

# Haben wir unsere Änderungen diesmal getestet?

Erfahrungen mit Test-Gap-Analyse bei der Munich Re

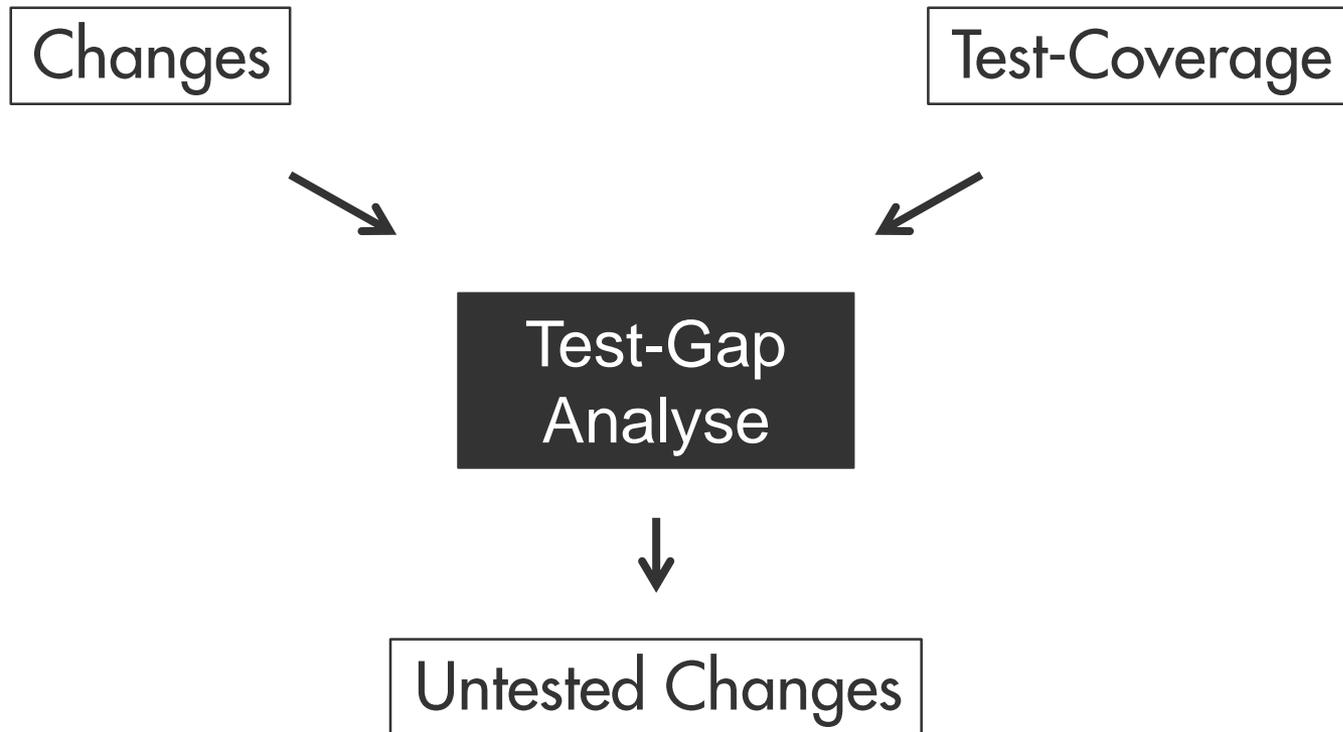
Uwe Proft (Munich Re)

Elmar Jürgens (CQSE GmbH)

