

# Ist das wichtig oder kann das weg?

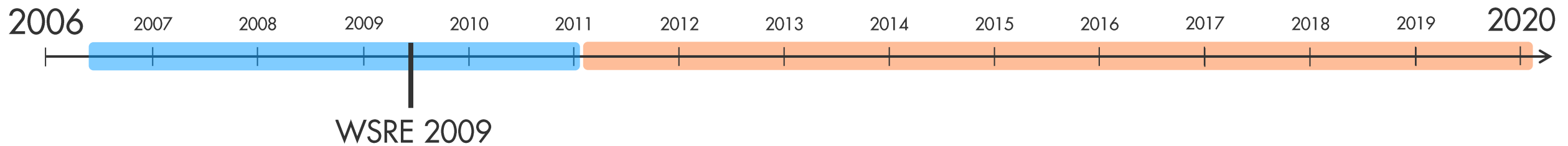
## Überraschende Ergebnisse dynamischer Analysen im Großen



TUM

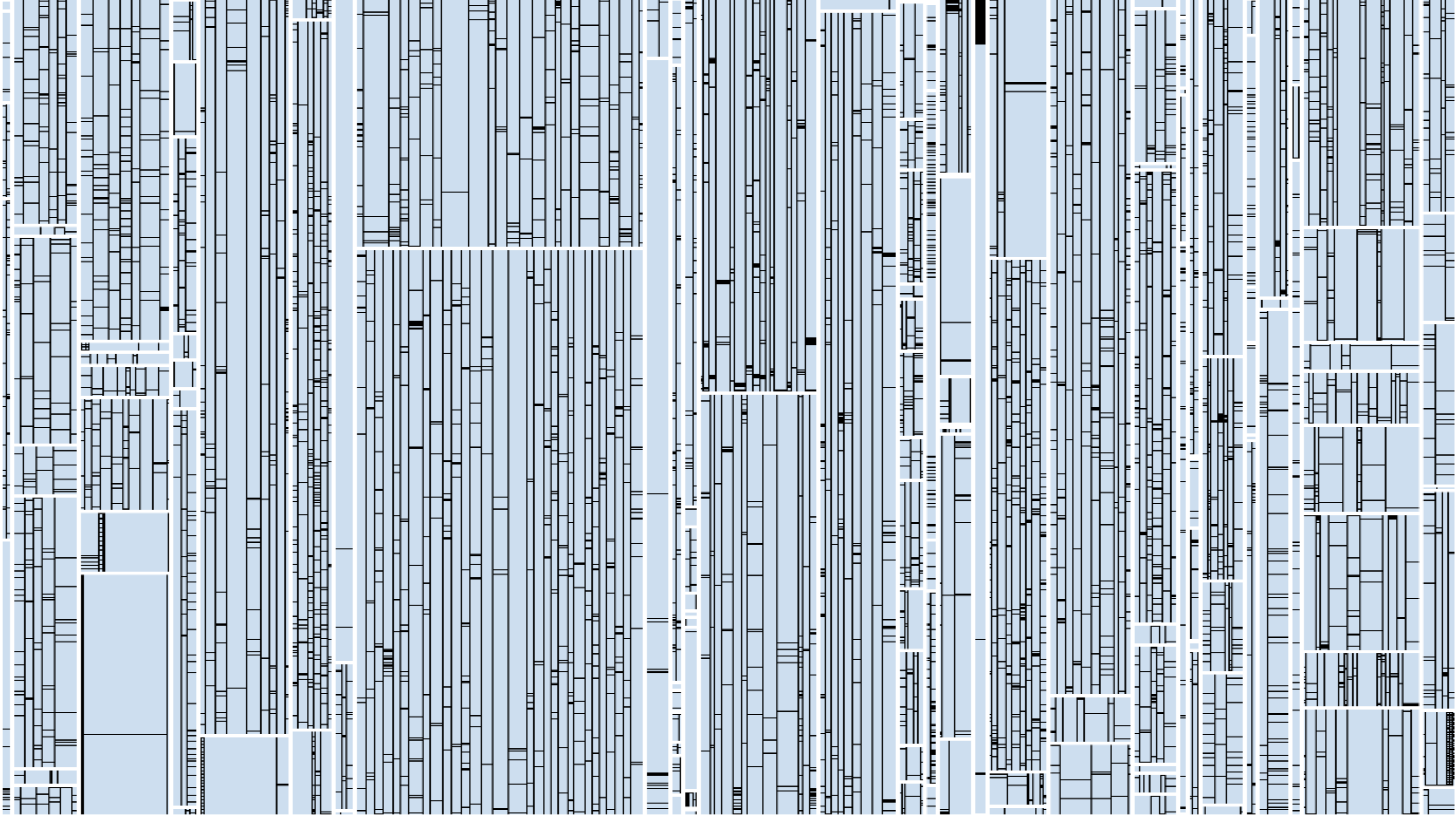


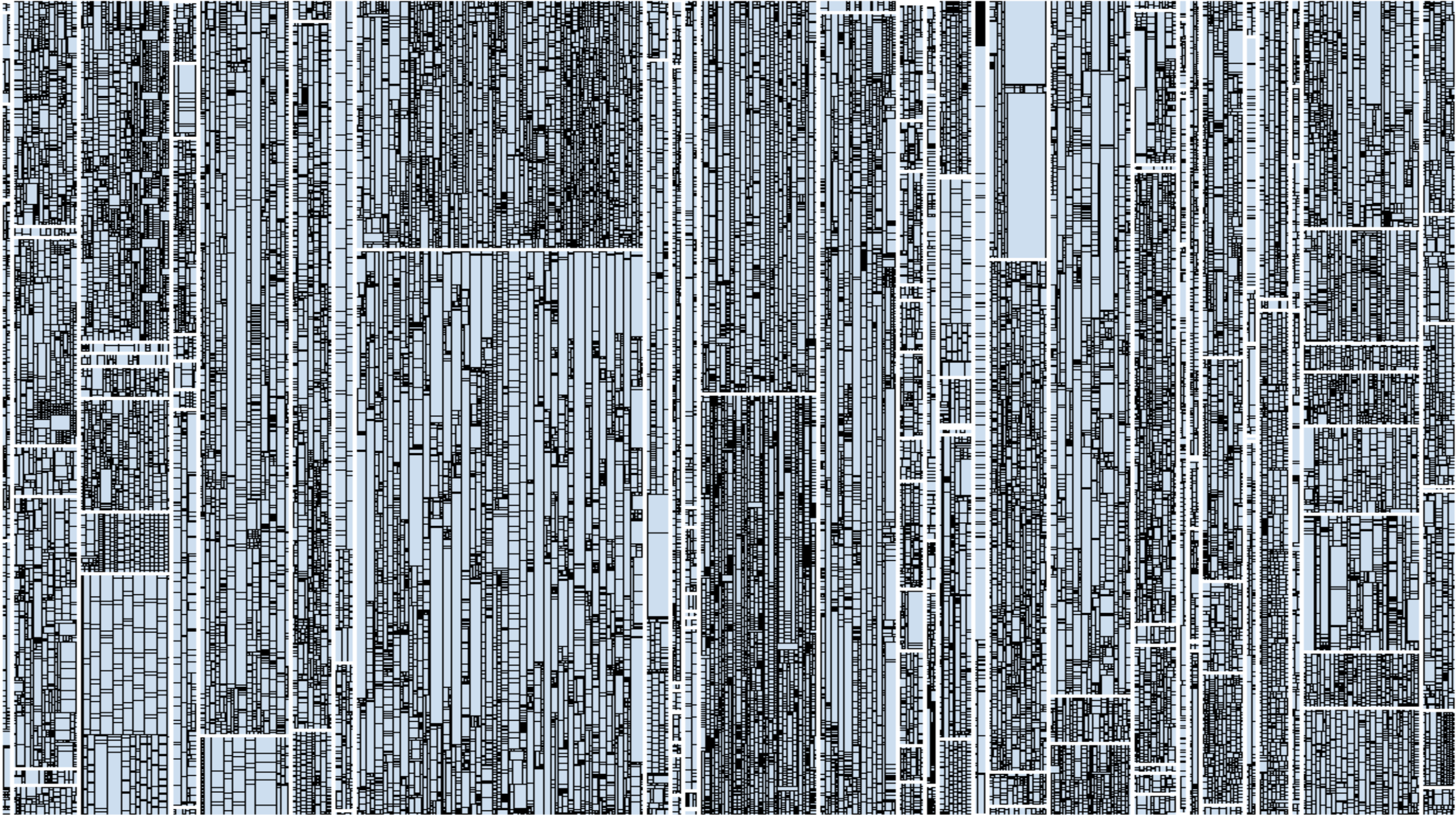
CQSE



```
1  protected void calculateIndirectAmbiguities() {
2      Map<NucleotideCompound, List<NucleotideCompound>> equivalentMap = new HashMap<>>();
3
4      List<NucleotideCompound> ambiguousCompounds = new ArrayList<NucleotideCompound>();
5      for (NucleotideCompound compound : getAllCompounds()) {
6          if (!compound.isAmbiguous()) {
7              continue;
8          }
9          ambiguousCompounds.add(compound);
10     }
11
12     for (NucleotideCompound sourceCompound : ambiguousCompounds) {
13         Set<NucleotideCompound> sourceConstituents = sourceCompound.getConstituents();
14         for (NucleotideCompound targetCompound : ambiguousCompounds) {
15             Set<NucleotideCompound> targetConstituents = targetCompound.getConstituents();
16             if (targetConstituents.containsAll(sourceConstituents)) {
17                 NucleotideCompound lcSourceCompound = toLowerCase(sourceCompound);
18                 NucleotideCompound lcTargetCompound = toLowerCase(targetCompound);
19                 checkAdd(equivalentMap, sourceCompound, targetCompound);
20                 checkAdd(equivalentMap, sourceCompound, lcTargetCompound);
21                 checkAdd(equivalentMap, targetCompound, sourceCompound);
22                 checkAdd(equivalentMap, lcTargetCompound, sourceCompound);
23                 checkAdd(equivalentMap, lcSourceCompound, targetCompound);
24                 checkAdd(equivalentMap, lcSourceCompound, lcTargetCompound);
25             }
26         }
27     }
28
29     for (NucleotideCompound key : equivalentMap.keySet()) {
30         List<NucleotideCompound> vals = equivalentMap.get(key);
31         for (NucleotideCompound value : vals) {
32             addEquivalent((C) key, (C) value);
33             addEquivalent((C) value, (C) key);
34         }
35     }
36 }
```

