

09. – 12.12.2019
Frankfurt am Main



Dr. Sven Amann

Müssen wir wirklich schon wieder alles testen?

Trotz lang laufender Testsuiten Fehler
schnell und zuverlässig aufdecken

#ittage



Praxis

Software-**Audits**

Kontinuierliche **Qualitäts-**
und **Testkontrolle**



CQSE GmbH



Forschung

17+ **Promotionen** in
Software Engineering

Eigene **Forschung**

Enger Kontakt zu
Universitäten

 Release Early & Often

 Fail Fast

 Instant Feedback

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

Runs: 1/227  Errors: 0  Failures: 0

▼  com.teamscale.test.ui.architecture.ArchitecturePerspectiveTest [Runner: JUnit 5]

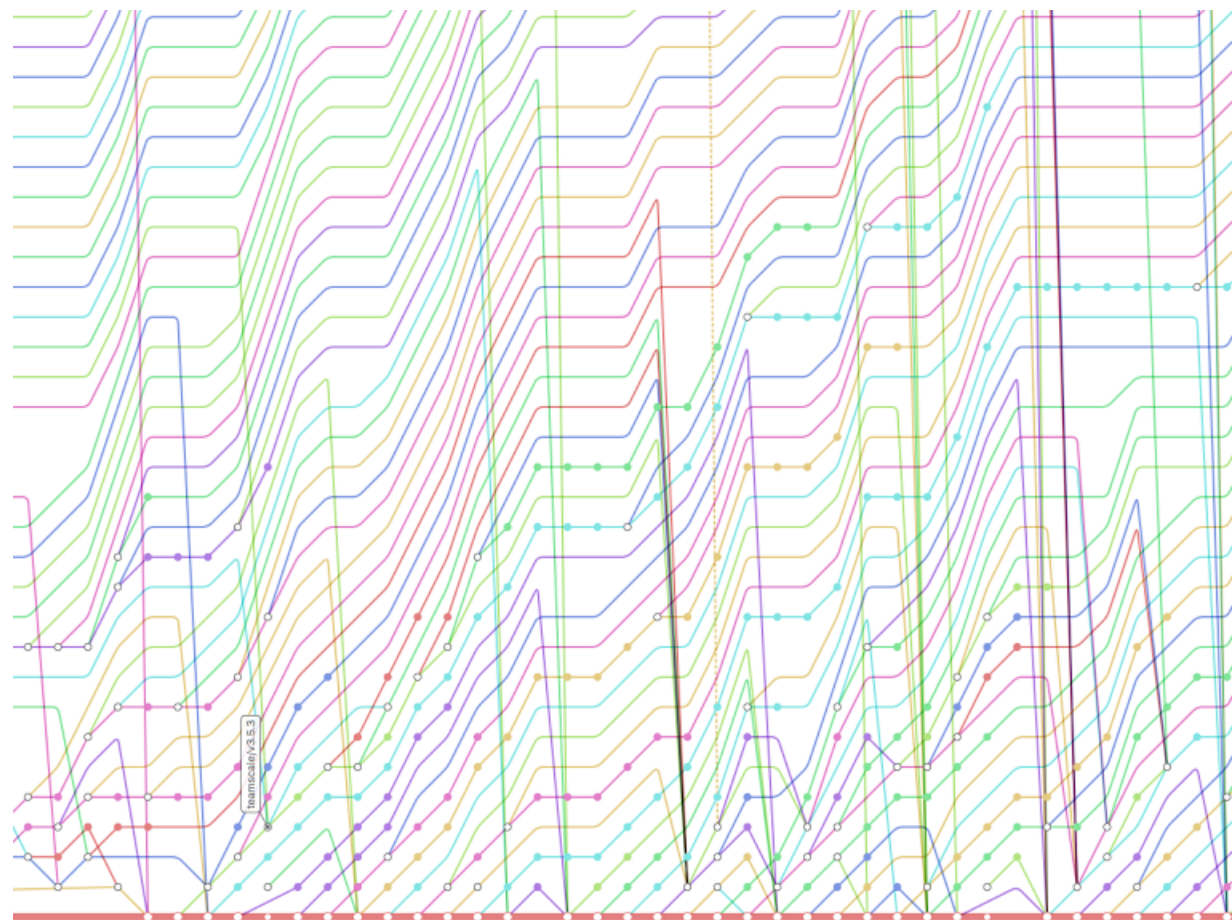
-  testViewingArchitectureElements
-  createArchitectureTest
-  testTimetravelMode
-  testUndoRedo
-  testArchitecturePolicyUpdate
-  testOpenCodeOfOrphans

Failure Trace

Wie lange laufen Ihre Tests?



Eine Stunde?





Eine Stunde?



Eine Nacht?



Eine Woche?