# Agenda

1. Test-Gap-Analyse
2. Erfahrungen aus dem Einsatz bei der Munich Re
3. Diskussion

# Anatomie Test-Gap Analyse





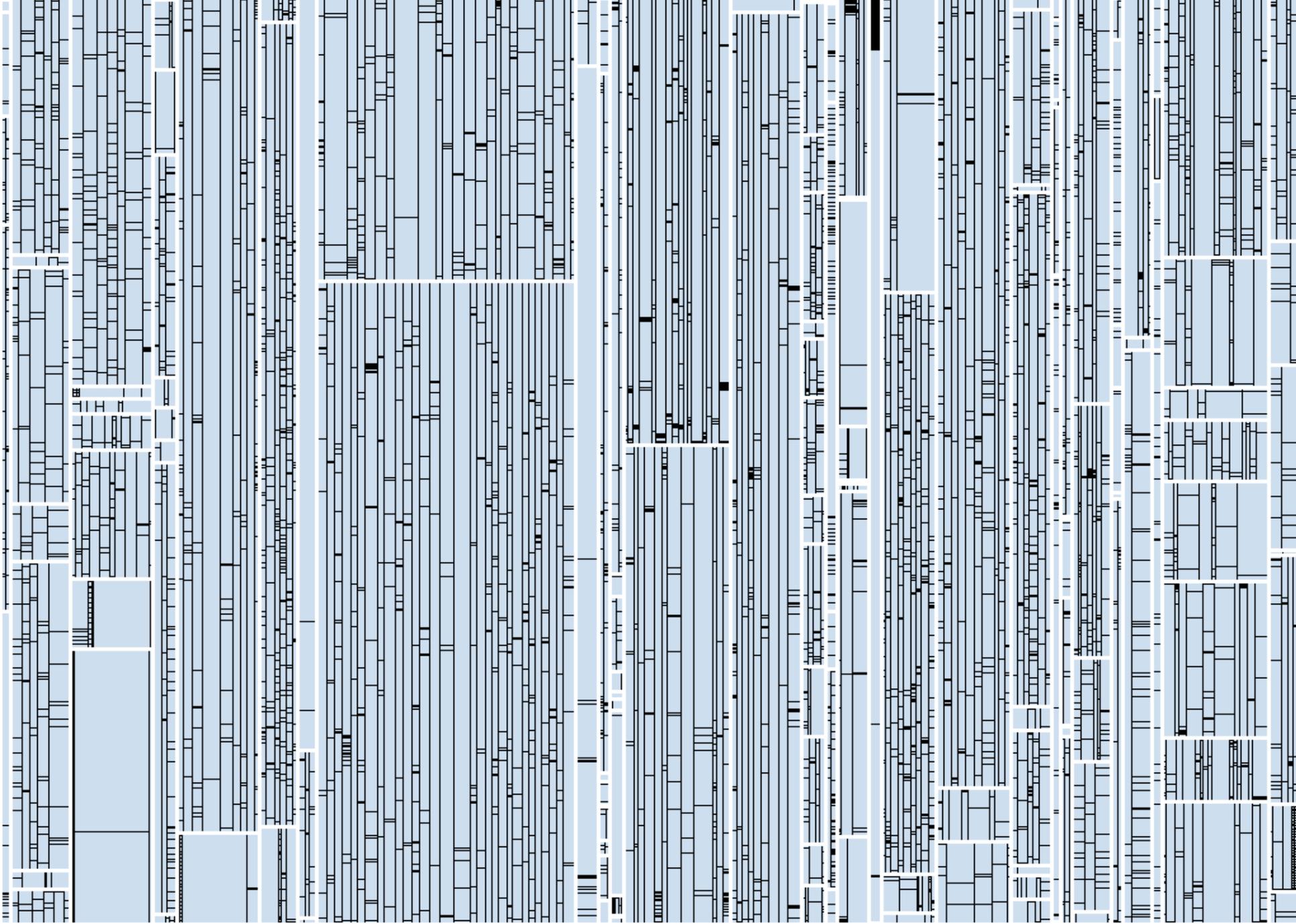
**GUI.Dialogs**

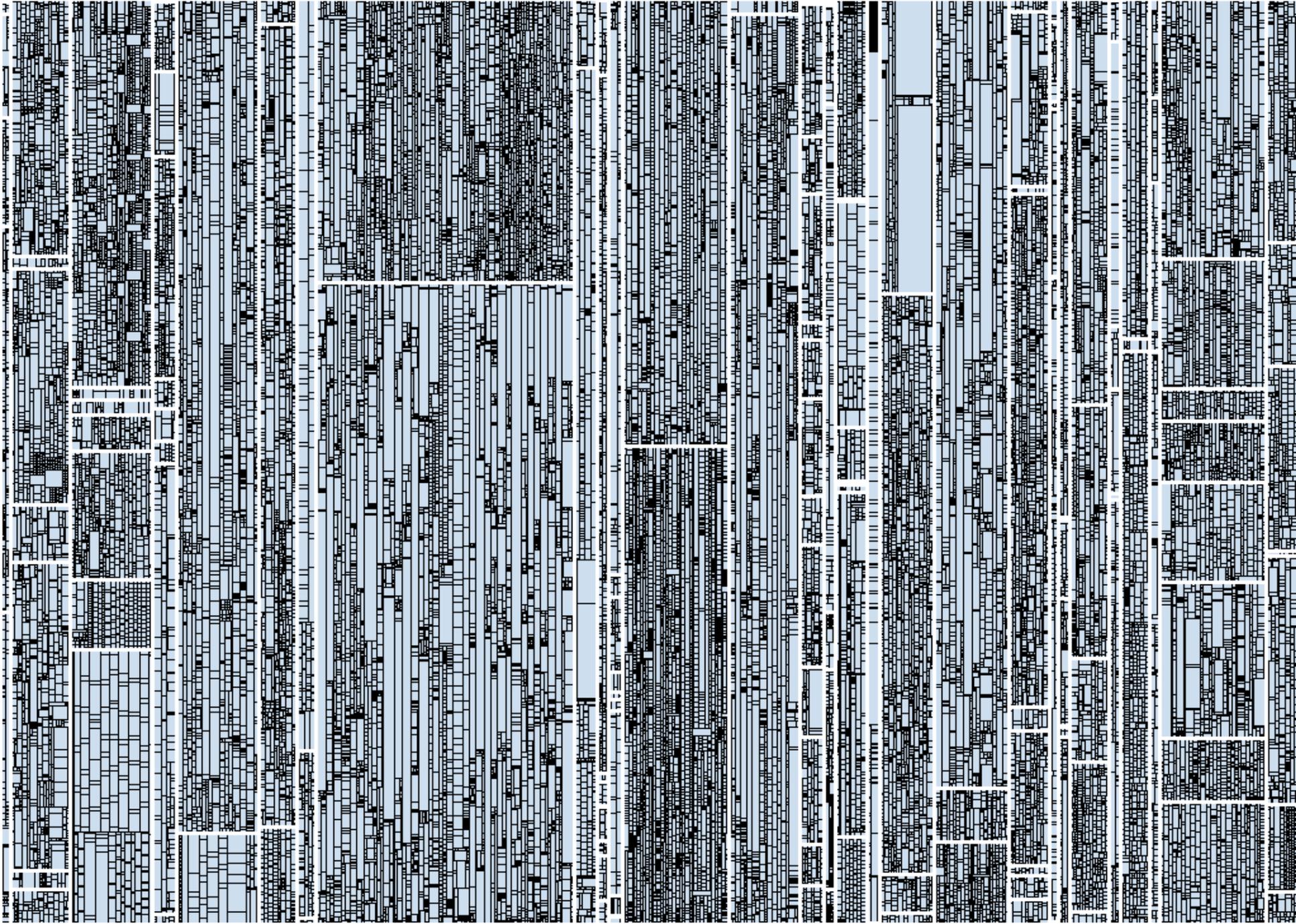
**Authentication**

**UI Controls**

**GUI.Base**

**Data  
Validation**





Changes

Test-Coverage

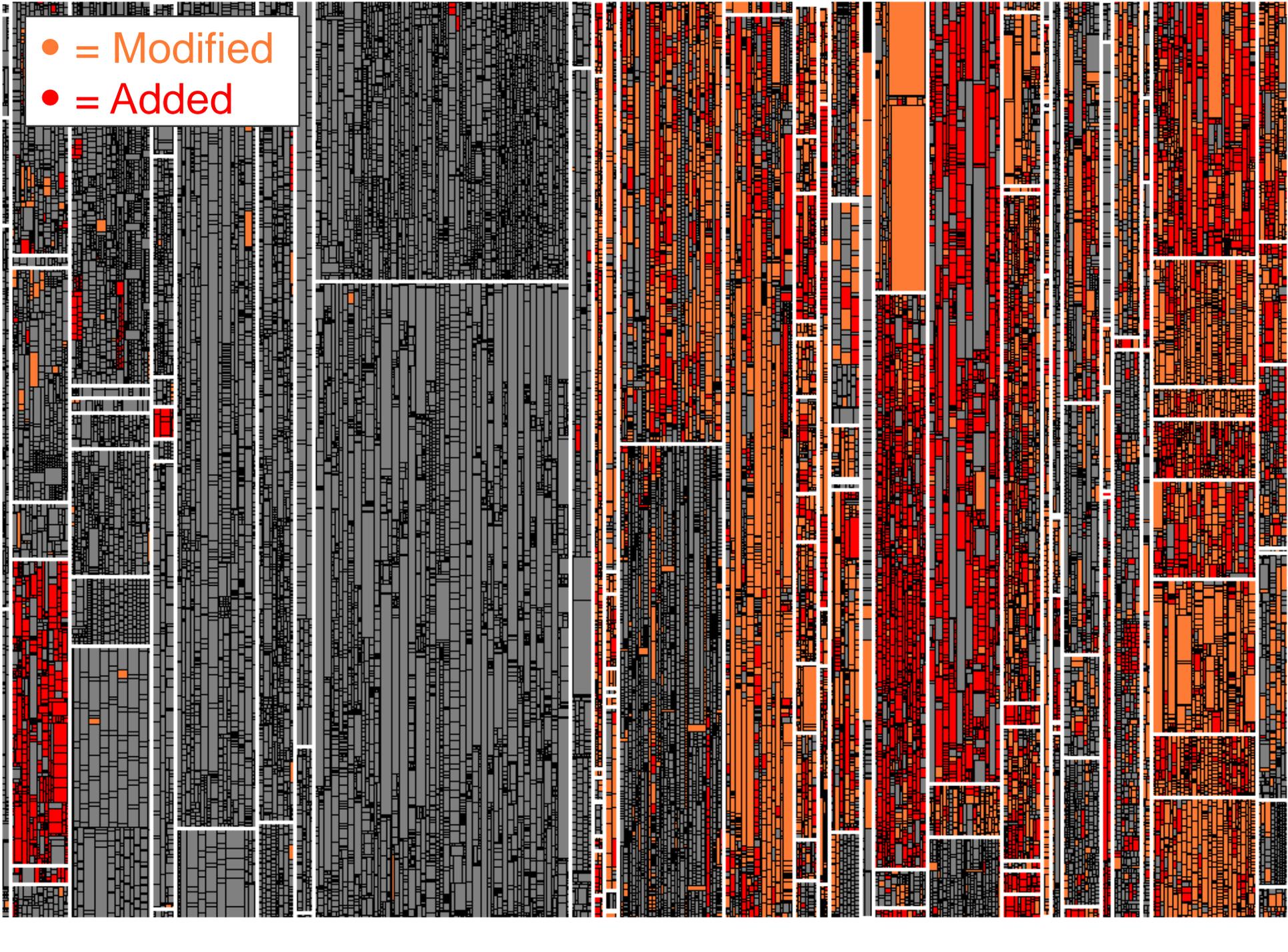


Test-Gap  
Analyse



Untested Changes

- = Modified
- = Added



Changes

Test-Coverage



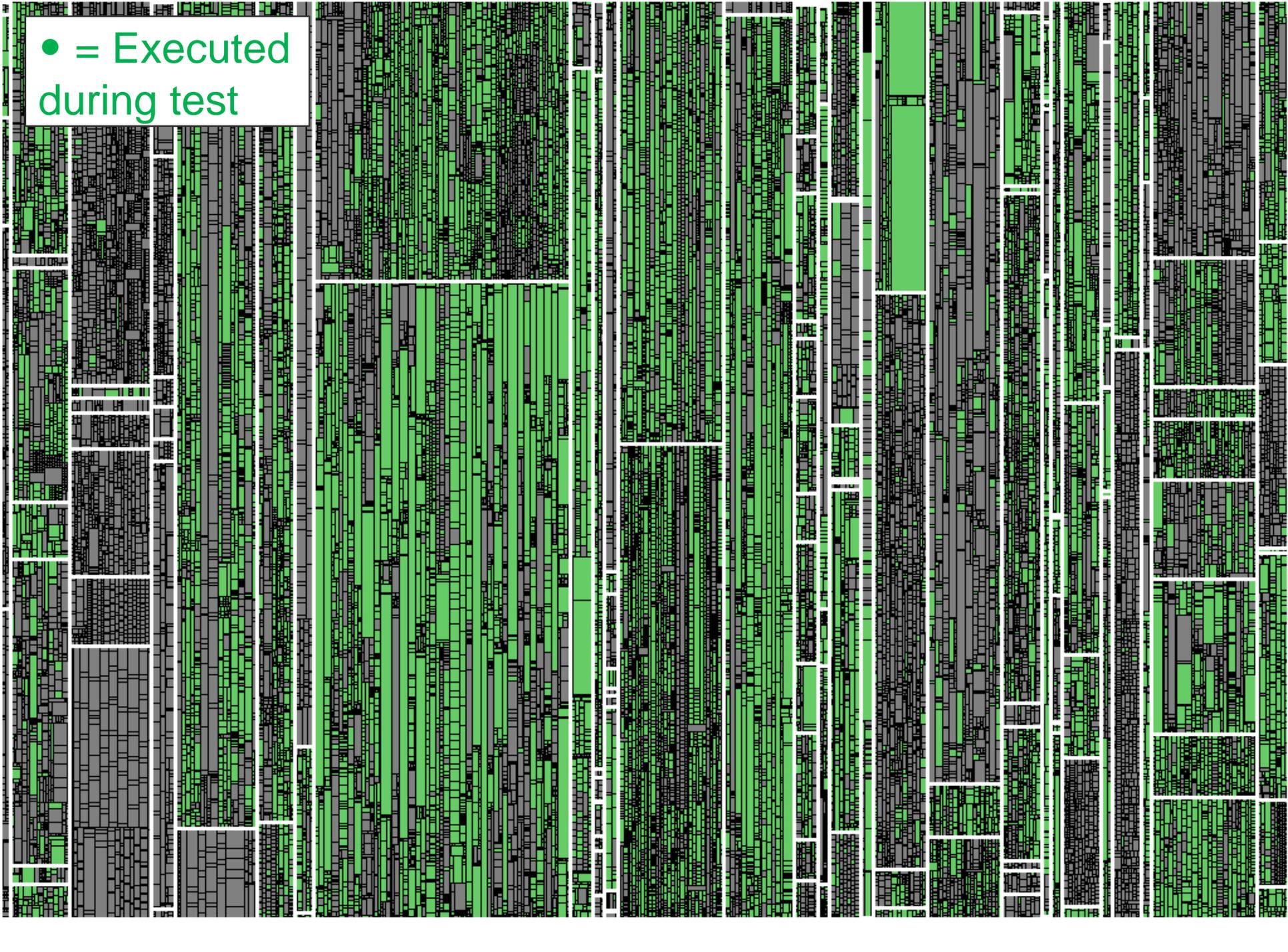
Test-Gap  
Analyse



Untested Changes



● = Executed during test



Changes

Test-Coverage

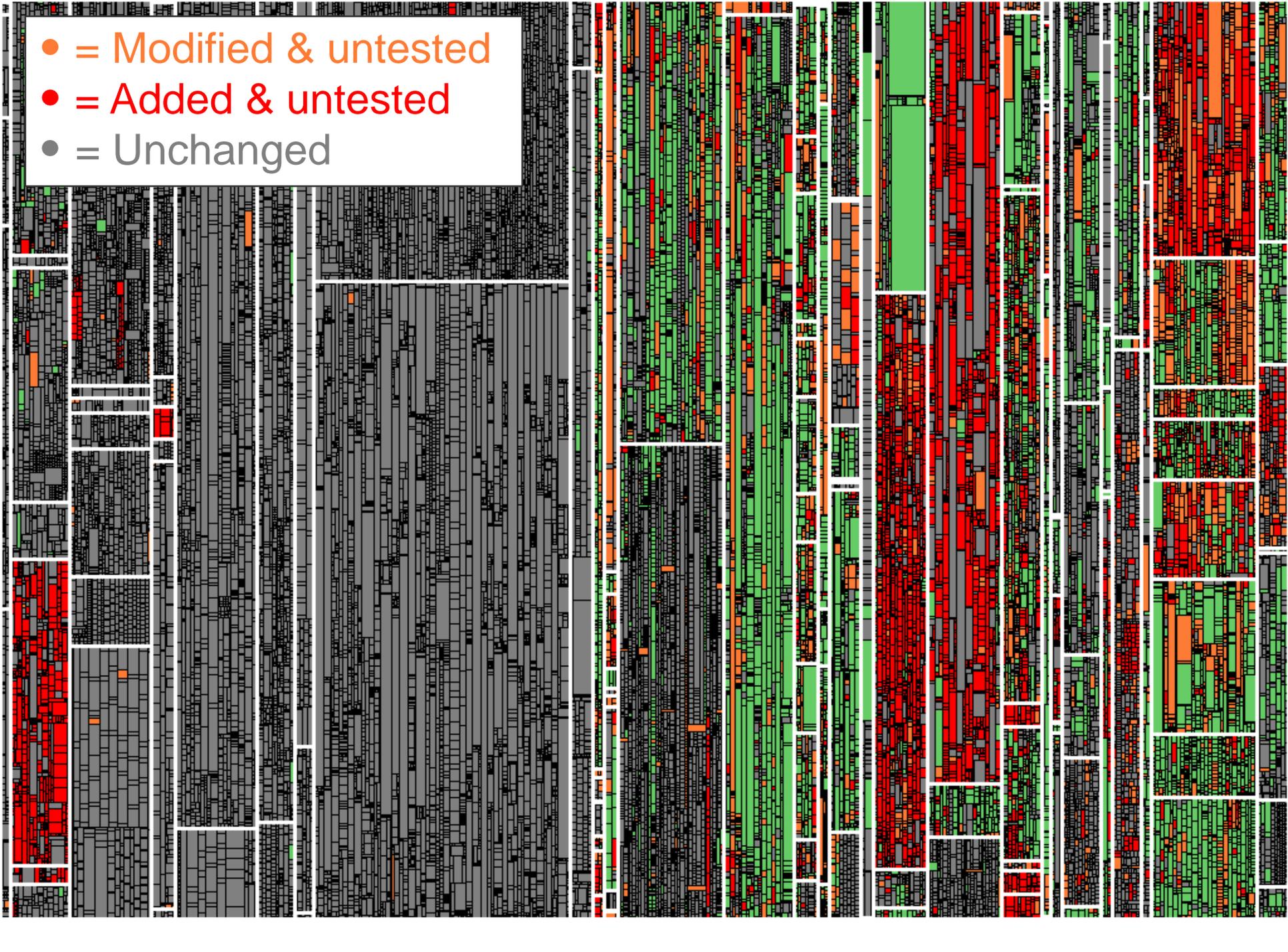


Test-Gap  
Analyse



Untested Changes

- = Modified & untested
- = Added & untested
- = Unchanged



Overview

Test Gap Analysis

- Changed Methods
- Executed Methods
- Test Gaps
- Details

Debug

- Traces
- Snapshots

Related Dashboards

- All
- Dev
- Test
- UAT

KPIs

- Churn
- Execution
- Test State

Info

- Log
- Config
- Execution Time
- Version

Test Gap Analysis

Changed Methods

Executed Methods

Test Gaps

Details

C#,

ABAP

General Information

Element	info
System	
Reference Version	20131111.1544.9.1.2.0
Newest Version	20140527.1545.9.2.300.6
First Test Run	2013-11-11 14:51:28
Latest Test Run	2014-05-27 16:32:15

Python,

Machine Overview

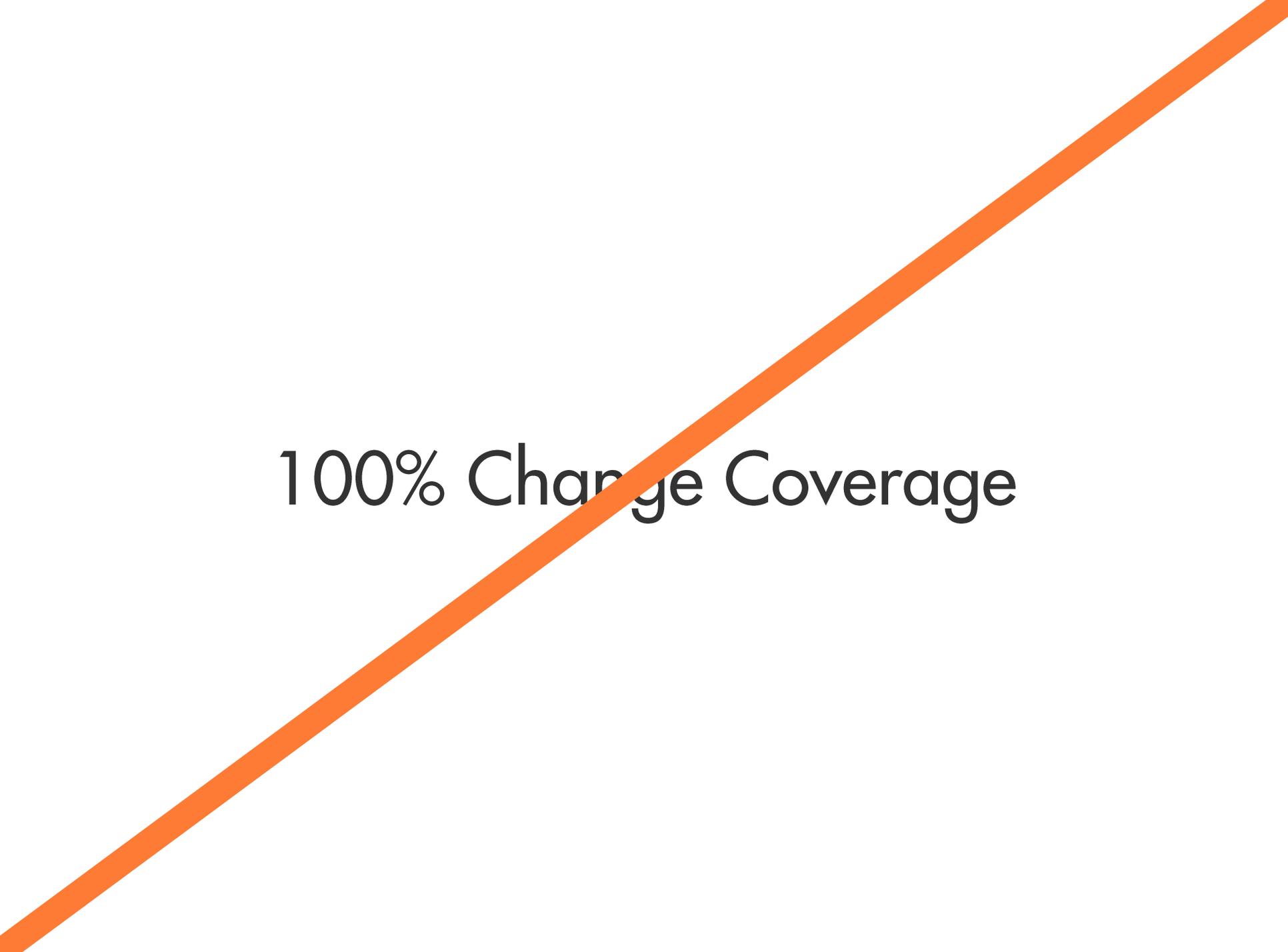
Machine	# Traces	# Versions	Latest Trace	Latest Version
devs805565	61	56	2014-05-27 13:58:39	20140527.1545.9.2.300.6
mucs801253	8	5	2014-05-19 11:06:44	20140519.1238.9.2.300.3
mucs801624	8	4	2014-05-27 16:32:15	20140519.1238.9.2.300.3

Java,

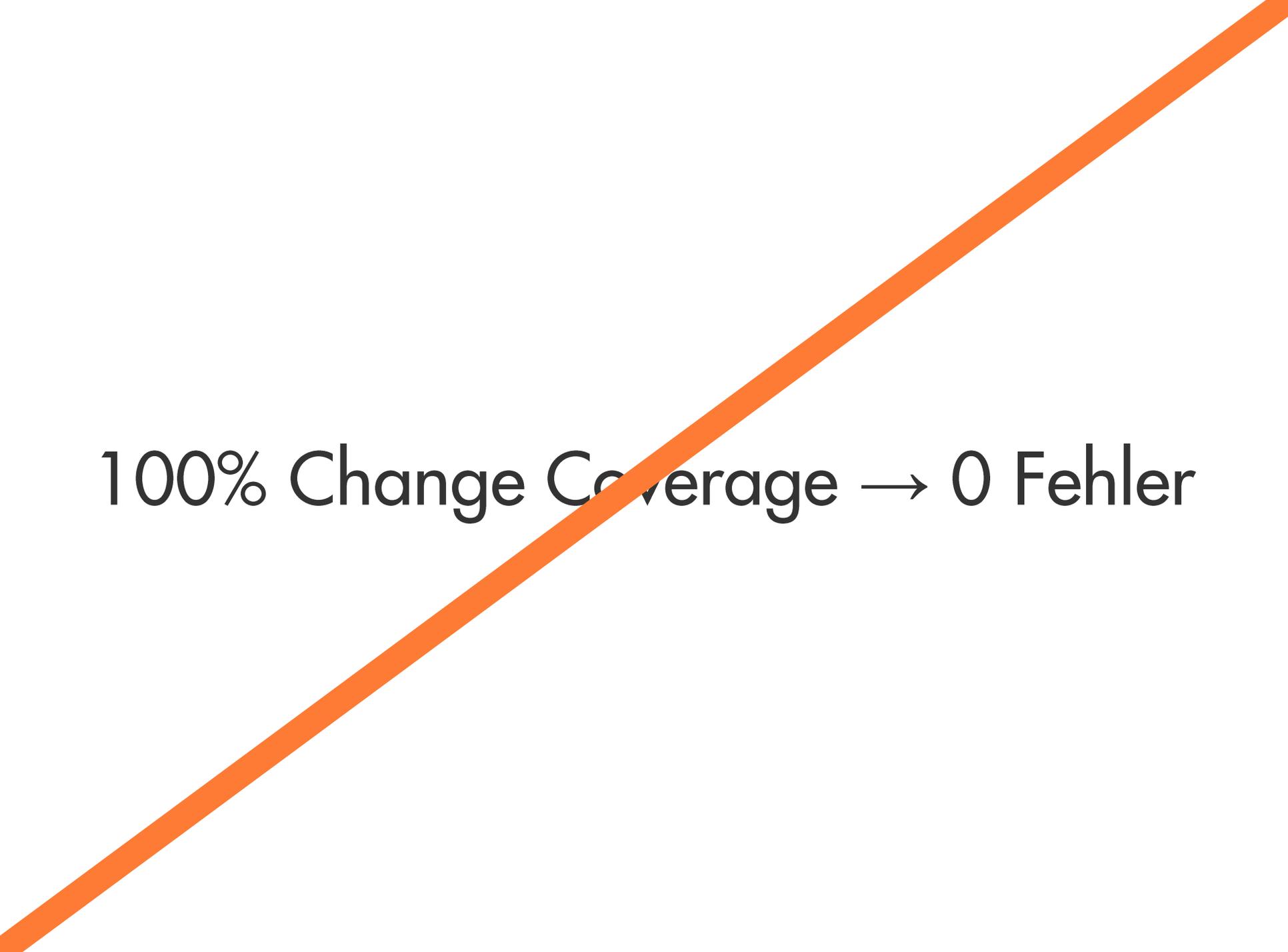
Version Overview

Version	# Traces	# Traced Machines	Latest Trace
20131111.1544.9.1.2.0	5	1	2014-02-28 16:59:11
20131205.0739.9.2.101.3	1	1	2013-12-06 11:05:54
20131210.1604.9.2.101.4	1	1	2013-12-10 15:09:09
20131211.0935.9.2.101.5	1	1	2013-12-11 08:40:53
20131211.1245.9.2.101.5	3	2	2014-01-30 13:05:58
20131220.1032.9.2.101.6	1	1	2013-12-20 09:46:55
20140102.0820.9.2.101.7	1	1	2014-01-02 07:25:44
20140107.1152.9.2.101.8	1	1	2014-01-07 11:04:26
20140110.0838.9.2.101.9	1	1	2014-01-10 07:51:55

C++

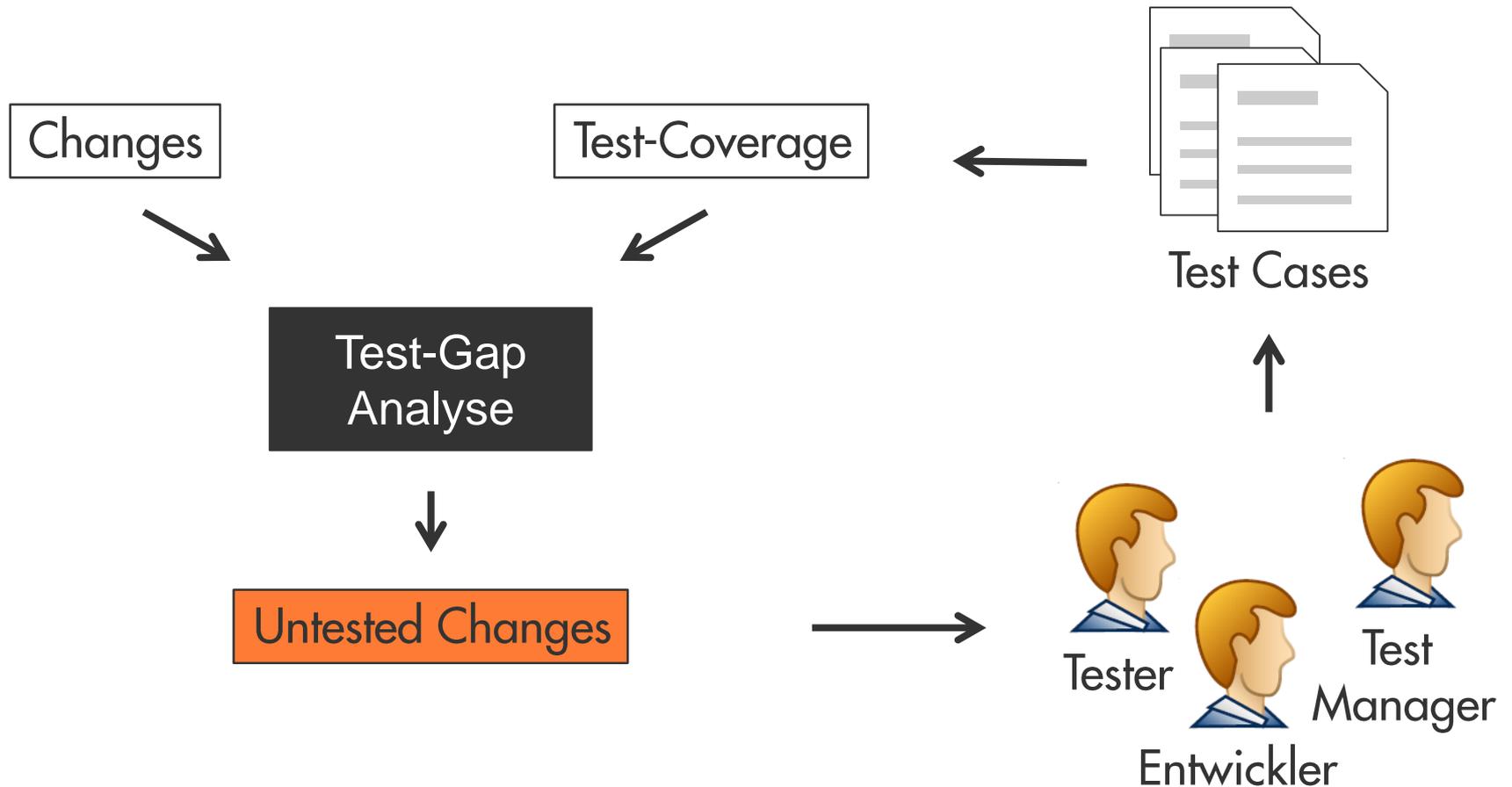
A thick orange diagonal line runs from the bottom-left corner to the top-right corner of the image, passing through the text.