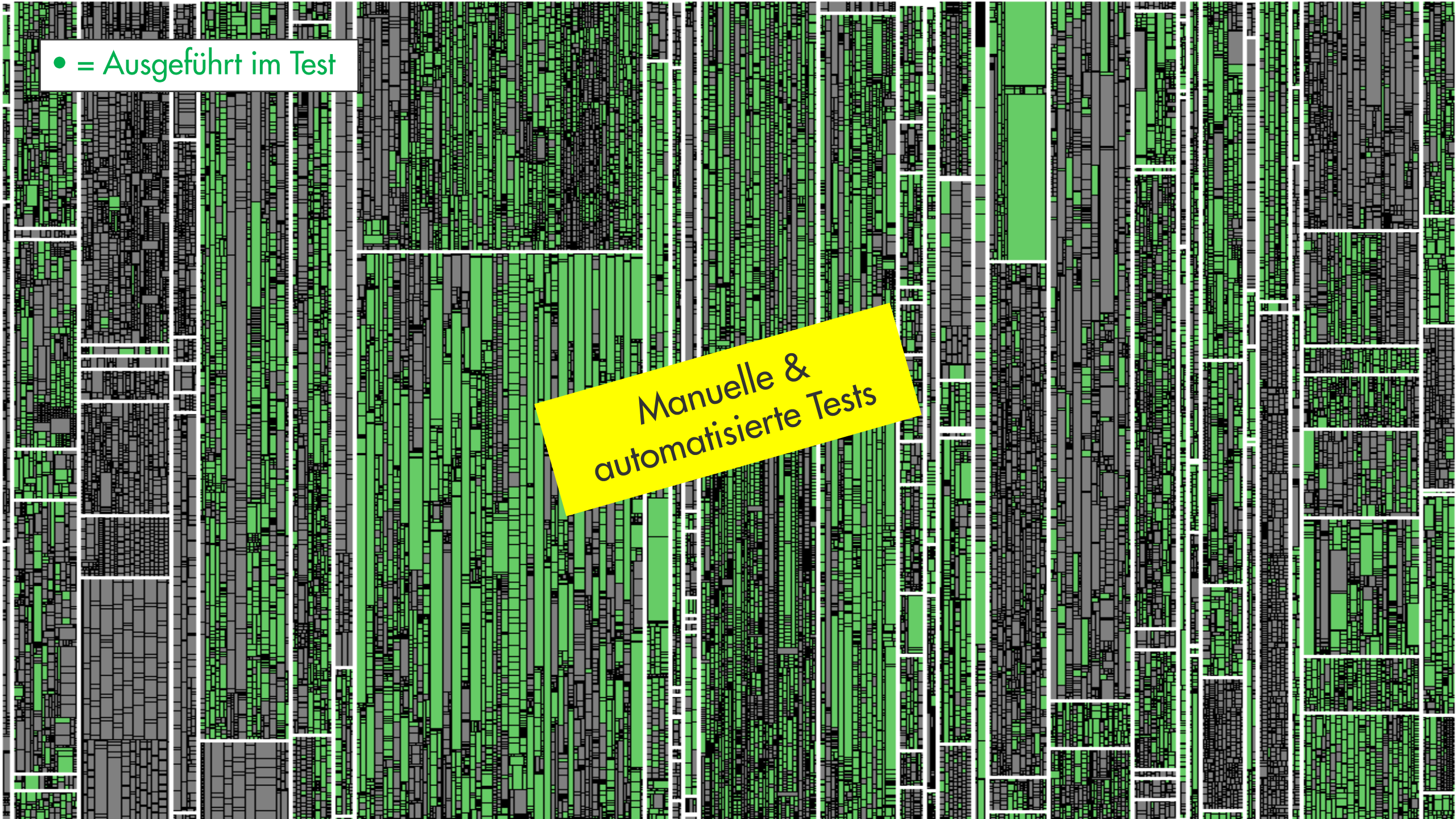




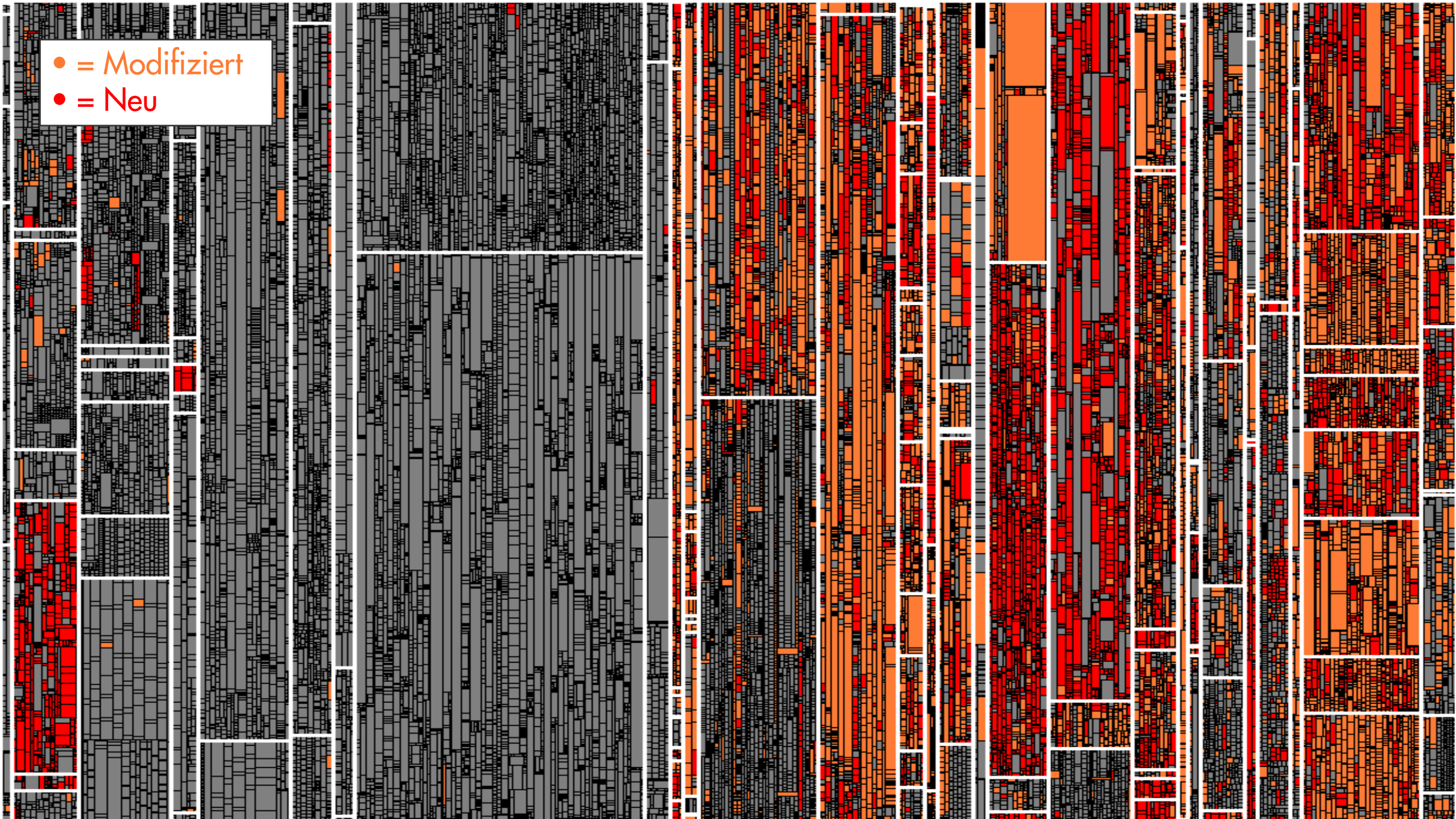


● = Ausgeführt im Test

Manuelle &  
automatisierte Tests

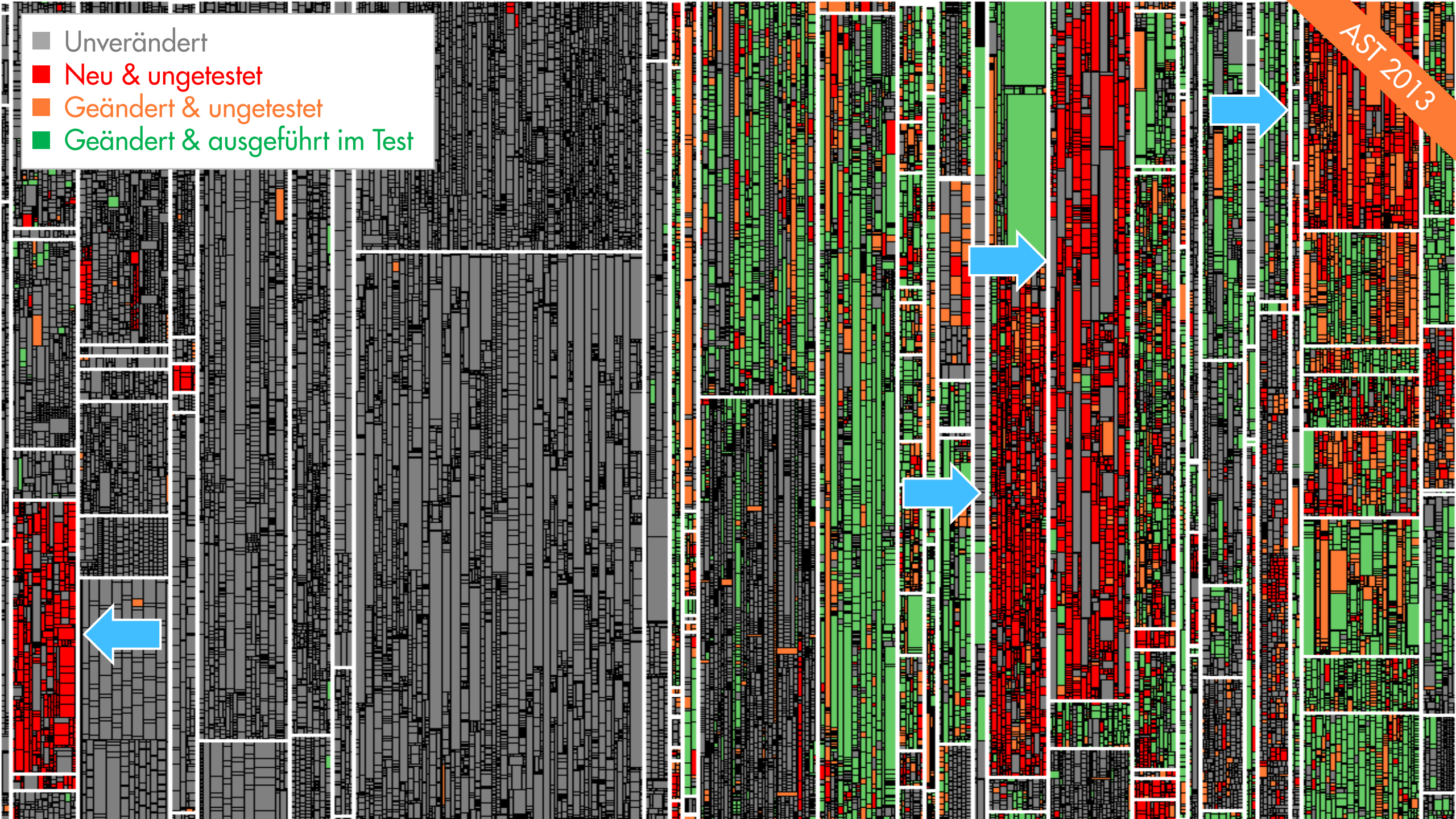


- = Modifiziert
- = Neu

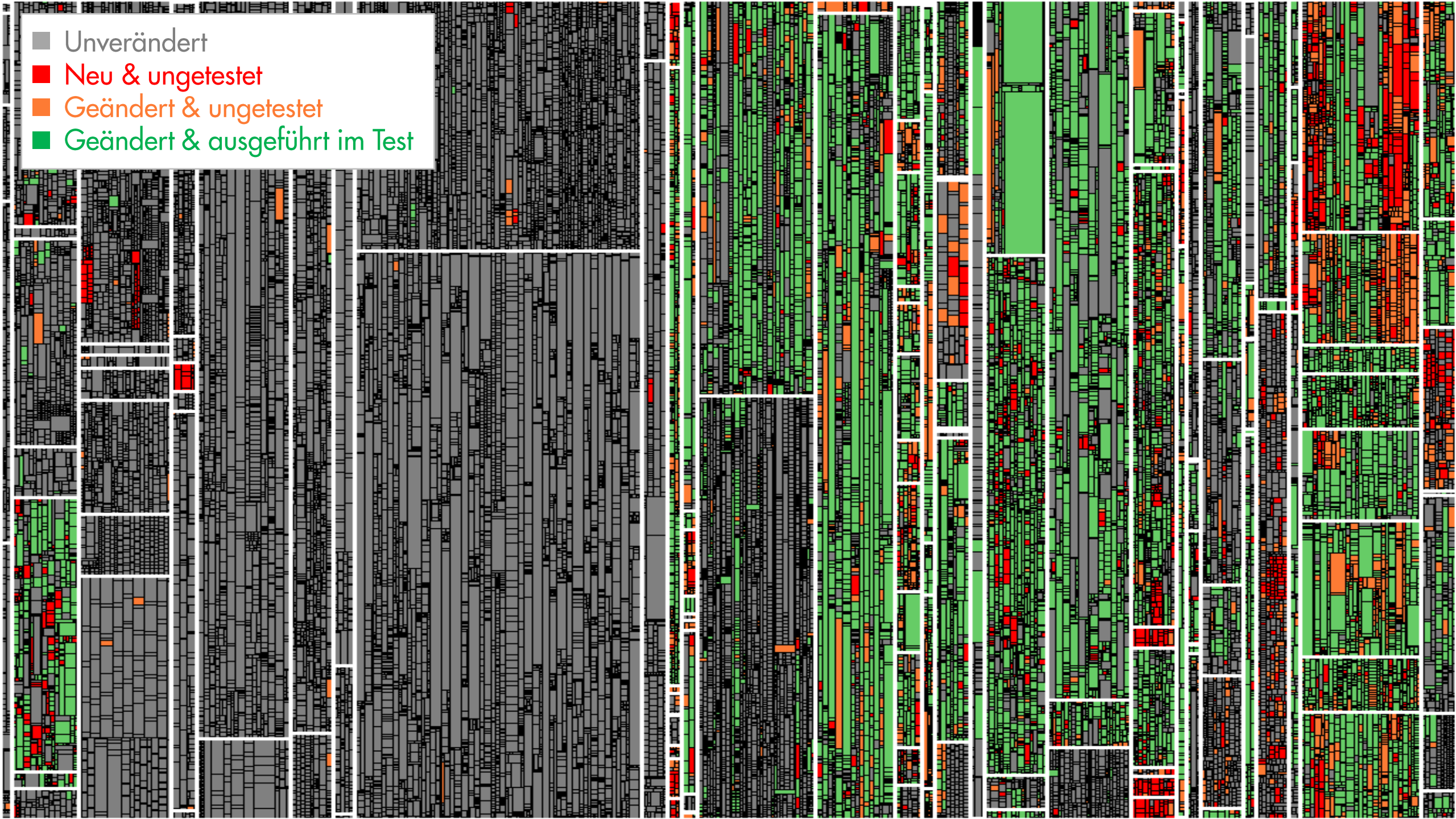




