

Gestern lief's doch noch

Muss ich heute wirklich schon wieder alles testen?
Forschungsergebnisse & eigene Erfahrungen mit Test-Impact-Analyse

Dr. Elmar Juergens (CQSE)
Alexander Kaserbacher (TNG)

TNG CQSE

Über Mich

Forschung

- Statische Analyse, Test-Gap-Analyse, ...
- PC Mitglied von MSR, ICPC, ...

Beratung

- Gründer und Mitglied der Geschäftsleitung
- Qualitäts-Bewertung & Qualitäts-Controlling





com.teamscale.test.ui.architecture.ArchitecturePerspectiveTest - testViewingArchitectureElements

Runs: 1/227

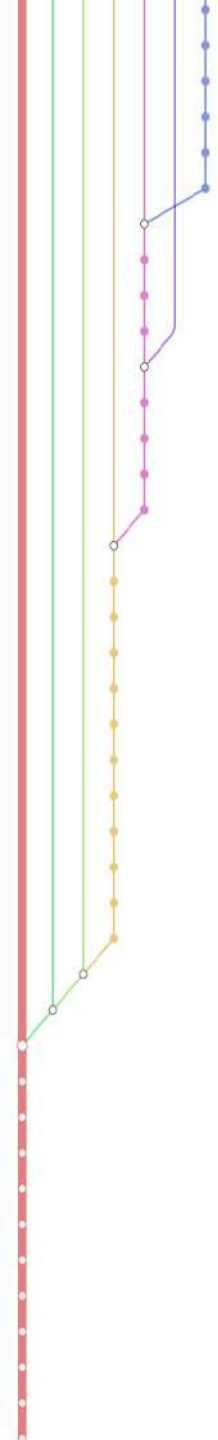
 Errors: 0

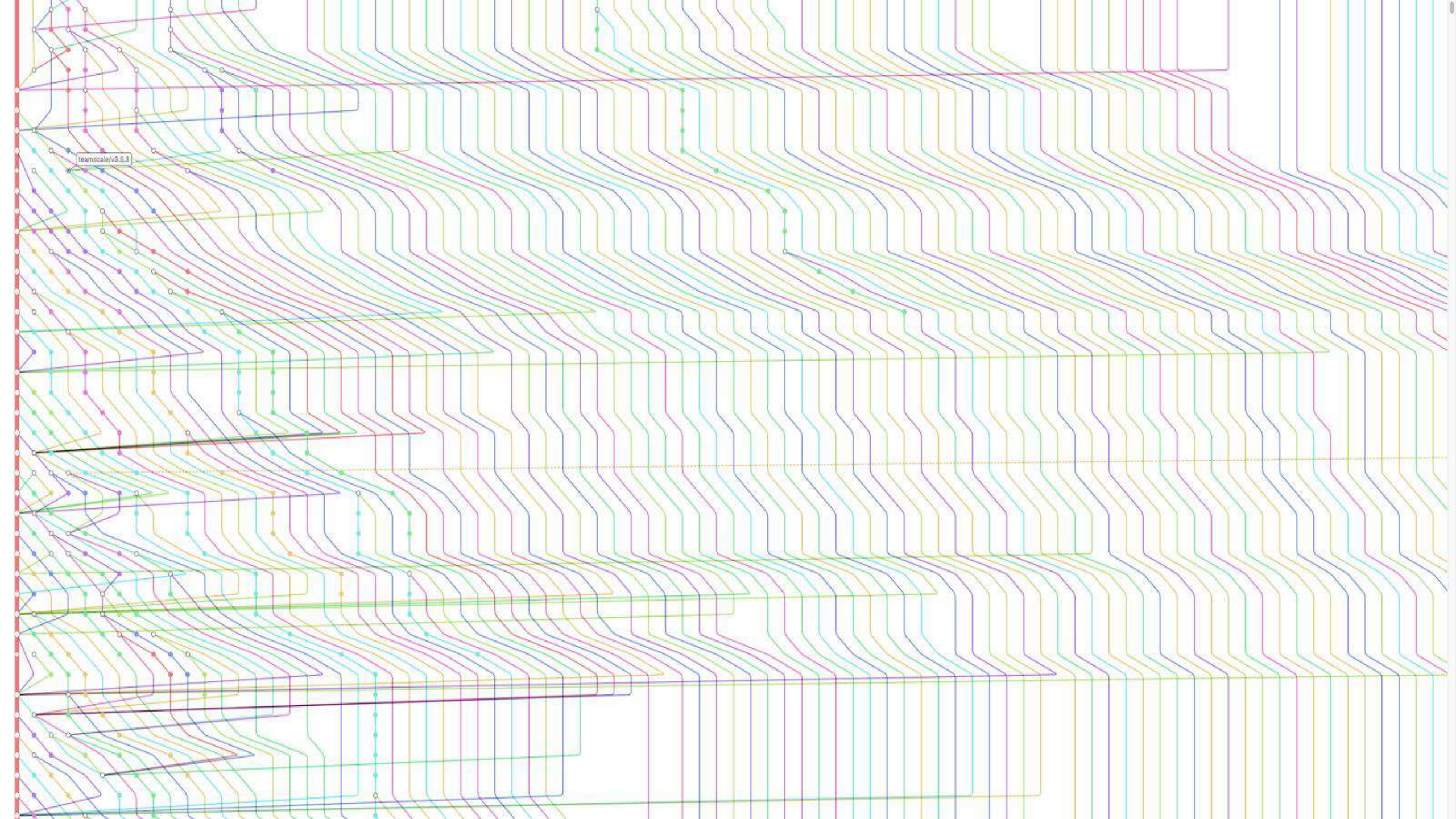
 Failures: 0

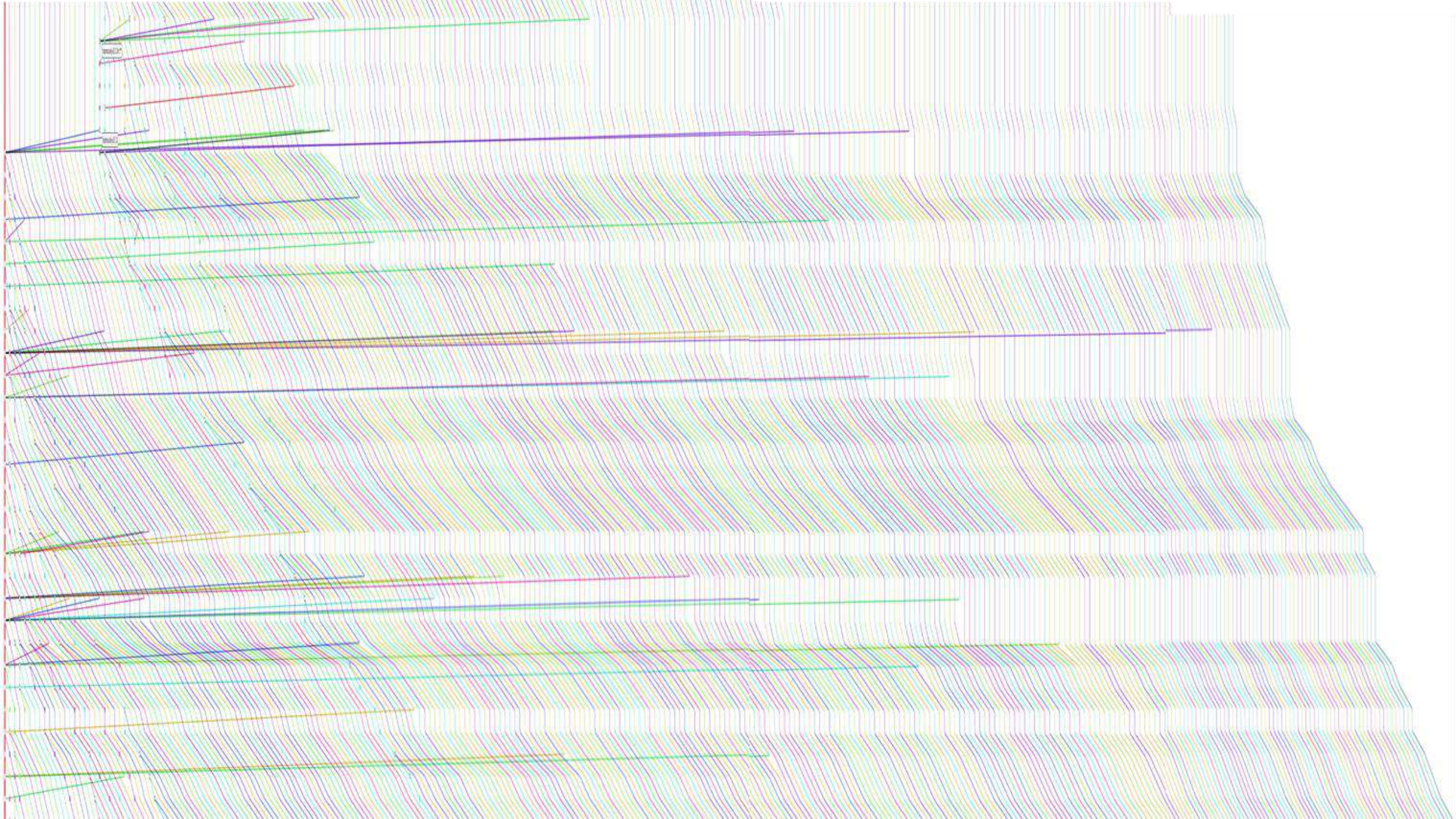
- ▼  com.teamscale.test.ui.architecture.ArchitecturePerspectiveTest [Runner: JUnit 5]
 -  testViewingArchitectureElements
 -  createArchitectureTest
 -  testTimetravelMode
 -  testUndoRedo
 -  testArchitecturePolicyUpdate
 -  testOpenCodeOfOrphans

