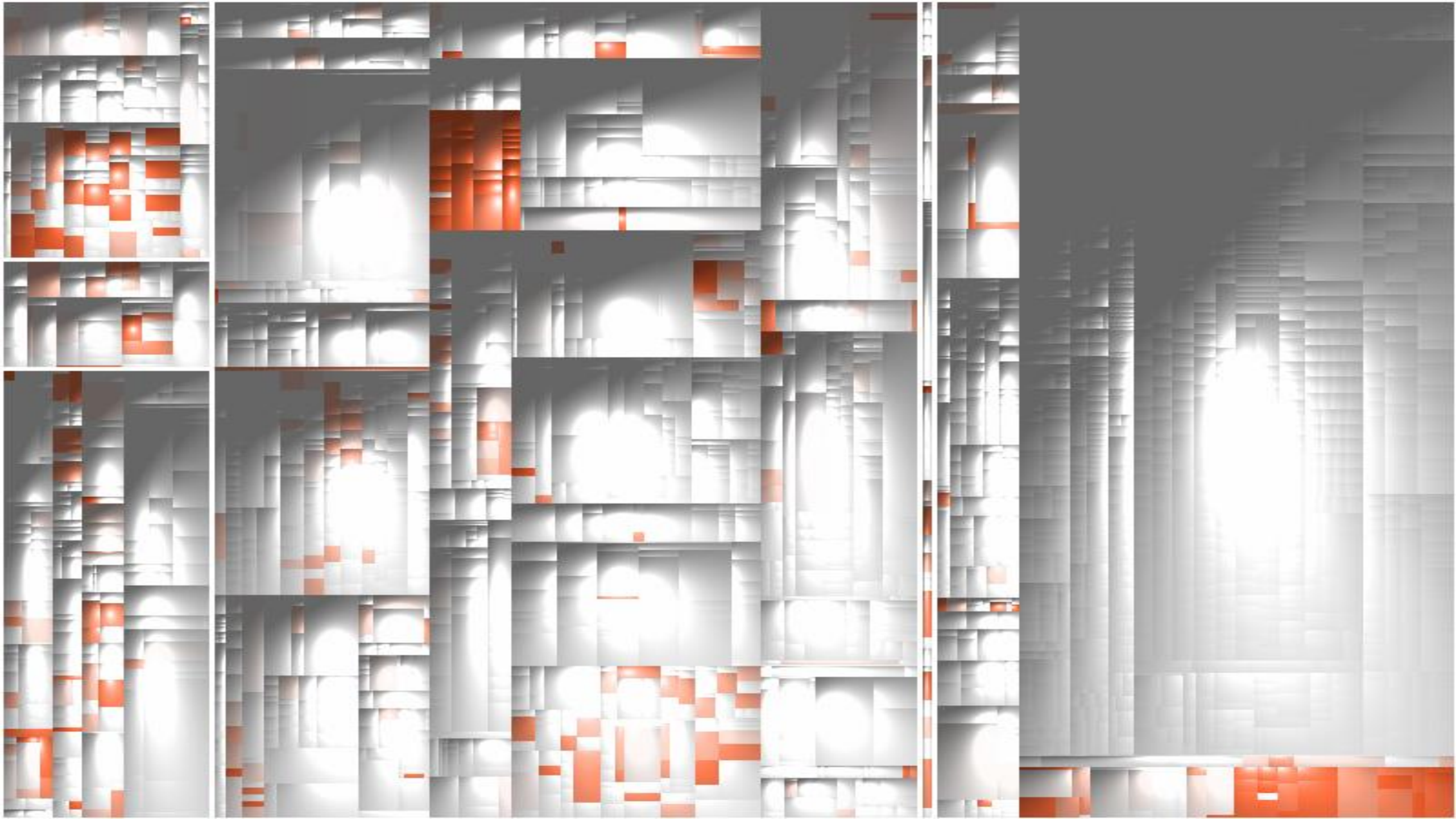
Product 1

Product 2

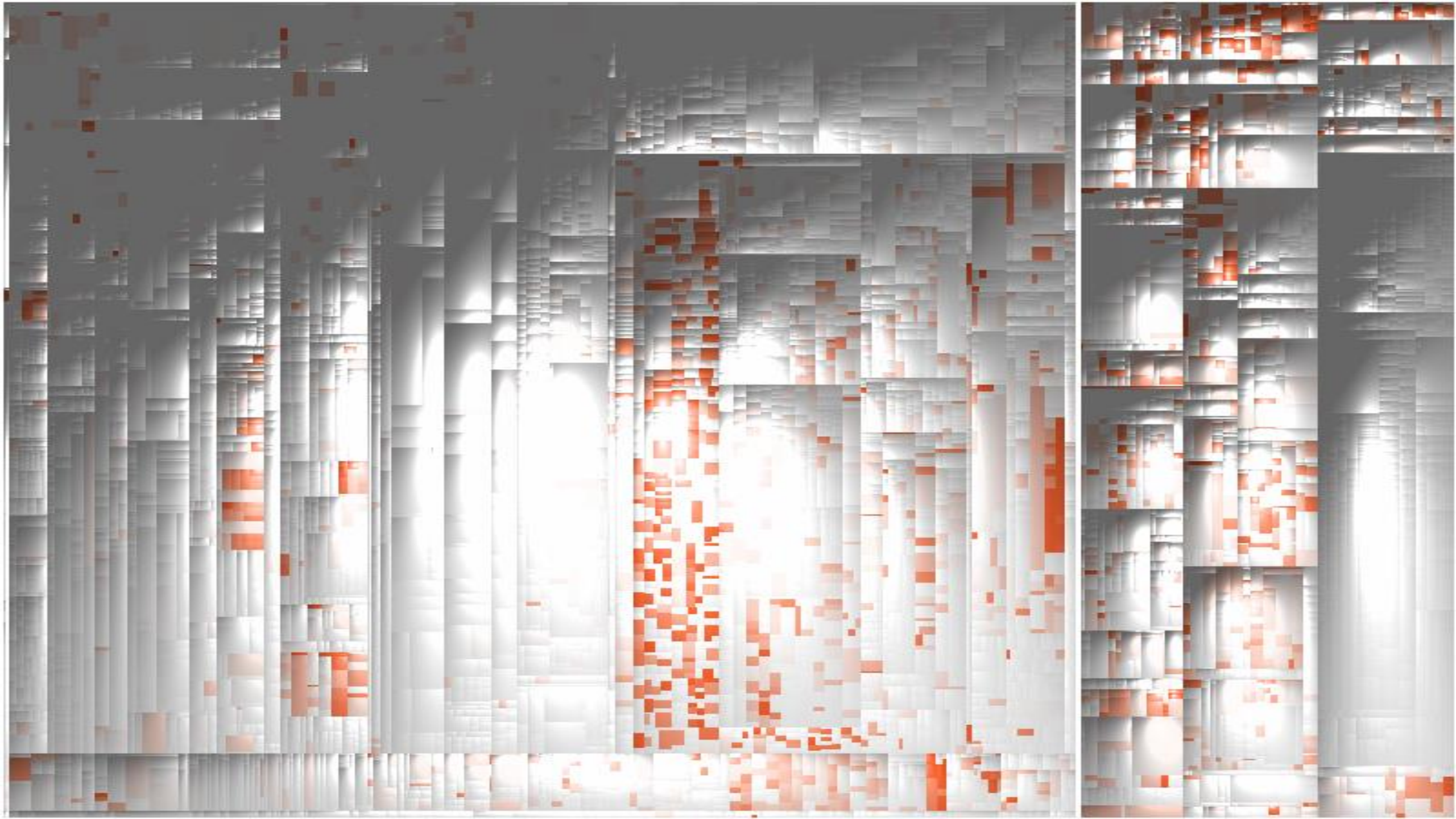
Product 3