- Unverändert
- Neu & ungetestet
- Geändert & ungetestet
- Geändert & ausgeführt im Test



- Unverändert
- Neu & ungetestet
- Geändert & ungetestet
- Geändert & ausgeführt im Test



# Wieviele Änderungen sind ungetestet?

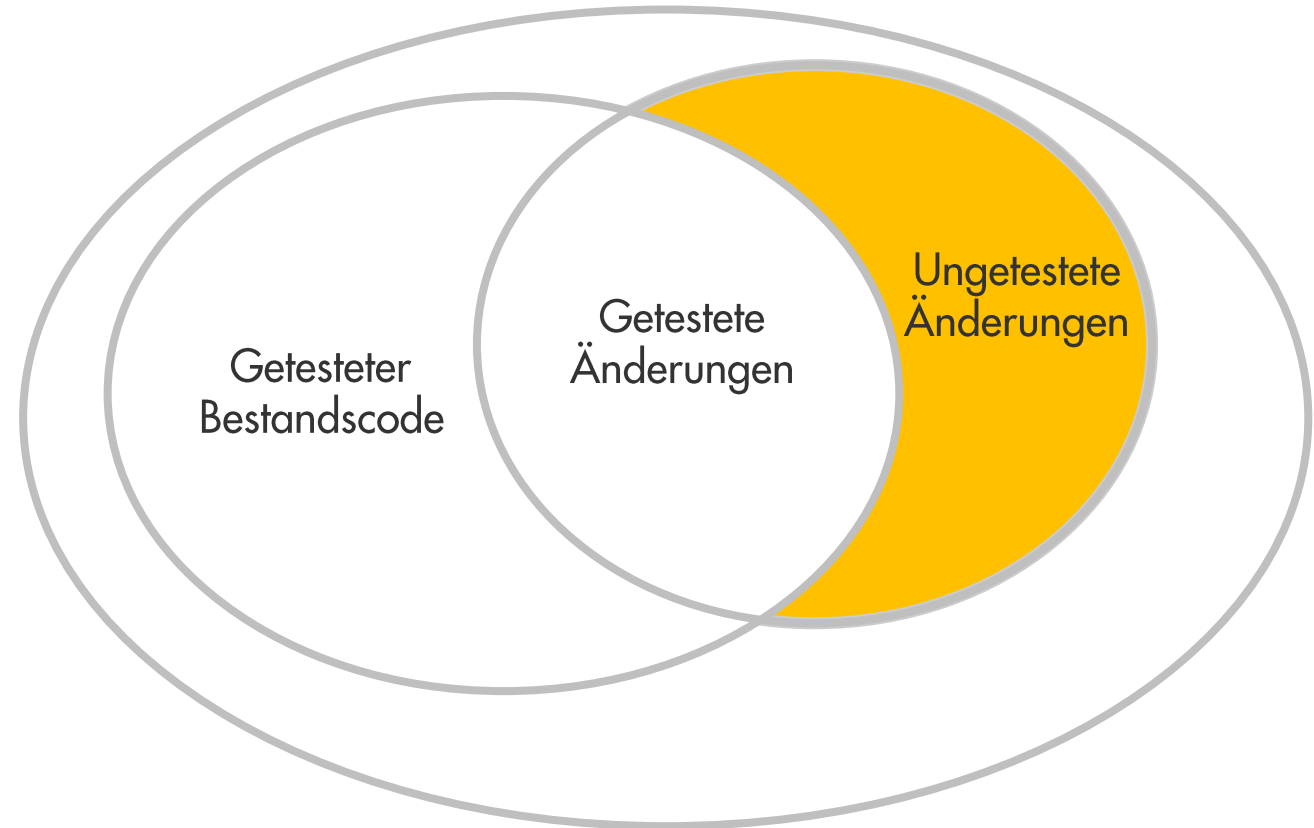