100% Change Coverage



100% Change Coverage → 0 Fehler

# Einsatz im Testprozess



# Agenda

1. Test-Gap-Analyse
2. **Erfahrungen aus dem Einsatz bei der Munich Re**
3. Diskussion

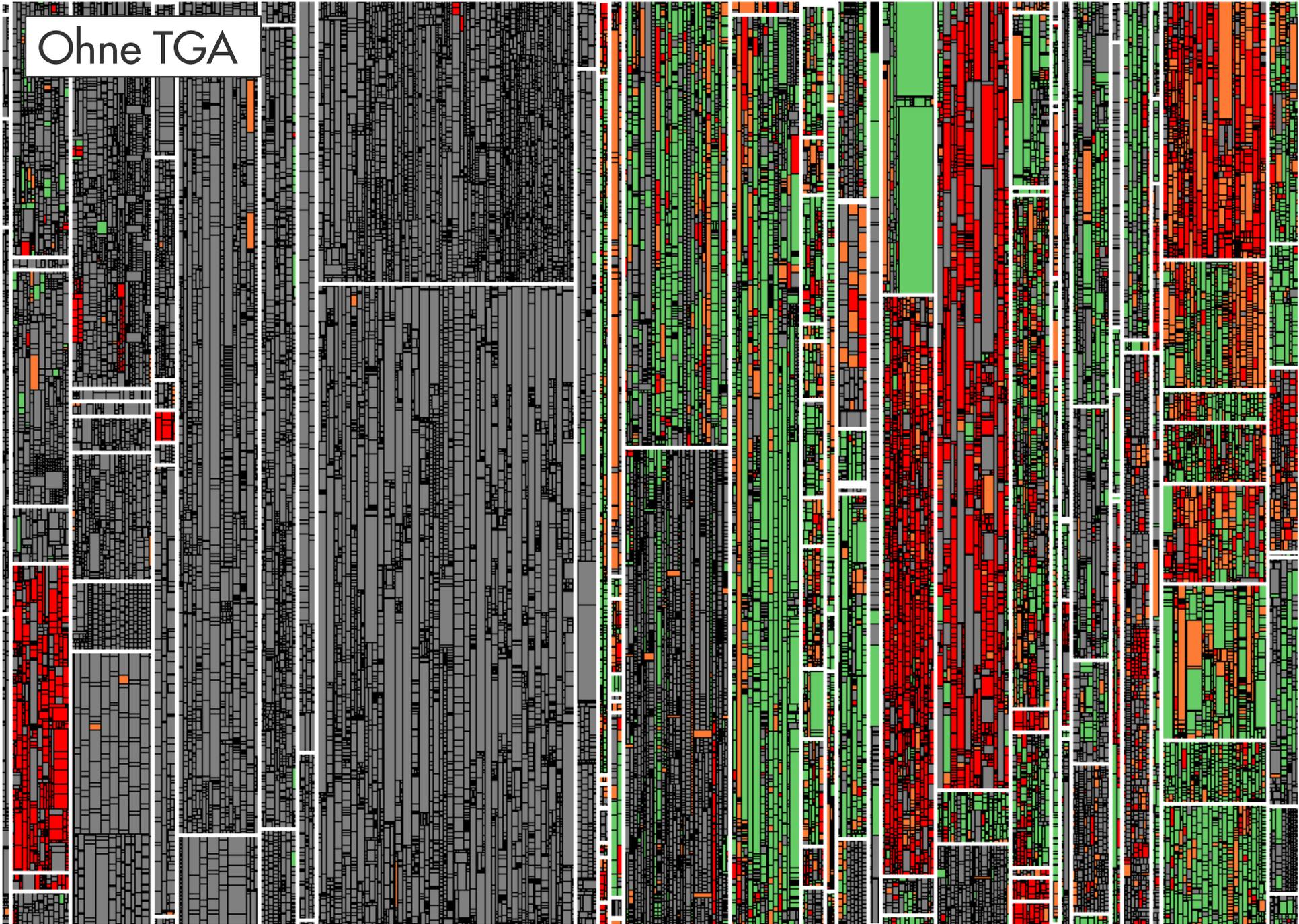
# Portfolio-Dashboard

- Zentrale TGA-Informationen für alle Anwendungen auf einen Blick
- Direkte Links auf Projekt-Dashboards mit detaillierter Information
- Macht TGA-Dashboards für alle einfach zu finden
- Erhöht Transparenz über Anwendungen hinweg

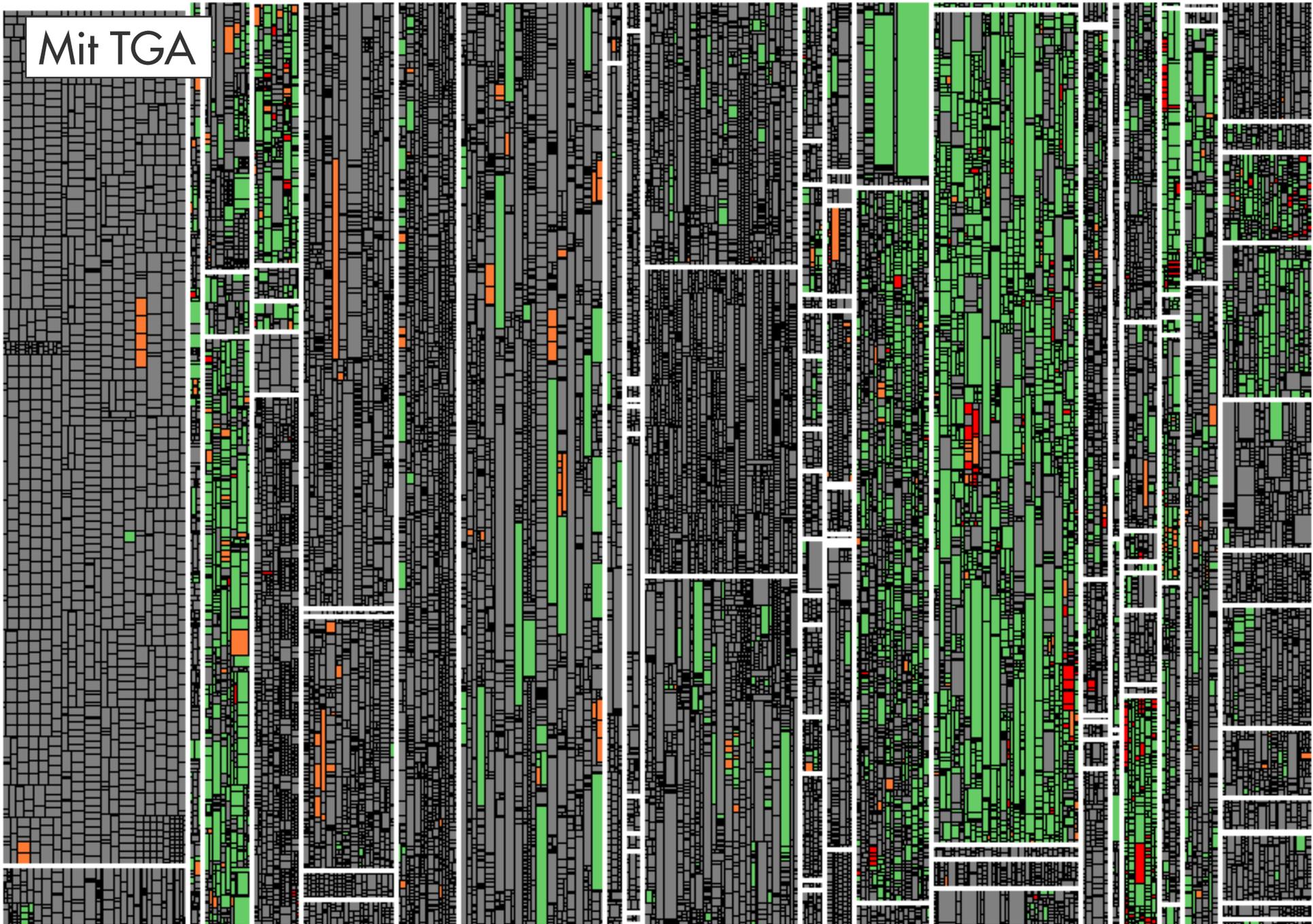
## Portfolio Overview

System	TGA	Technology	Test Gap
...	Link	.NET	71.4%
...	Link	ABAP/BW	28.7%
...	Link	.NET	38.7%
...	Link	.NET	5.9%
...	Link	.NET	7.6%
...	Link	.NET	11.7%
...		.NET	
...	Link	.NET	not yet run
...	Link	.NET	15.5%
...		OScript	
...	Link	ABAP	32.4%
...	Link	ABAP/BW	28.7%
...	Link	.NET	10.7%
...	Link	.NET	42.8%
...	Link	ABAP	51.1%
...		JavaScript(DB)	
...	Link	.NET	not run
...			
...			
...			
...	Link	.NET	50.2%
...	Link	.NET	41.1%
...	Link	.NET	93.4%
...	Link	ABAP/BW	2.8%
...			
...	Link	.NET	15.6%
...	Link	.NET	30.2%
...	Link	.NET	15.6%
...	Link	ABAP/BW	28.7%
...	Link	.NET	5.6%
...		SAP BPM	

Ohne TGA



Mit TGA



# Lesson Learned

Test-Gap-Analyse schafft Transparenz. Diese Transparenz wird von Teams genutzt, um ungewollt ungetesteten Code zu vermeiden.

# Test-Arten

## Hotfix-Test

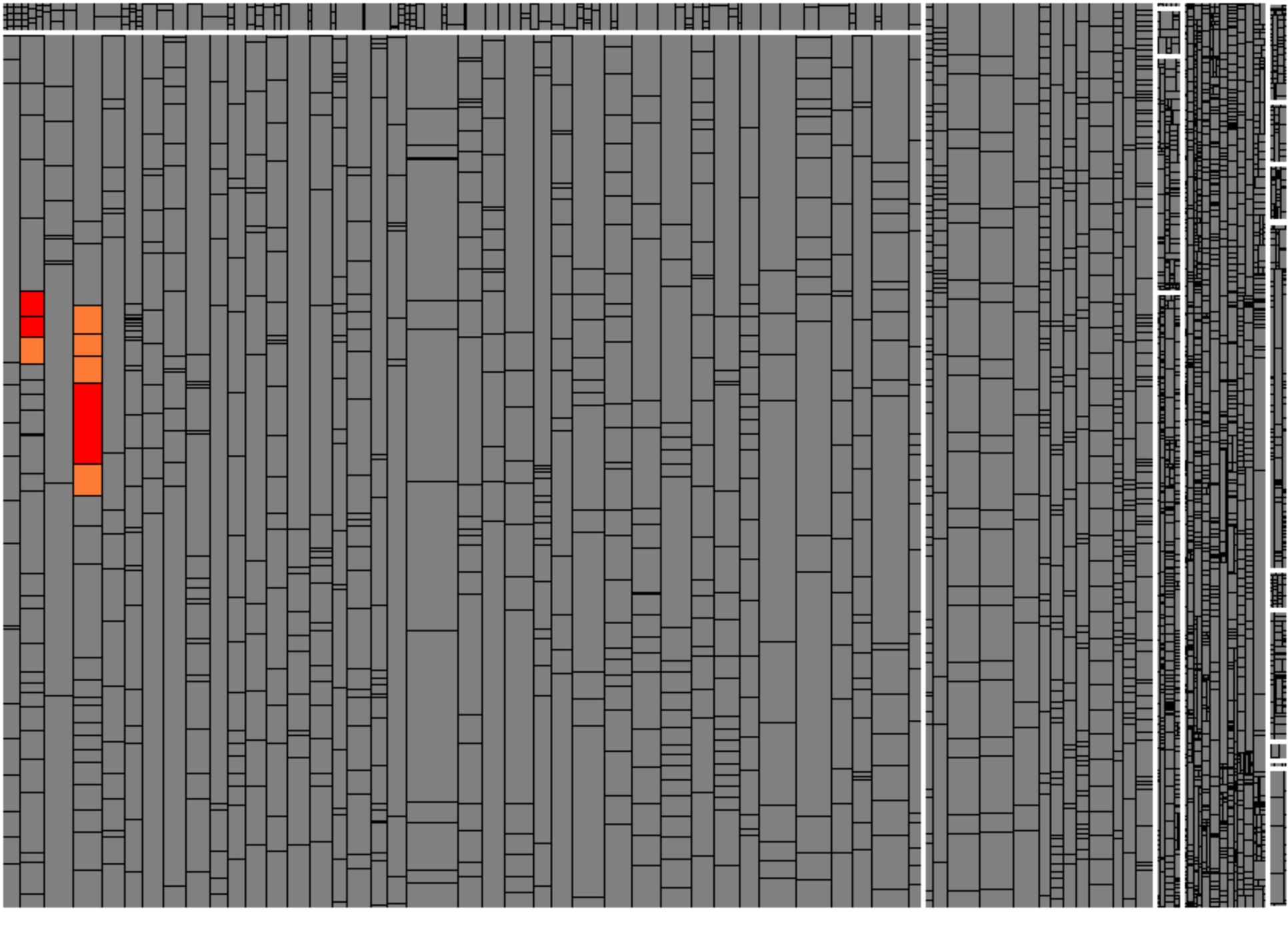
- On-Demand, wenig Zeit
- Getestet werden nur Änderungen des Hotfix

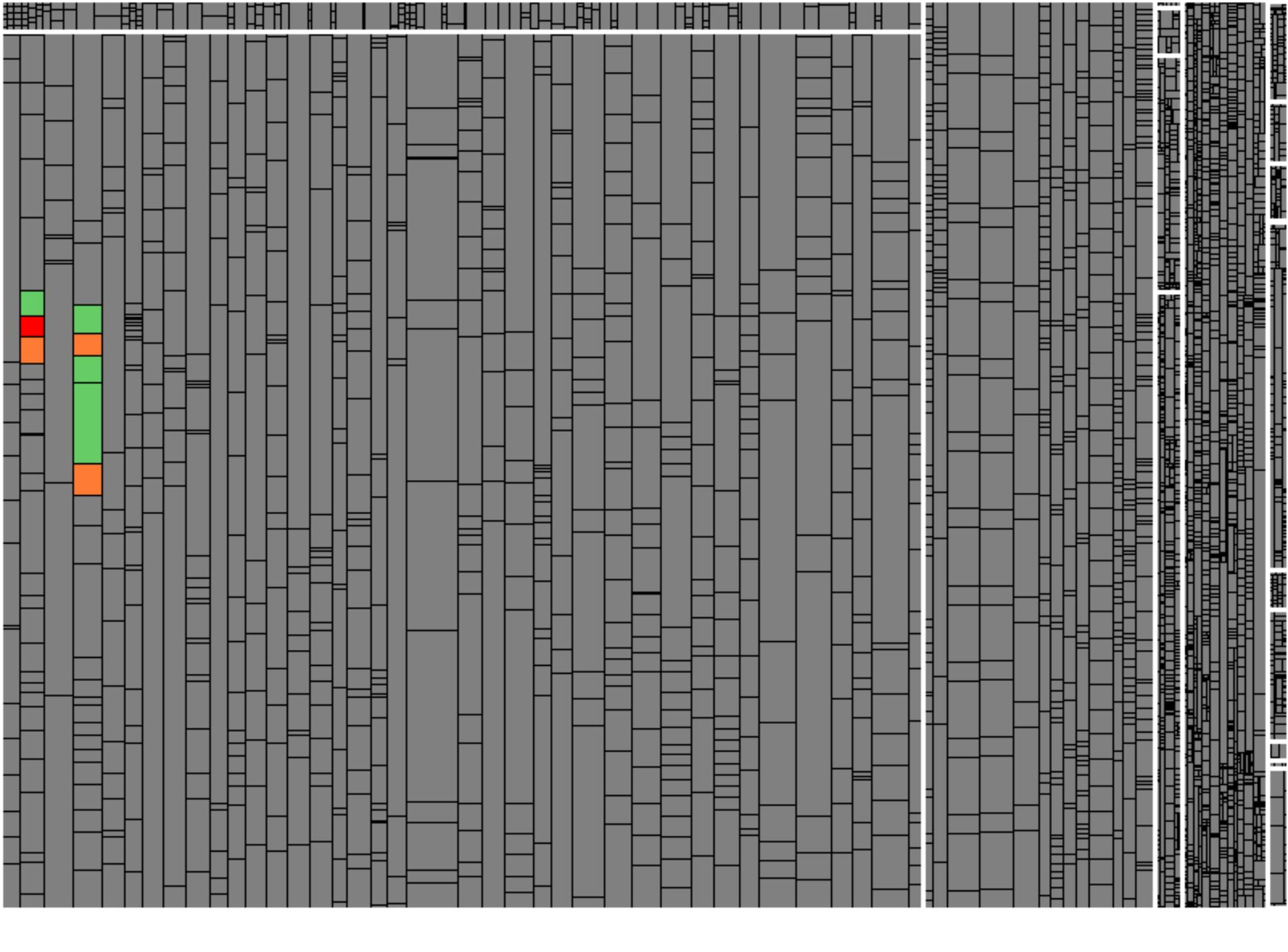
## Release-Test

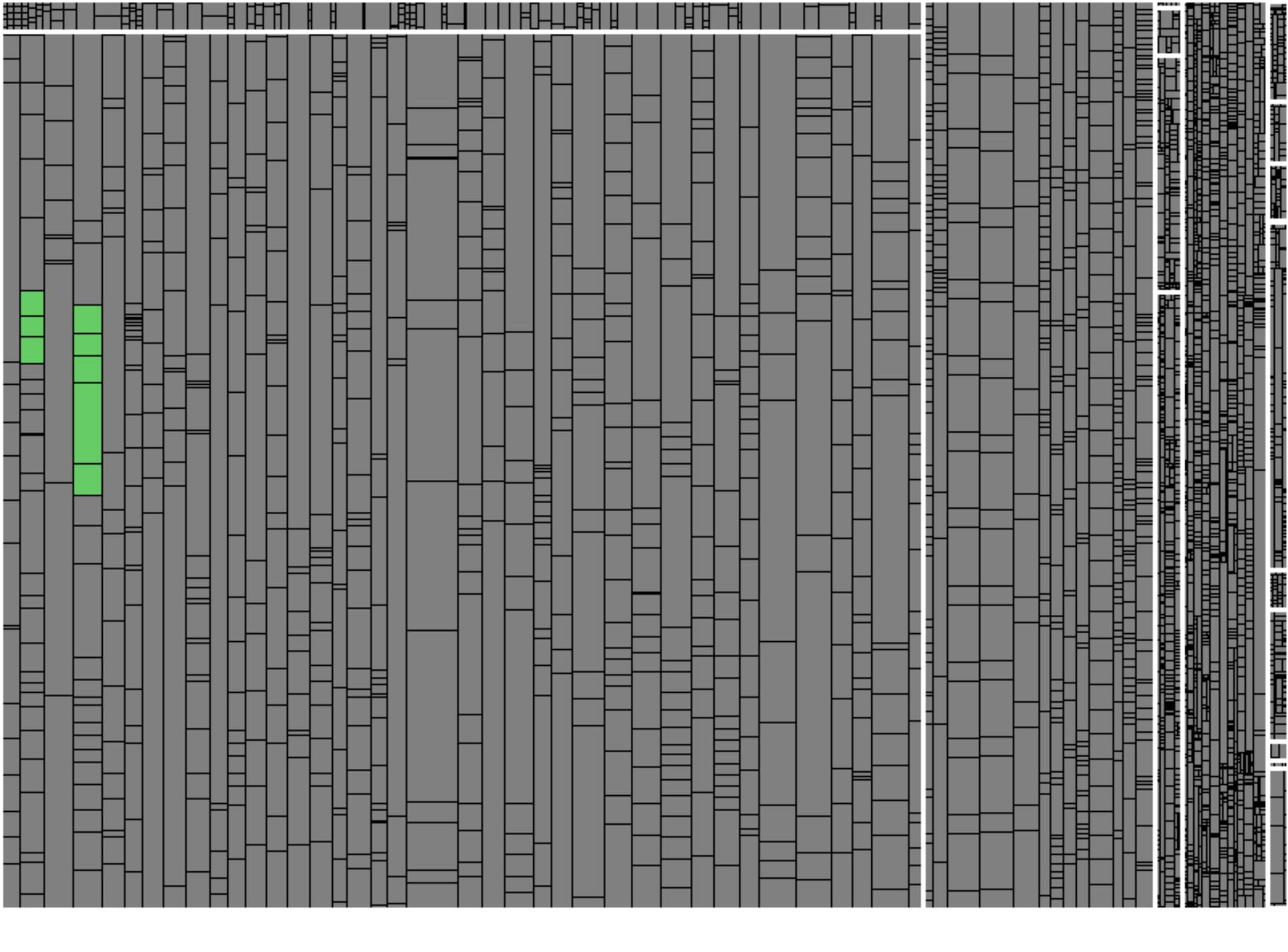
- Nur vor Release, Dauer mehrere Wochen
- Getestet werden alle Änderungen seit letztem Release
- 2-4 Mal pro Jahr (abh. Von Release-Zyklus der Anwendung)