Product 4

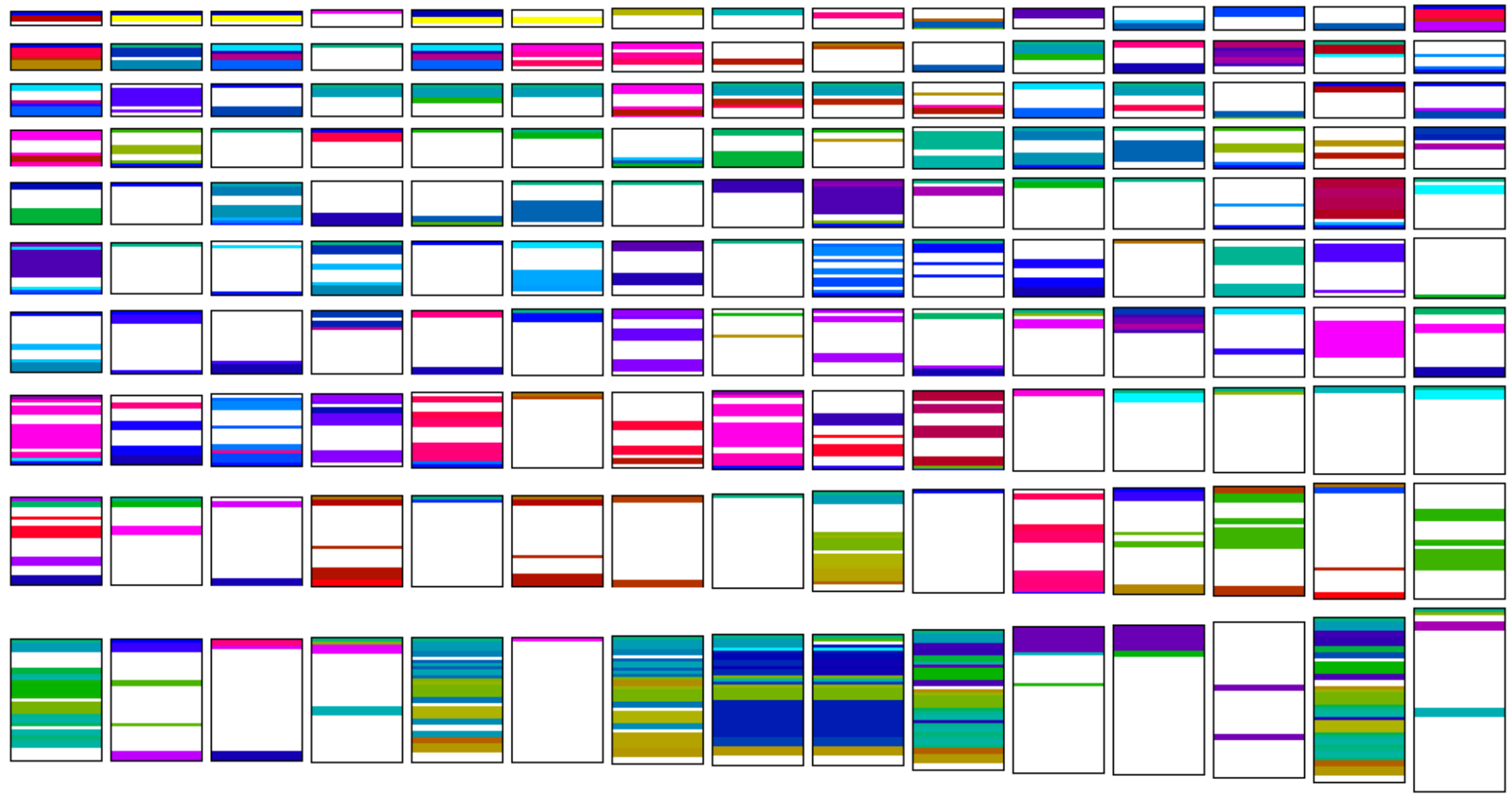
Product 5







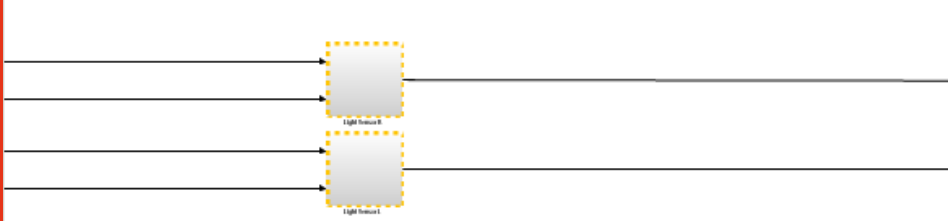