Studie: C# System @ Munich Re

## Release A:

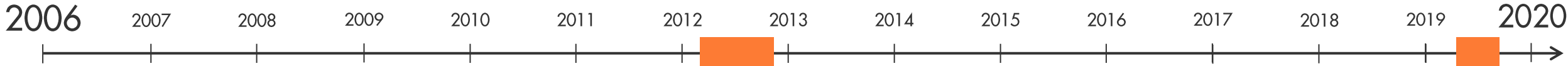
15% Code neu/geändert,  
**>50% ungetestet**

## Release B:

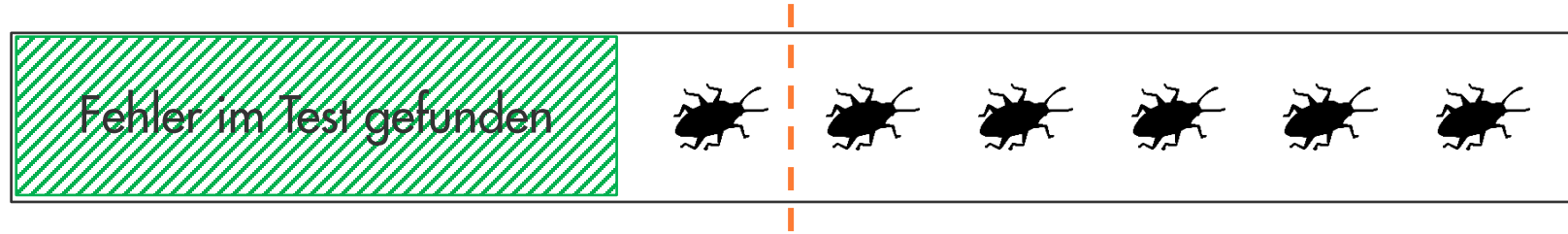
15% Code neu/geändert,  
**>60% ungetestet**