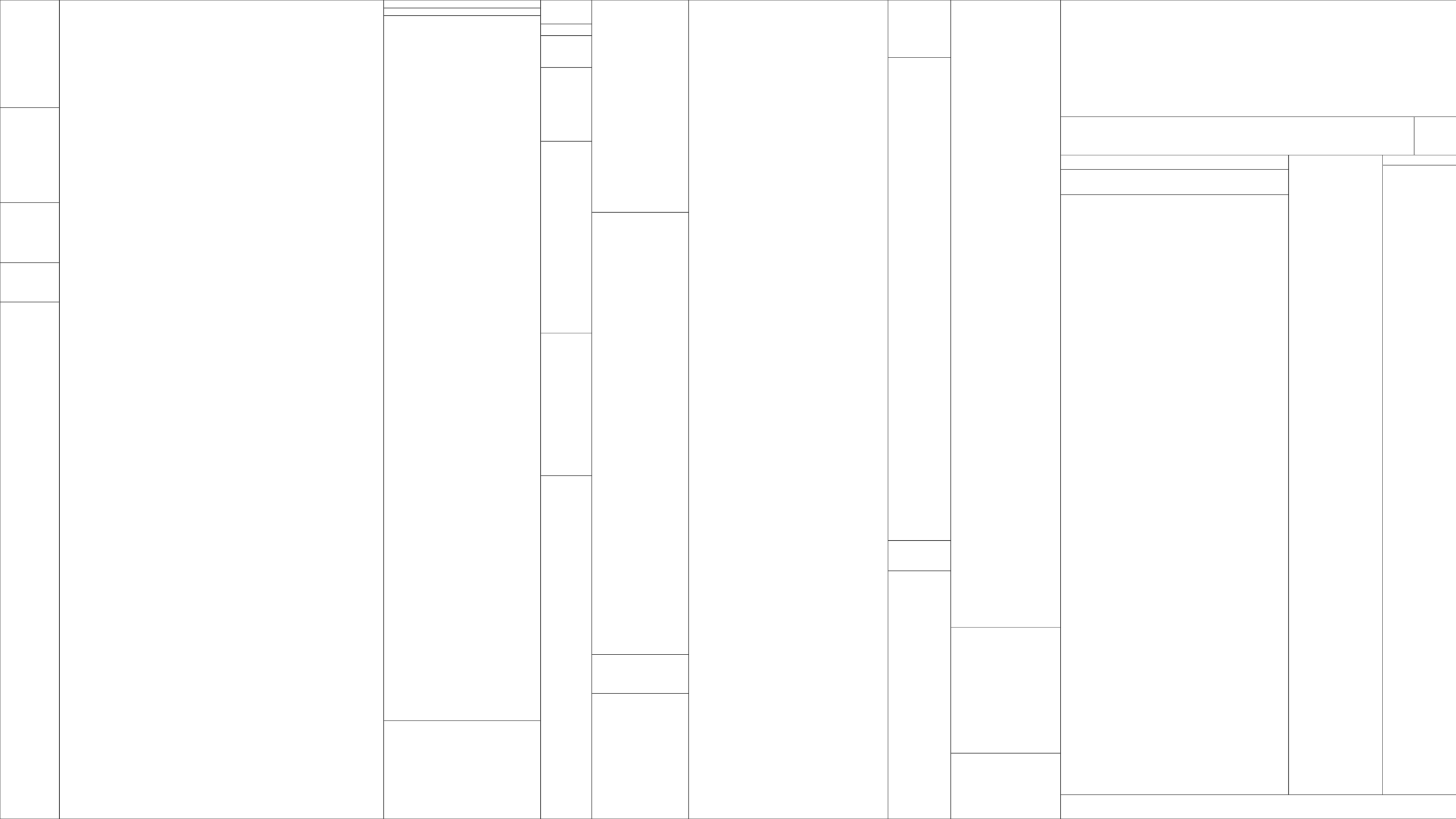