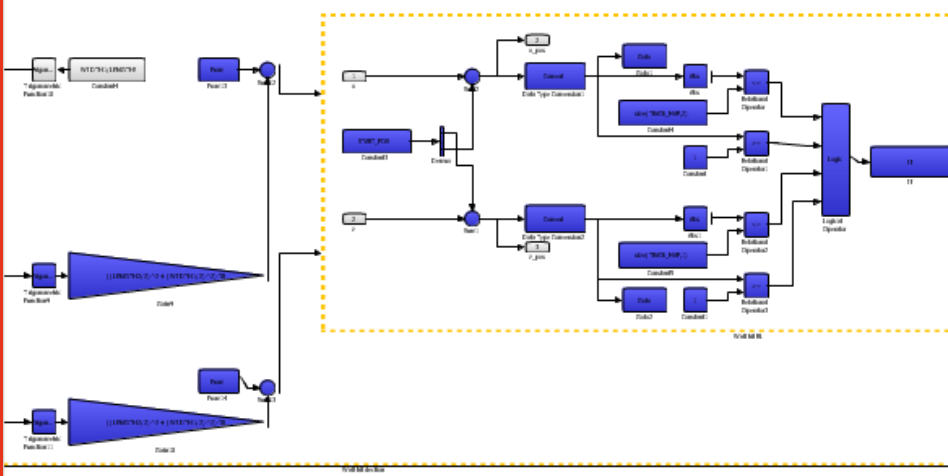
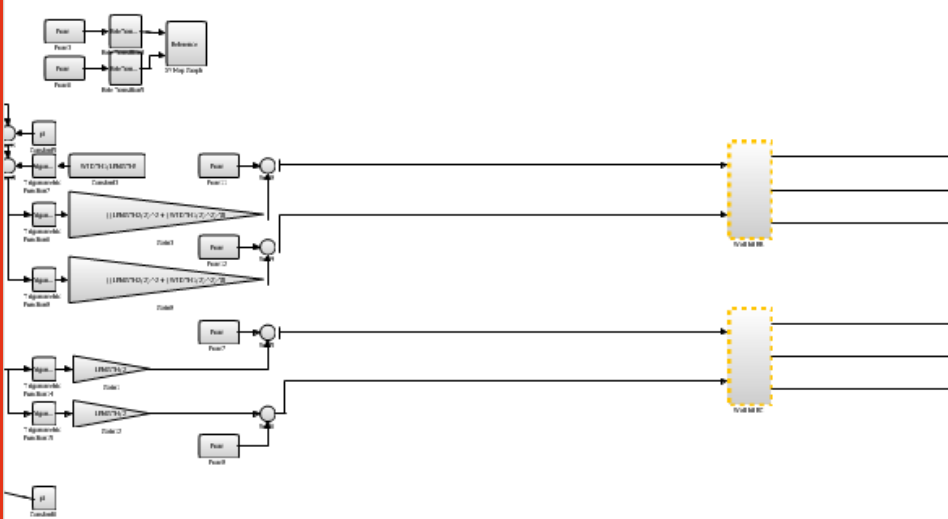
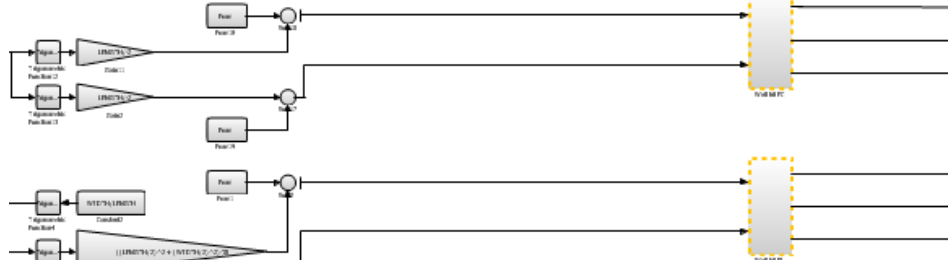