**Feldfehlerwahrscheinlichkeit 5x höher für ungetestete Änderungen!**



$\% \text{Restfehler} = 60\%$



$\% \text{Restfehler} = 28\%$

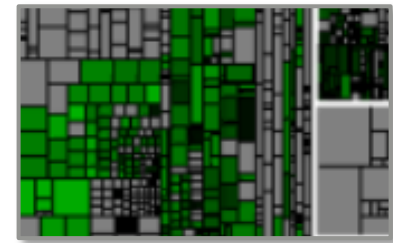


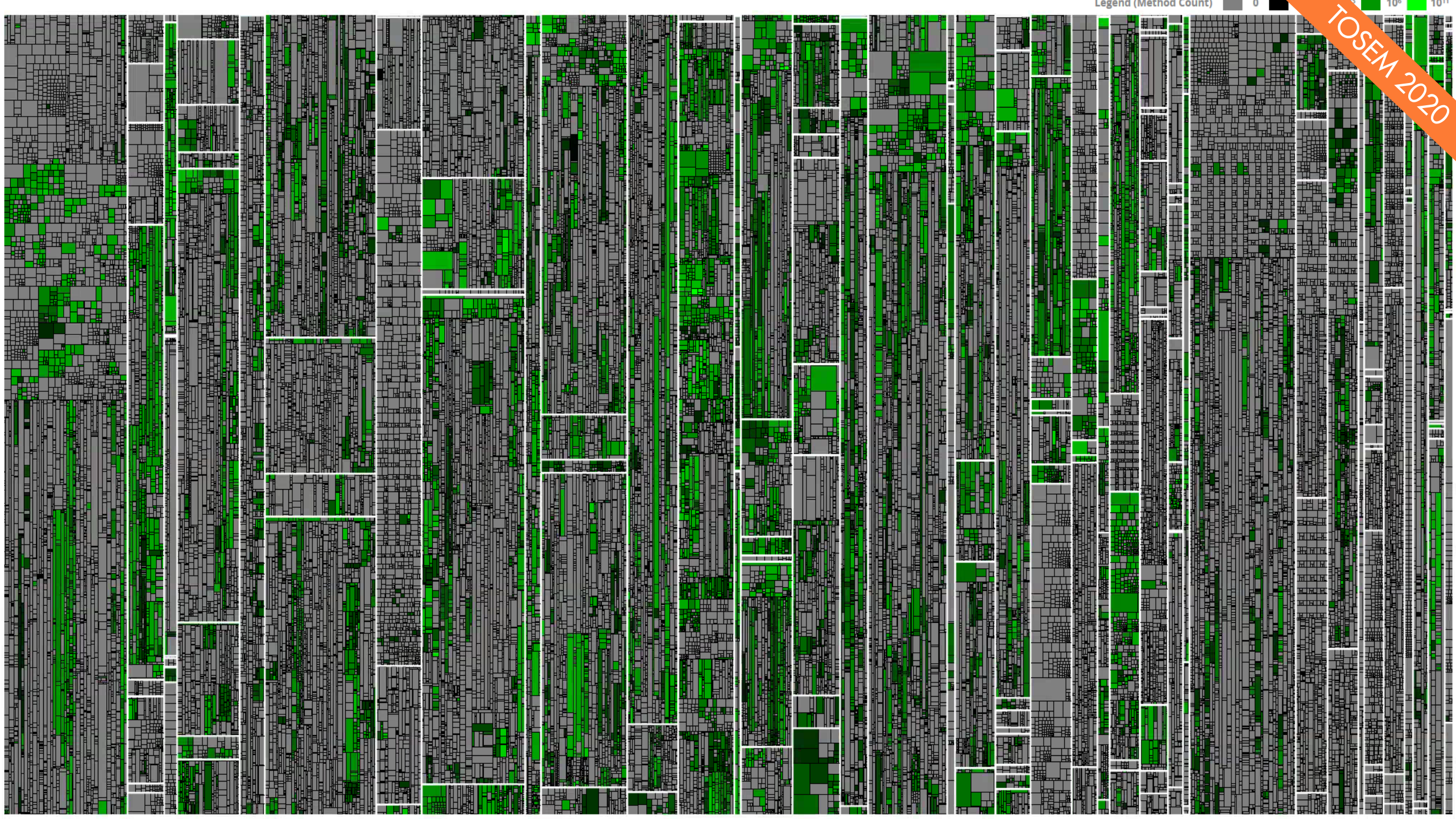


Nutzer

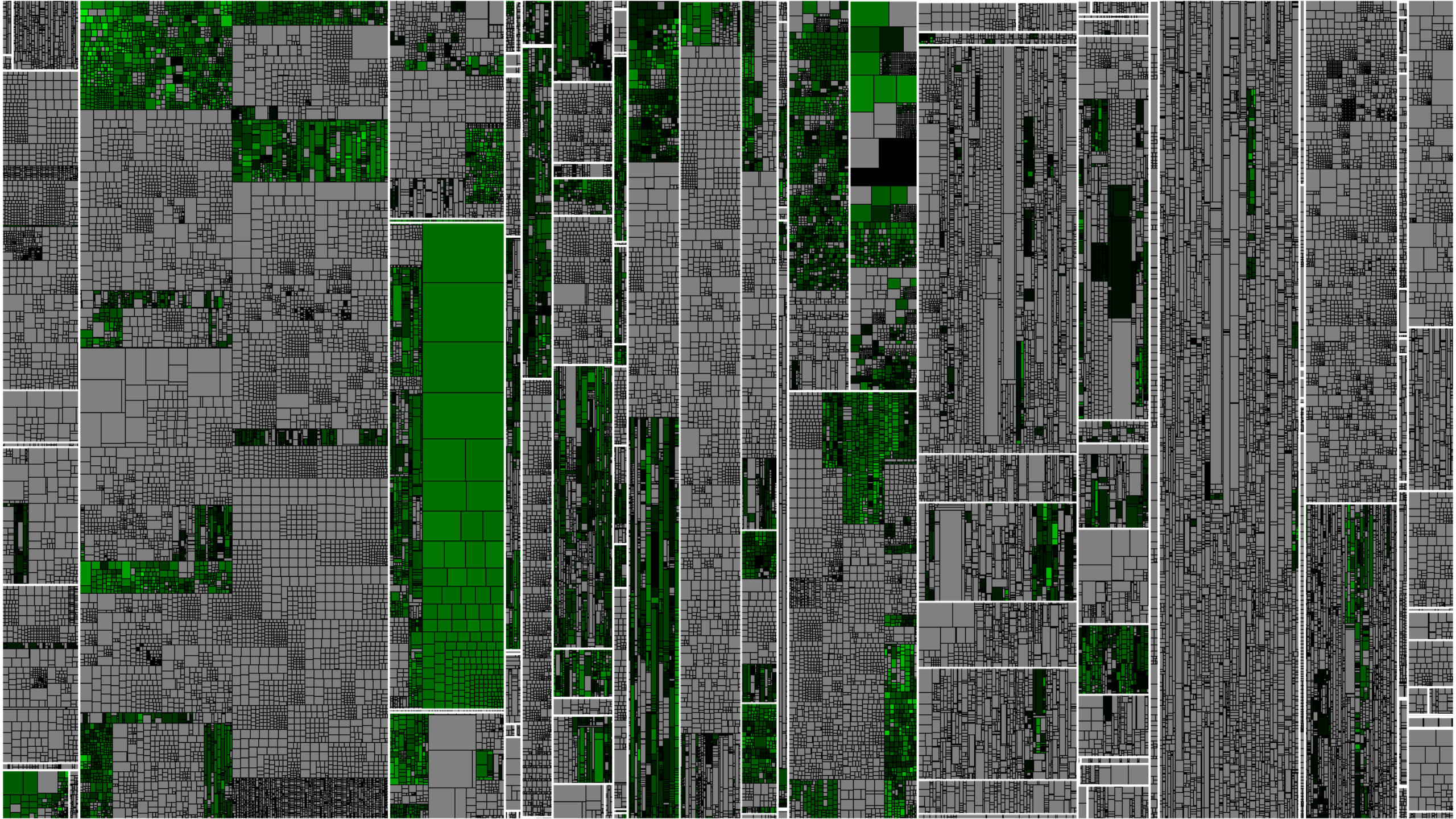


Produktivsystem



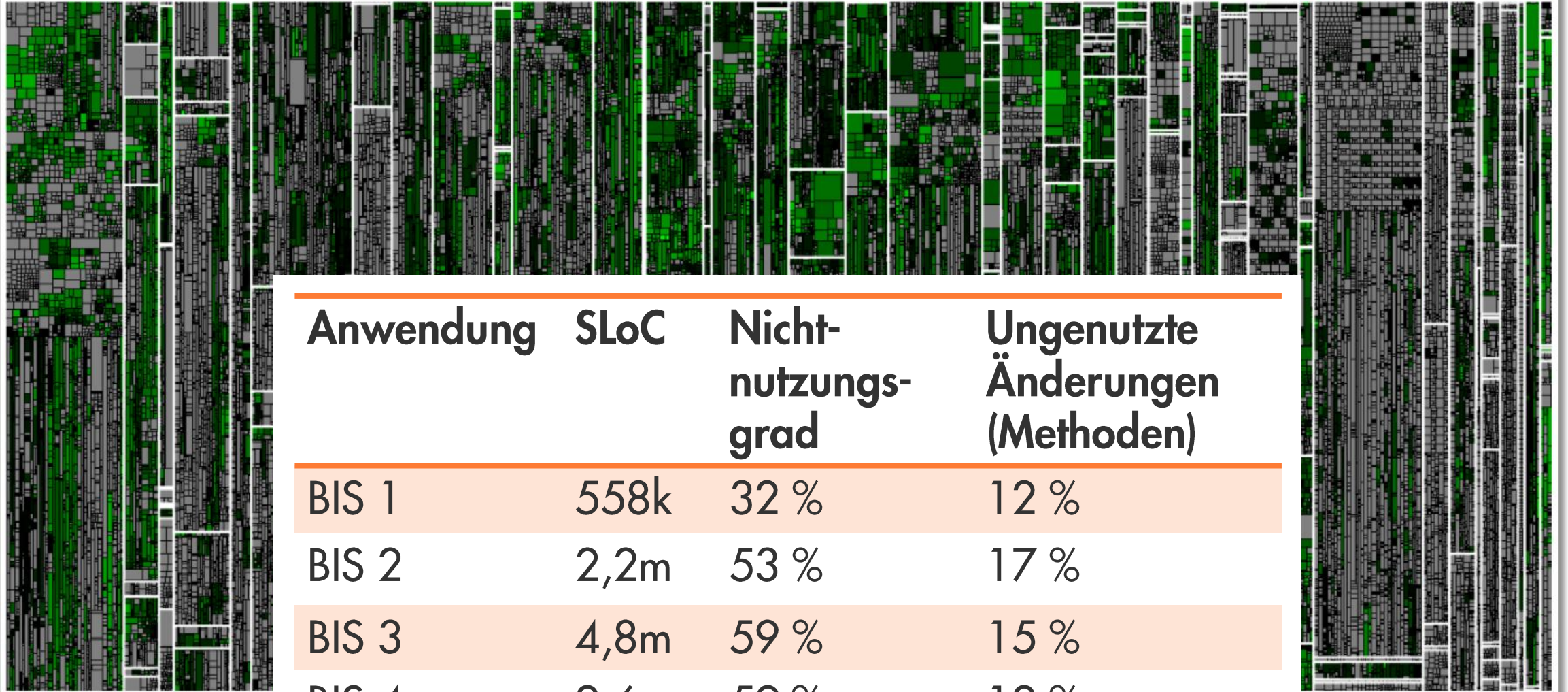