 Failure Trace



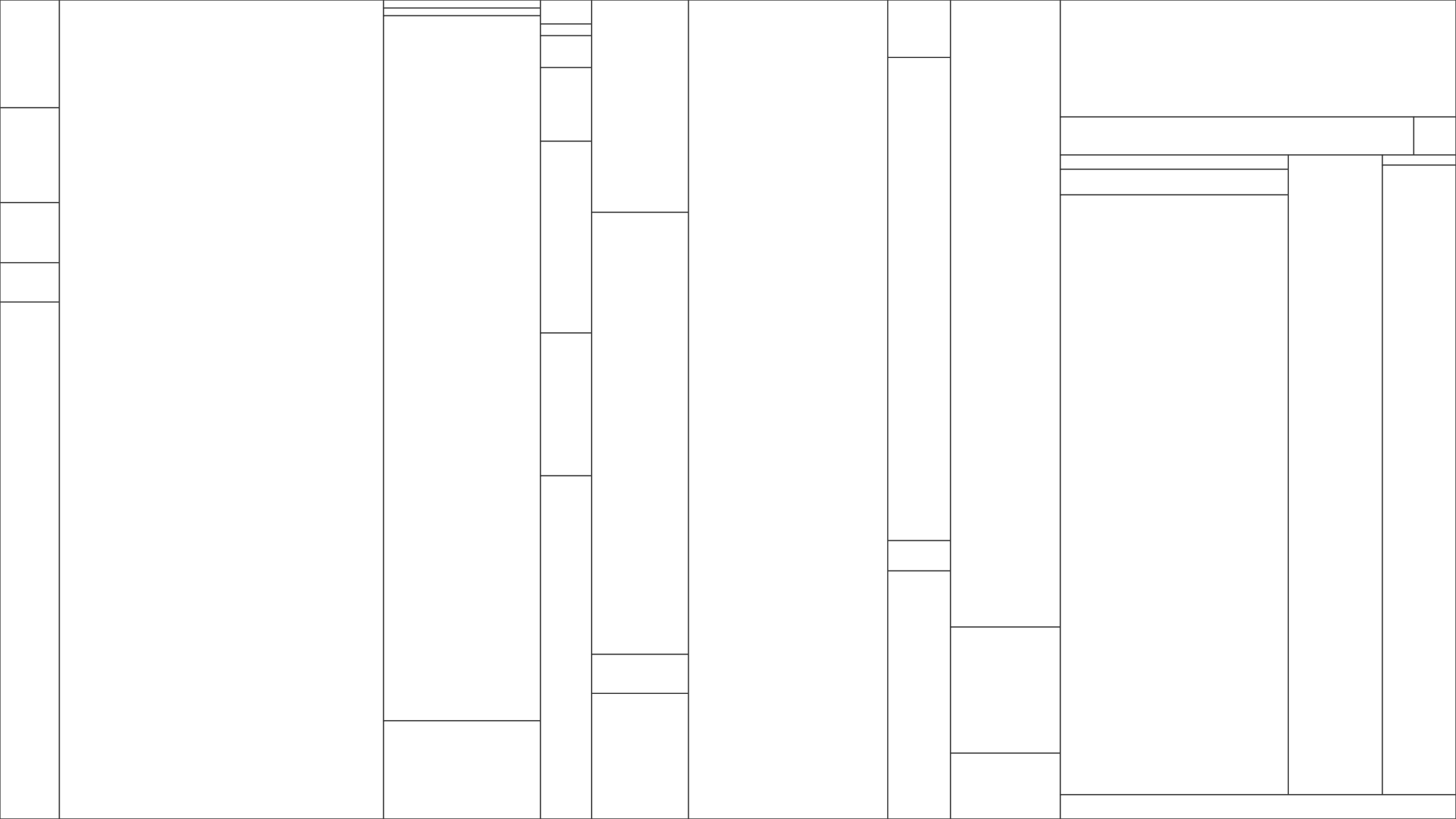


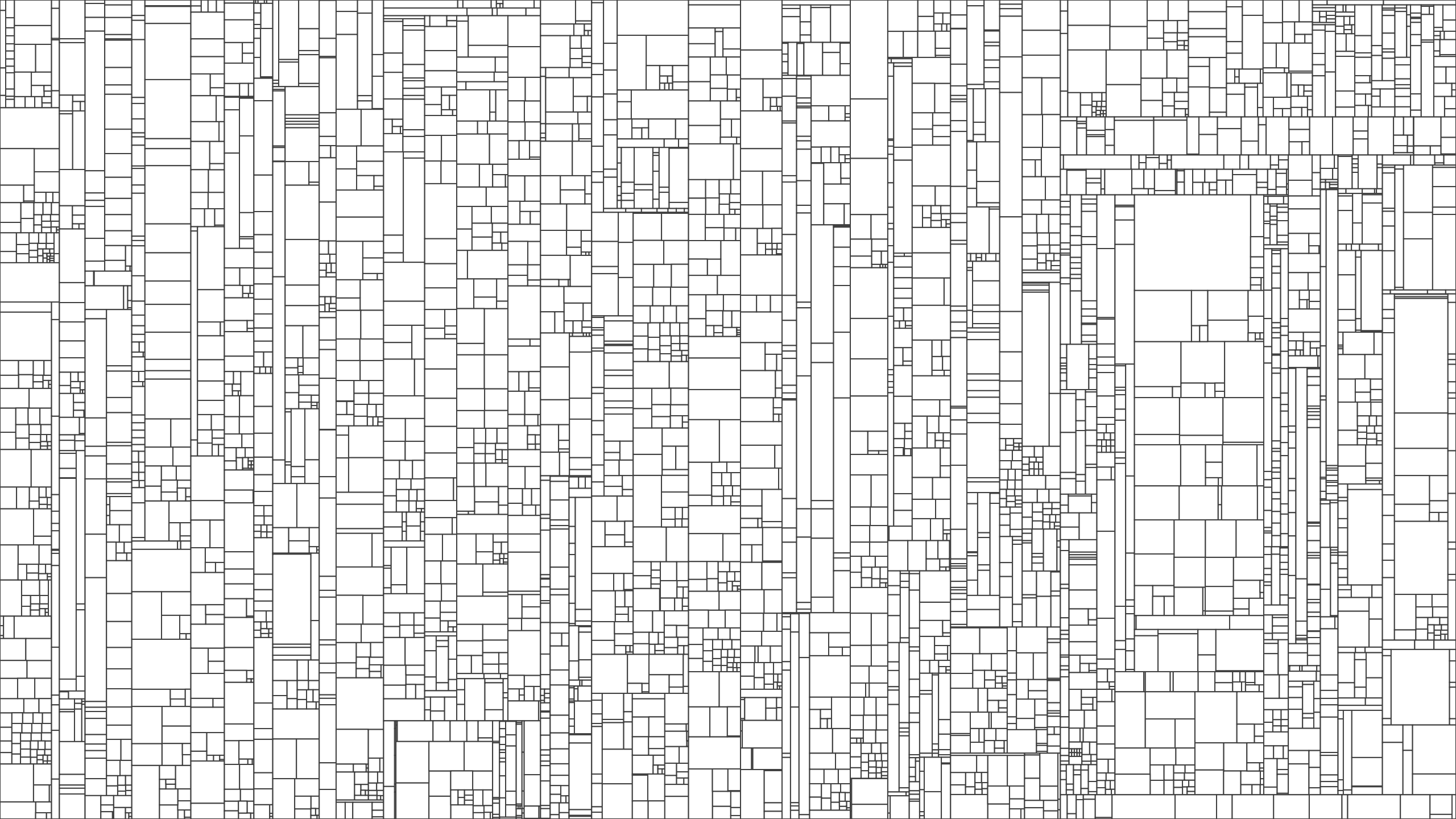


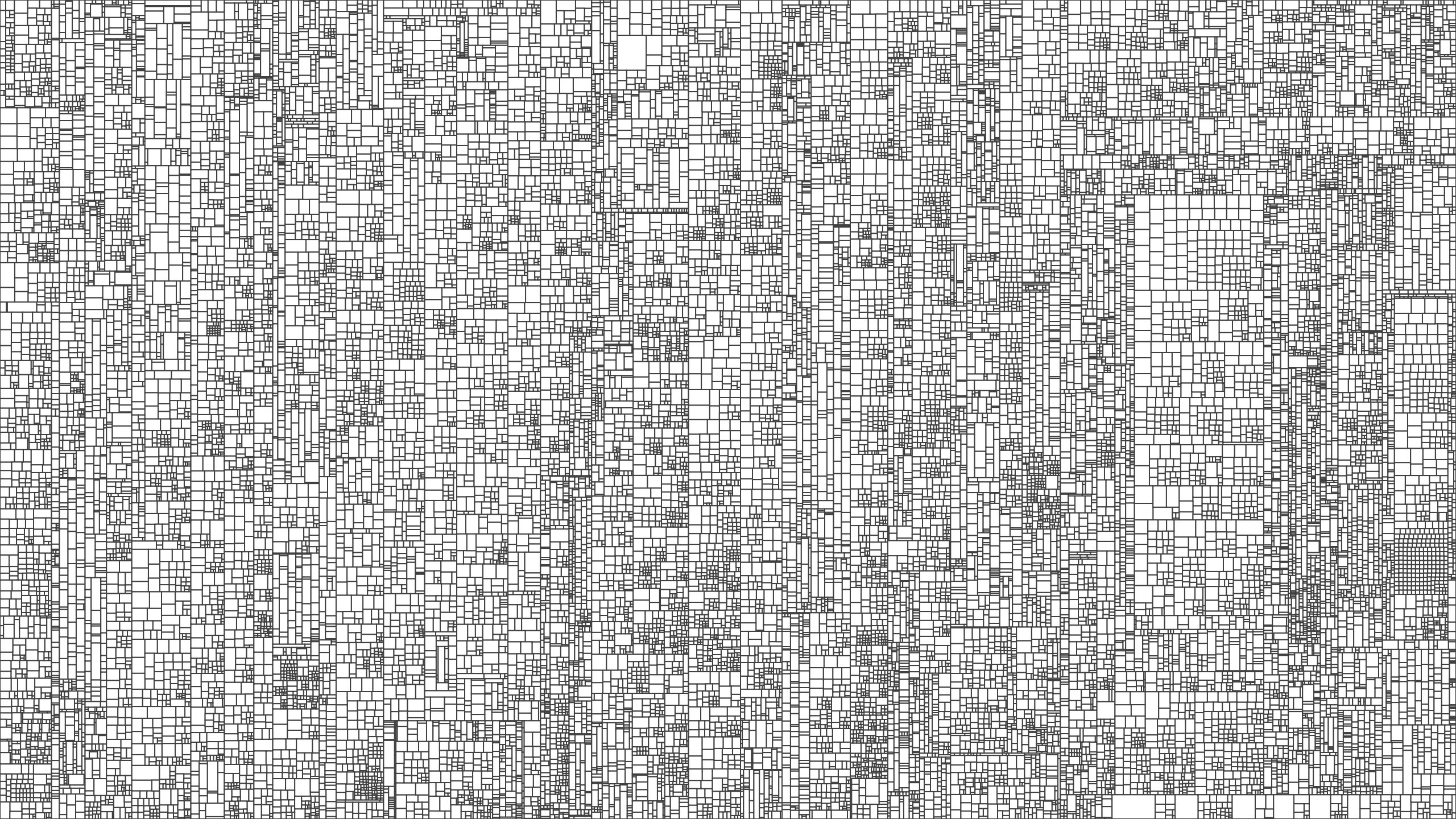














Data Flow Analysis can not handle java lambdas (logs many errors currently)

- Edit
- Comment
- Assign
- More ▾
- Back to New



Details

Type:	■ Bug	Status:	DONE (View Workflow)
Priority:	↑ High	Resolution:	Green
Component/s:	Backend	Fix Version/s:	Teamscale 4.3
Labels:	dataflow java		
Affected Version:	master		
Merge Request:	https://git.cqse.eu/cqse/teamscale/merge_requests/2734		
PDash Task:	#4886		

People

Assignee:	Andreas Sewe Assign to me
Reporter:	Rainer Niedermayr
QA-Contact:	Alexander von Rhein
Votes:	2 Vote for this issue
Watchers:	3 Start watching this issue

Description

Analysis profile: Java
 Repository: JabRef, start revision 9efd23b71871747fe5e18e915e637891d7e55b6d

Dates

Created:	24/Nov/16 8:35 AM
Updated:	4 days ago
Resolved:	08/May/18 12:42 PM

Time Tracking

Estimated:	<div style="width: 100%; height: 10px; background-color: #ccc;"></div> Not Specified
Remaining:	<div style="width: 100%; height: 10px; background-color: #ccc;"></div> 0m
Logged:	<div style="width: 100%; height: 10px; background-color: #4CAF50;"></div> 4d 1h 57m

Agile

[View on Board](#)