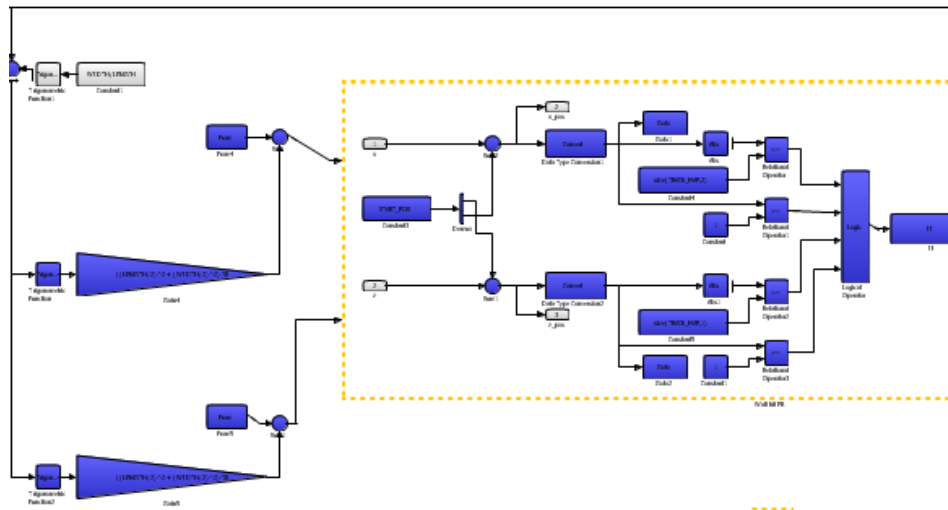
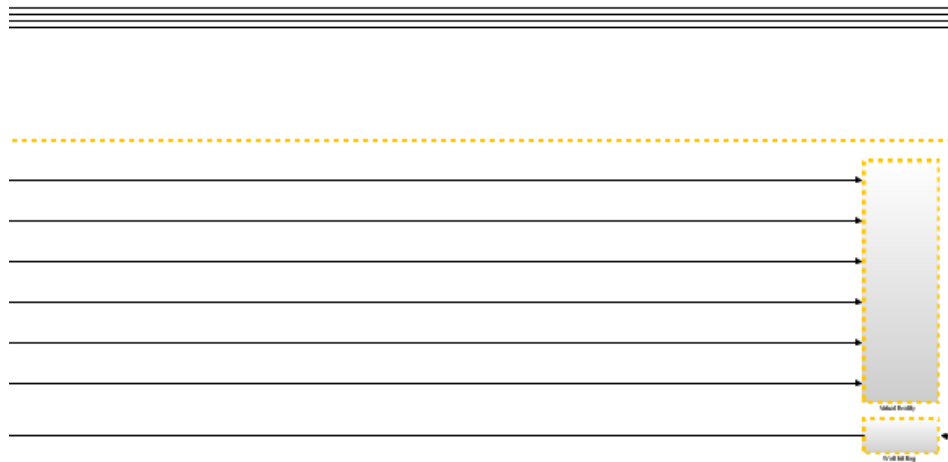
**Software ist mehr als Code**