TOSEM 2020





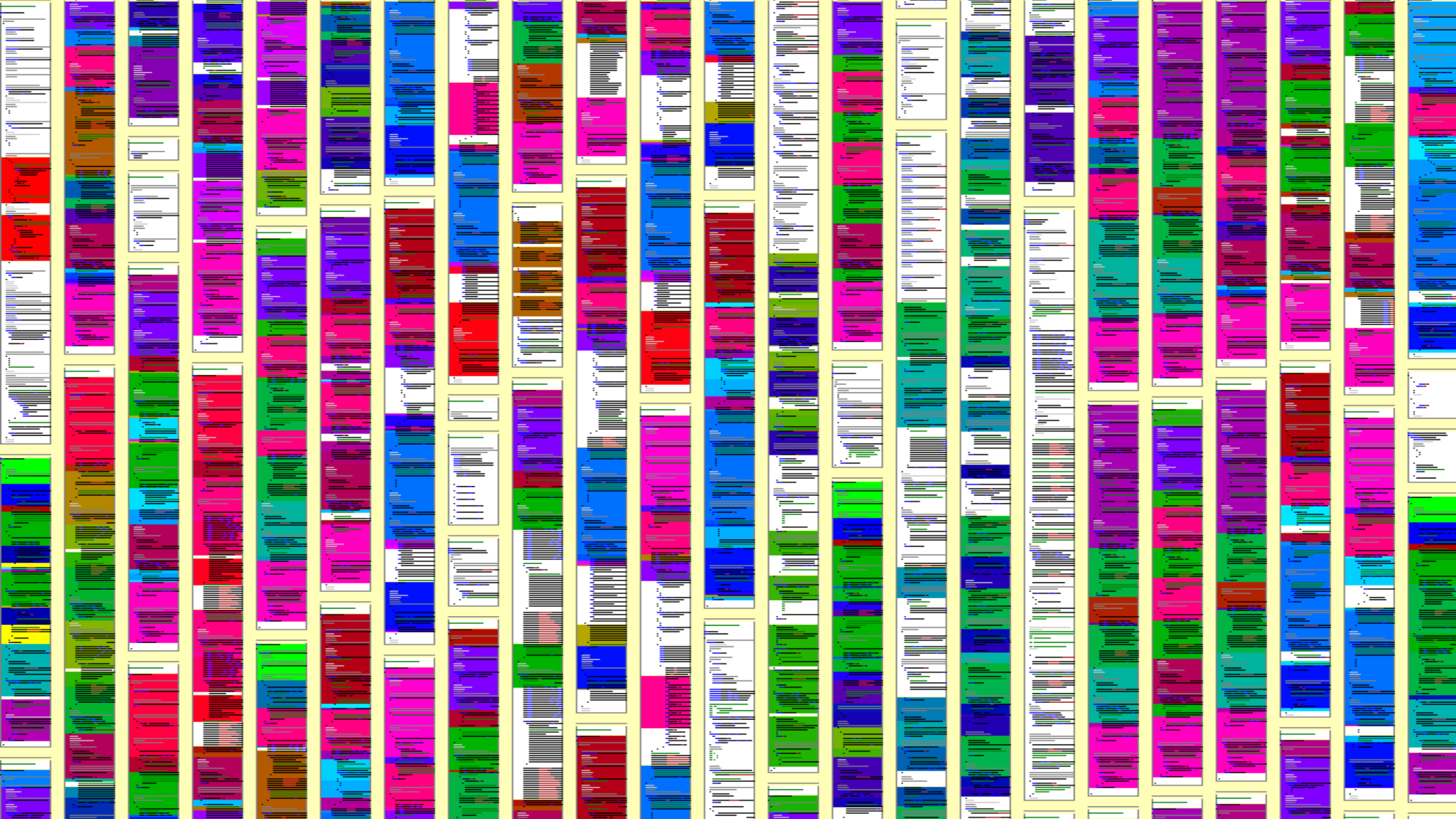
Execution Count: 0 1 10<sup>3</sup> 10<sup>6</sup> 10<sup>10</sup> | Unexecuted: 64.09%



Anwendung	SLoC	Nicht-nutzungs-grad	Ungenutzte Änderungen (Methoden)
BIS 1	558k	32 %	12 %
BIS 2	2,2m	53 %	17 %
BIS 3	4,8m	59 %	15 %
BIS 4	2,6m	59 %	12 %
BIS 5	4,2m	64 %	17 %