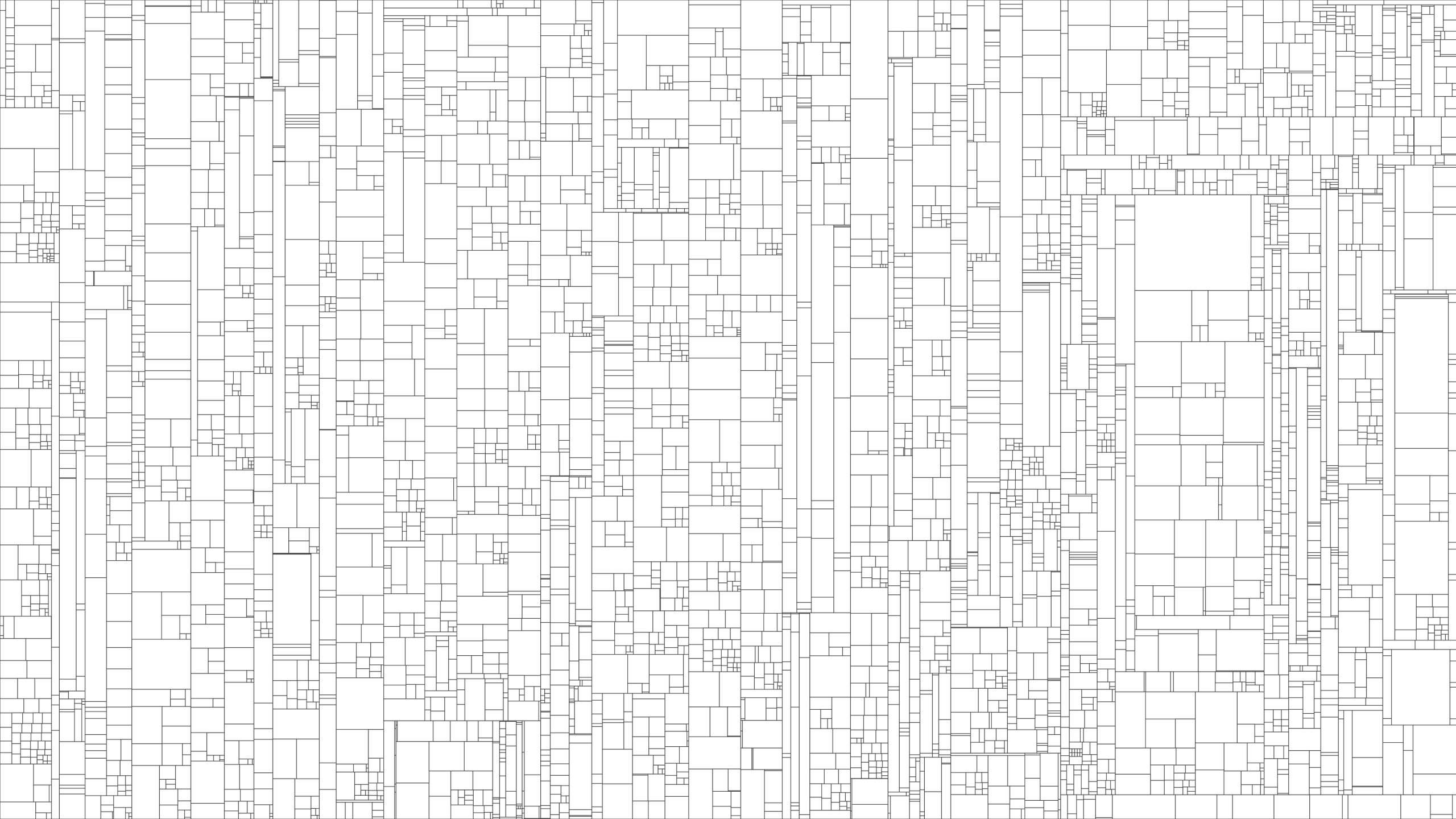
Einen Monat?