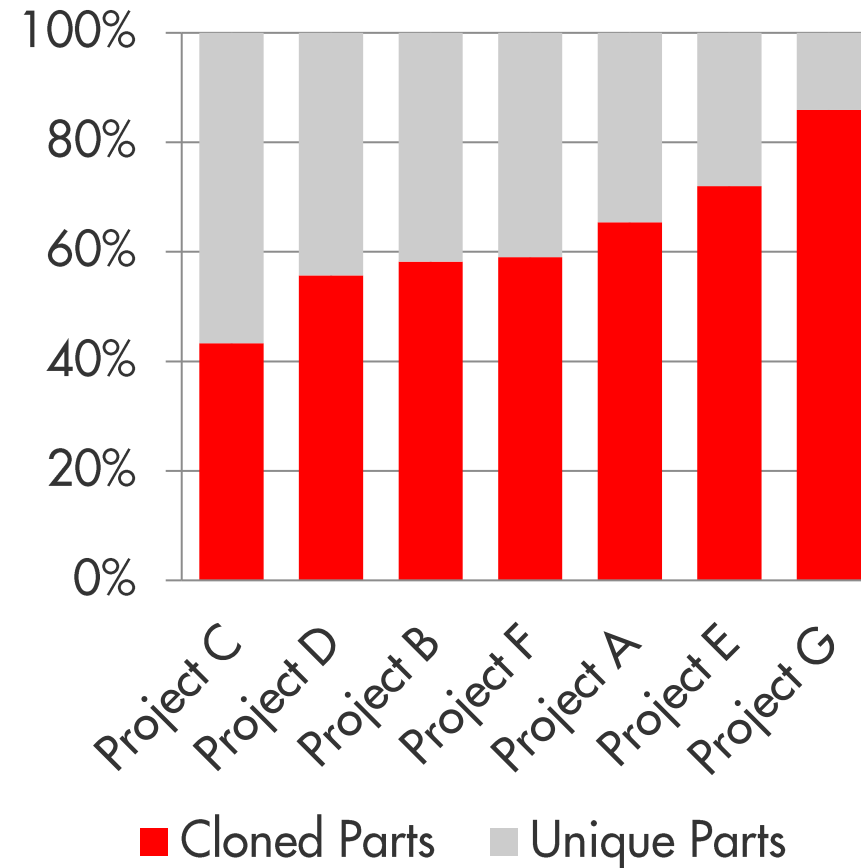
176 use cases in total, 150 contain cloning