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





# Wieviele Änderungen sind ungetestet?

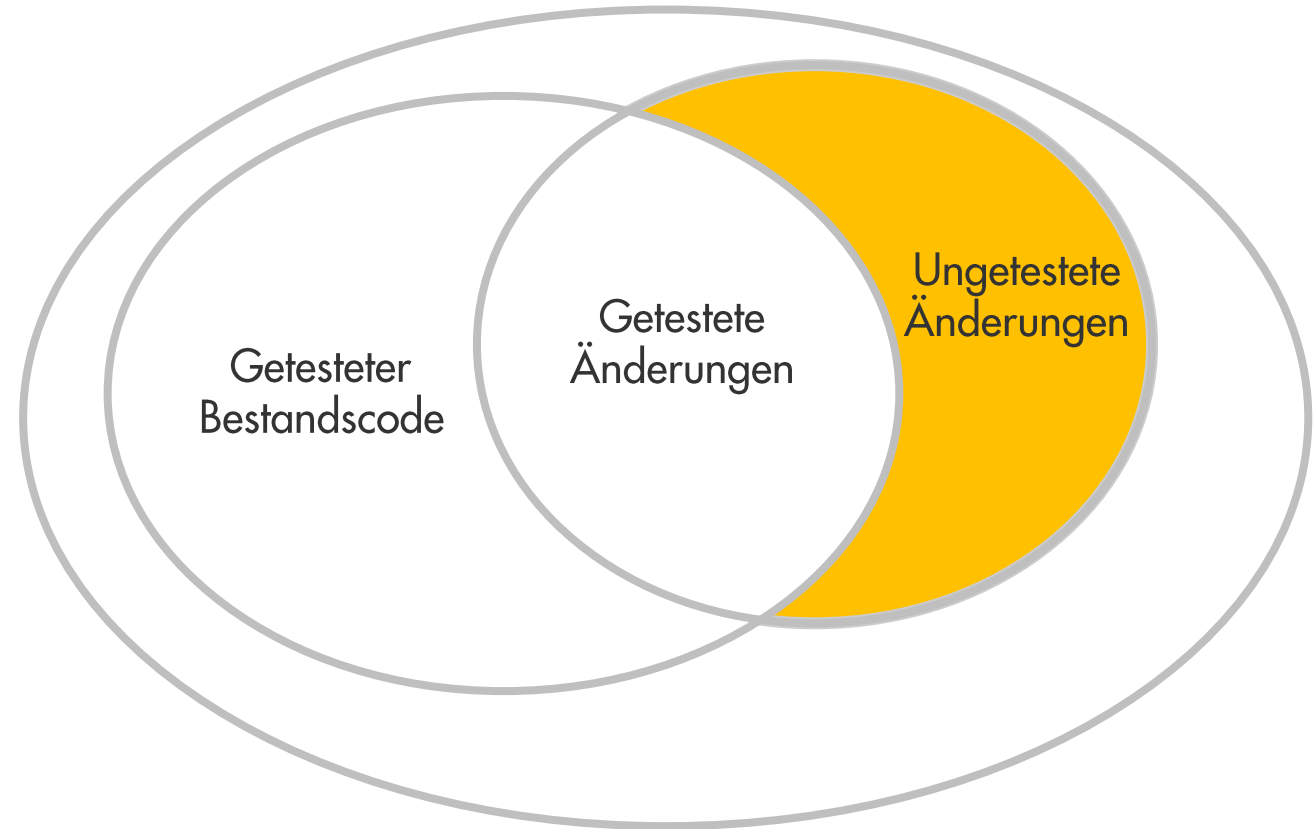
Studie: C# System @ Munich Re

## Release A:

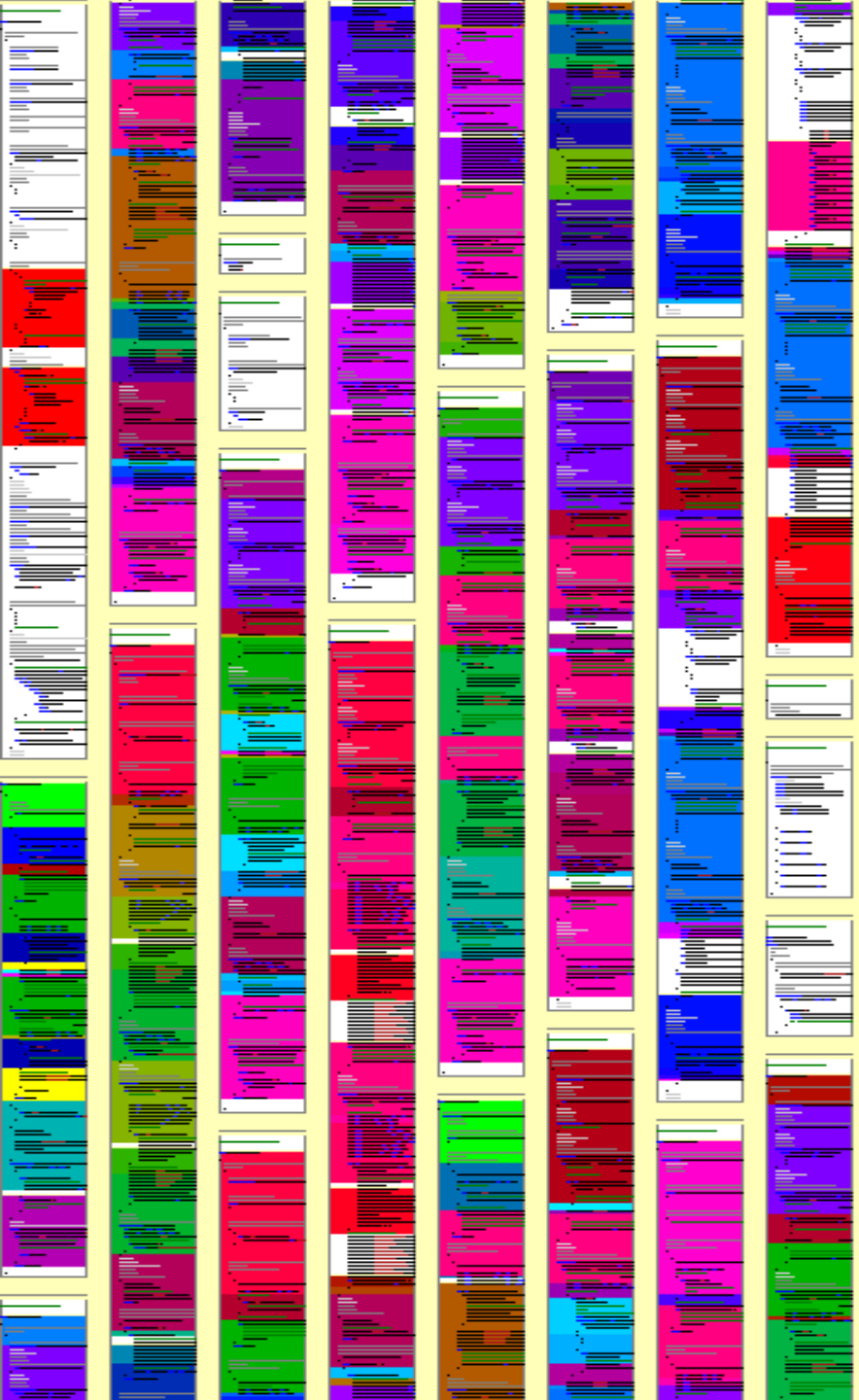
15% Code neu/geändert,  
**>50% ungetestet**