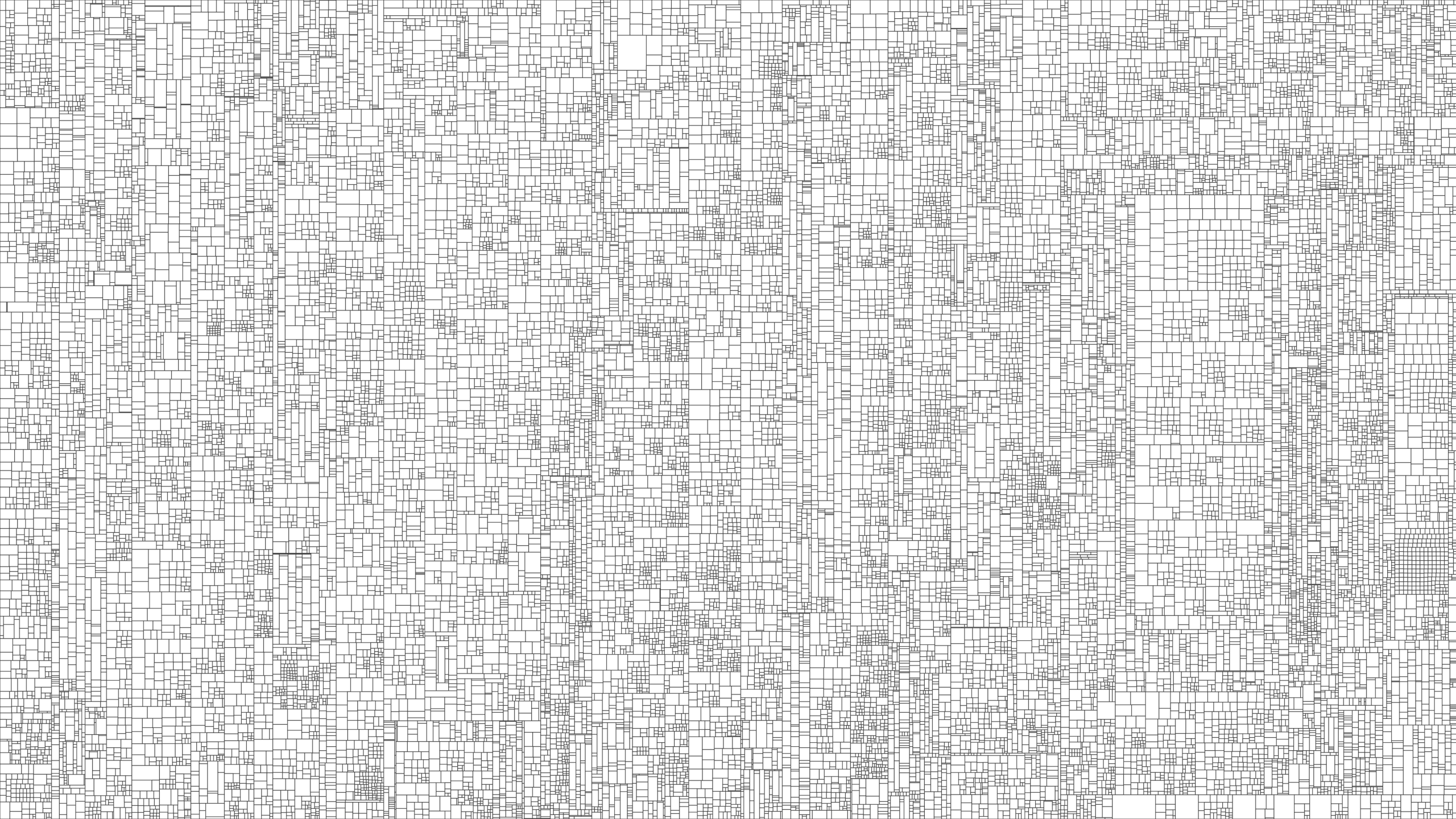


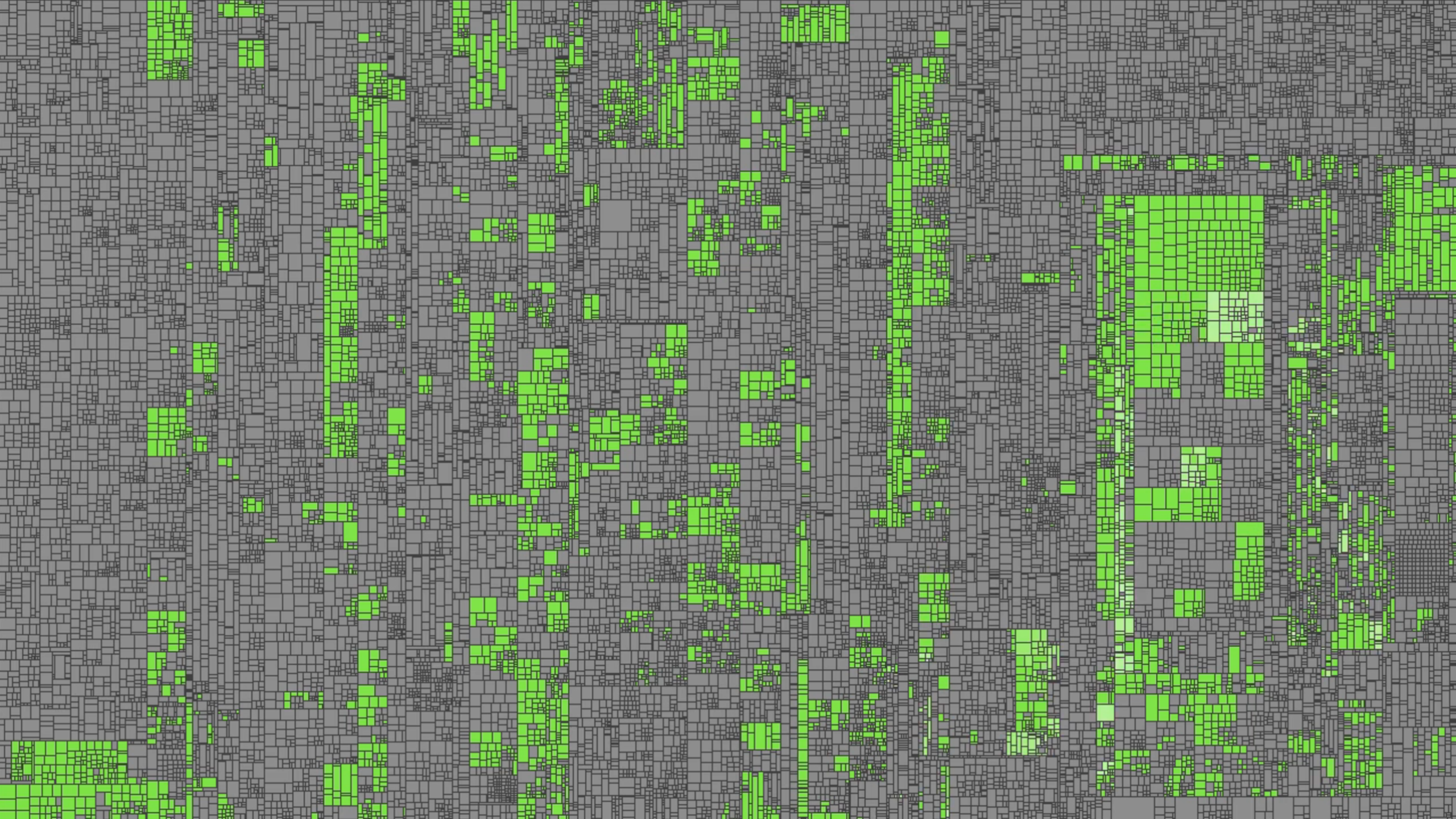
<https://www.technica-engineering.de/produkte/bts-body-electronic-test-system>