## Iterations-Test

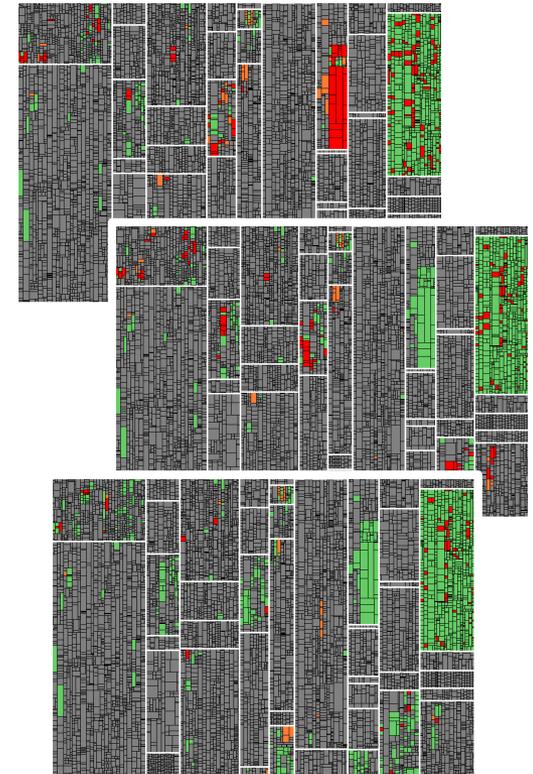
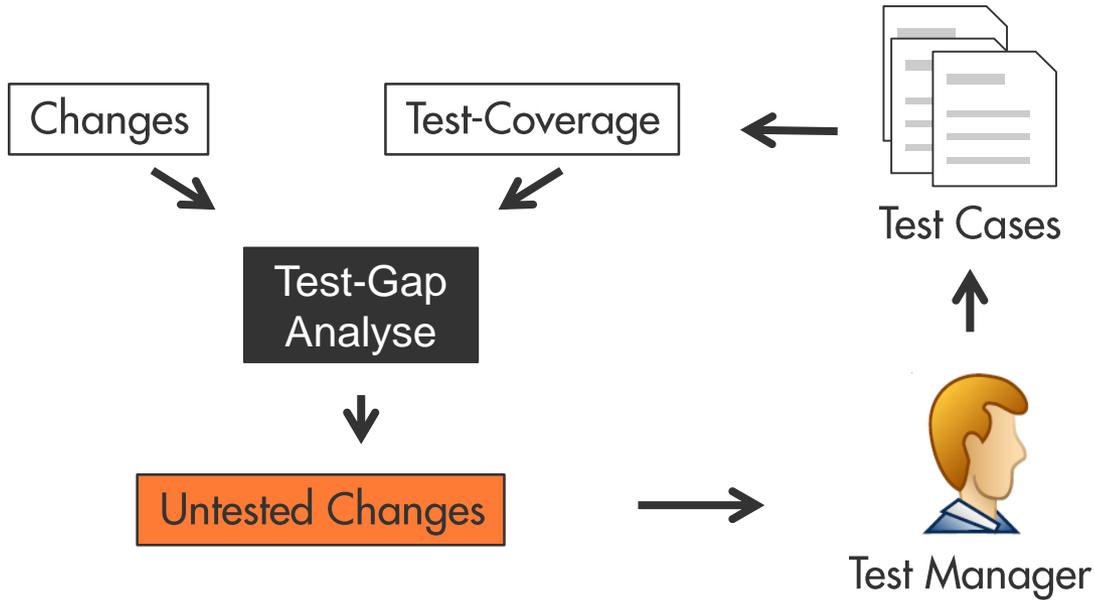
- Entwicklungsbegleitend, kontinuierlich, Iterationsdauer (2-4 Wochen)
- Getestet werden Änderungen aus der Iteration

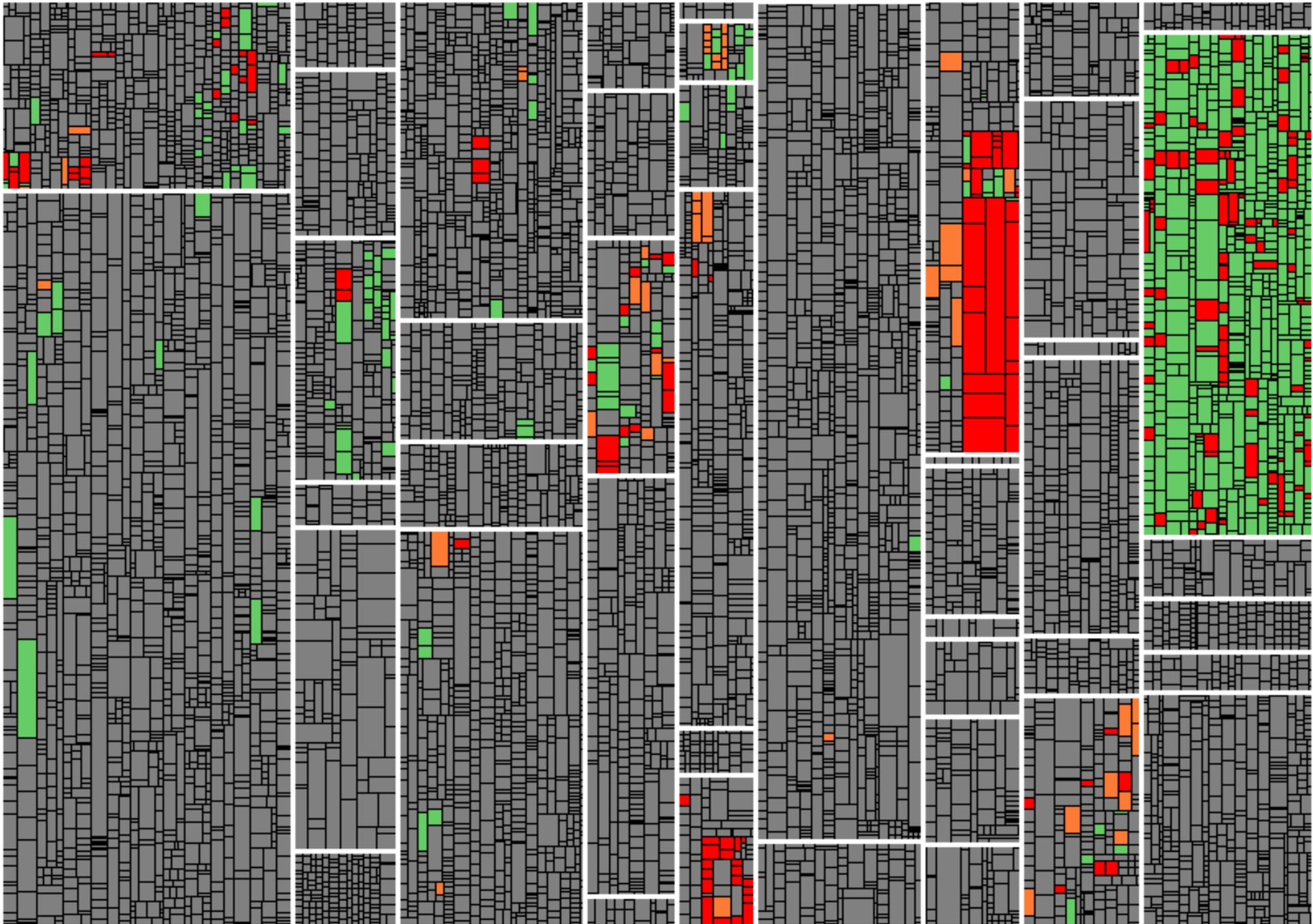


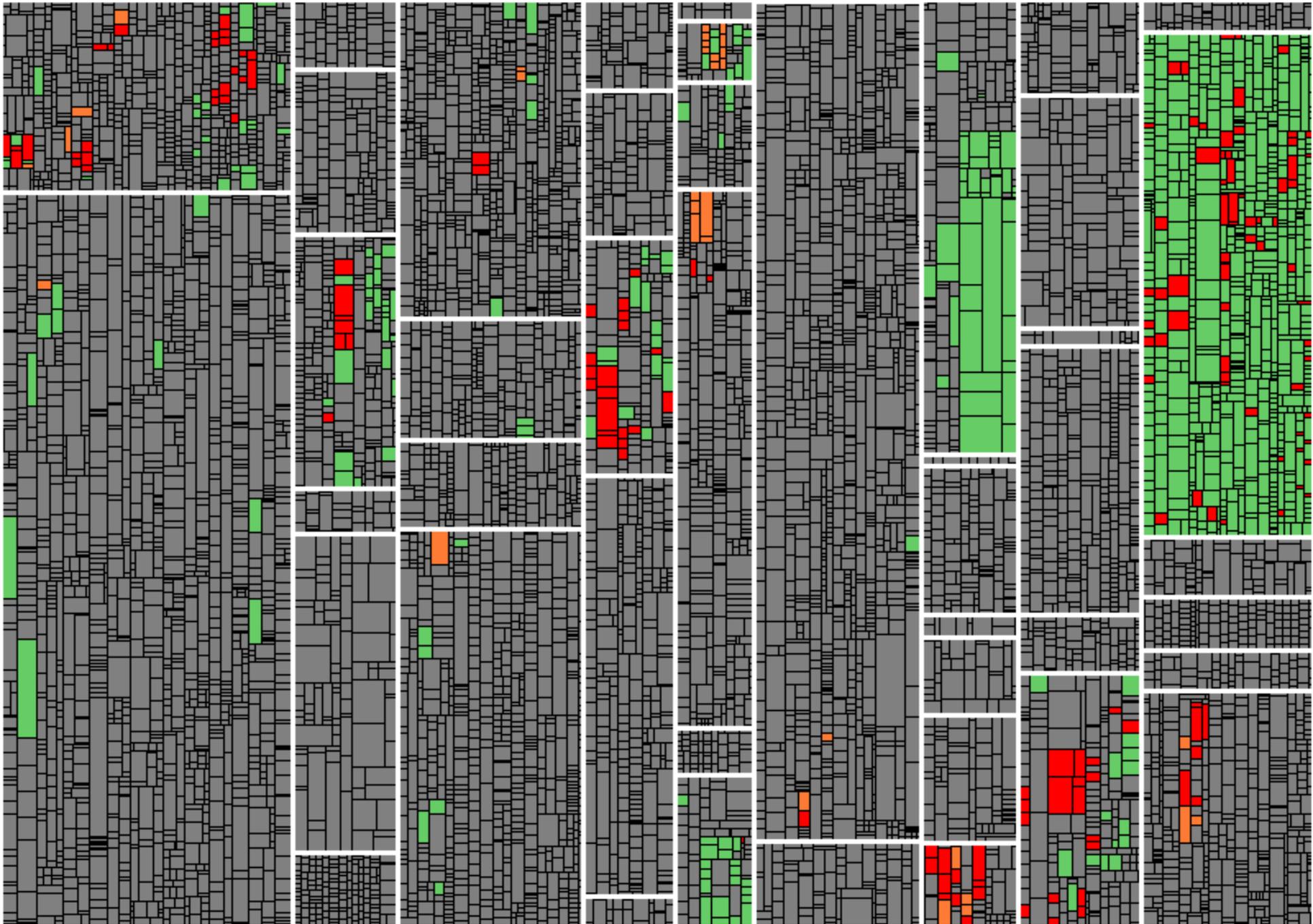


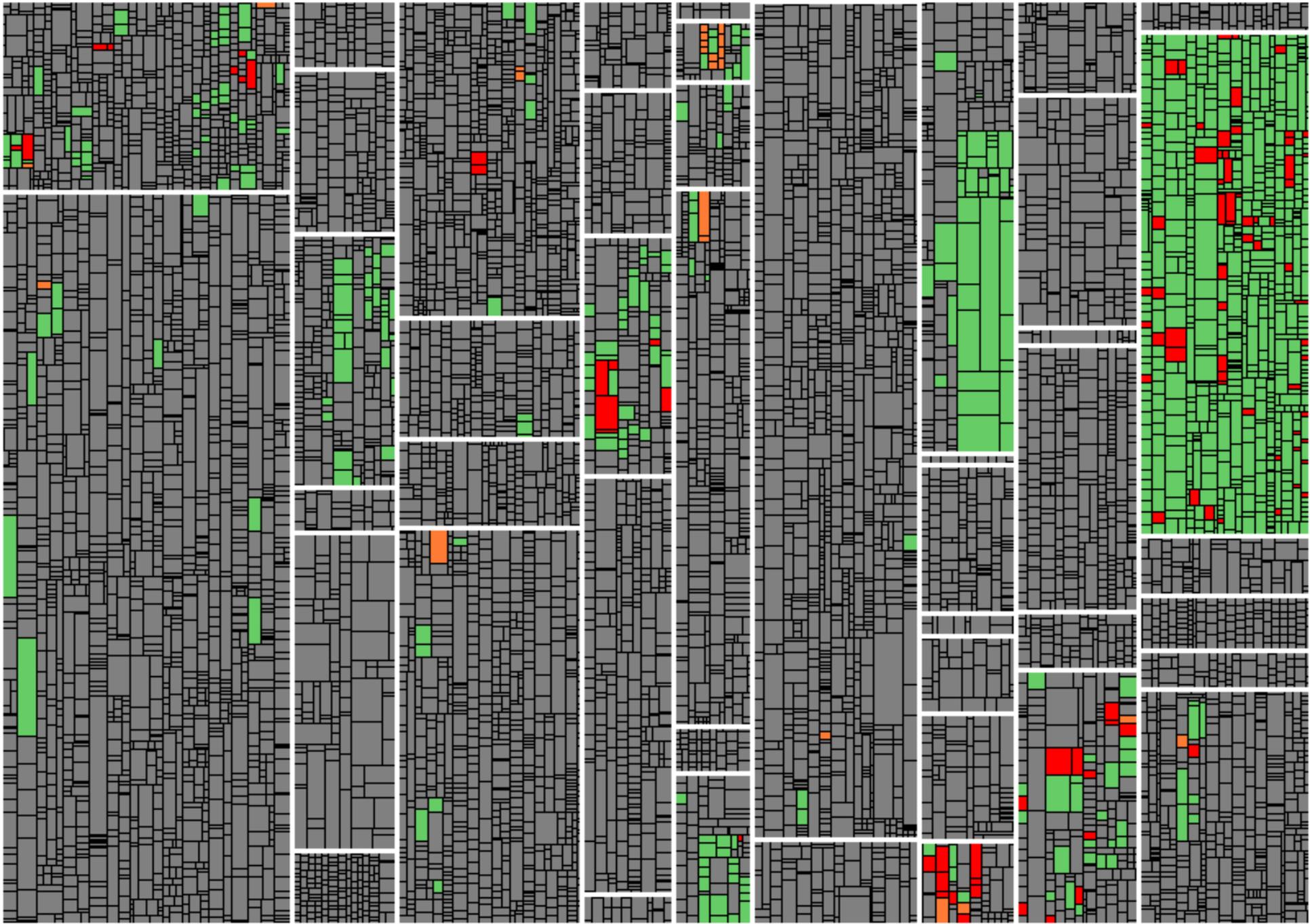


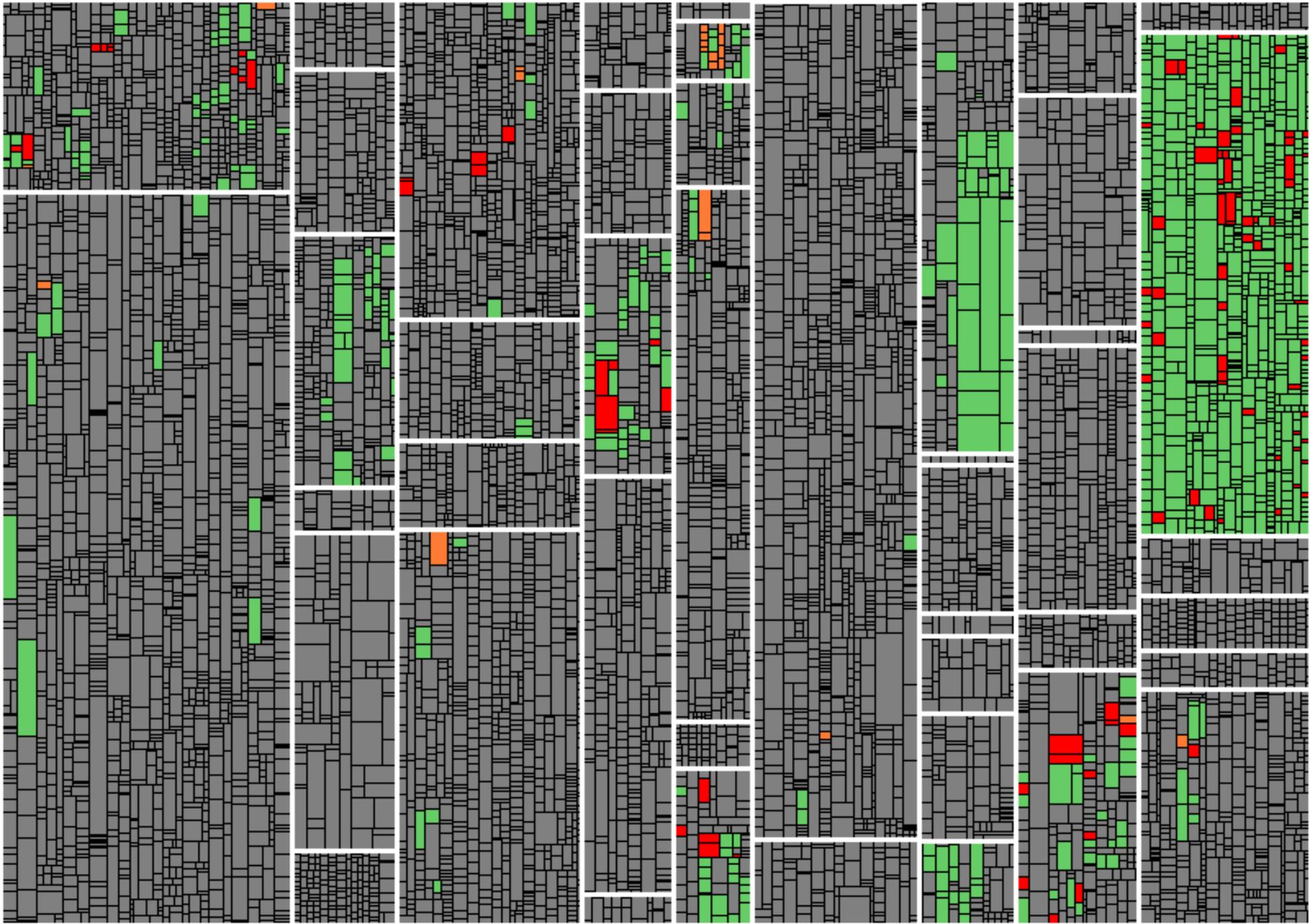
# Release-Test: Beispiel (1)