Gitlab CI

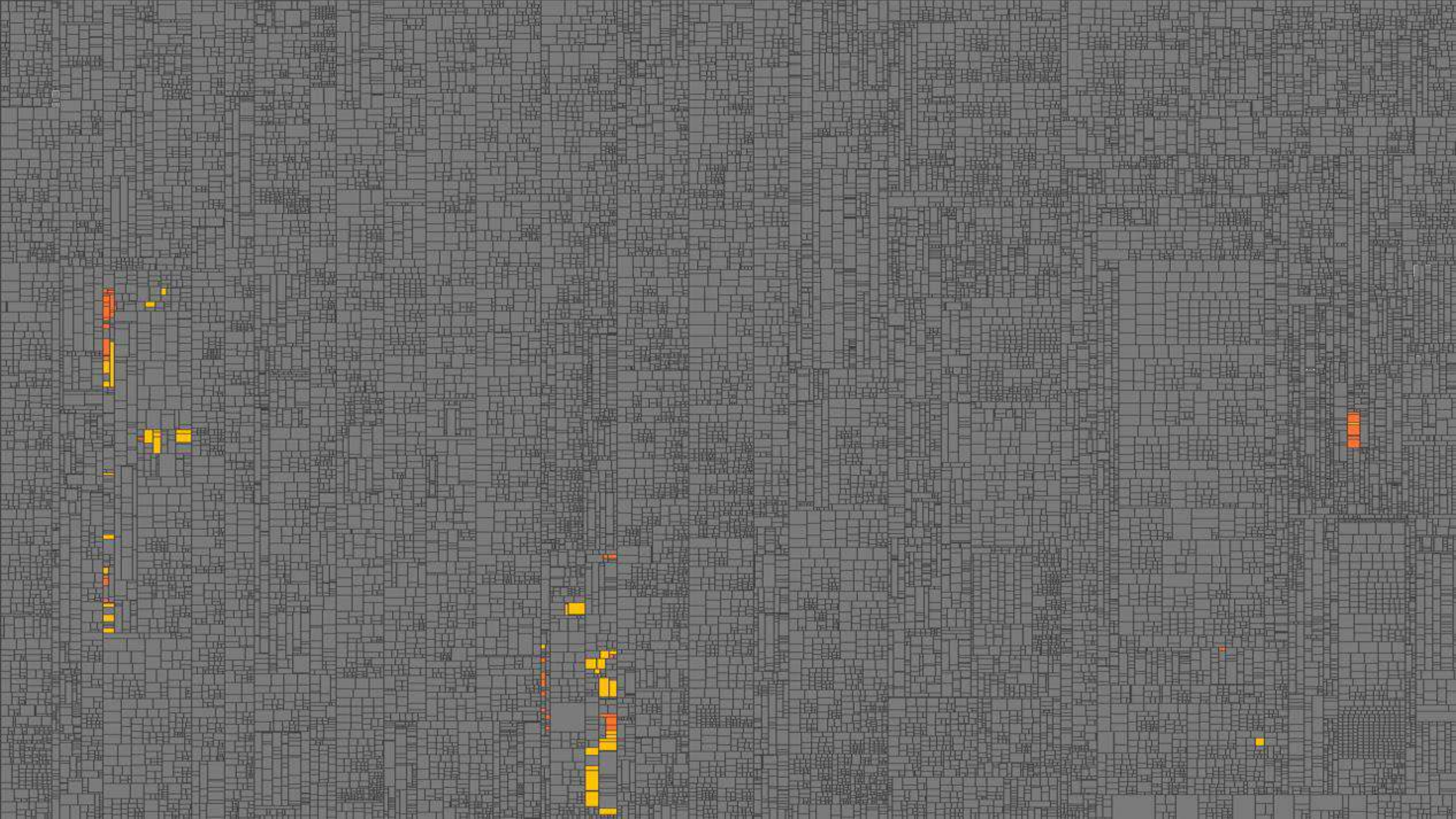


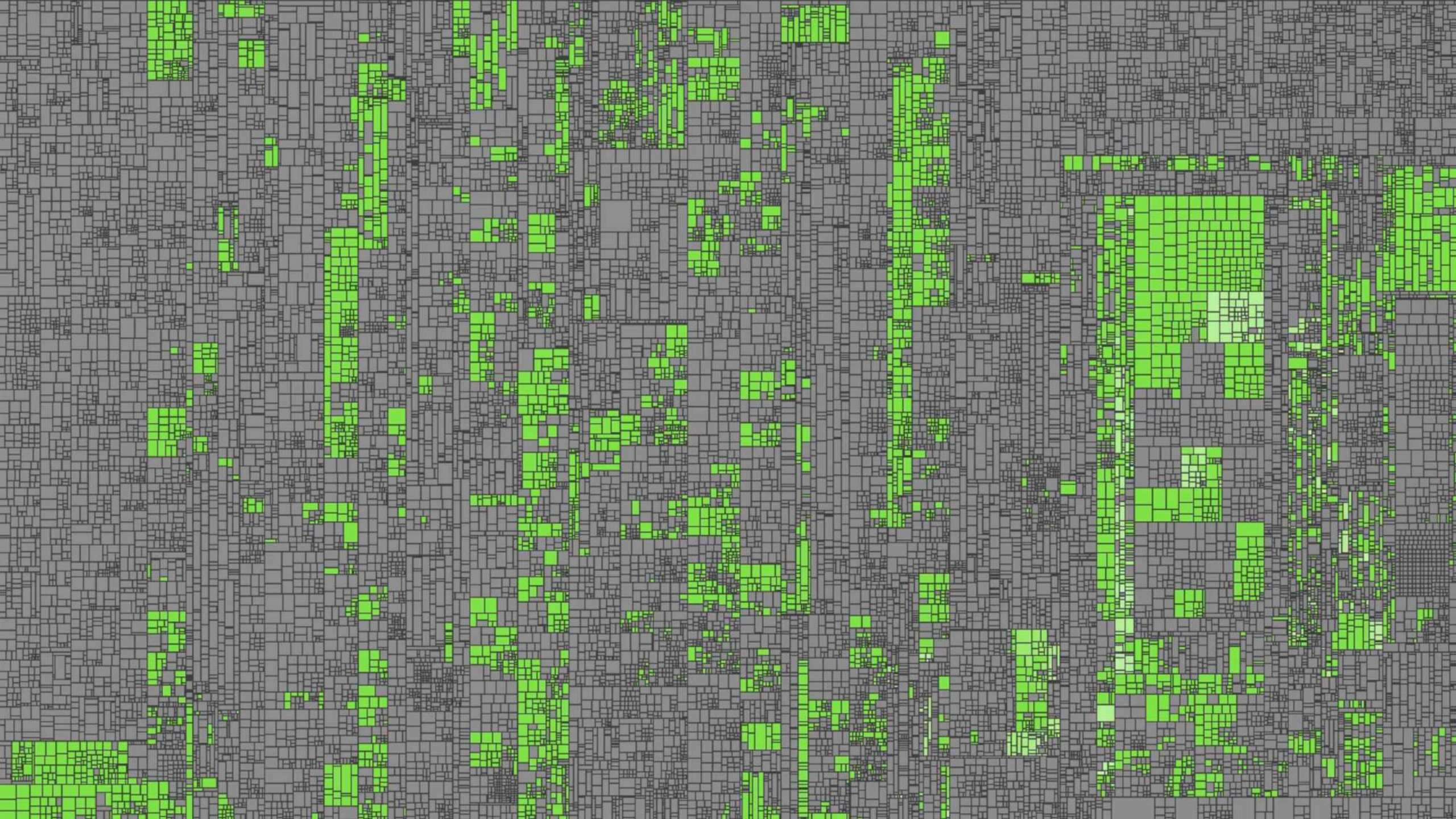
```

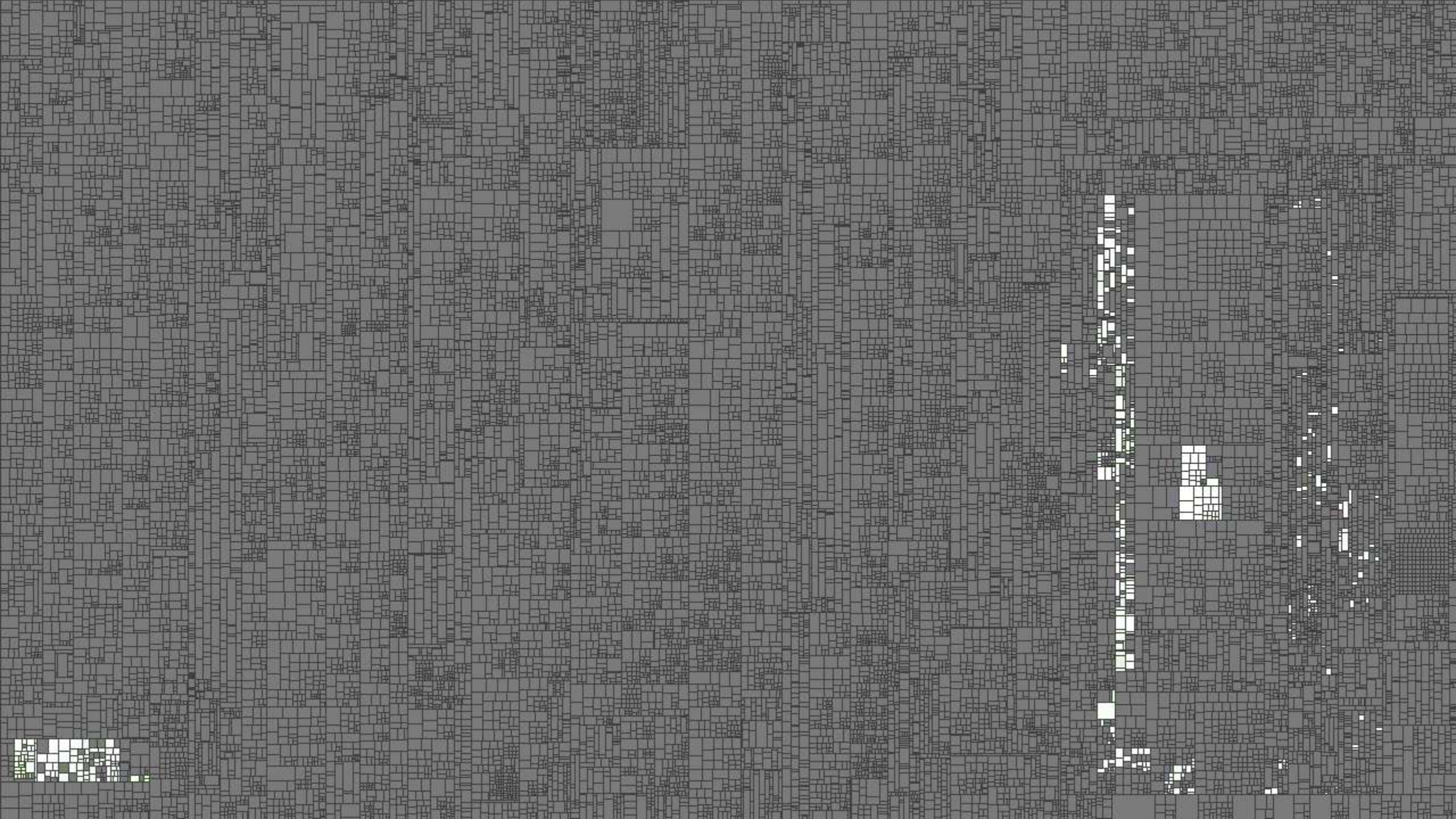
ERROR : An error occurred while trying to construct a CFG for function 'null' in element src/main/java/net/sf/jabref/exporter/layout/format/DOI
[STATEMENT: lambda expression: null (lines 33-33)
]
Tokens: doi . getURL ( )
Occurred in src/main/java/net/sf/jabref/exporter/layout/format/DOICheck.java:32-32 (com.teamscale.index.dataflow.DataFlowFindingsSynchronizer.c
org.conqat.engine.core.core.ConQATException: Could not find any rule that applies to the following entity list:
[STATEMENT: lambda expression: null (lines 33-33)
]
Tokens: doi . getURL ( )
Occurred in src/main/java/net/sf/jabref/exporter/layout/format/DOICheck.java:32-32
    at org.conqat.engine.sourcecode.dataflow.heuristics.ControlFlowCreator.findApplicableRule(ControlFlowCreator.java:164)
    at org.conqat.engine.sourcecode.dataflow.heuristics.ControlFlowCreator.transformOneStep(ControlFlowCreator.java:139)
    at org.conqat.engine.sourcecode.dataflow.heuristics.ControlFlowCreator.transform(ControlFlowCreator.java:92)
[...]

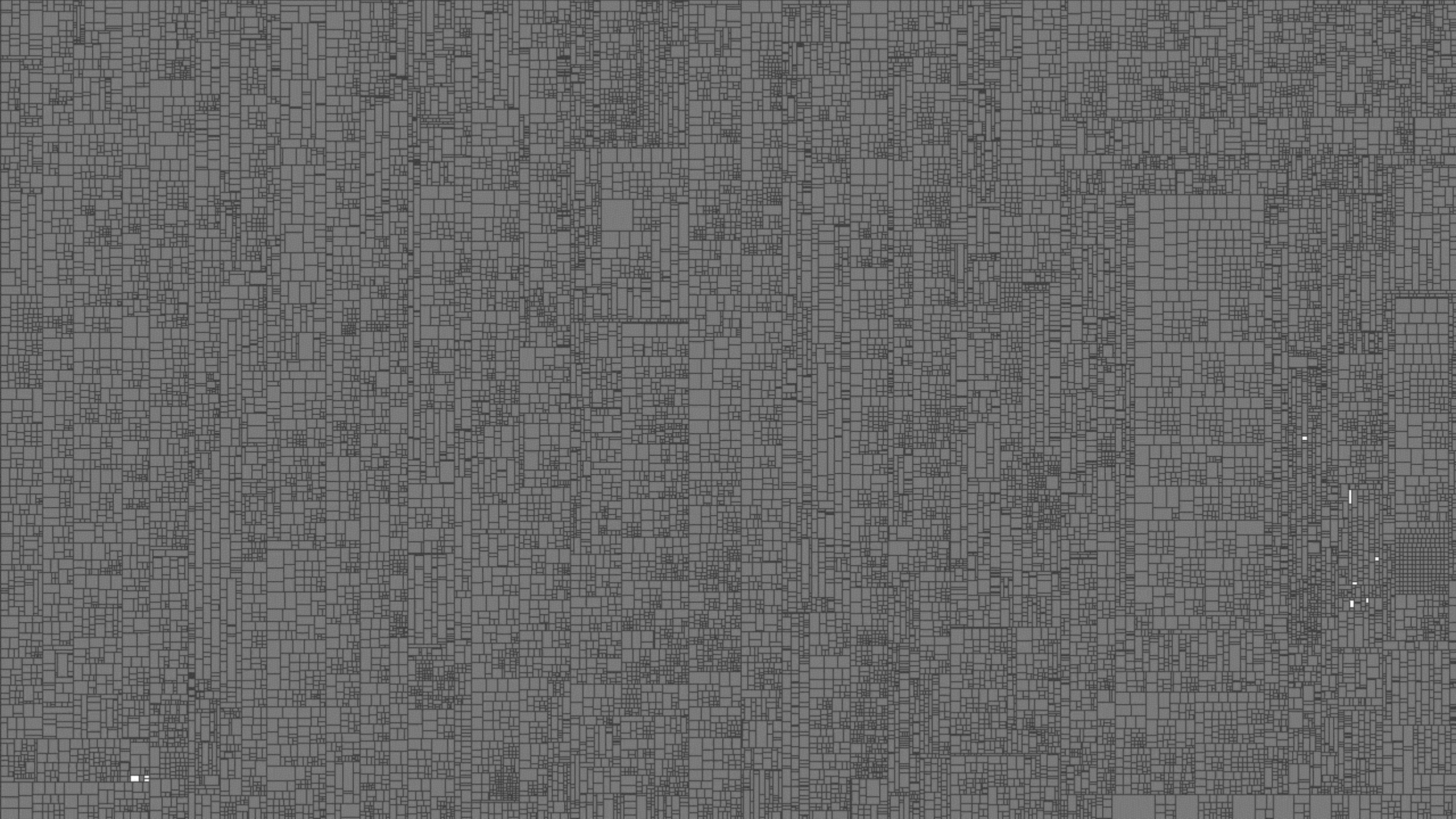
ERROR : An error occurred while trying to construct a CFG for function 'null' in element src/main/java/net/sf/jabref/exporter/layout/format/DOI
[STATEMENT: lambda expression: null (lines 31-31)
]
Tokens: doi . getDOI ( )
Occurred in src/main/java/net/sf/jabref/exporter/layout/format/DOIStrip.java:30-30 (com.teamscale.index.dataflow.DataFlowFindingsSynchronizer.c
org.conqat.engine.core.core.ConQATException: Could not find any rule that applies to the following entity list:

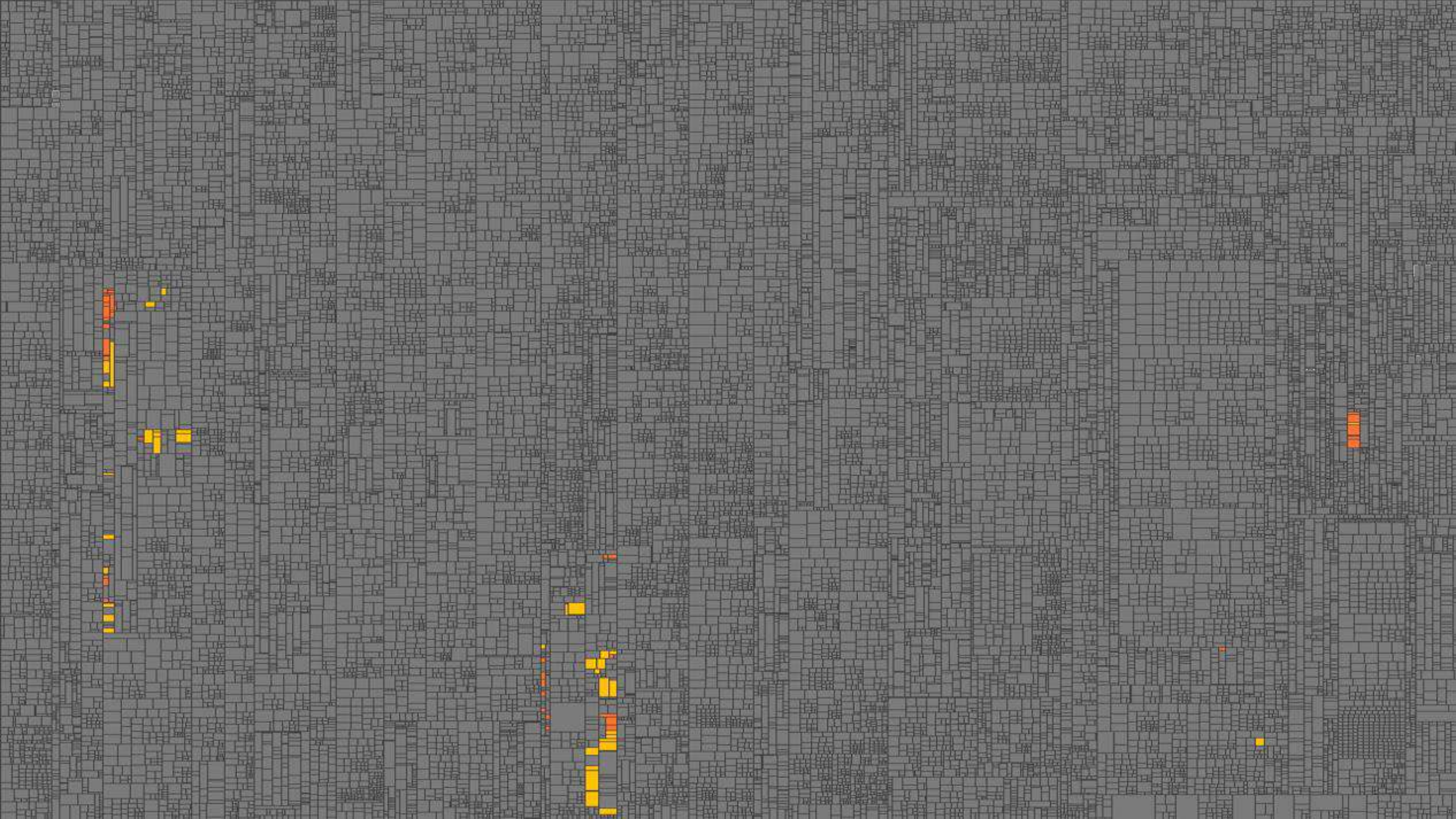
```

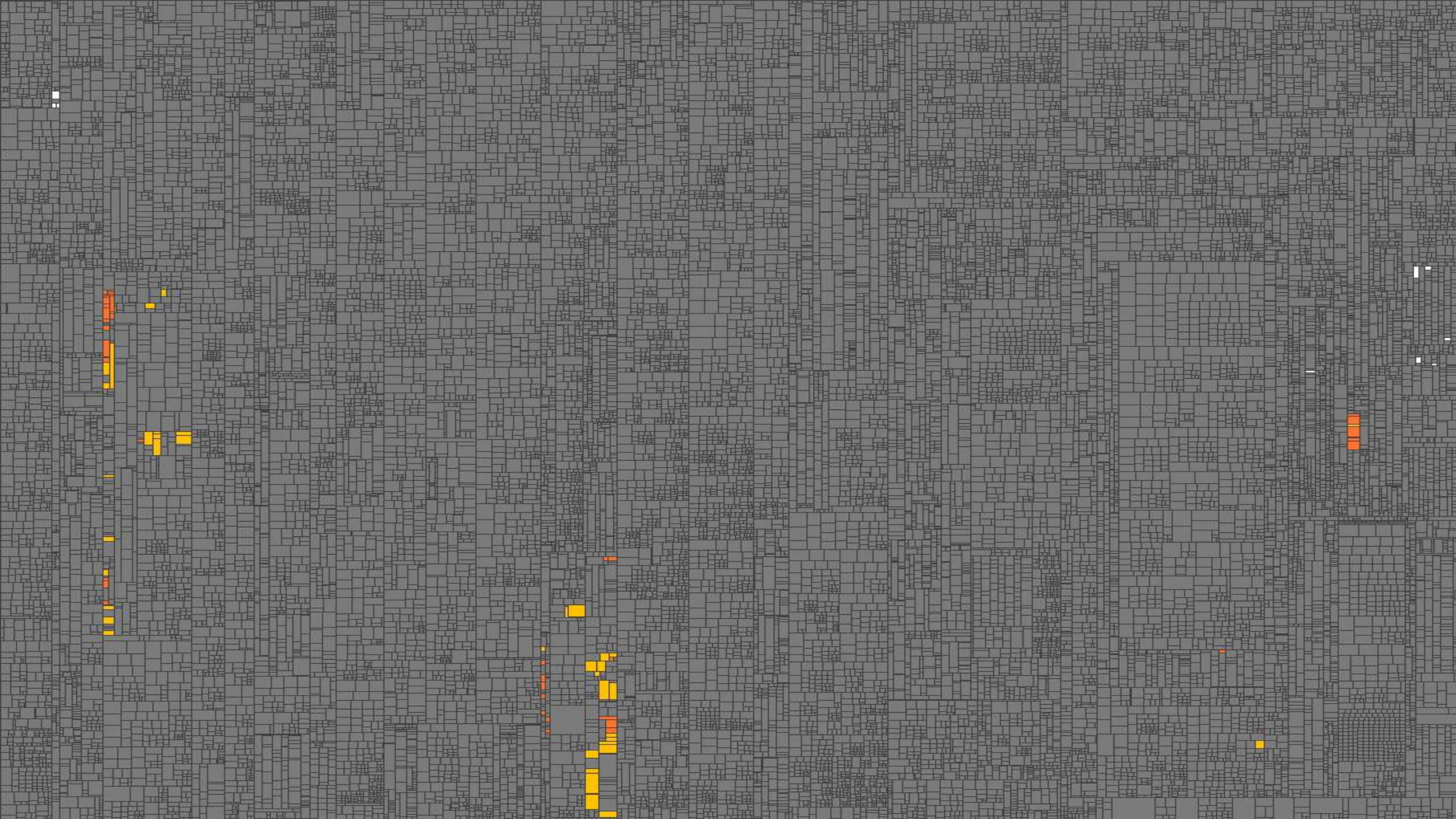


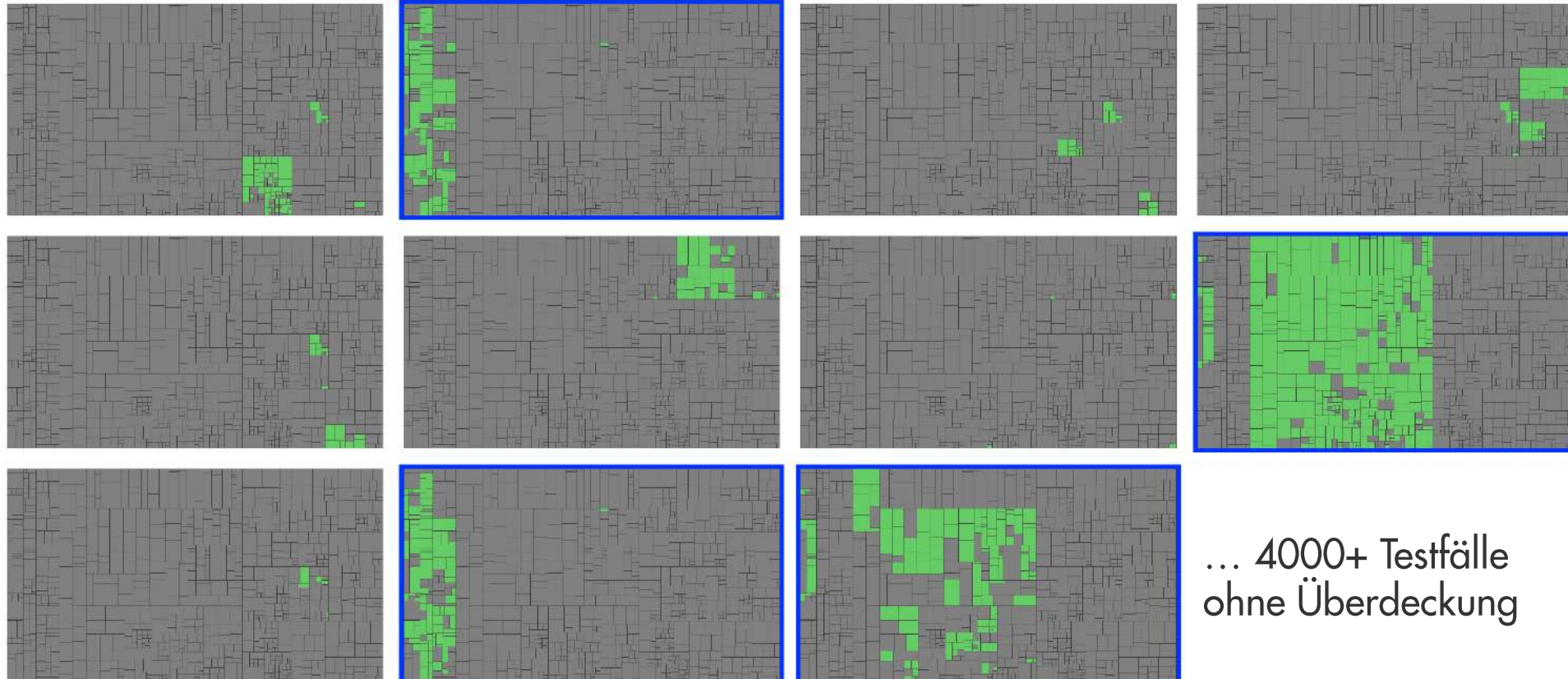
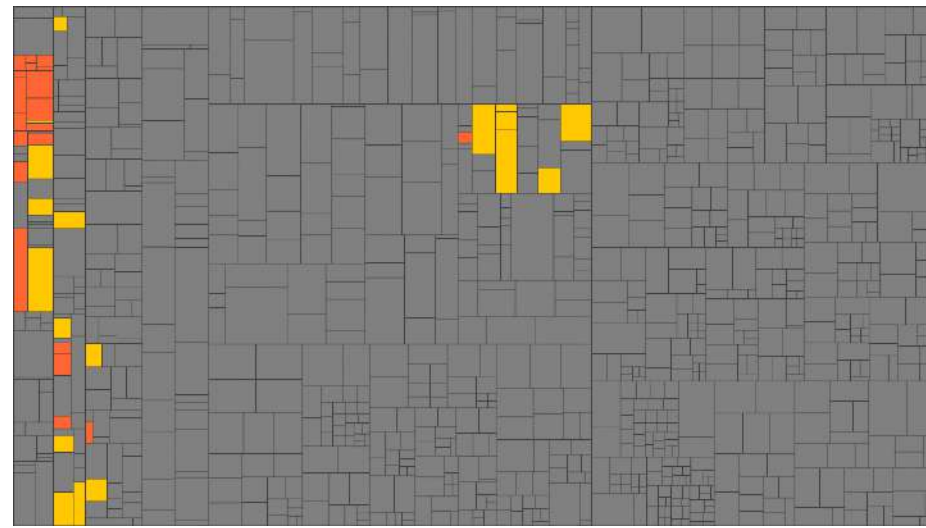