Warum laufen die Tests so lange?









Simplicity—the art of maximizing the amount of work not done—is essential.
(Agile Manifesto)



Teamscale / TS-10829

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

Description

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

```
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)
[...]
```

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](#)

Dates

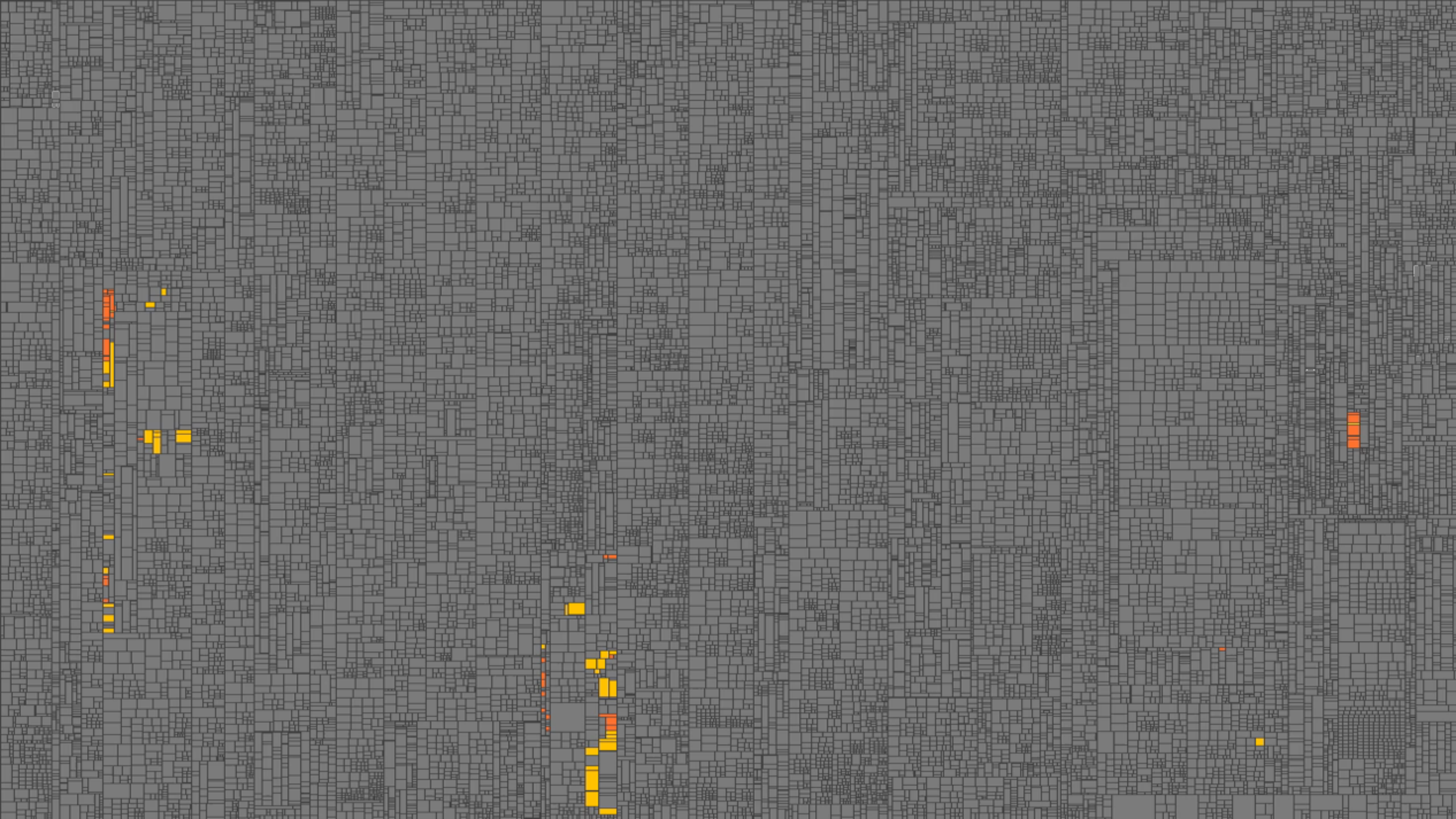
Created: 24/Nov/16 8:35 AM
Updated: 4 days ago
Resolved: 08/May/18 12:42 PM