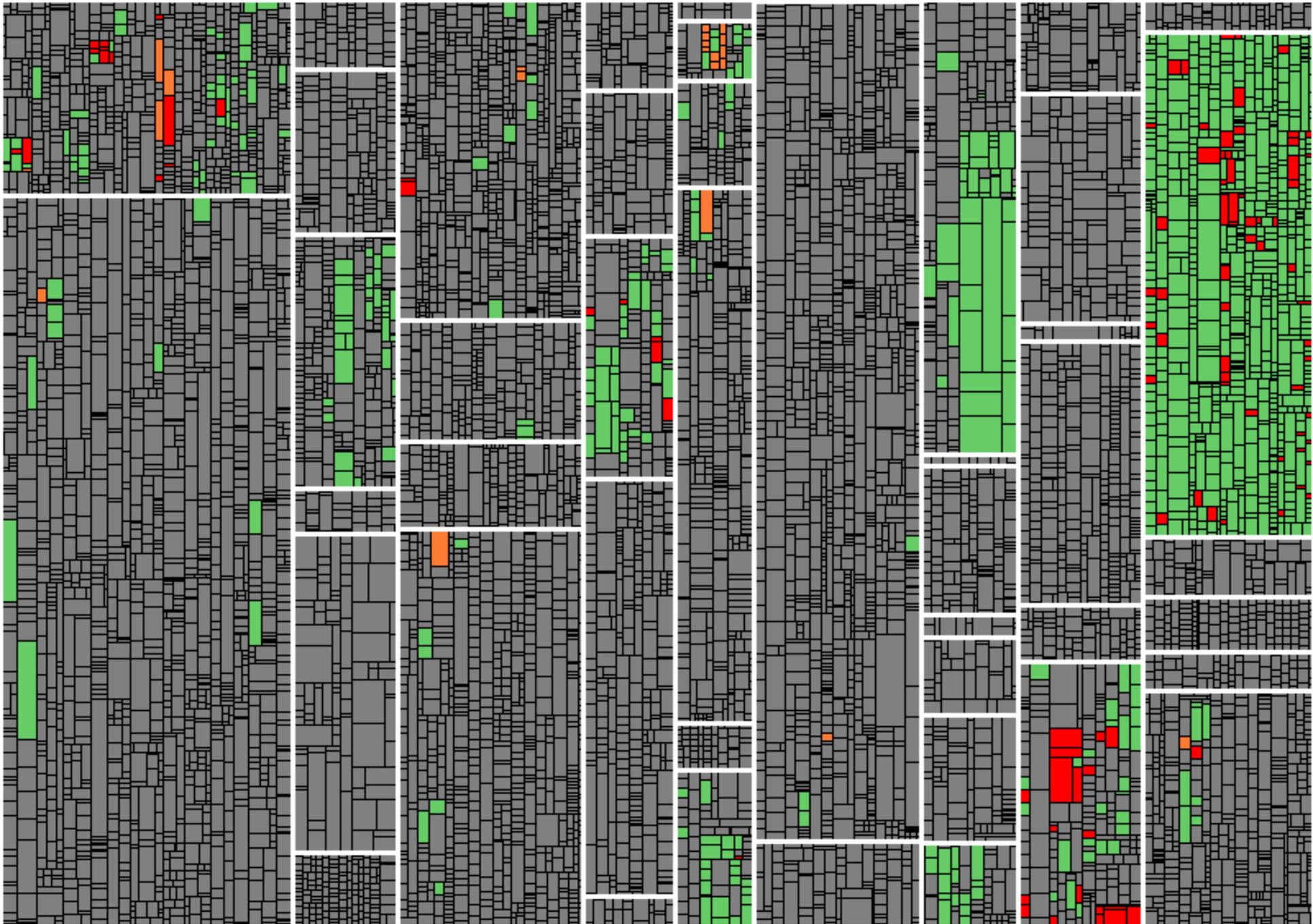


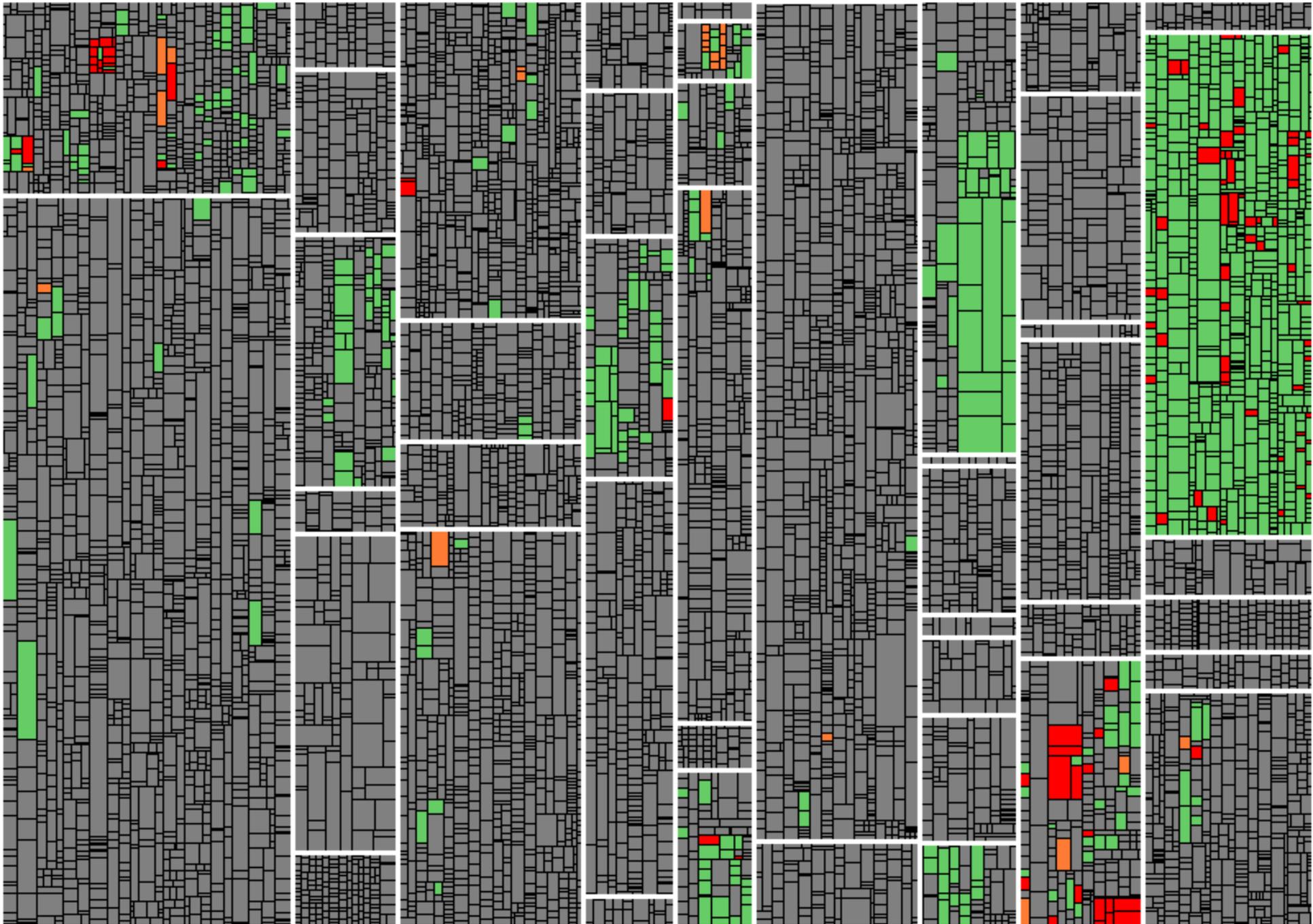


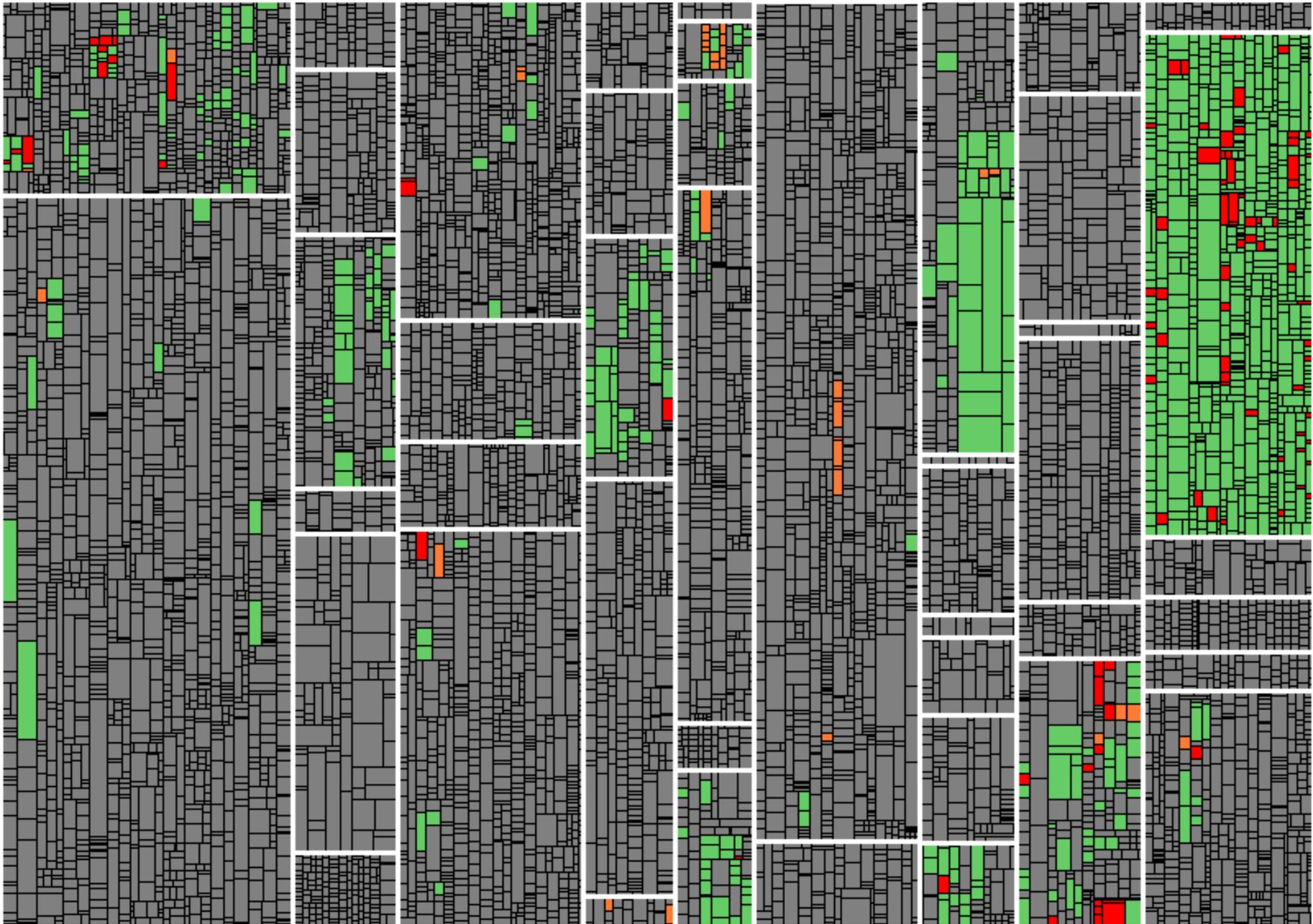


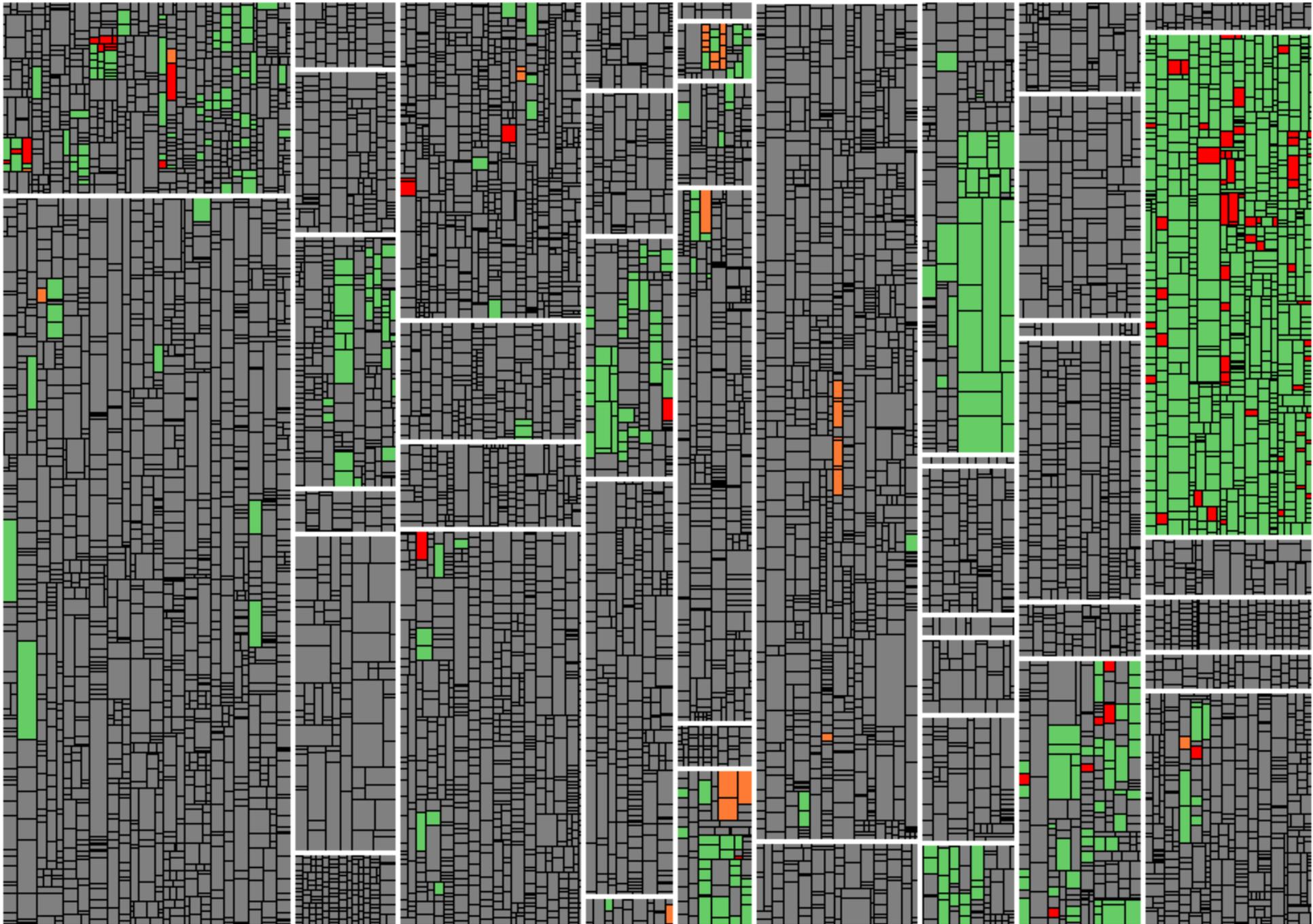




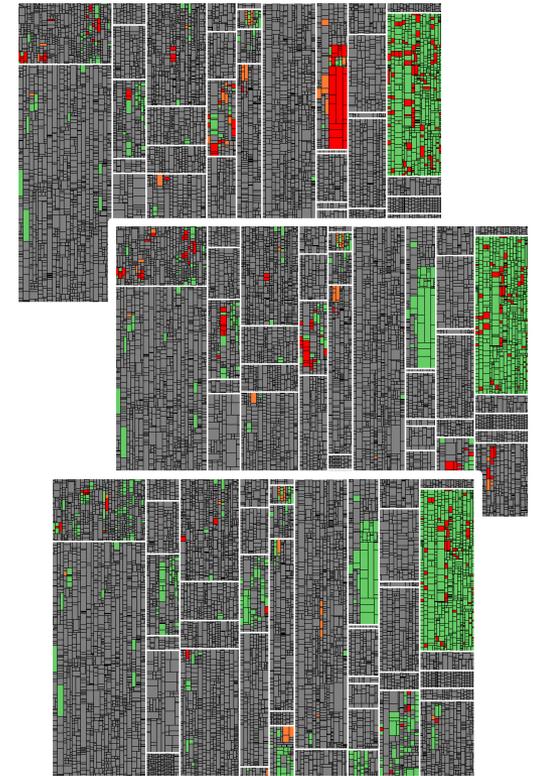
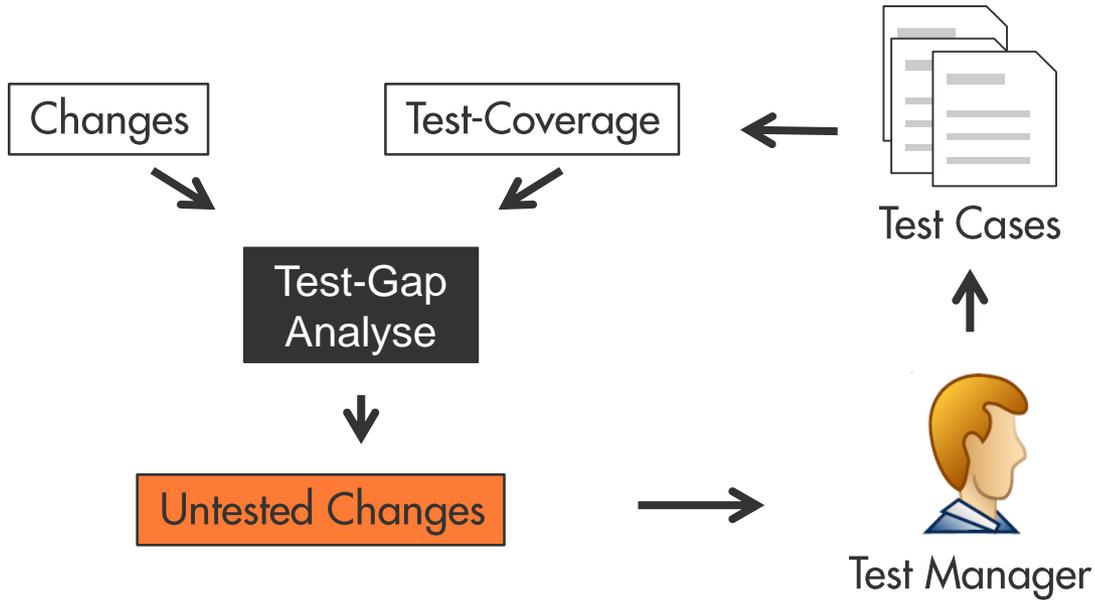