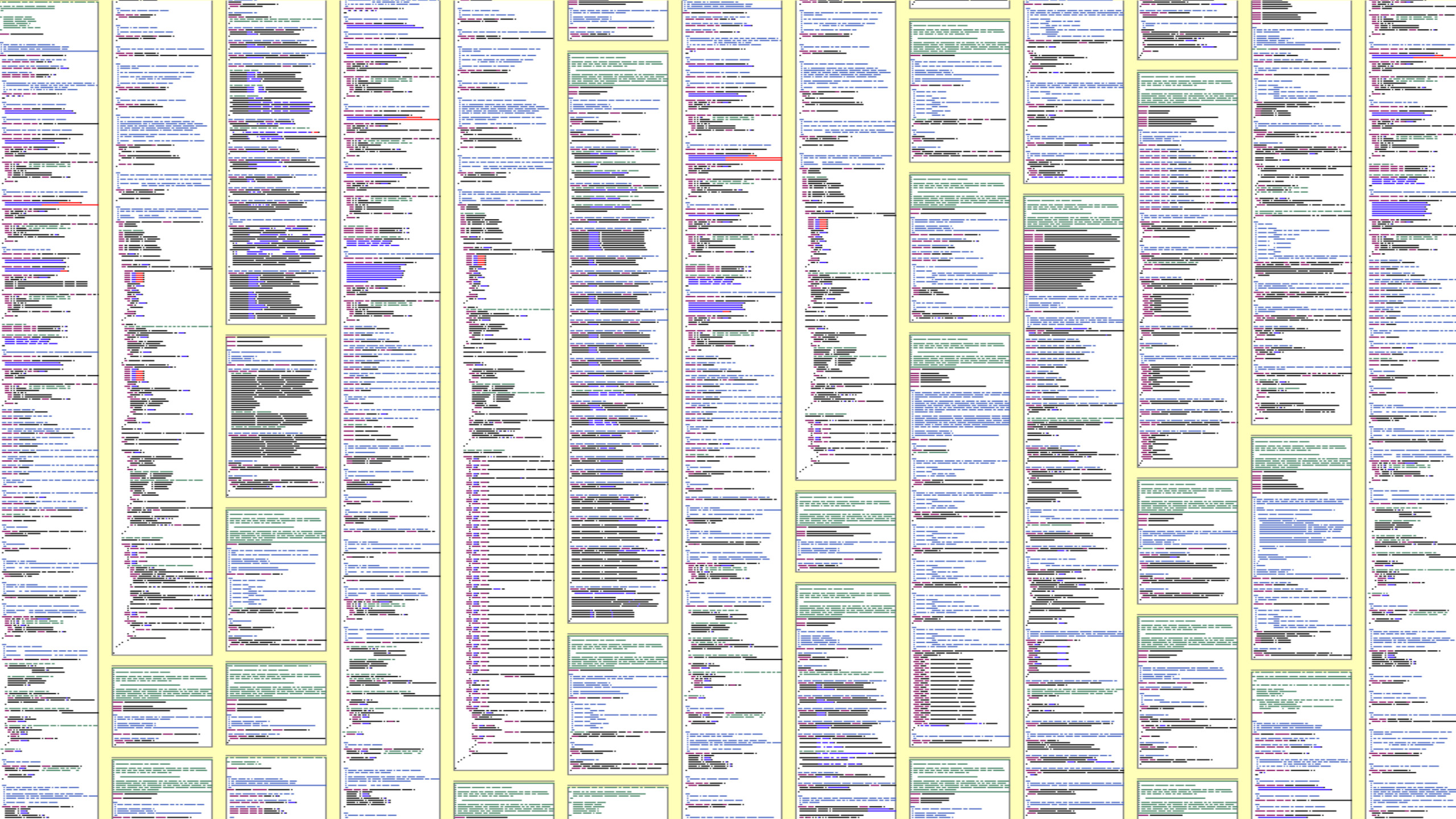
Spec	Pages	Words	Clone cov.	Clone groups	Clones	Blow-up relative	Blow-up words
A	517	41,482	35.0%	259	914	32.6%	10,191
B	1,013	130,968	8.9%	265	639	5.3%	6,639
C	133	18,447	18.5%	37	88	11.5%	1,907
D	241	37,969	8.1%	105	479	6.9%	2,463
E	185	37,056	0.9%	6	12	0.4%	161
F	42	7,662	51.1%	50	162	60.6%	2,890
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