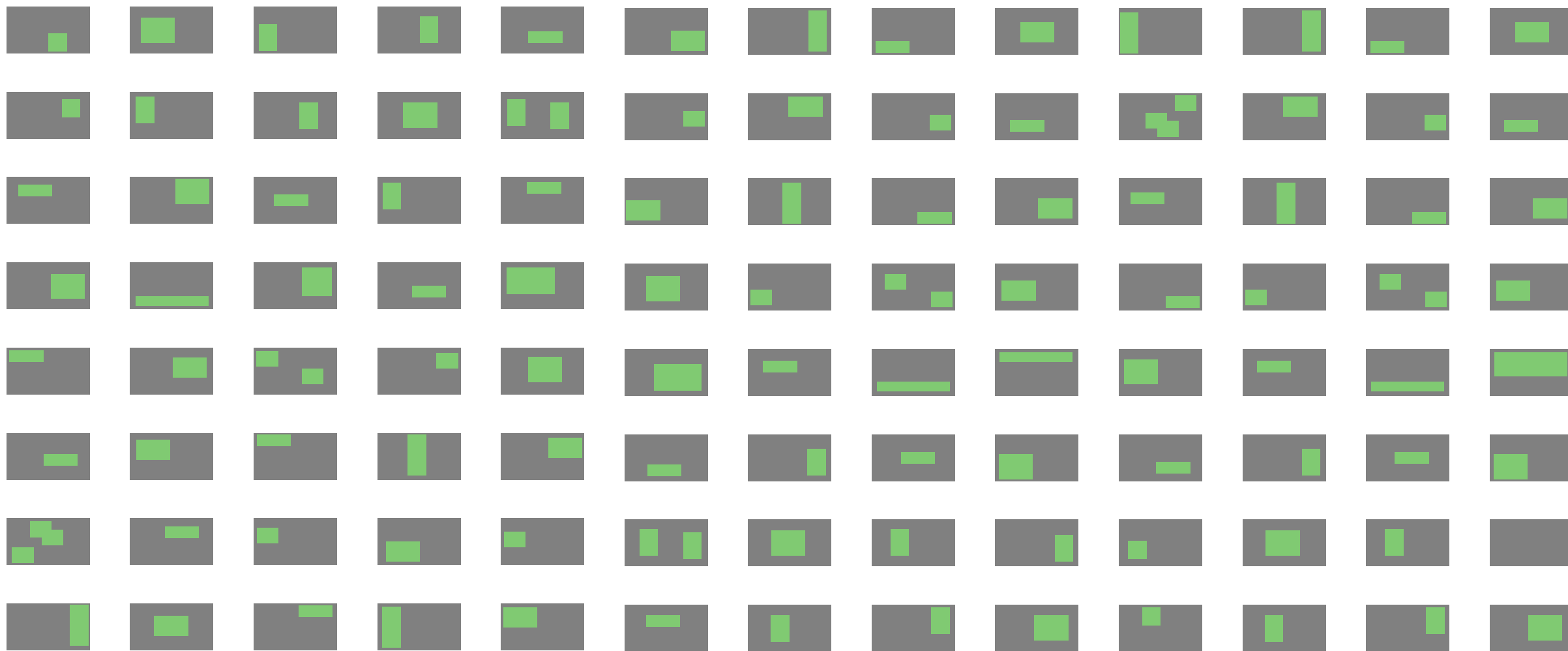








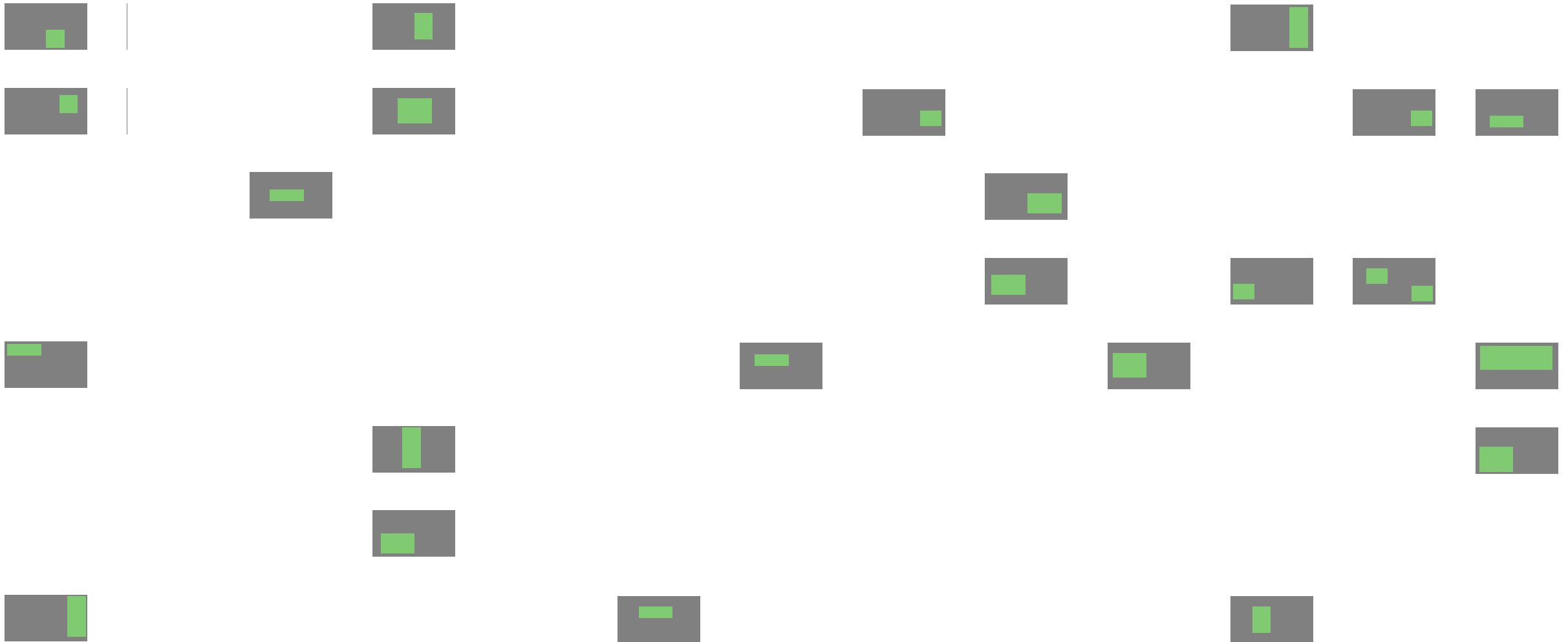
... 4000+ Testfälle
ohne Überdeckung



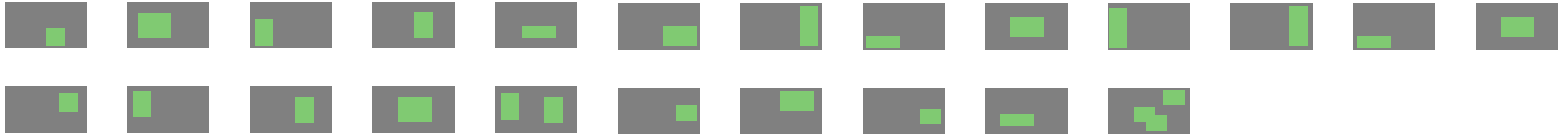
Schritt 1: Selektion betroffener Testfälle



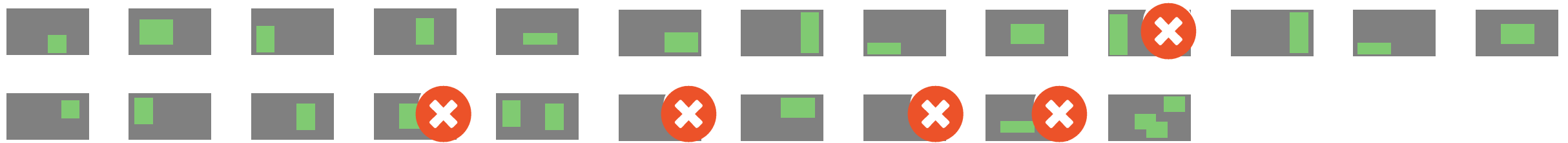
Schritt 1: Selektion betroffener Testfälle



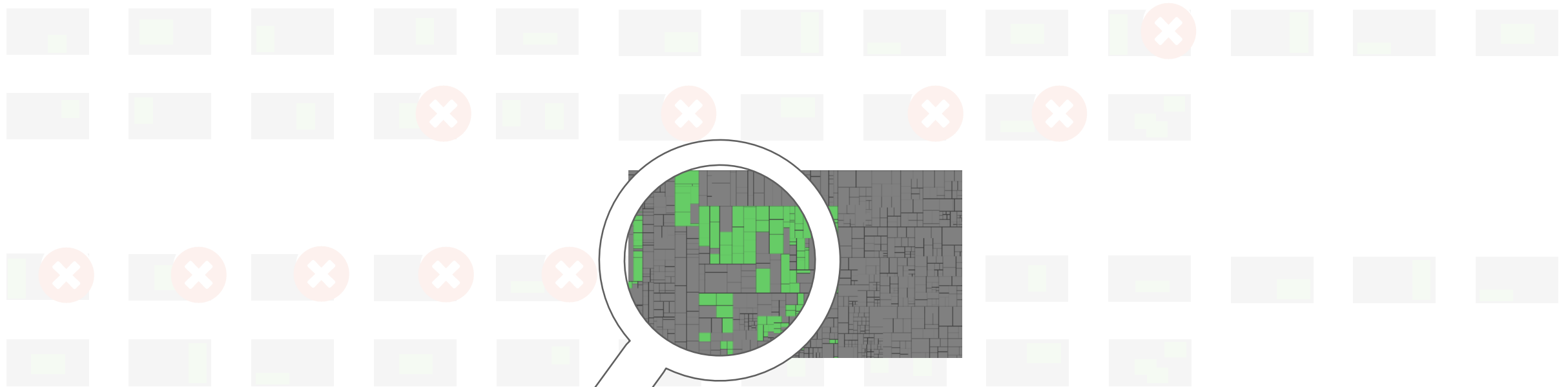
Schritt 1: Selektion betroffener Testfälle



Schritt 2: Priorisierung selektierter Testfälle

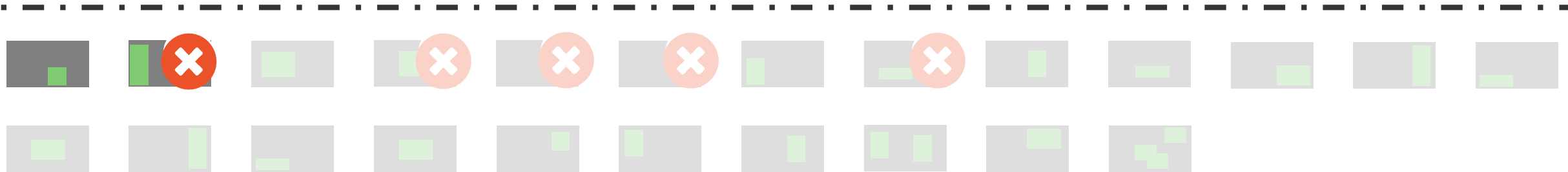


Schritt 2: Priorisierung selektierter Testfälle

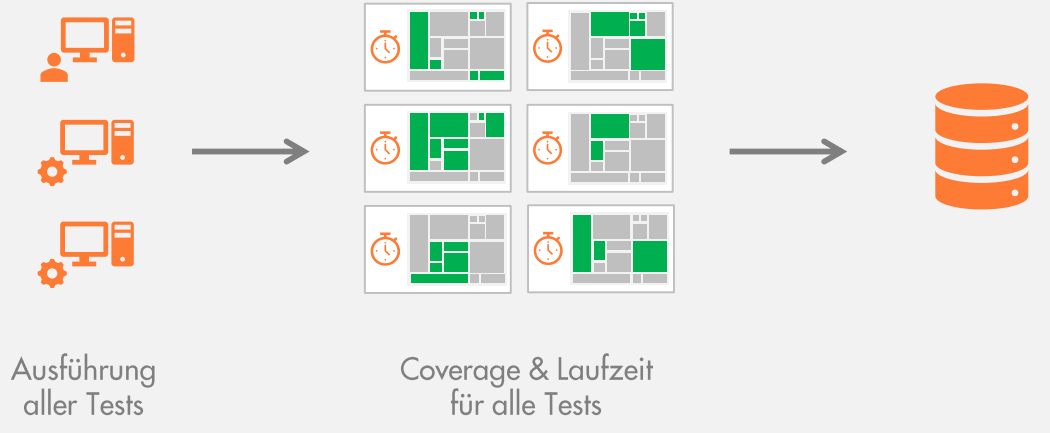


 Change coverage

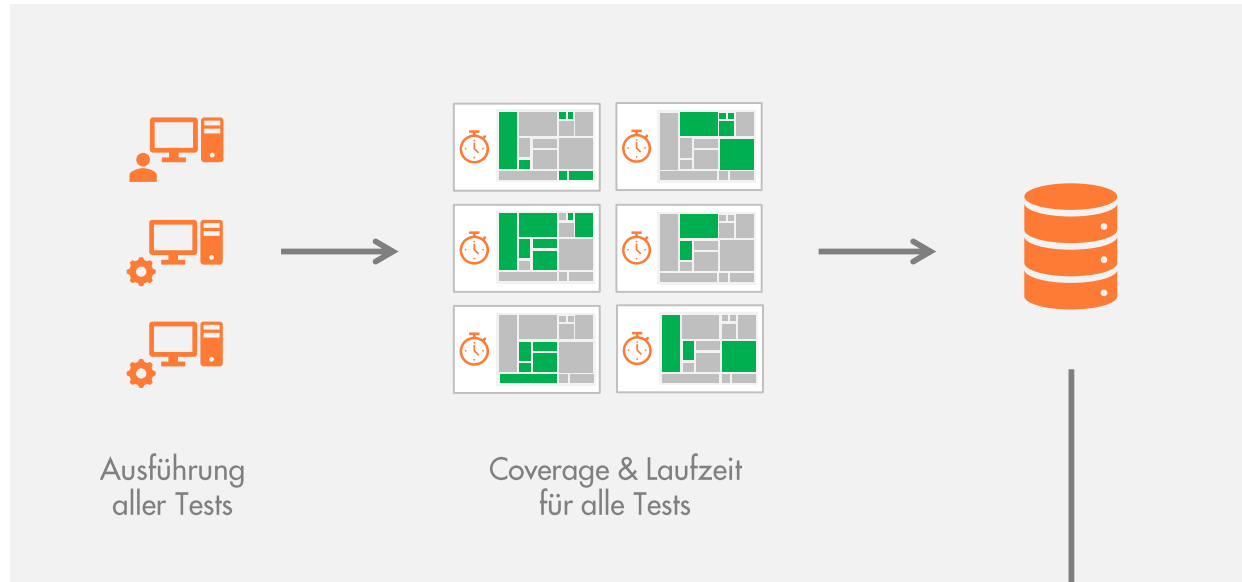
 Execution time



Initiale Aufzeichnung aller Tests



Initiale Aufzeichnung aller Tests



Testausführung nach Änderungen



Grenzen von Test-Impact-Analyse

- Manche Änderungen betreffen alle Tests
 - Änderungen von Konfigurationsinformation
 - Änderungen an Testdaten
- Wenn sich die Tests selbst ändern, müssen sie erneut ausgeführt werden
- Es gibt keine Garantie, dass die selektierten Tests alle Fehler finden