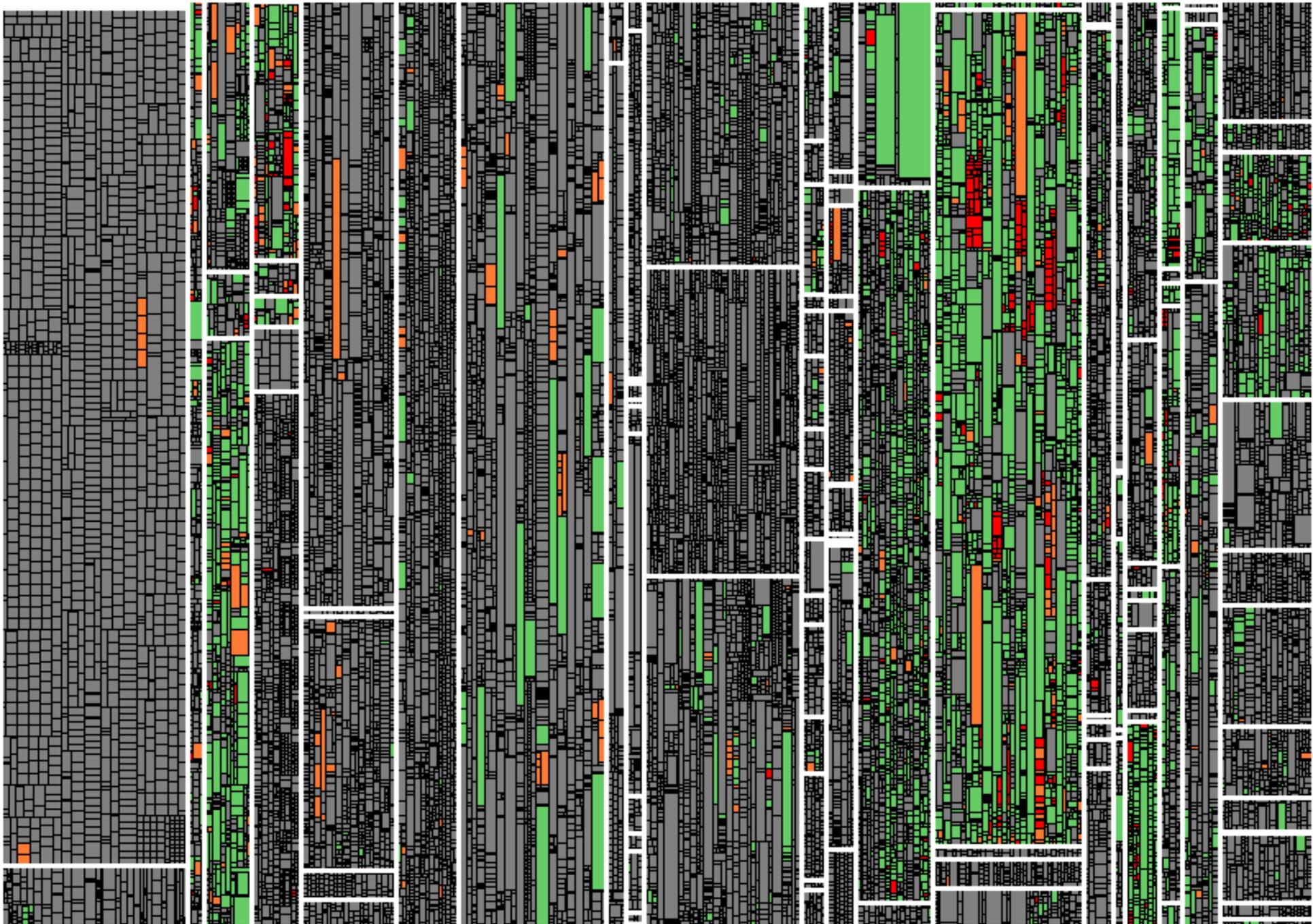


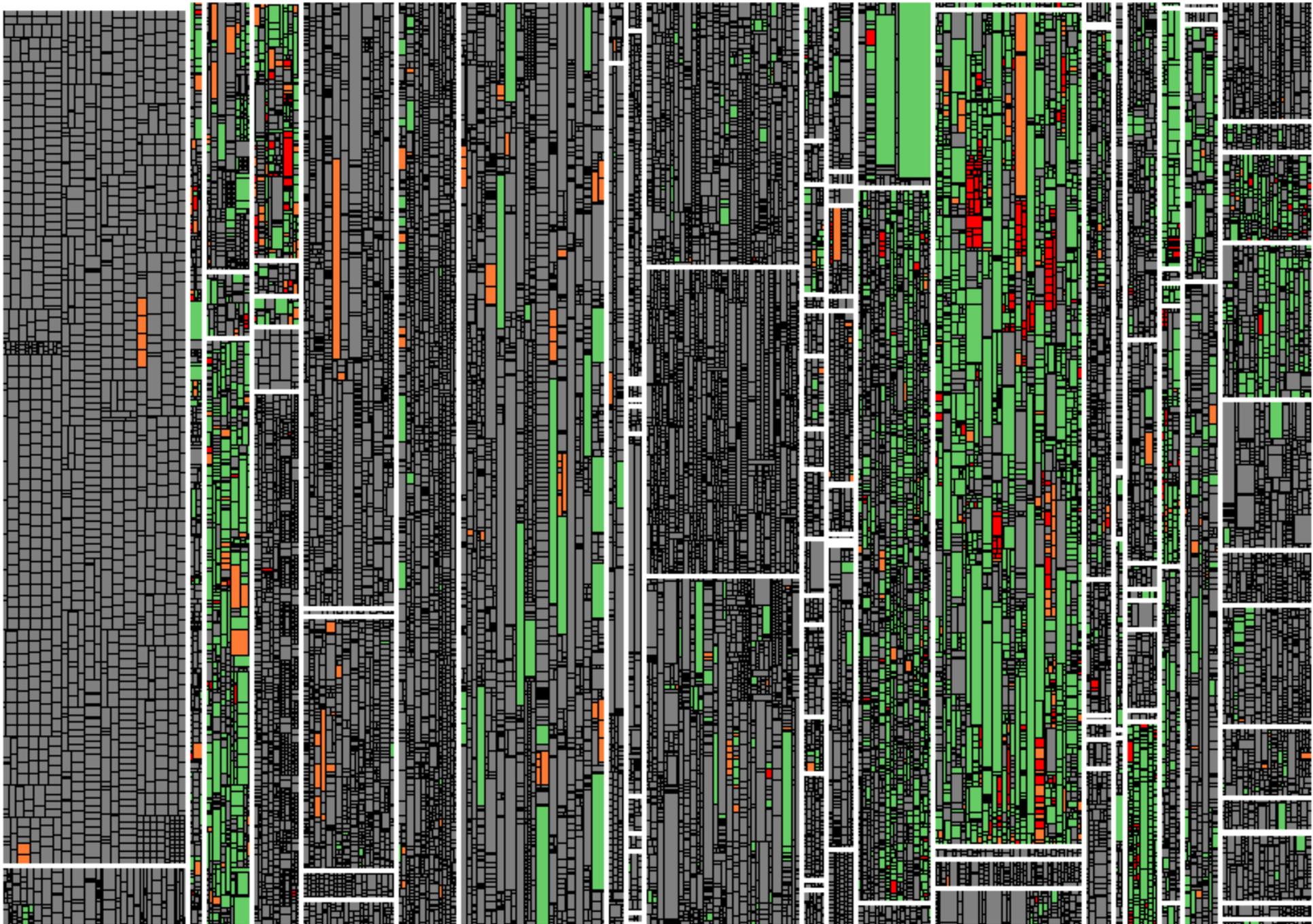


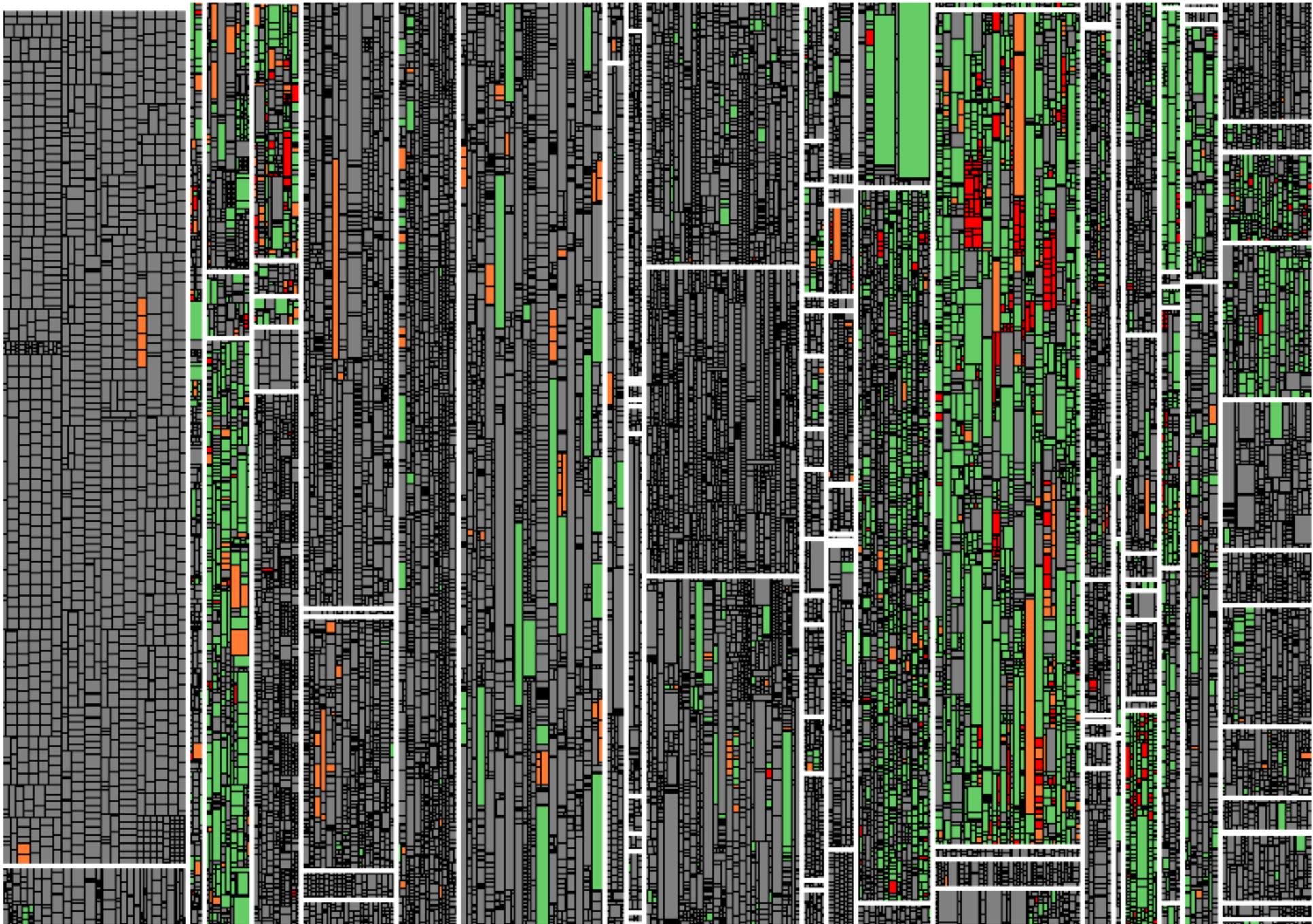


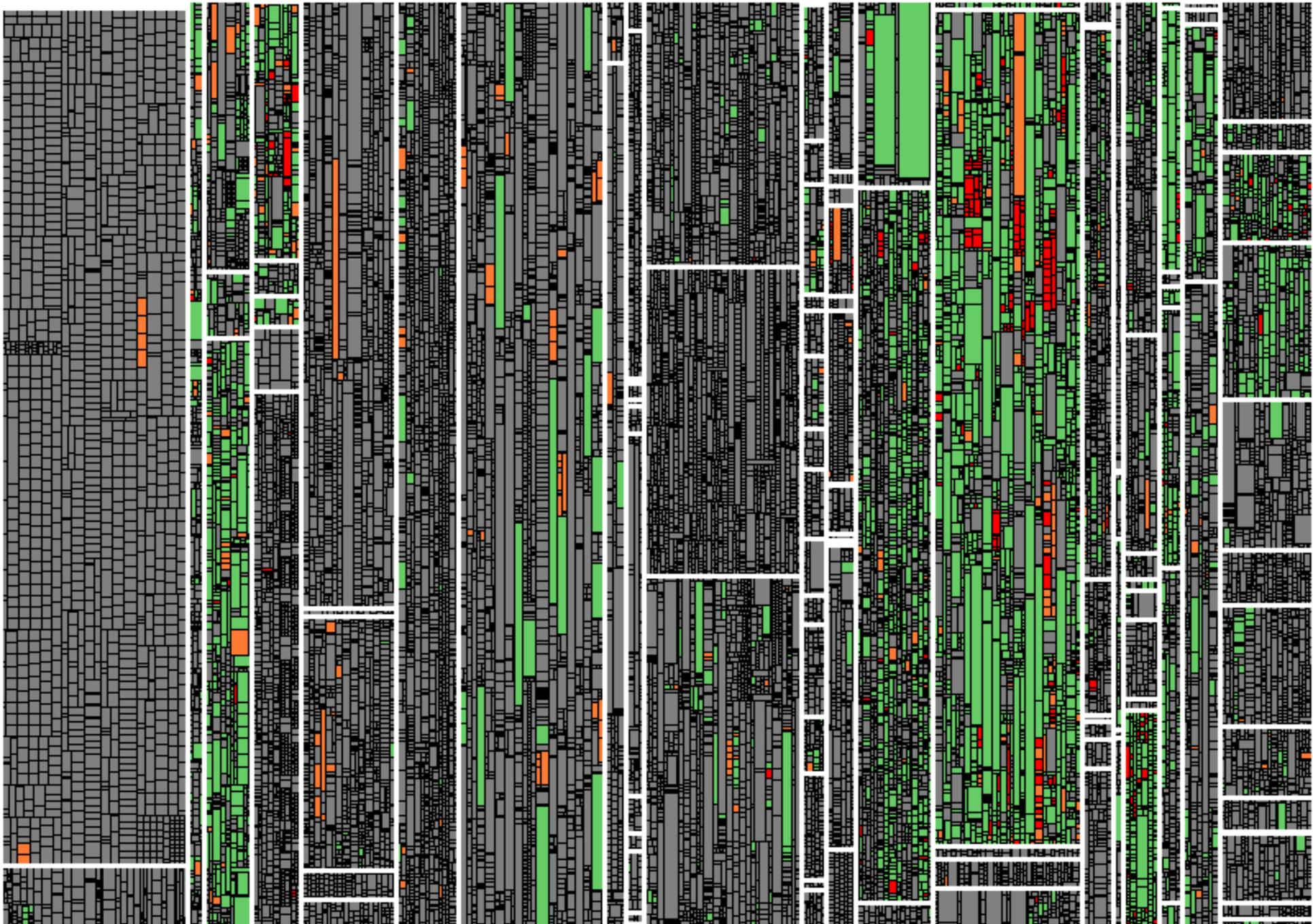
# Release-Test: Beispiel (2)