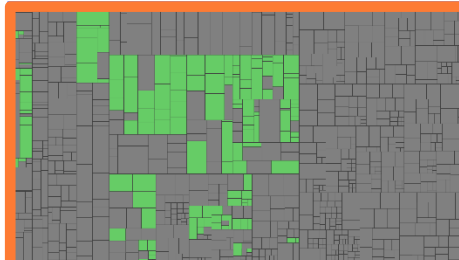
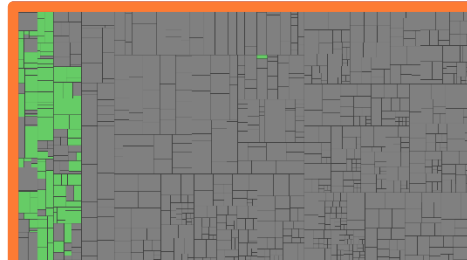
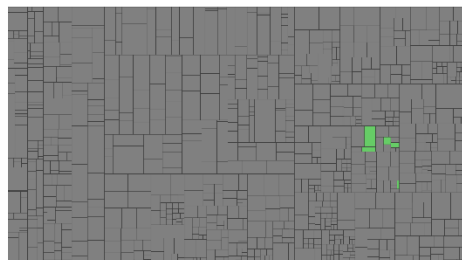
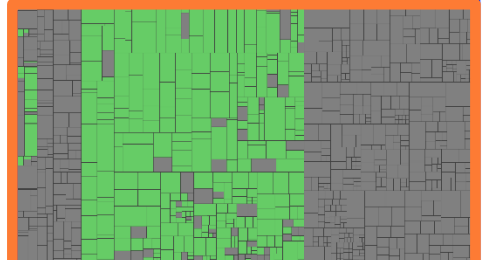
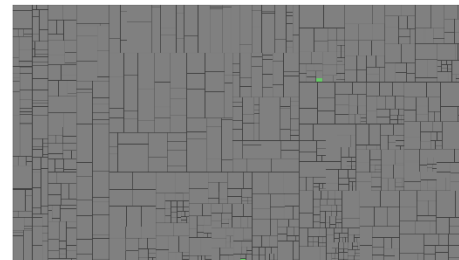
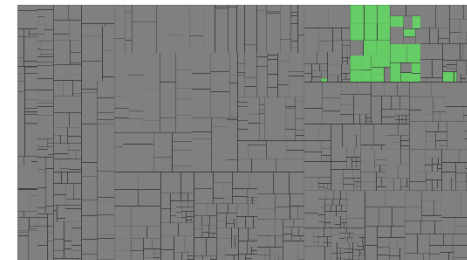
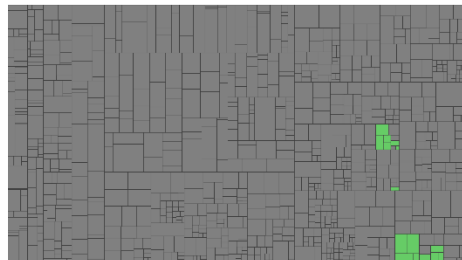
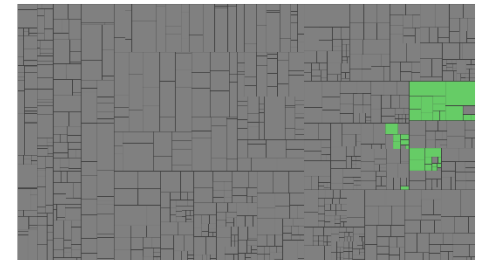
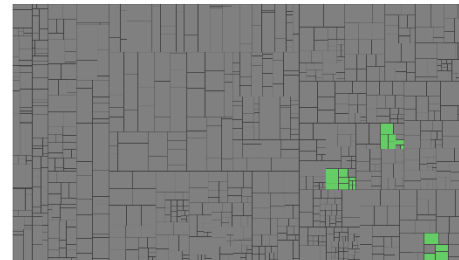
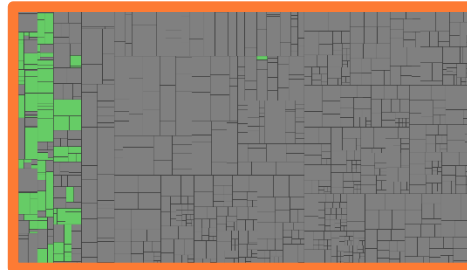
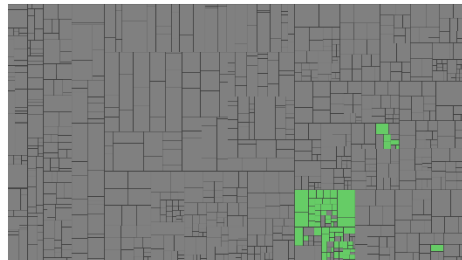
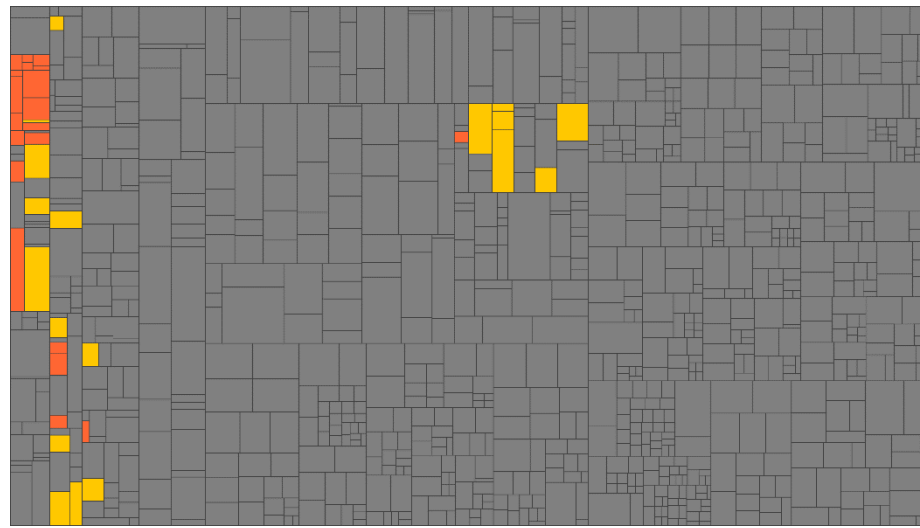
Time Tracking