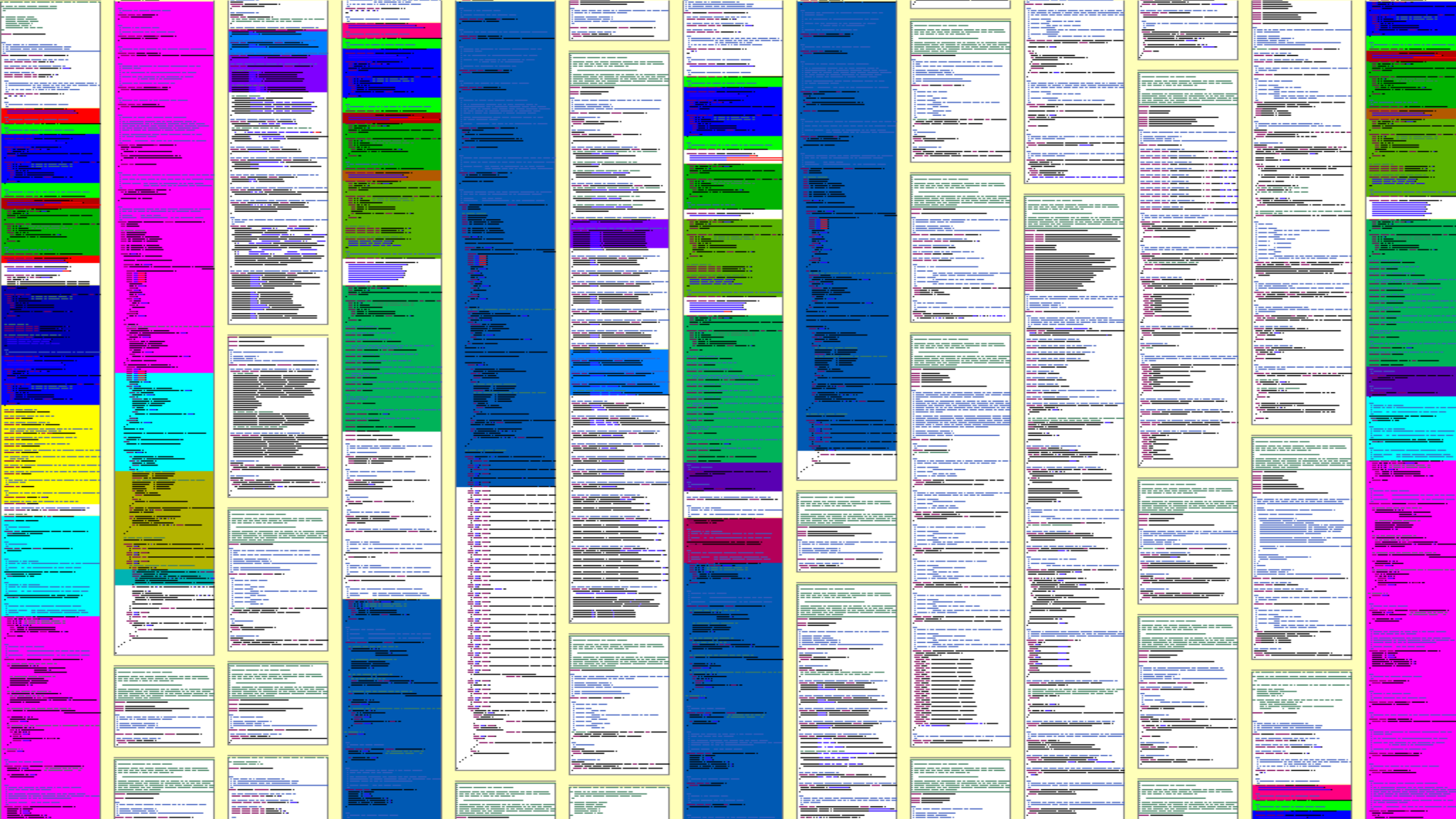


	System under Test	Test Suite
		# Tests
System A	330 kLoC	266
System B	580 kLoC	1,059
System C	150 kLoC	72
System D	430 kLoC	180
System E	760 kLoC	1,804
System F	1,400 kLoC	135
System G	160 kLoC	605