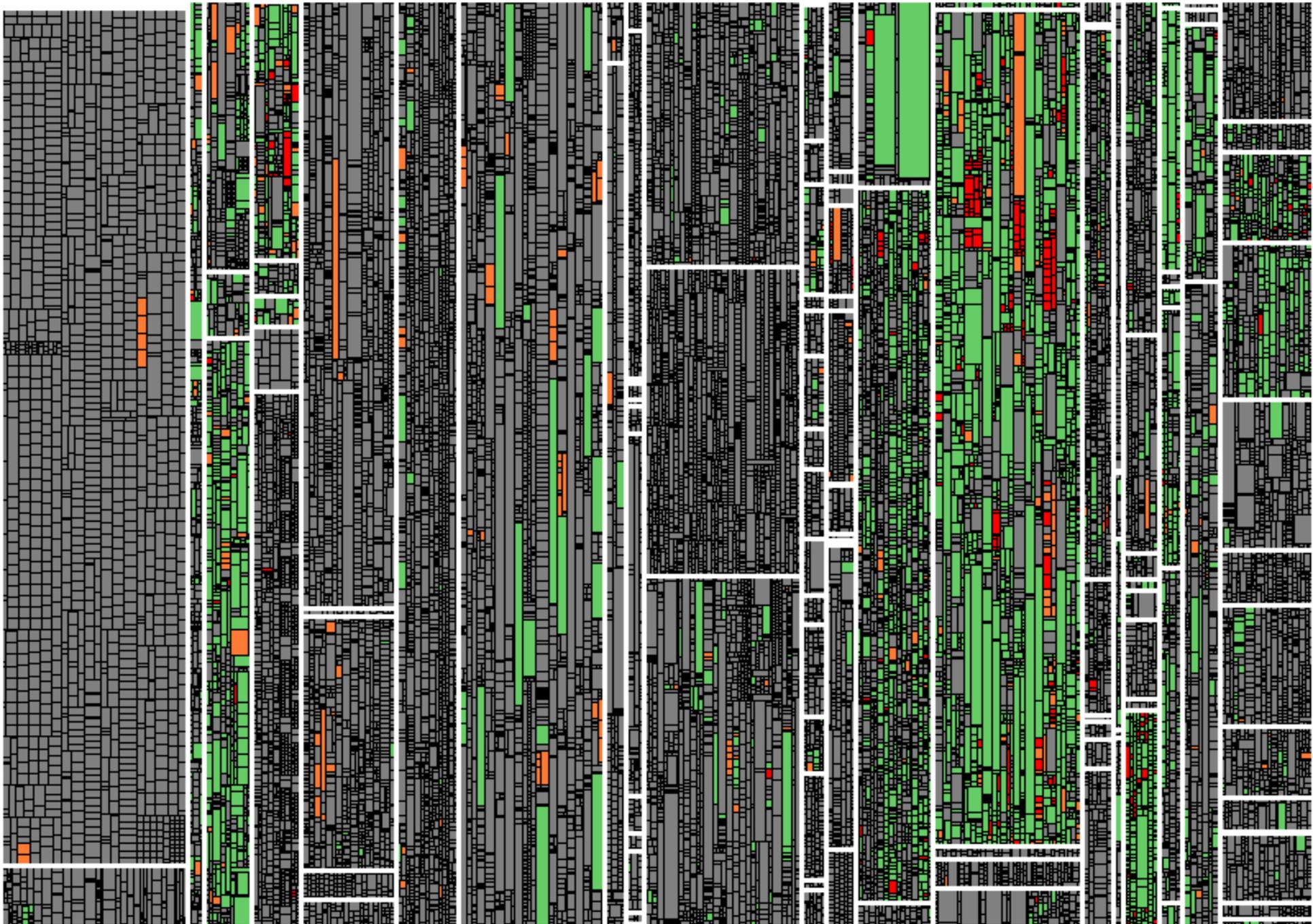


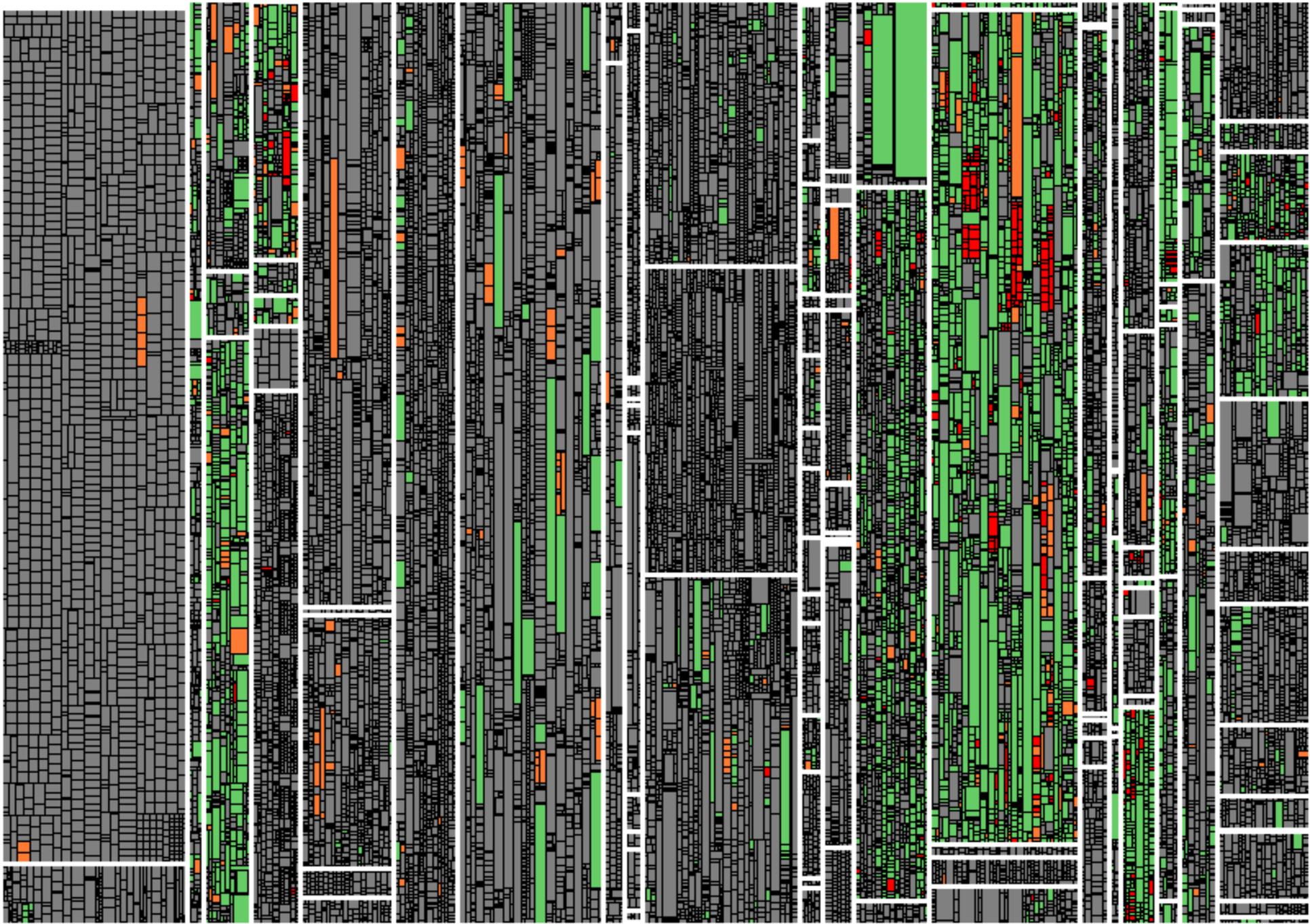


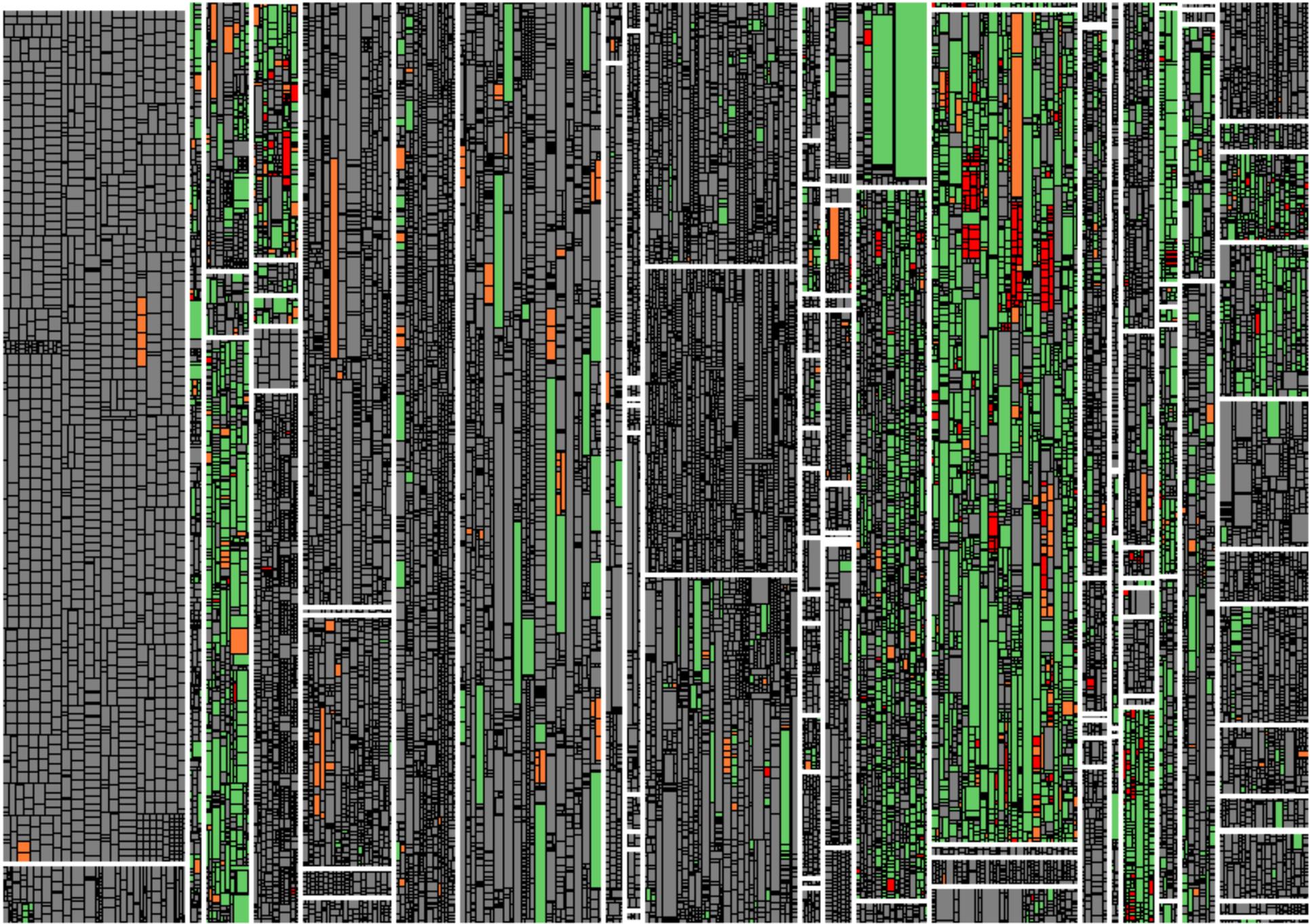


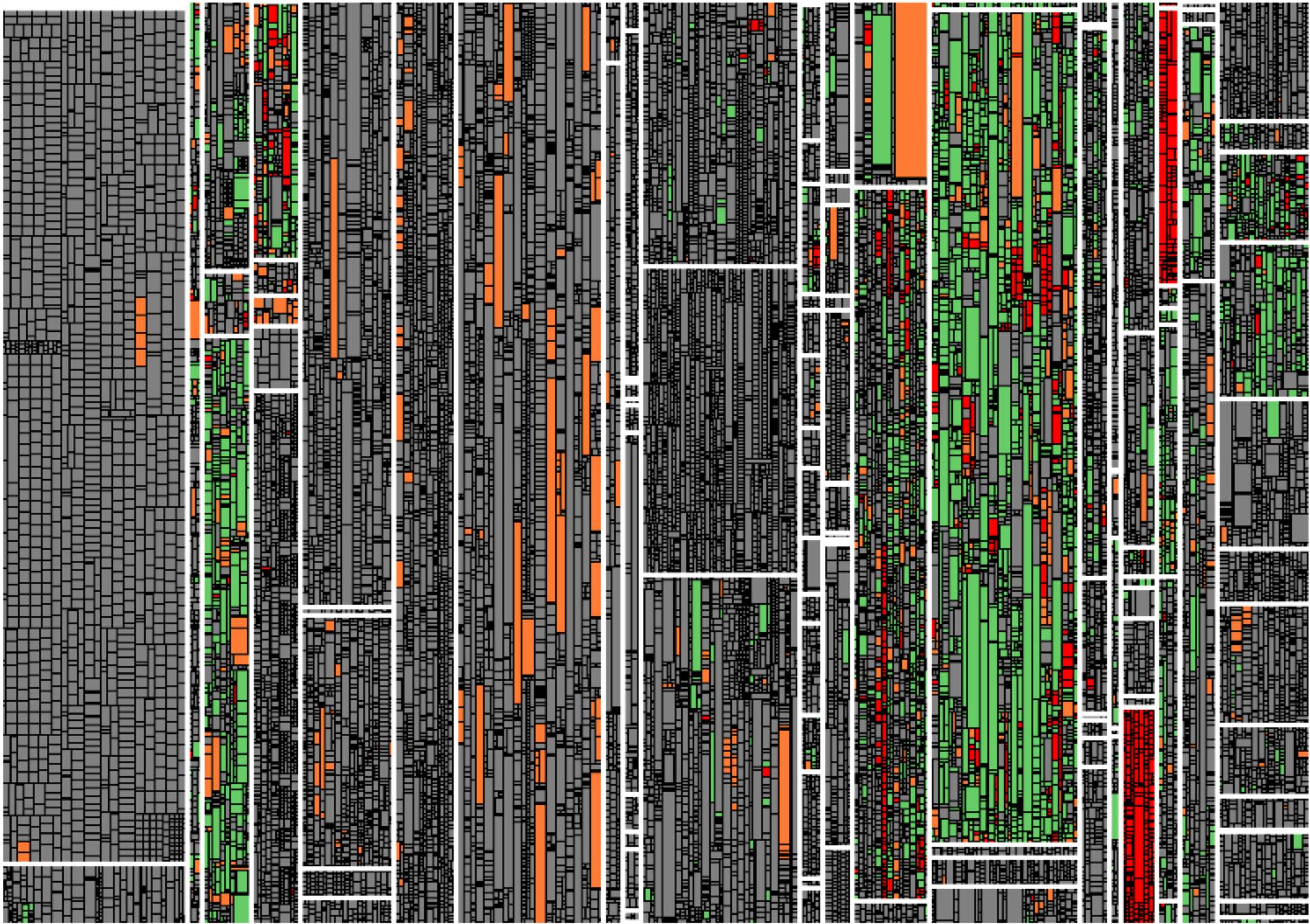




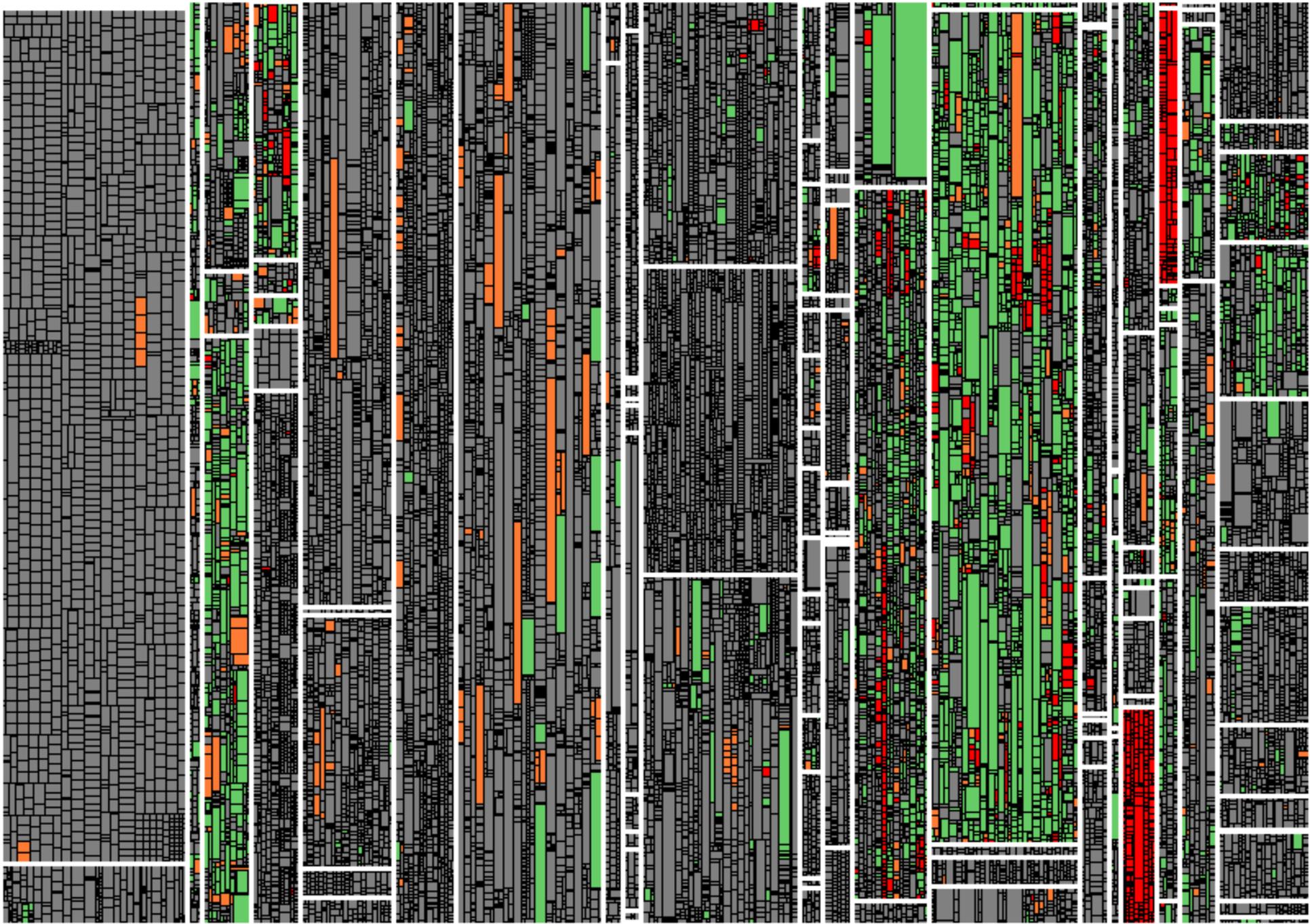


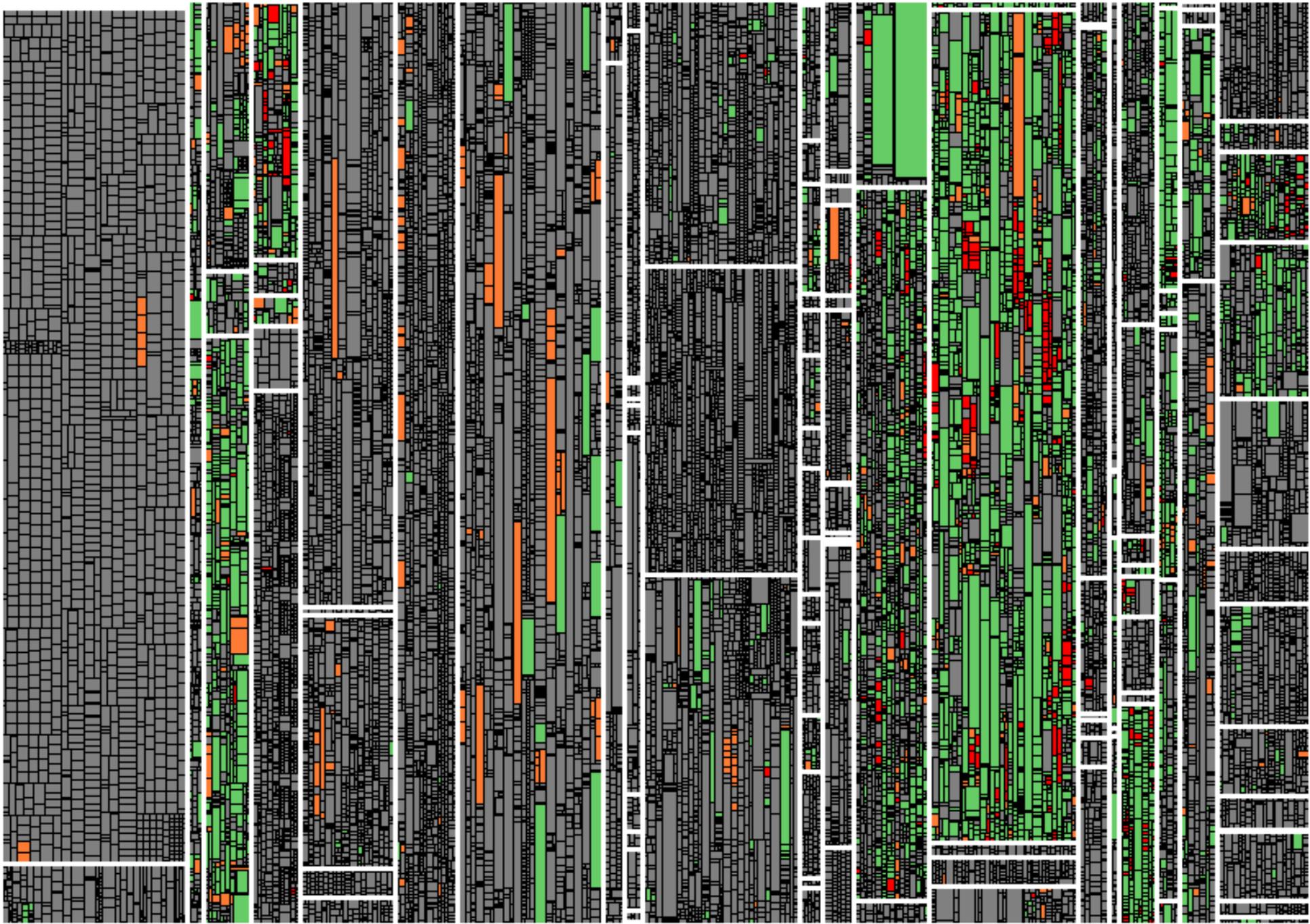


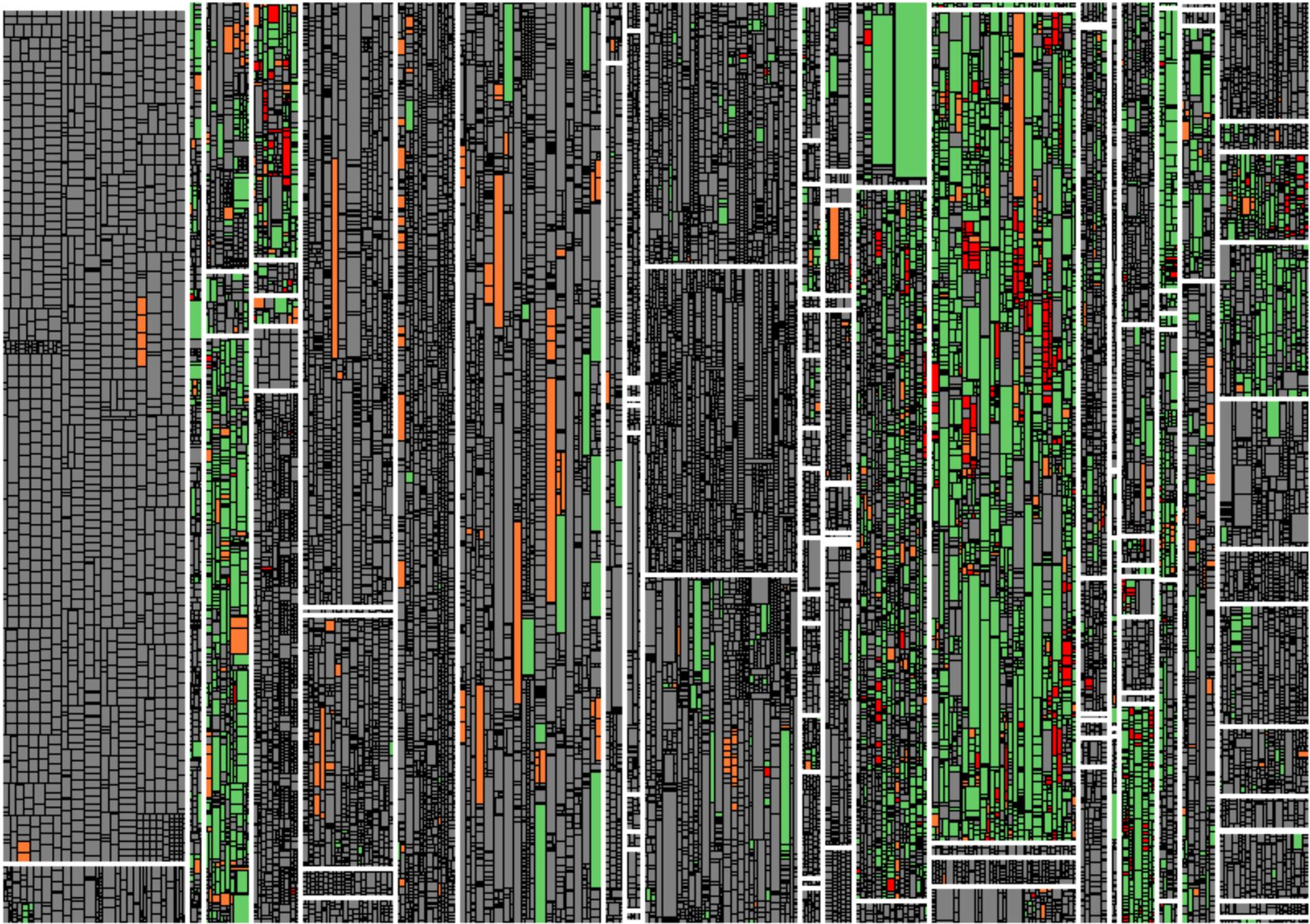


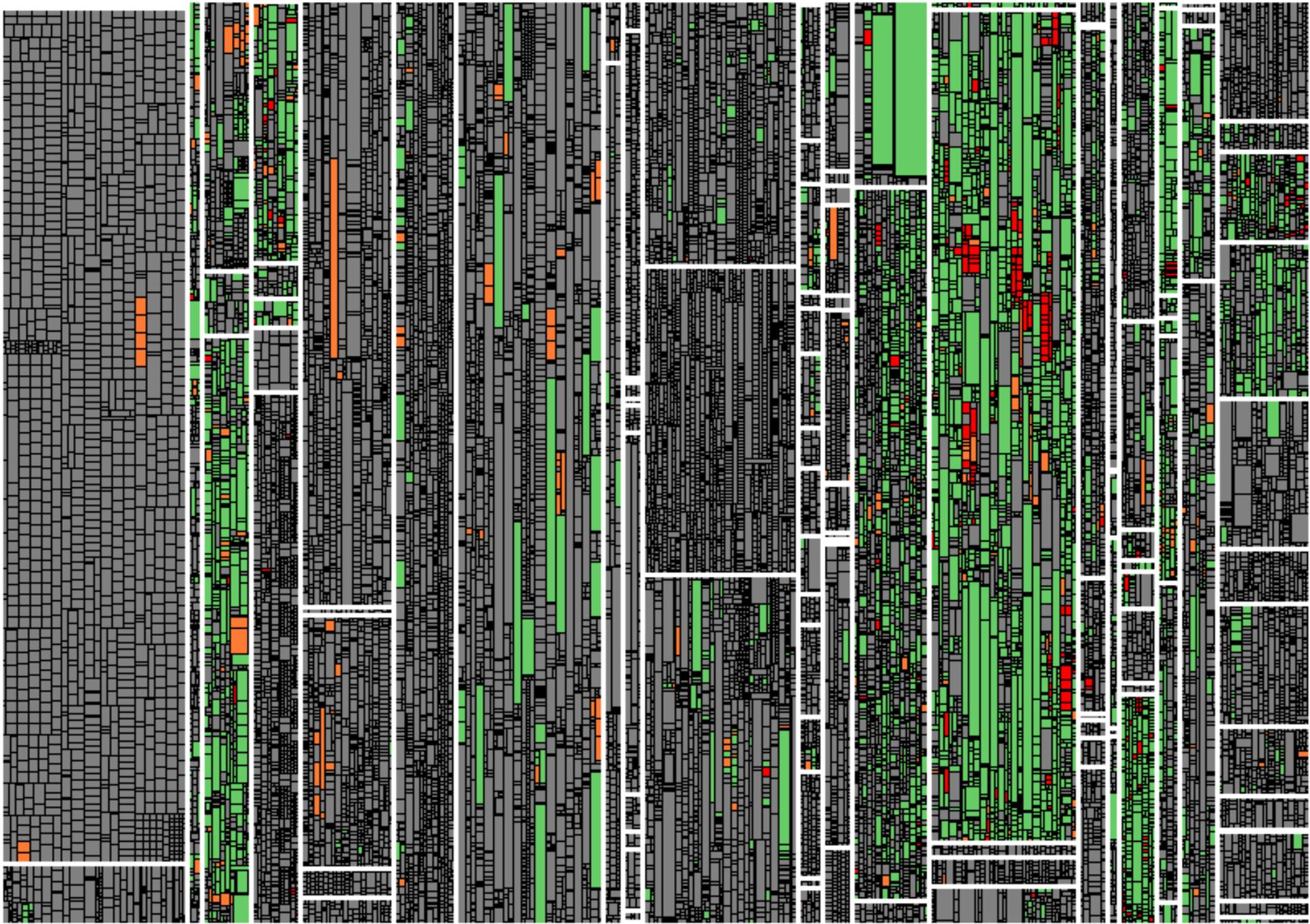


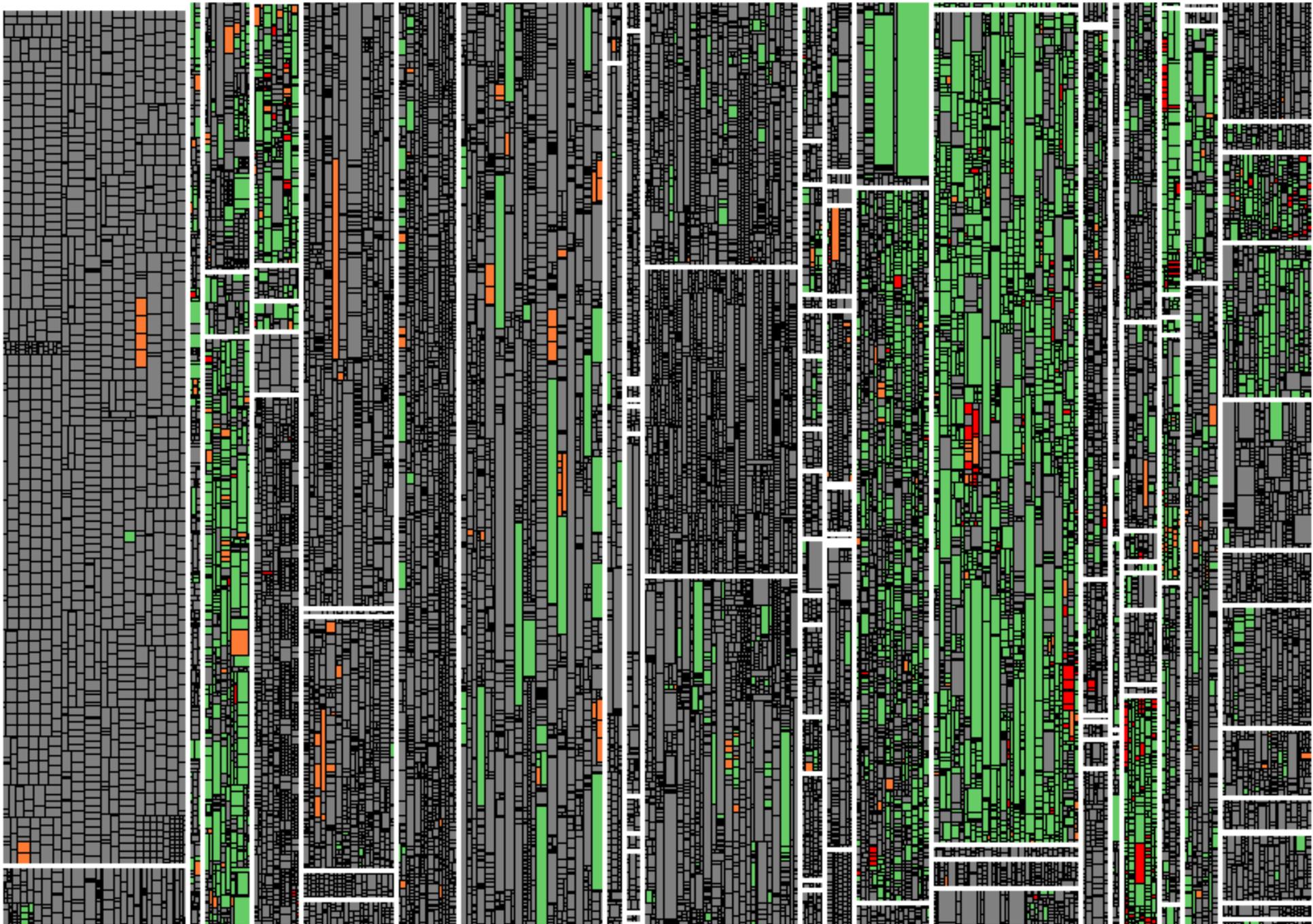












# TGA im Iterations-Test

- Arbeit in Iterationen in *Work Items* strukturiert. (a.k.a. Change Requests, Issues, Tickets, ...).
  - Test-Gap-Information liegt aktuell auf Ebene von Code vor, nicht auf Ebene von Work-Items.
  - Zusammenhang Test-Gaps <-> Work Items unklar.
- 
- Einige Teams wollen während dem Iterationstest nur *einzelne* Work Items testen.
  - Unklar, welche Test-Gaps das betrifft.
- ⇒ Roadmap: Filtern von Test-Gaps nach Work Items und Iterationen.

The image shows a screenshot of the Bugzilla Bug List interface. The header includes navigation links: Home | New | Browse | Search | [input field]. Below the header, there is a search filter section with the text "Hide Search Description" and "Status: UNCONFIRMED, NEW, ASSIGNED, REO". A message states "83 bugs found." Below this, a table lists bugs with columns for ID and Summary. The visible rows are:

ID ▲	Summary
<a href="#">3382</a>	Remove old filtering framework
<a href="#">4608</a>	Improve progress monitoring f
<a href="#">3613</a>	Make MetricOverview more flexi
<a href="#">3811</a>	Incorrect offset reported by tw

Below the table, there is a vertical double-headed arrow with "??" next to it, indicating a relationship or mapping. At the bottom of the image, there are two horizontal bar charts. The top chart is a dense, multi-colored bar chart with many small segments in red, green, and grey. The bottom chart is a similar but less dense bar chart with segments in green, orange, and grey.

# Lesson Learned

Für Hotfix-Tests und Release-Tests funktioniert die Betrachtung aller Änderungen seit dem letzten Release.

Für Iterationstests ist es notwendig, auf einzelne Änderungen filtern zu können.

# Phasen der Einführung

1. Mit den Projekten beginnen, die viel Interesse haben
2. Show-Cases für andere Projekte ableiten
3. Für alle Projekte einführen
  - Erhebung Messdaten in Testumgebungen integrieren
  - Einrichtung TGA-Dashboards sehr einfach machen

Kontinuierlich Aufwände und Fehlerdaten messen

Site Content

- Documents
- public
- TGA Backlog
- TGA dashboards
- TSA Backlog
- TSA dashboards
- TGA-TSA-Project

This is the public MOSS for Test Gap Analysis and Test Smell Analysis. It hosts documentation about method and tools and the links to all dashboards.

## Test Gap Analysis

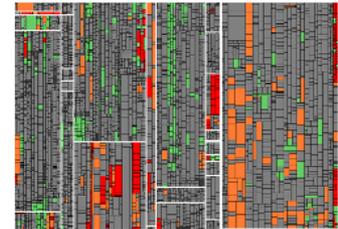
[Overview Presentation](#) gives a general introduction to the Test Gap method.

Links to all available [dashboards](#) are listed on a central page.