## Release B:

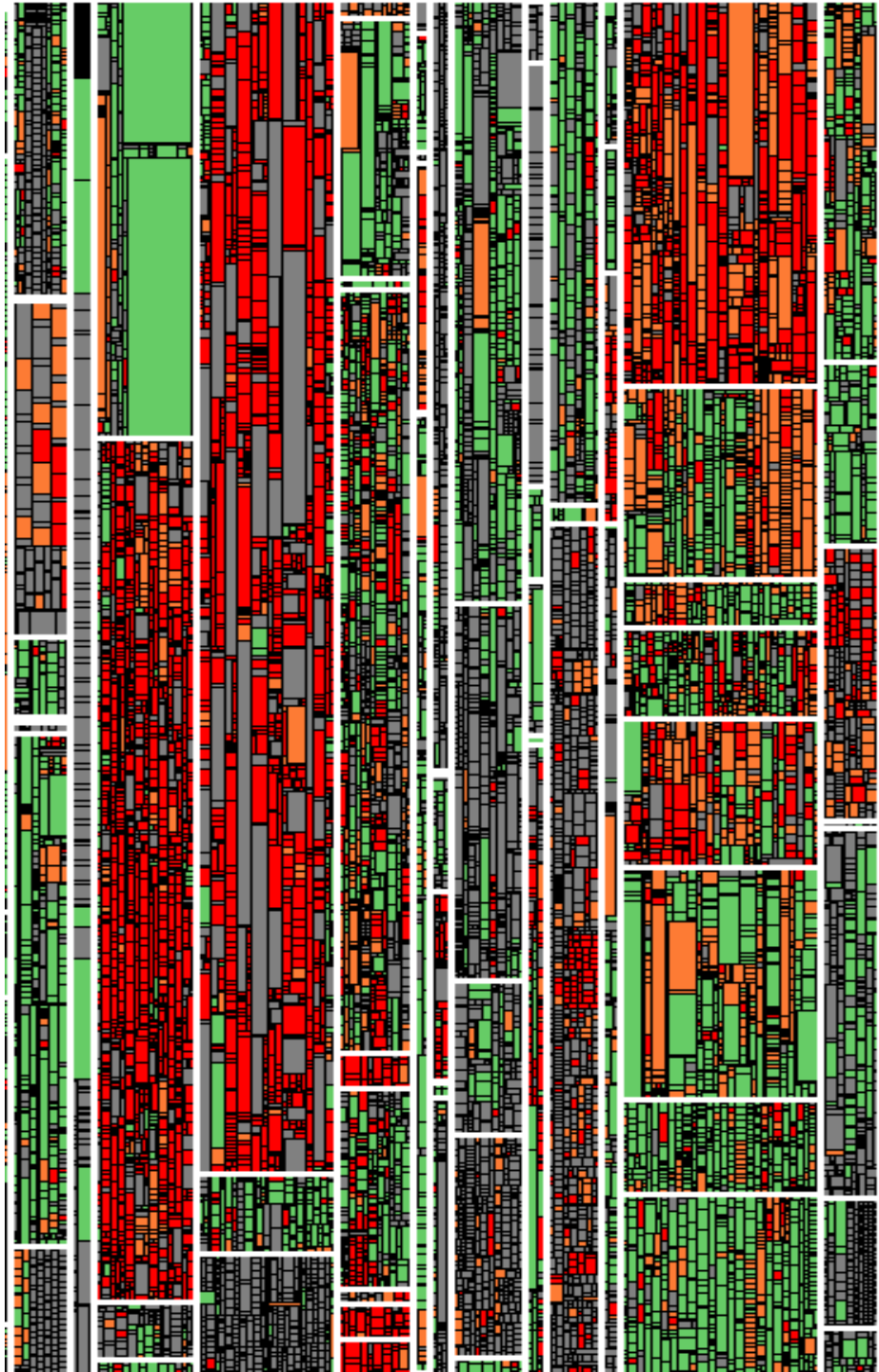
15% Code neu/geändert,  
**>60% ungetestet**

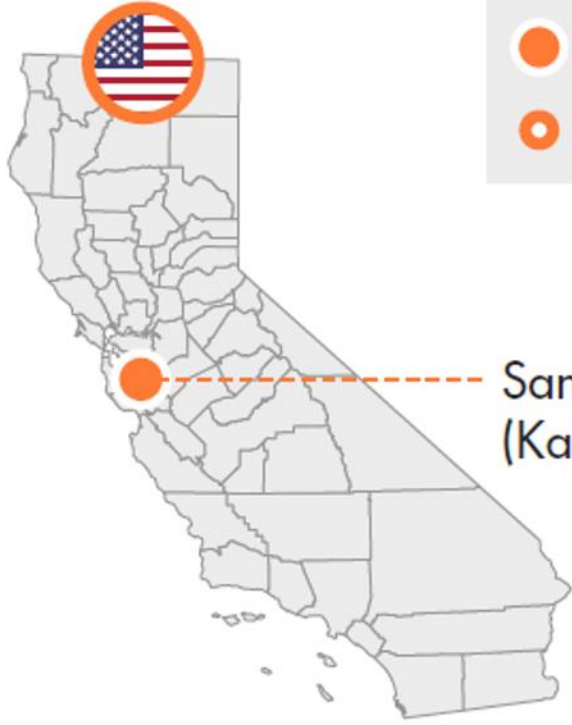


**Feldfehlerwahrscheinlichkeit 5x höher für ungetestete Änderungen!**

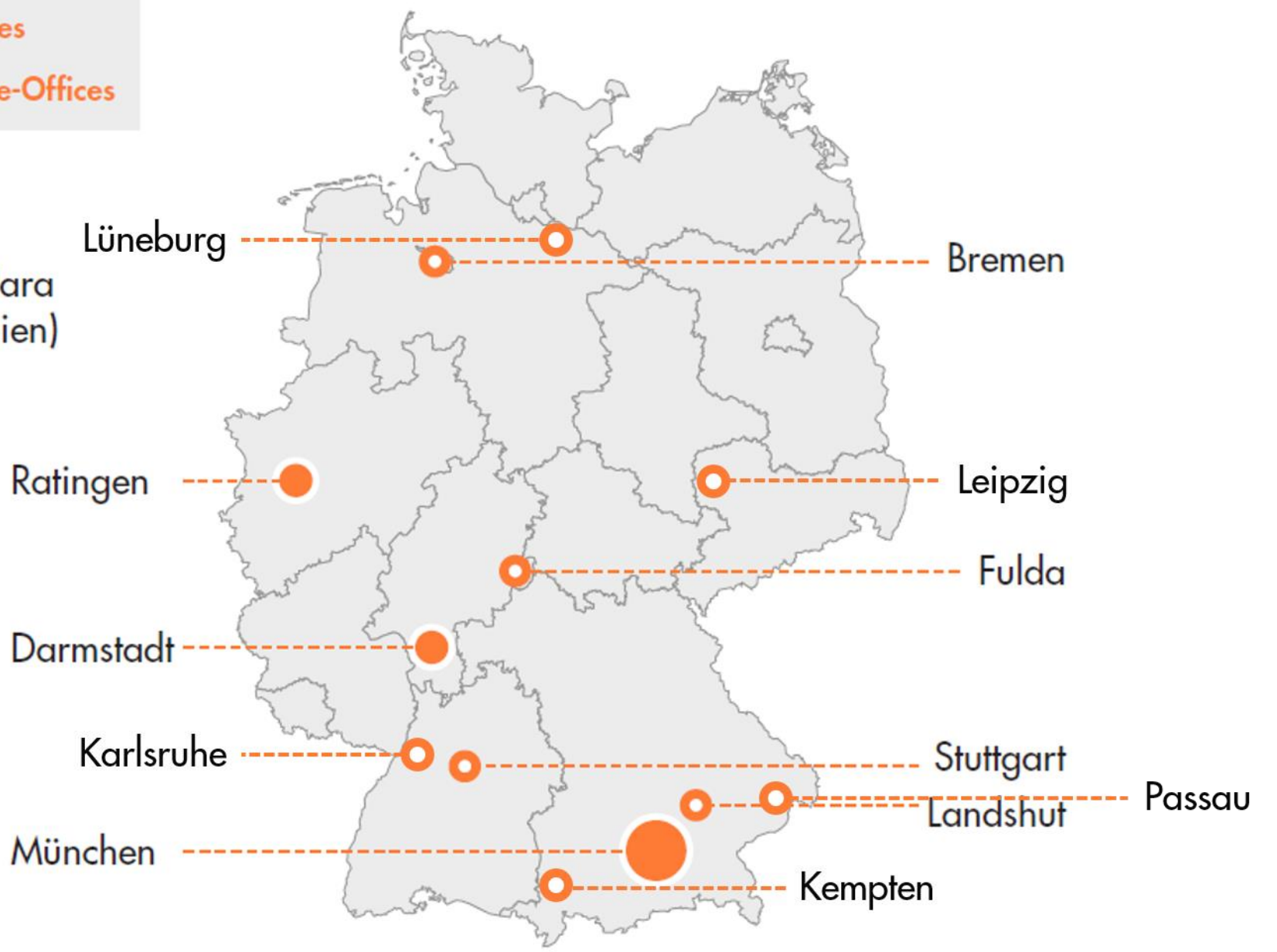


vs.





- Offices
- Home-Offices



# Kontakt – Ich freue 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)