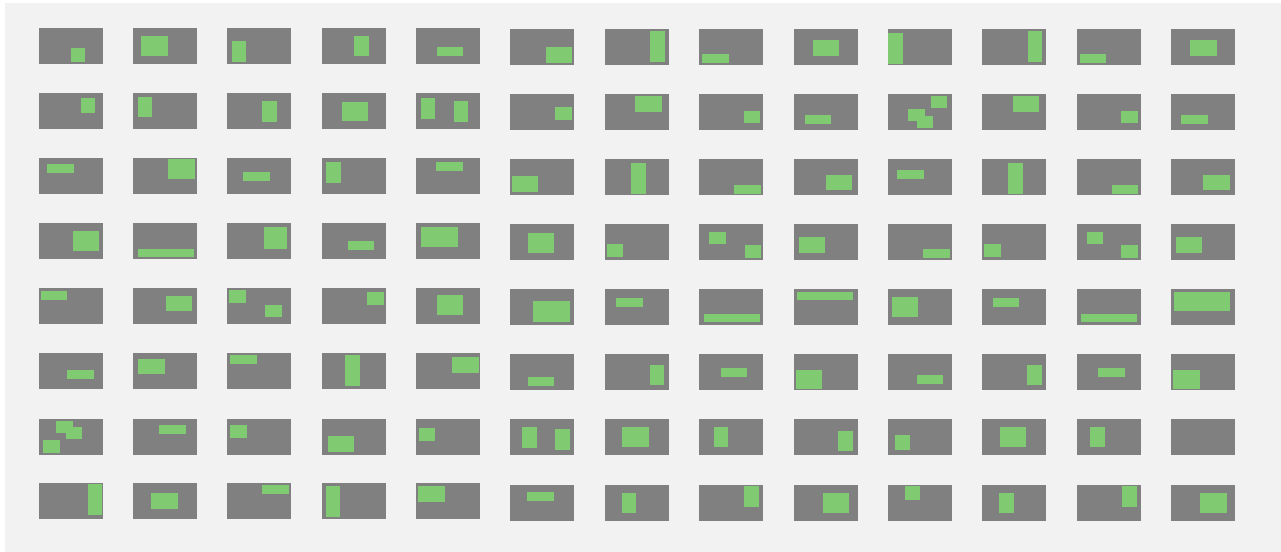
⇒ In regelmäßigen Abständen alle Tests ausführen

⇒ Hält testfallspezifische Coverage-Daten aktuell

Ziel: Tests in der **Continuous Integration** wirksam machen



Trotzdem: Regelmäßig **alle Tests** laufen lassen



Neue Tests immer ausführen



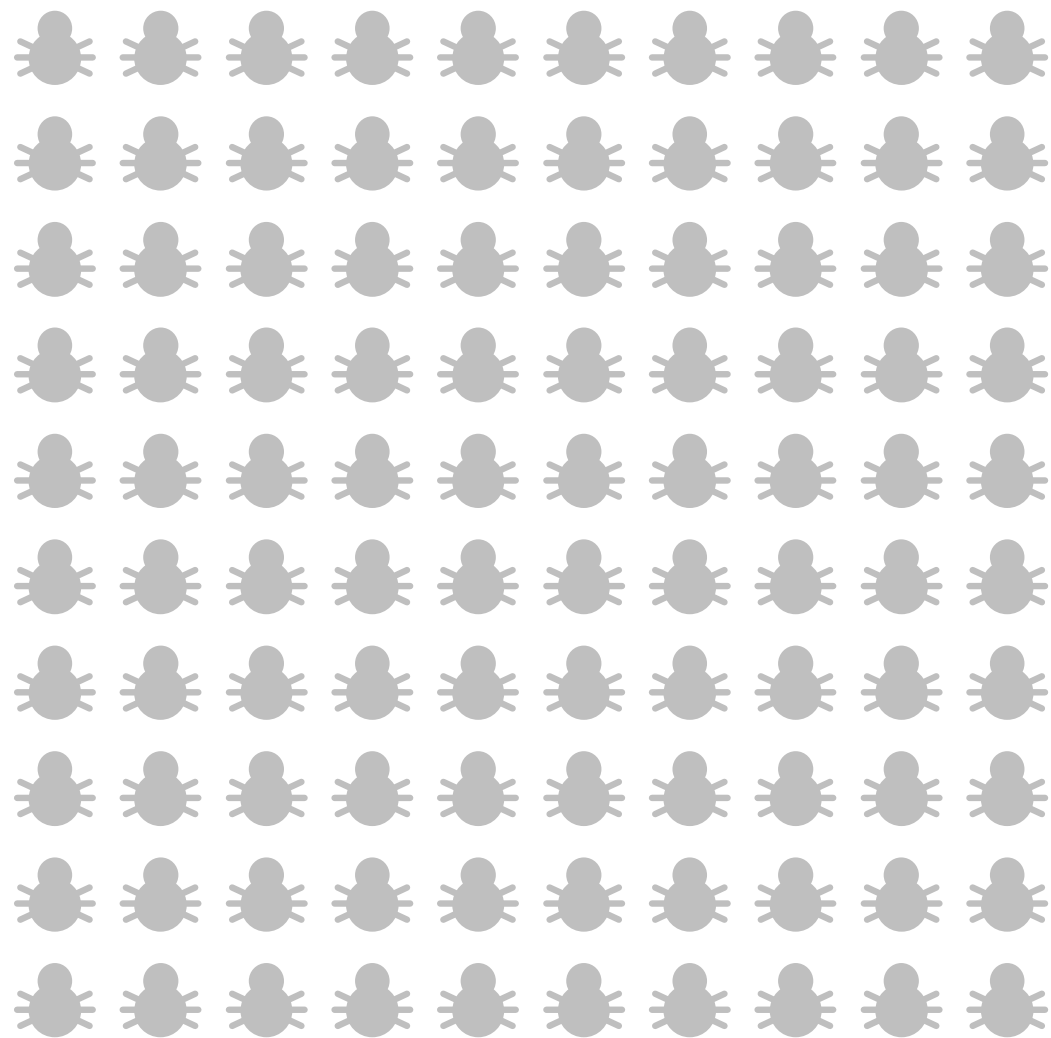
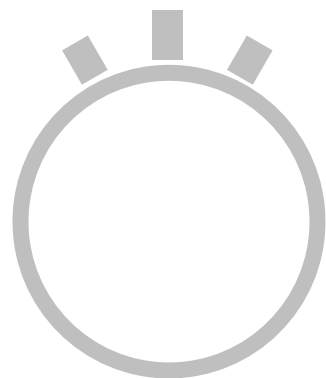
Geänderte Tests immer ausführen

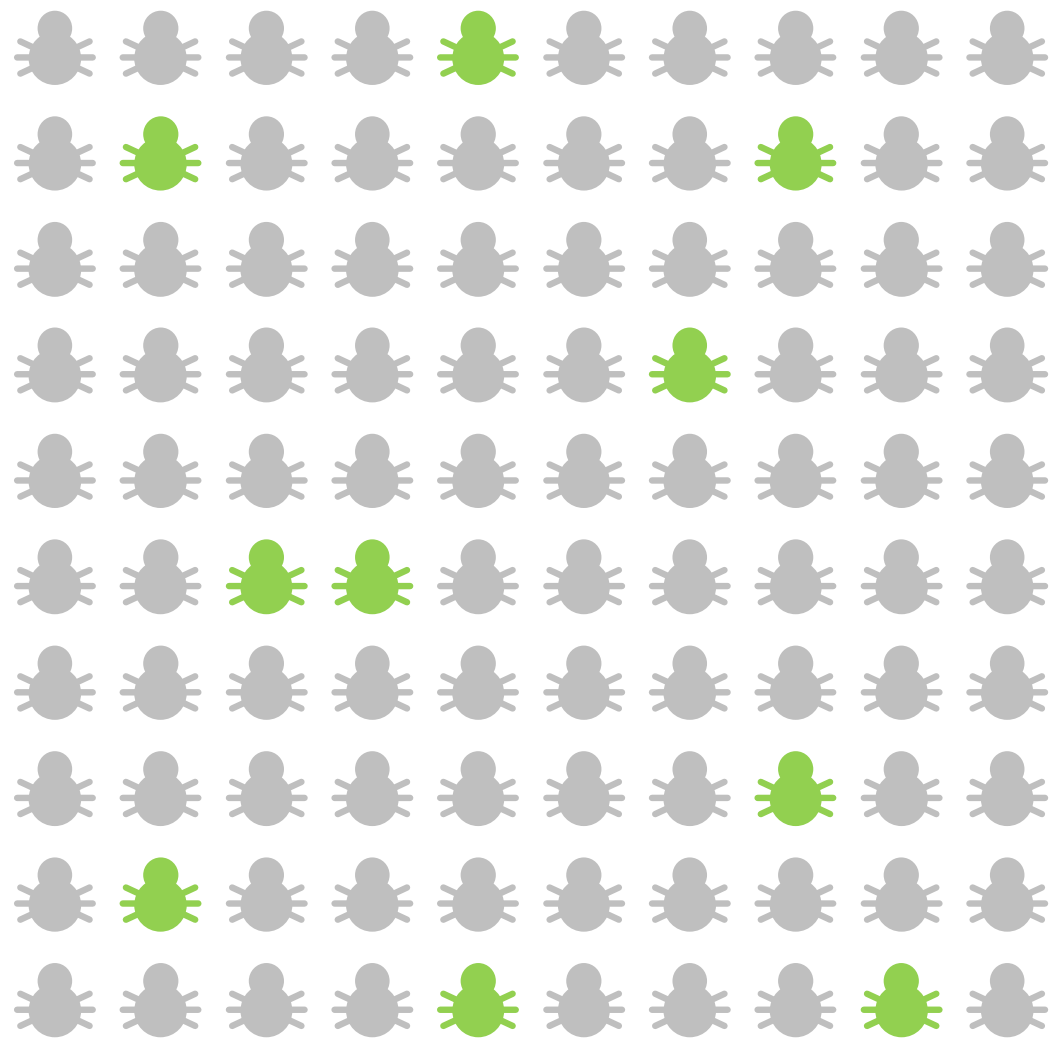


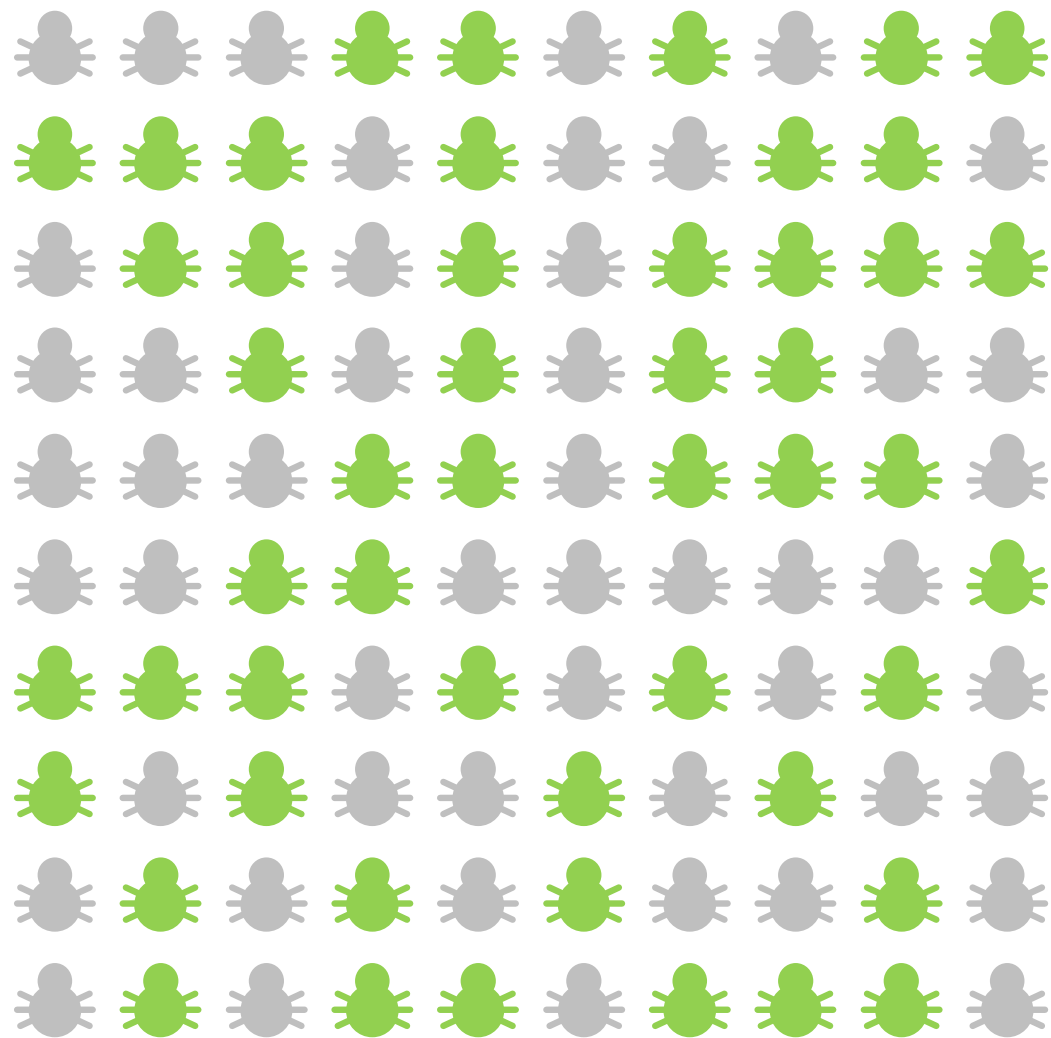
... wenn sich Daten ändern

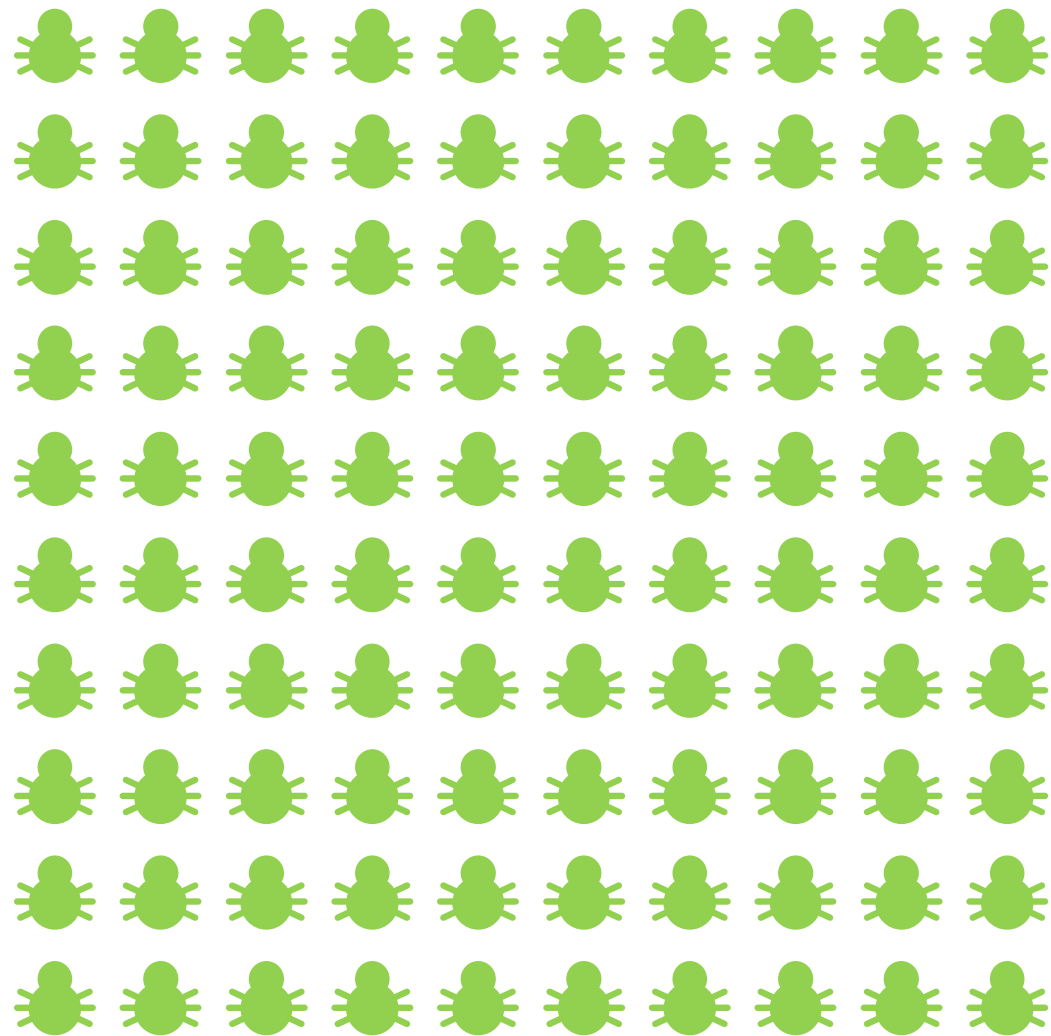
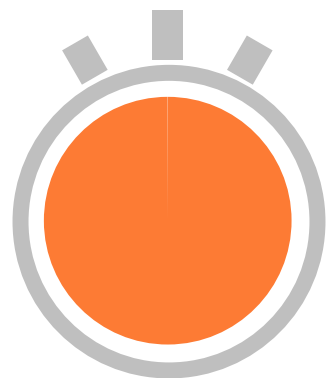