The [Usage Concept](#) describes how to work with Test Gap Analysis.

For any project / system a ramp-up and installation needs to be done. There are two check lists for necessary information - [.NET / SAP](#).

There is continuous improvement of TGA - see [product backlog](#). If you have ideas for enhancements please [propose there](#).



# IT Test Community



▼ TGA / TSA user guide & best practices

Test Gap Analysis

▼ Test Smell Analysis

Core Concepts

TSA for Managers

TSA for Test Analysts

## Core Concepts

This section describes core concepts related to Test Smell analysis. In general, Test Smell analysis aims at helping projects to write better test cases by applying several heuristics. Some of these use natural language processing. By raising awareness about possible problems in test cases, teams are enabled to take conscious decisions about where improvements make sense.

## What is Measured?

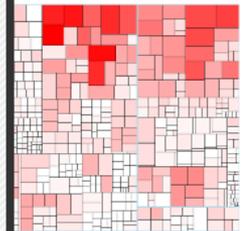
The analysis tries to find possible problems in test case descriptions using smell detection, i.e. searching for certain patterns that might be problematic during the execution or maintenance of test cases. The categories we currently look at are:

- *Vague sentences* that leave room for interpretation and may therefore not lead to repeatable test execution.
- *Conditional sentences* that introduce complexity into a single test step, where an additional test case may be more appropriate.
- *Long steps* that may be split up into separate steps, so that additional check points are available in-between.
- *Long sentences* that may be hard to understand by the tester.
- *Clones* that introduce additional maintenance effort should the cloned part need to be changed in the future.

The parameters for these analyses are shown in every dashboard. See the following tables for the current defaults.

## Configurable Parameters

Parameter Name	Description	Value

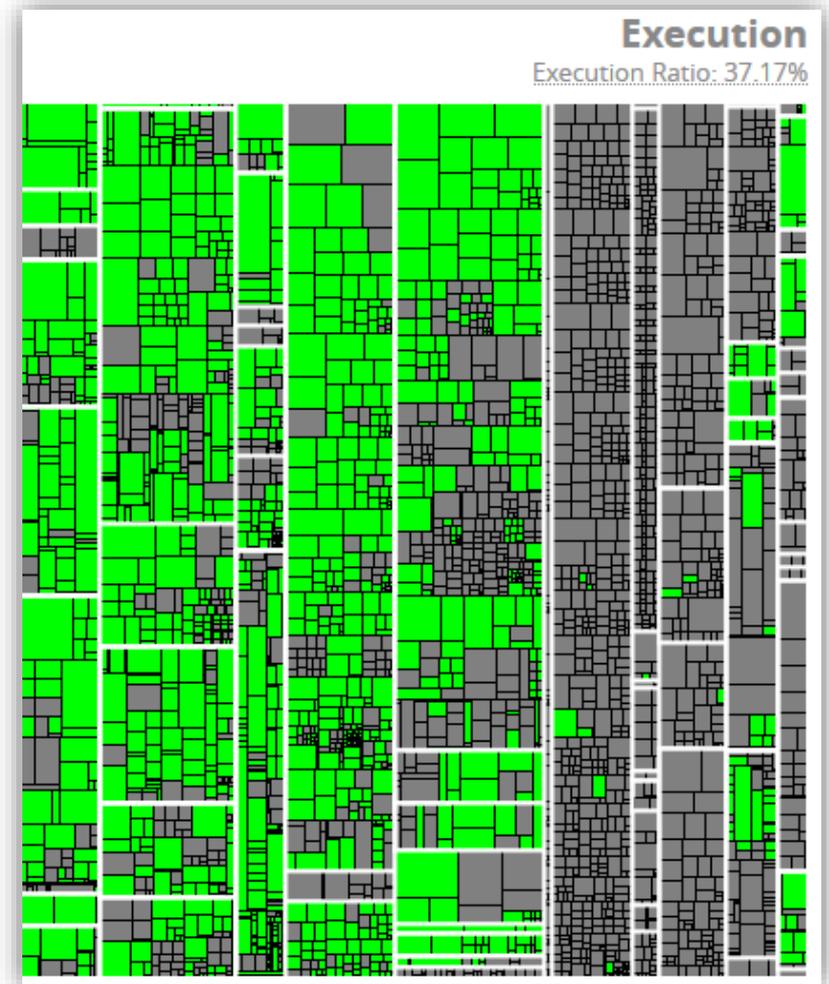


# Lesson Learned

Damit Test-Gap-Analyse langfristig eingesetzt wird, muss Change-Management betrieben werden.

# Live-Demo: TGA mit Teamscale

- Heute um 14:35 Uhr
- Solution Provider Forum (Treppe in der Lobby hoch)
- Live-Demo an Open-Source System
- Q&A
- Kostenlose Evaluierungslizenz für Teilnehmer



# Fazit

Test-Gap-Analyse schafft Transparenz. Sie wird von vielen Teams genutzt, um ungewollt ungetestete Änderungen zu vermeiden.

Jede Test-Art hat unterschiedliche Anforderungen. Test-Gap-Analyse ist am einfachsten bei Hotfix- & Release-Tests einsetzbar.

Damit Test-Gap-Analyse langfristig eingesetzt wird, muss Change-Management betrieben werden.

# Kontakt

Wir sind noch länger hier und freuen uns auf Diskussionen!

Uwe Proft · UProft@munichre.com

Dr. Elmar Juergens · juergens@cqse.eu · +49 179 6753863

@ElmarJuergens

[www.cqse.eu/en/blog](http://www.cqse.eu/en/blog)