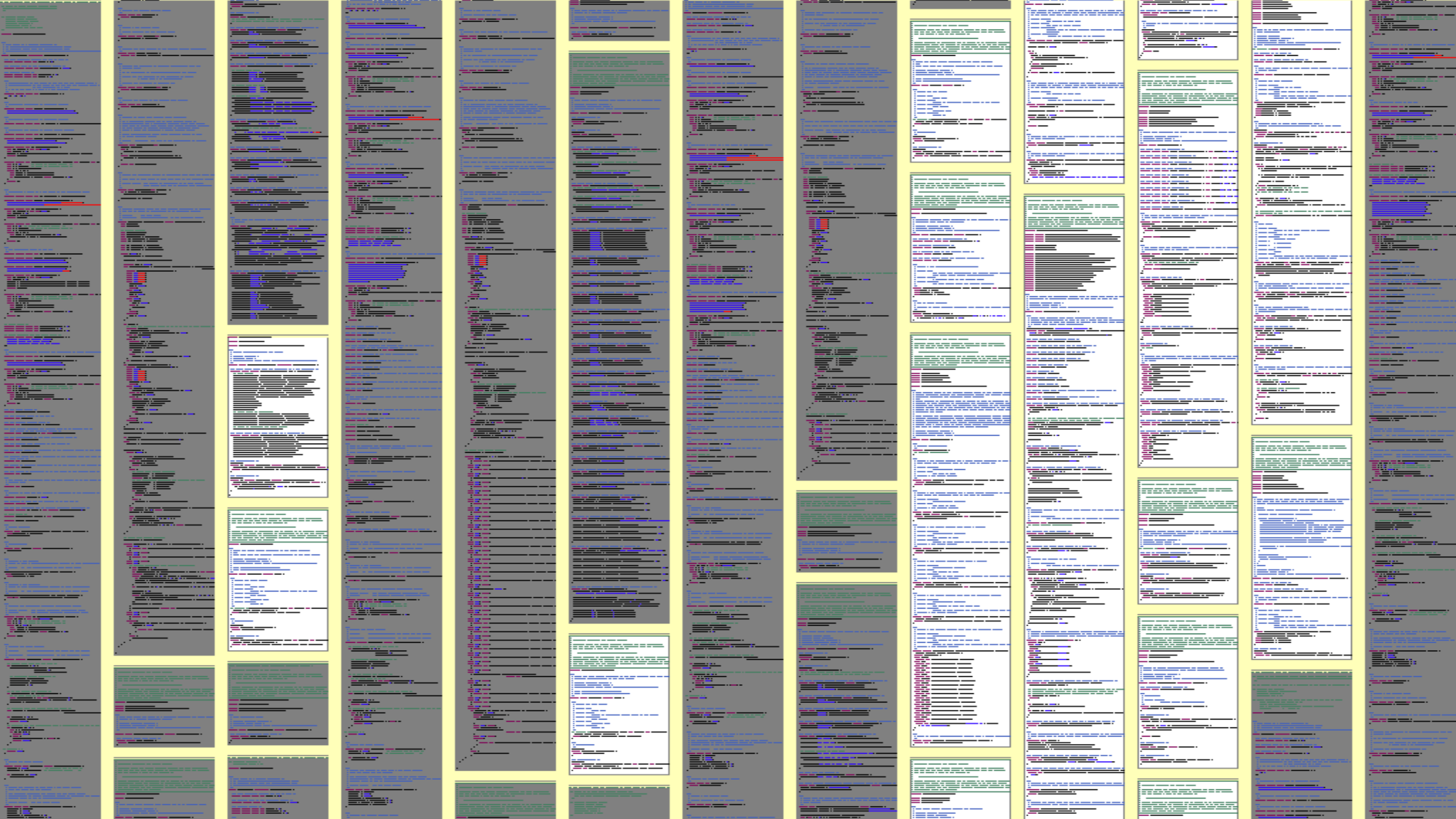


**Wie ausprobieren?**











# CQSE Workshop

## Copy, Paste und dann ...?

Erfahrung aus 15 Jahren Clone-Detection und -Management



18. Oktober  
10:30-12:00 Uhr  
[cqse.eu/clone-2311-oopst](https://cqse.eu/clone-2311-oopst)



# Kontakt – Freude mich auf Diskussionen 😊



Dr. Elmar Jürgens · [juergens@cqse.eu](mailto:juergens@cqse.eu) · +49 179 675 3863

CQSE GmbH  
Lichtenbergstraße 8  
85748 Garching bei München  
[www.cqse.eu](http://www.cqse.eu)

**CQSE**