... wenn sich die Konfiguration ändert











Baseline

Working Copy



```
int I = 0;
i++;
void foo() {
  do();
  c = a + b;
}
int get() {
  return fo();
}
close();
```



- Test 1
- Test 2
- Test 3
- Test 4
- Test 5

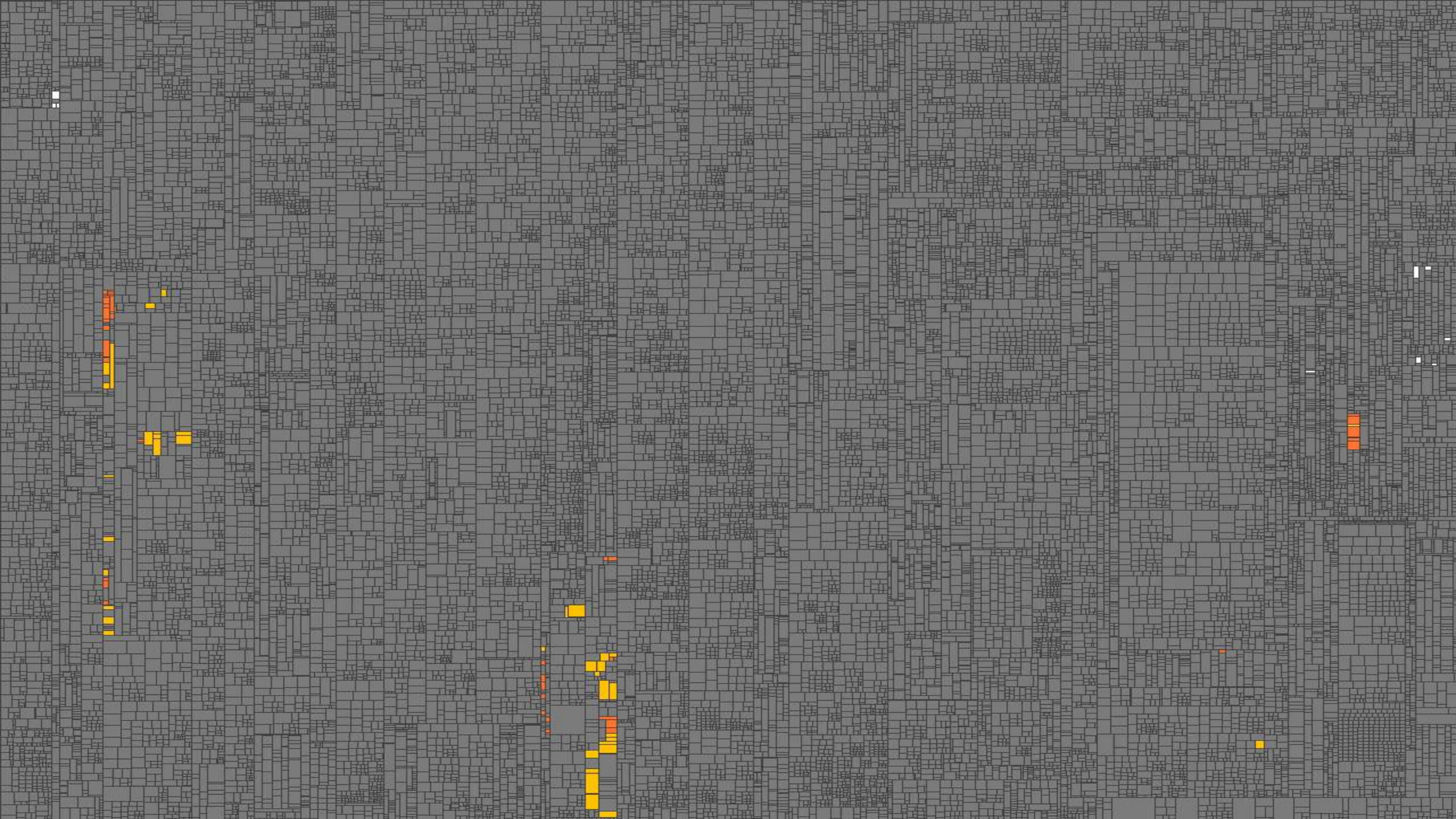
Werden mit den selektierten Tests
die gleichen Fehler gefunden,
wie mit allen Tests?

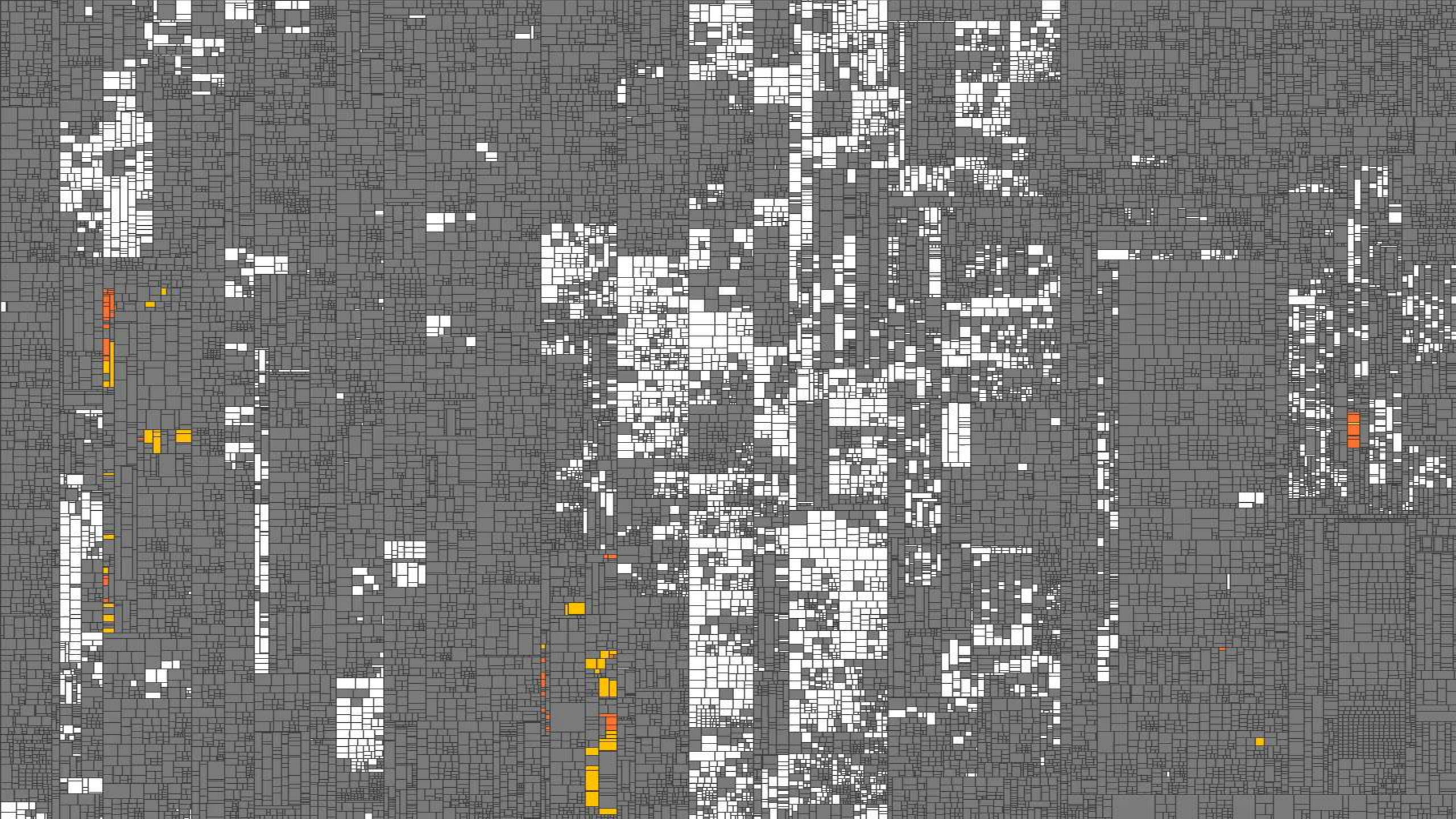
Wieviel Zeit kann mit selektierten Tests eingespart werden?

System	Alle Tests [s]
Apache Commons Collections	25
Apache Commons Lang	25
Apache Commons Math	160
Histone Template Engine	35
JabRef	120
Joda-Time	21
Lightweight-Stream-API	2
LittleProxy	155
OkHttp	97
RxJava	464
Symia Commons Math Parser	1
Teamscale	1249

System	Alle Tests [s]	Selektierte Tests [s]	Einsparung
Apache Commons Collections	25	1	98 %
Apache Commons Lang	25	3	88 %
Apache Commons Math	<div style="text-align: center;"> <p>Im Durchschnitt</p> <p>64%</p> <p>Einsparung</p> </div>	114	29 %
Histone Template Engine		32	7 %
JabRef		41	66 %
Joda-Time		1	94 %
Lightweight-Stream-API		1	68 %
LittleProxy		150	3 %
OkHttp		97	76
RxJava	464	171	63 %
Symia Commons Math Parser	1	1	0 %
Teamscale	1249	197	84 %

Einsparung bei Erkennung aller fehlschlagenden Tests schwankt stark





Welcher Anteil fehlerhafter Builds wird
in begrenzter Zeit erkannt?

System	1 %
Apache Commons Collections	84 %
Apache Commons Lang	94 %
Apache Commons Math	80 %
Histone Template Engine	88 %
JabRef	96 %
Joda-Time	98 %
Lightweight-Stream-API	61 %
LittleProxy	74 %
OkHttp	91 %
RxJava	92 %
Symia Commons Math Parser	78 %
Teamscale	92 %

System	1 %	2 %
Apache Commons Collections	84 %	97 %
Apache Commons Lang	94 %	96 %
Apache Commons Math	80 %	86 %
Histone Template Engine	88 %	88 %
JabRef	96 %	96 %
Joda-Time	98 %	99 %
Lightweight-Stream-API	61 %	91 %
LittleProxy	74 %	77 %
OkHttp	91 %	91 %
RxJava	92 %	95 %
Symia Commons Math Parser	78 %	86 %
Teamscale	92 %	96 %

System	1 %	2 %	5 %	10 %
Apache Commons Collections	84 %	97 %	100 %	100 %
Apache Commons Lang	94 %	96 %	98 %	99 %
Apache Commons Math	80 %	86 %	92 %	93 %
Histone Template Engine	88 %	88 %	88 %	88 %
JabRef	96 %	96 %	96 %	96 %
Joda-Time	98 %	99 %	99 %	100 %
Lightweight-Stream-API	61 %	91 %	95 %	97 %
LittleProxy	74 %	77 %	80 %	80 %
OkHttp	91 %	91 %	91 %	97 %
RxJava	92 %	95 %	95 %	95 %
Symia Commons Math Parser	78 %	86 %	89 %	89 %
Teamscale	92 %	96 %	97 %	97 %

Über Mich

Studium

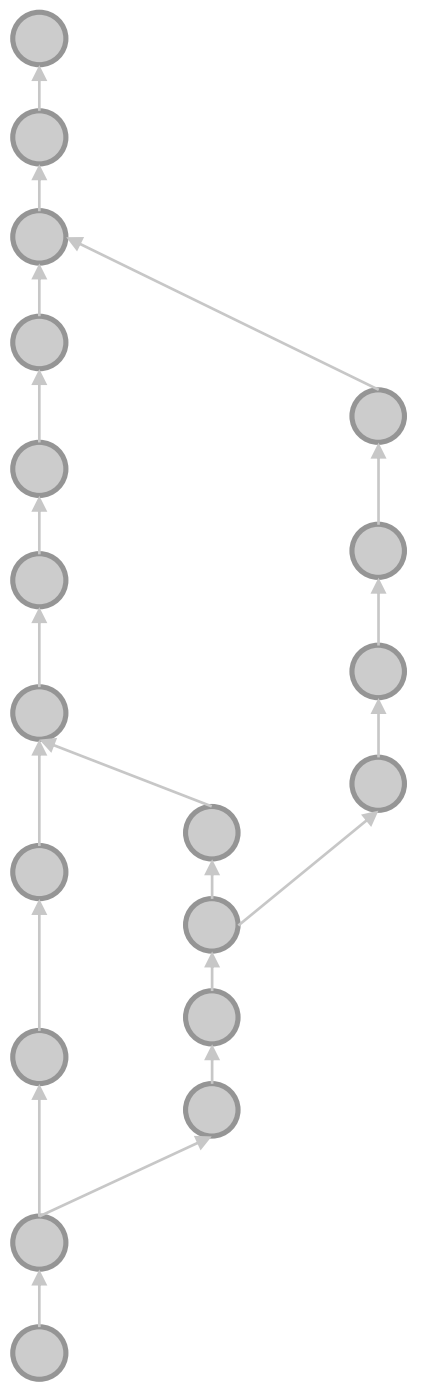
- Master Informatik an der TU München
- Masterarbeit zu Test-Impact-Analyse

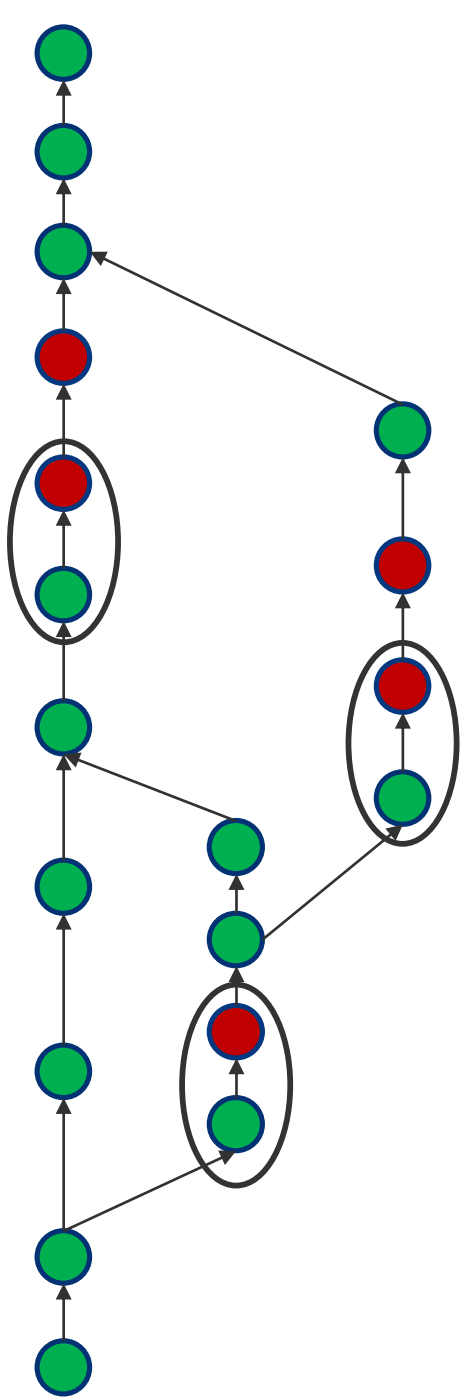
Consulting

- Software Consultant bei TNG Technology Consulting



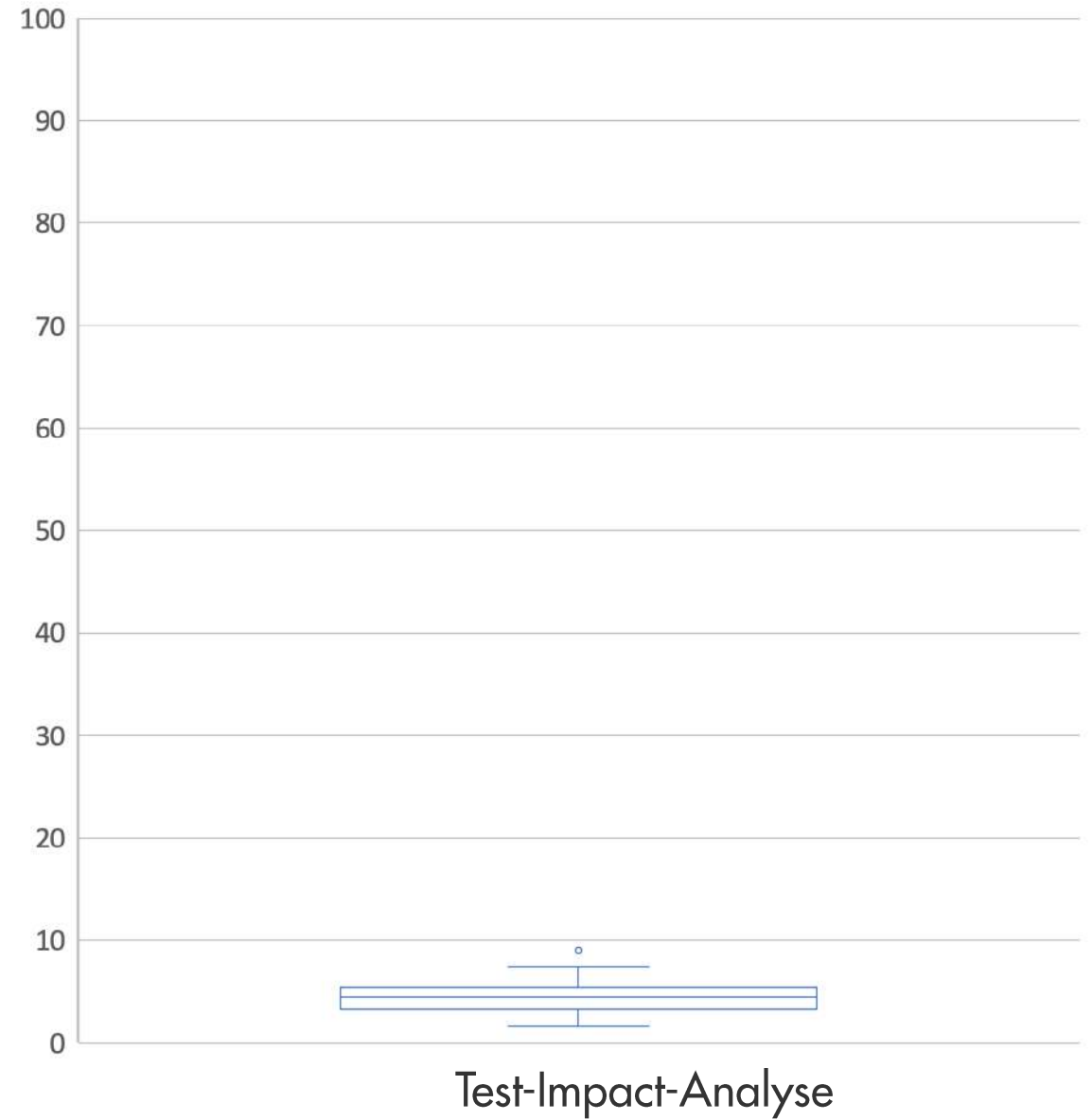
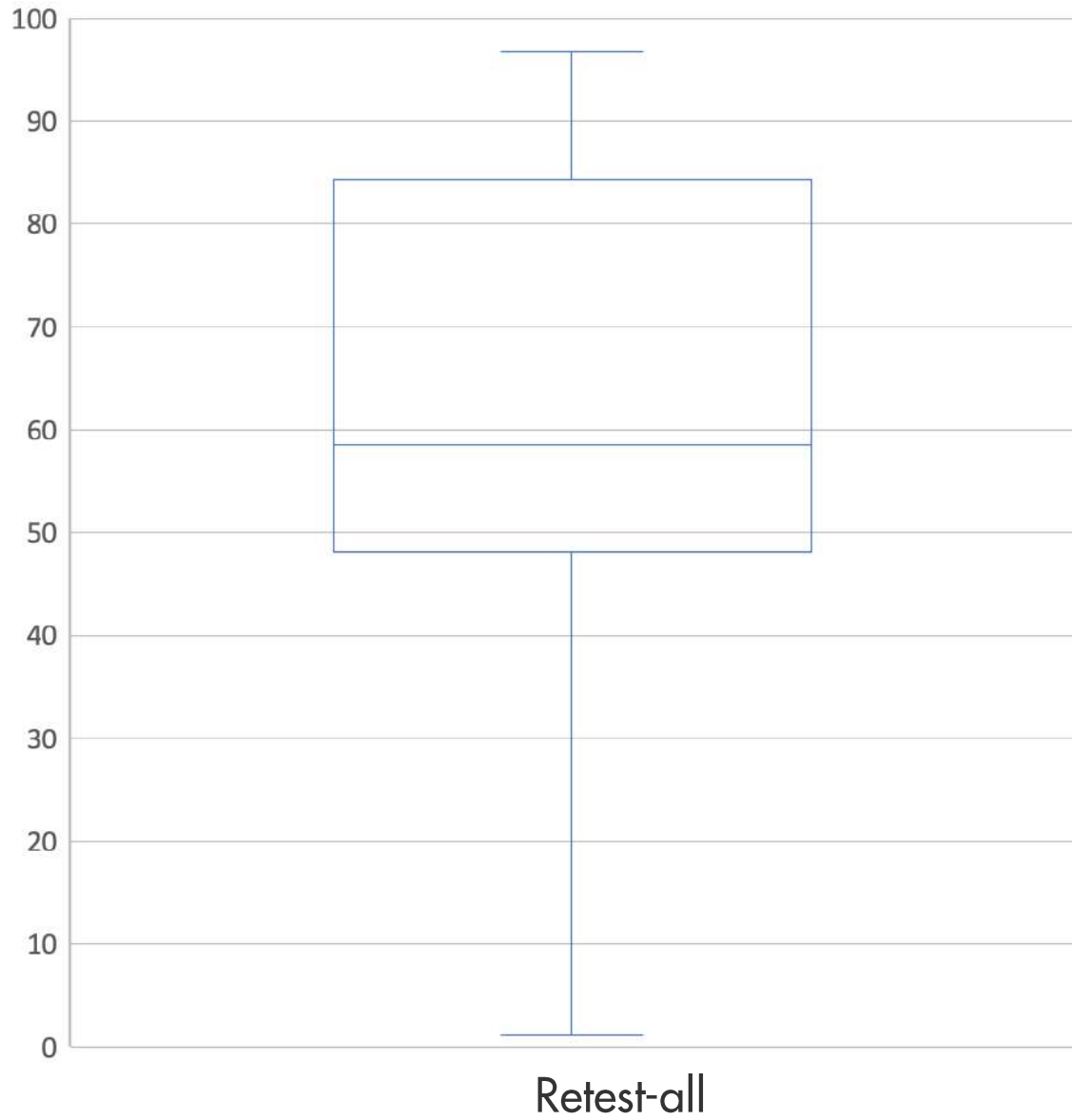
Lines of Code	4,3 millionen
Technologien	Java, Spring, Hibernate, Maven, ...
Anzahl Maven Submodule	736
Entwicklung seit	2001
Anzahl aktiver Entwickler	60
Laufzeit lokale Tests	ca. 37 Minuten
Laufzeit End-to-end Tests	ca. 7 Minuten



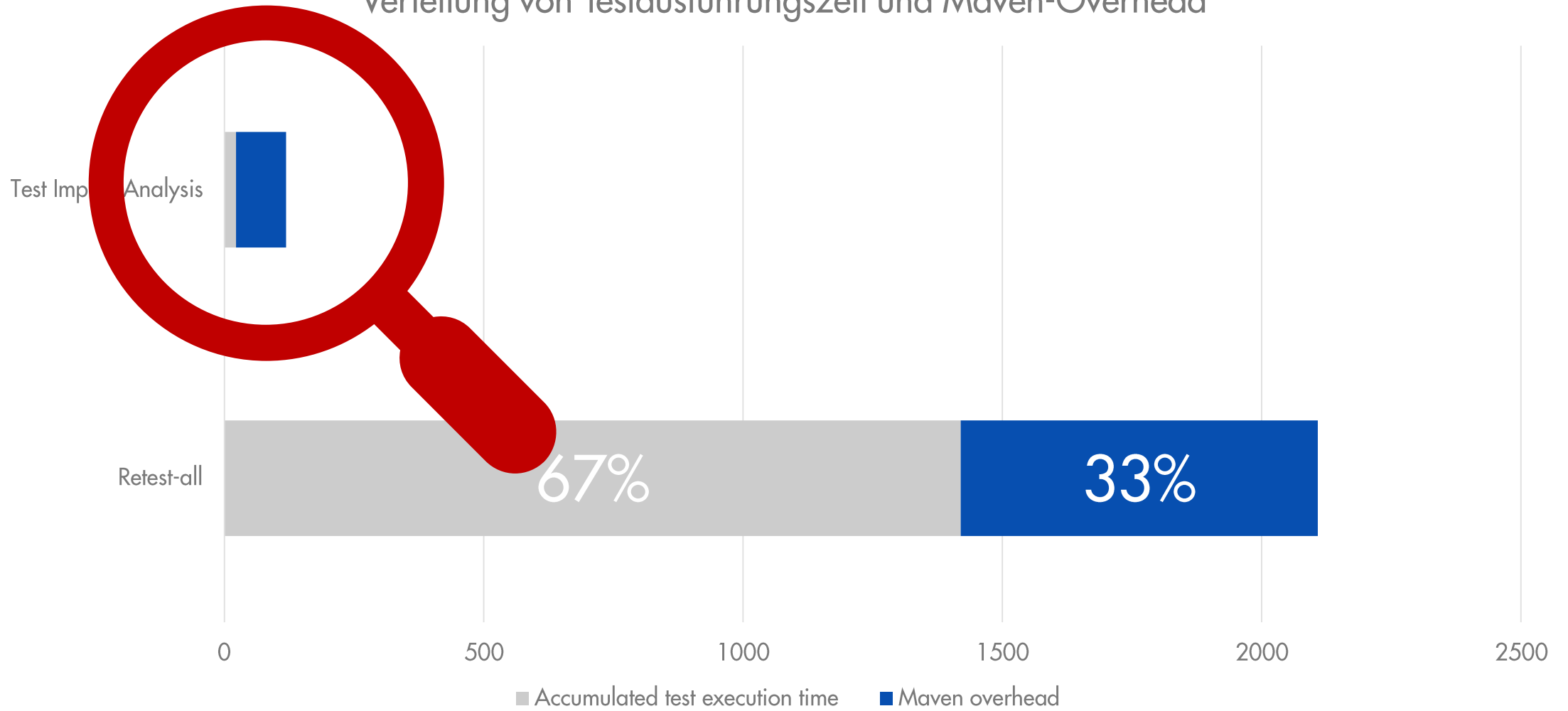


Untersuchte Commits	1605
Fehlerverursachende Änderungen	50

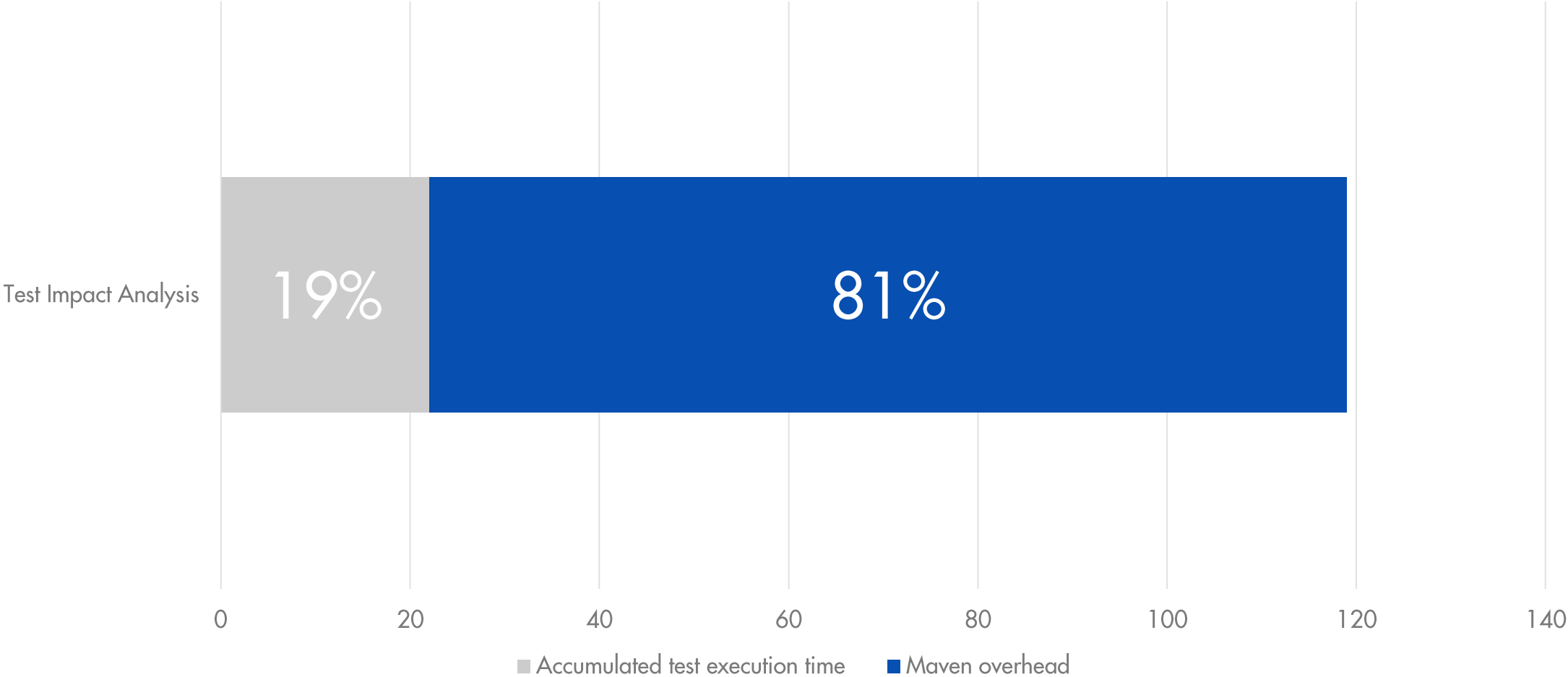
Zeitspanne bis erster Fehler gefunden wird, relativ zu gesamter Ausführungszeit

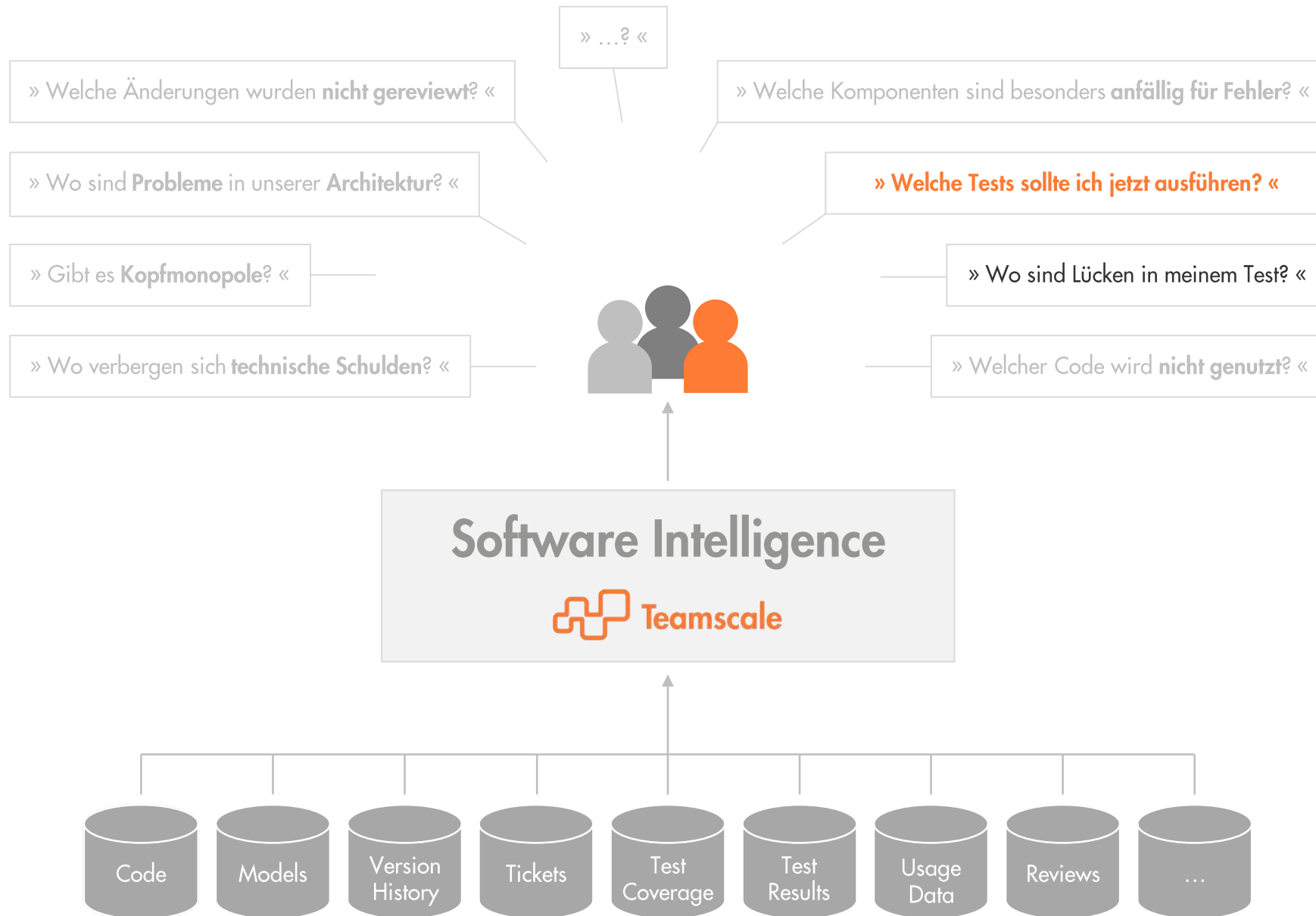


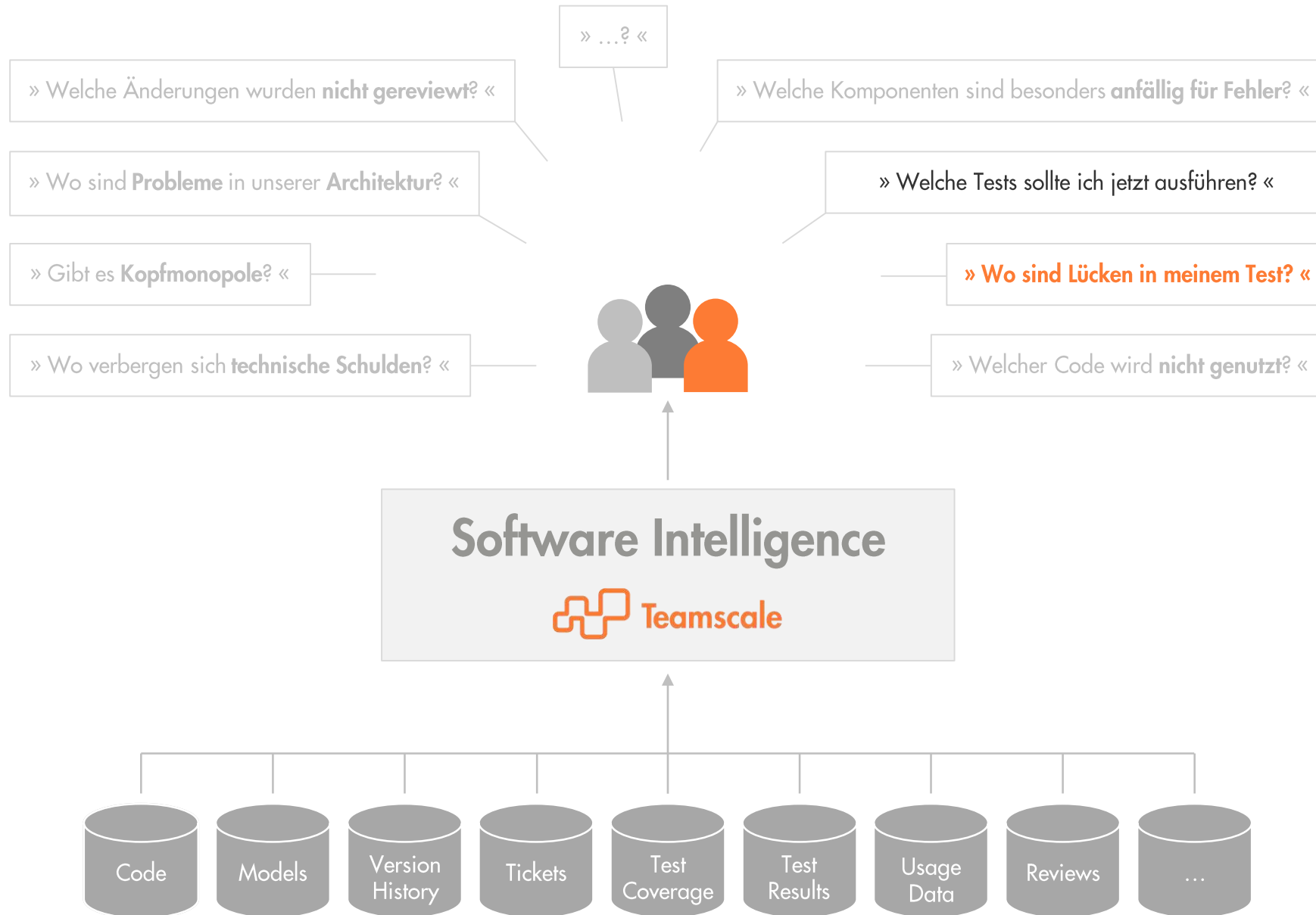
Verteilung von Testausführungszeit und Maven-Overhead



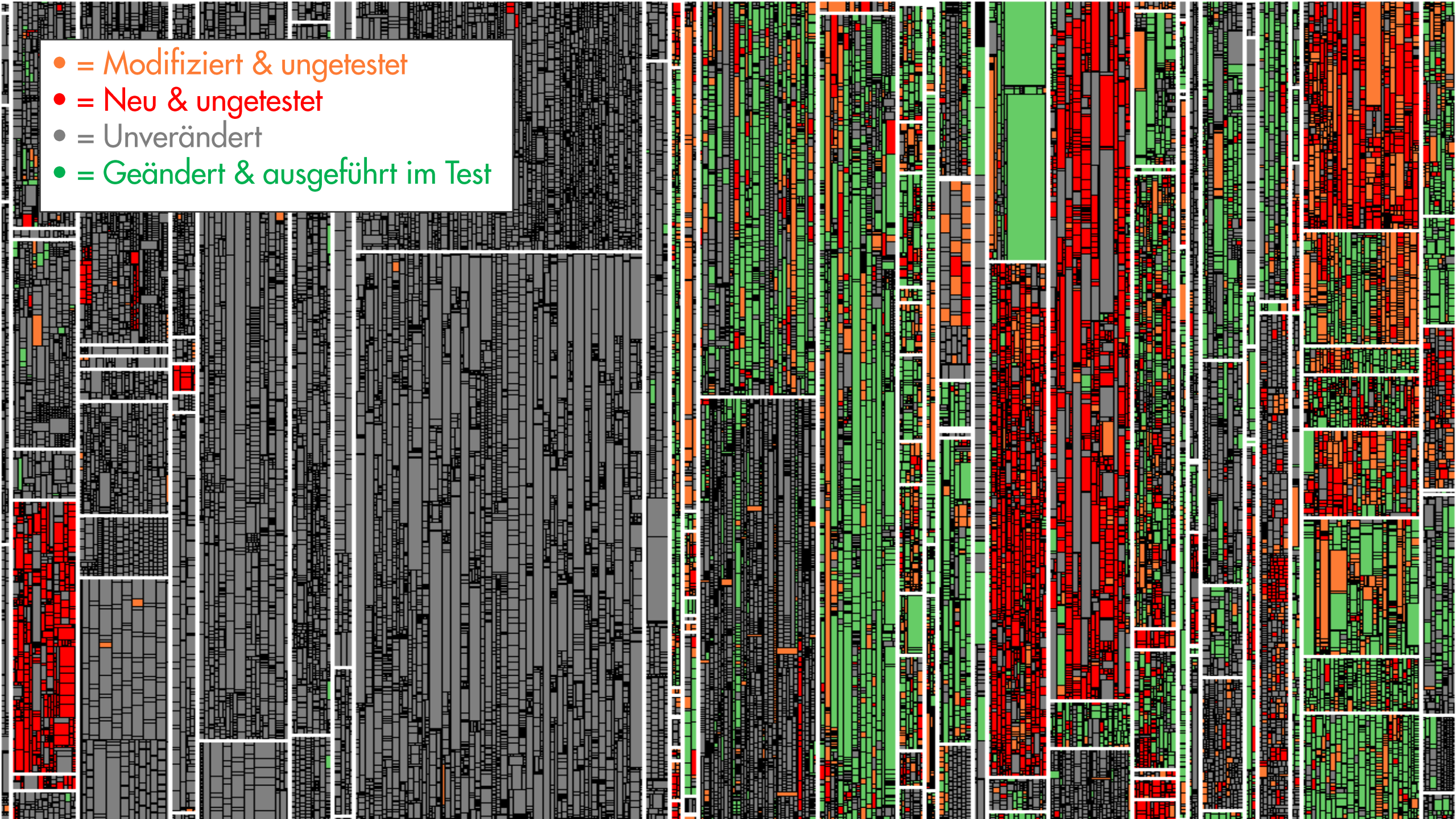
Verteilung von Testausführungszeit und Maven-Overhead




























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




Issue # ▾	Subject	Done	Avatar	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% 

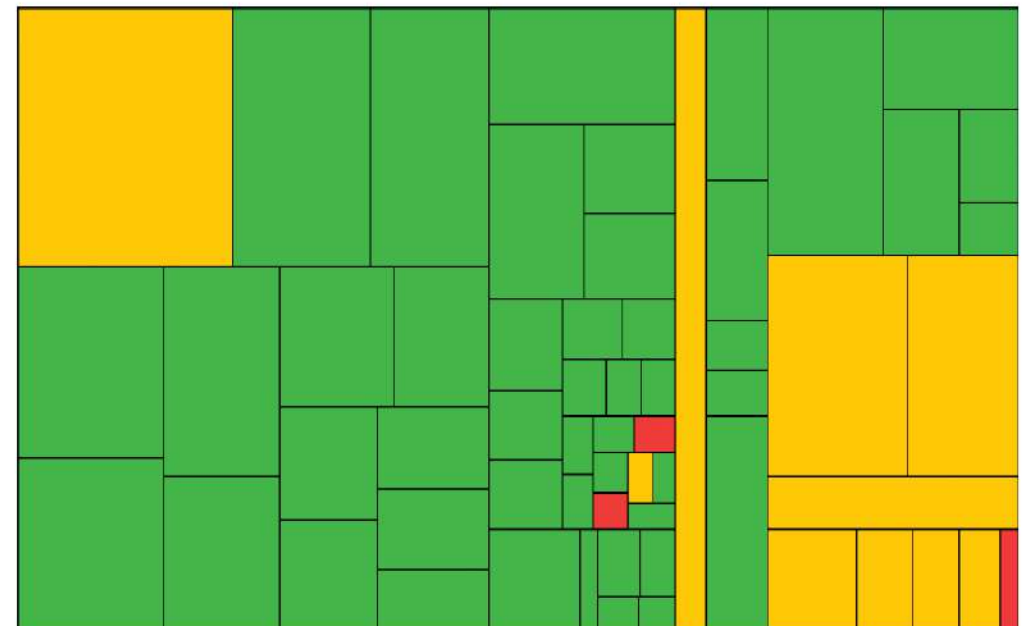
Done Issue TS-14733 - Remove Dataflow blocks

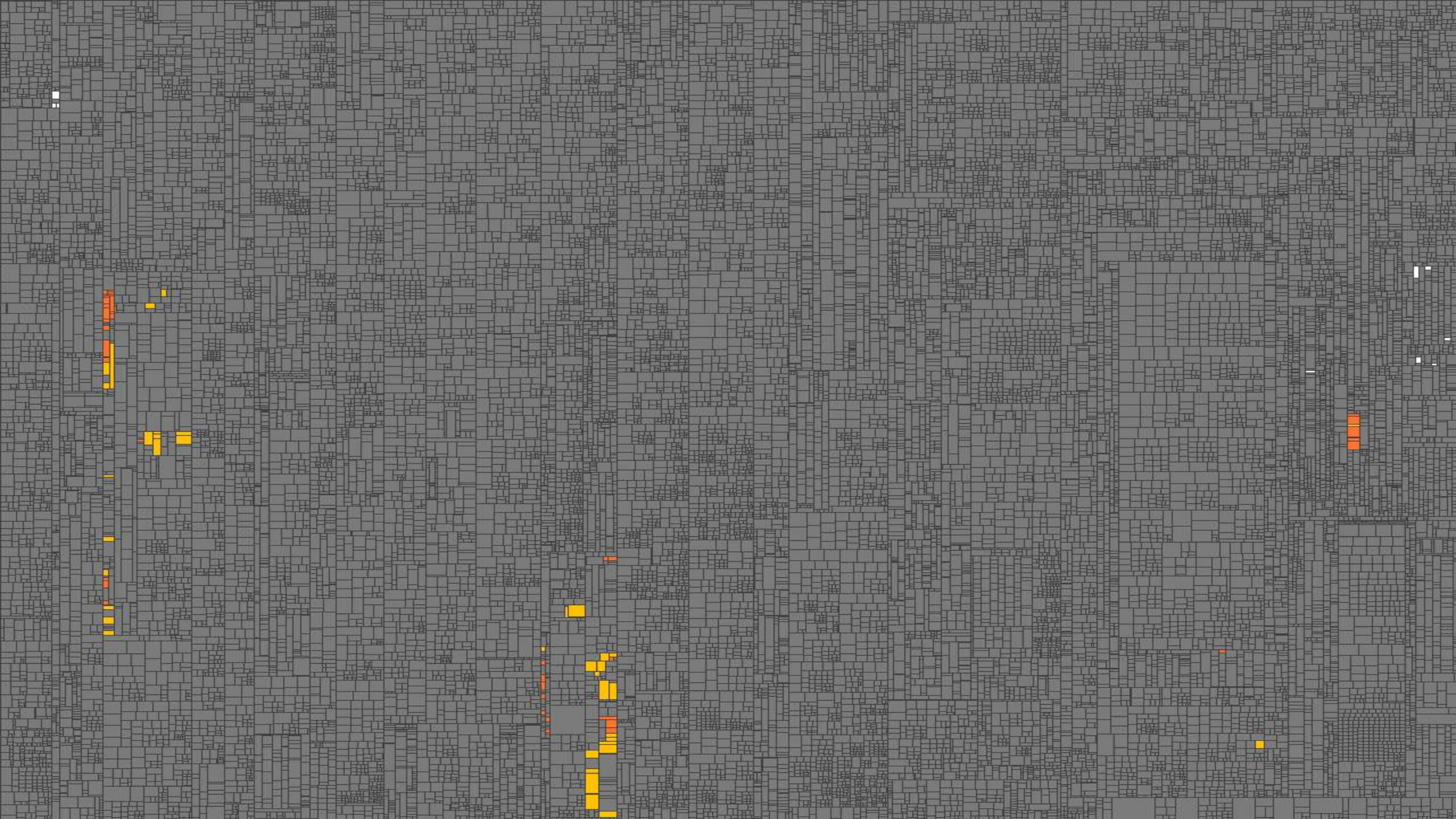
Creator:  (on Apr 06 2018 19:44) Last update: Aug 24 2018 09:32

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%





Vielen Dank!



Kontakt – Wir freuen uns auf Diskussionen 😊



Dr. Elmar Jürgens · juergens@cqse.eu

CQSE



Alexander Kaserbacher · alexander.kaserbacher@tngtech.com

TNG