Estimated: Not Specified
Remaining: 0m
Logged: 4d 1h 57m

Agile

[View on Board](#)

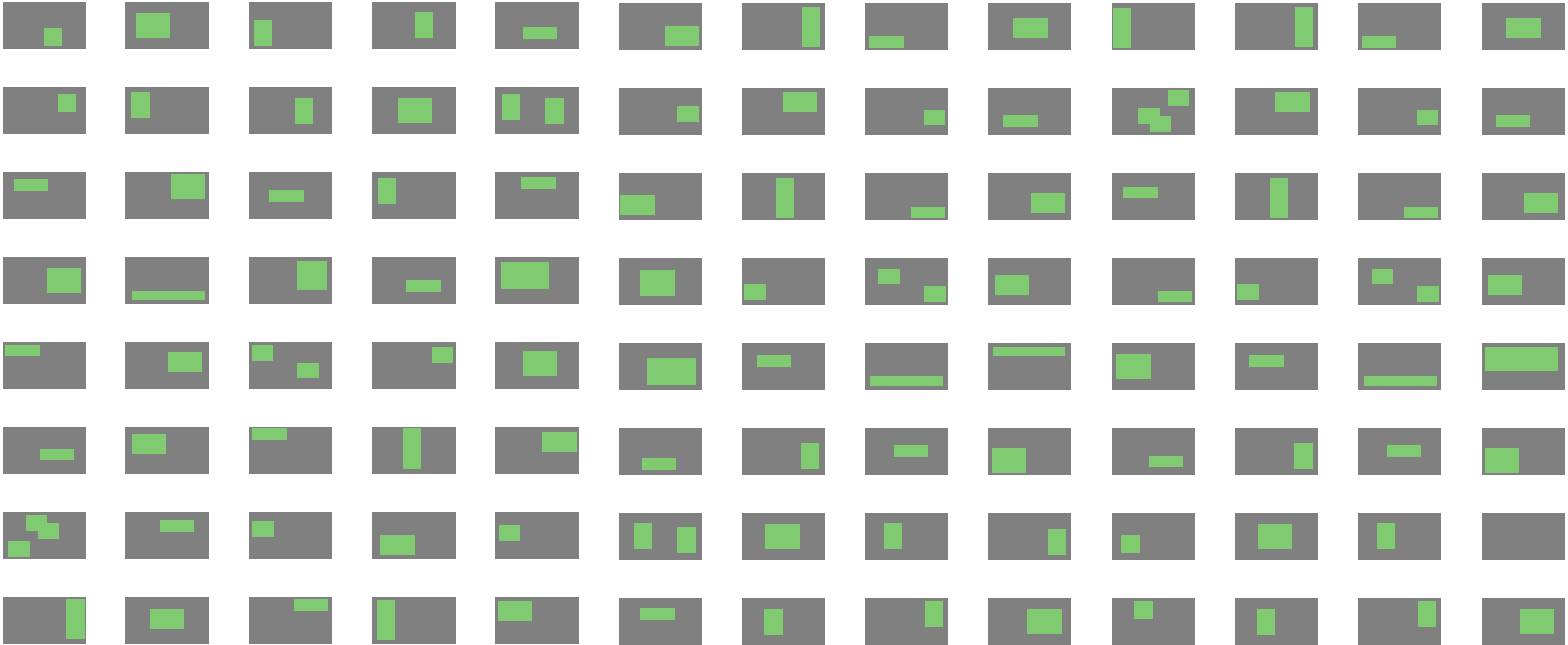




... 4000+ Testfälle
ohne Überdeckung



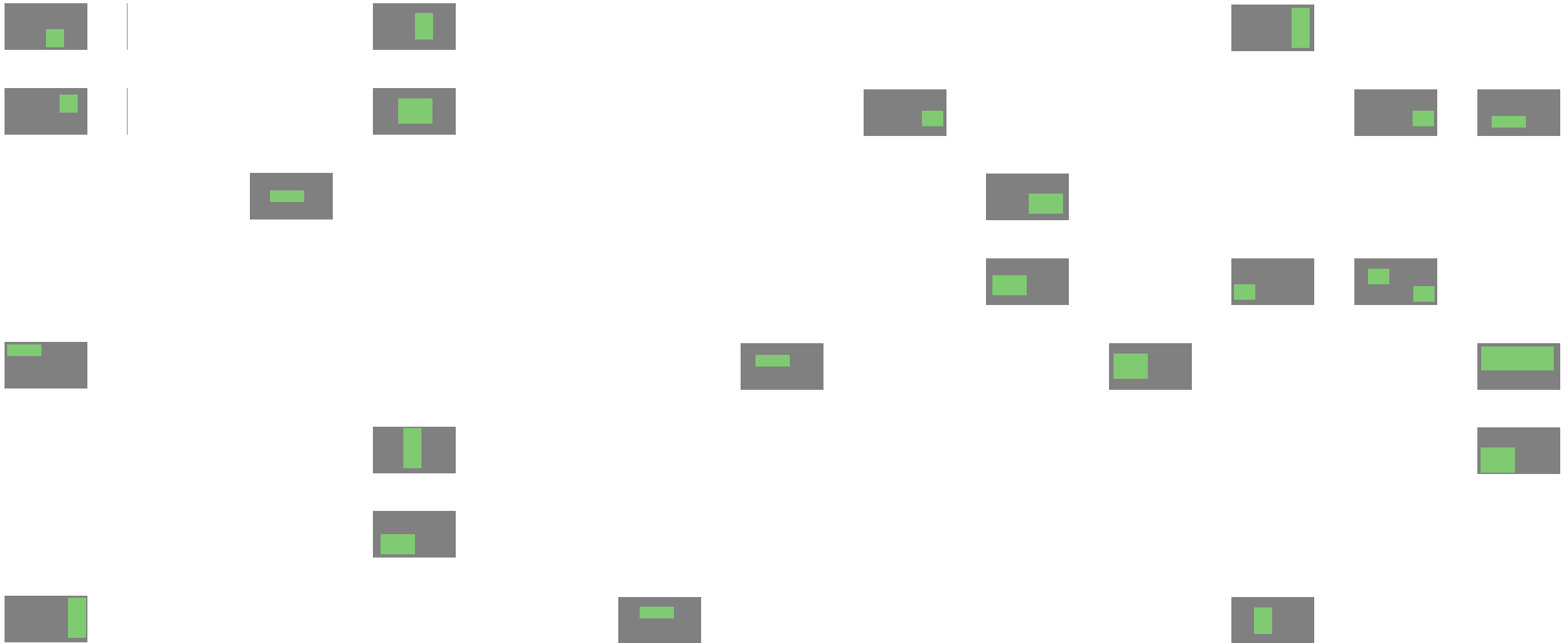
Test-Impact- Analyse



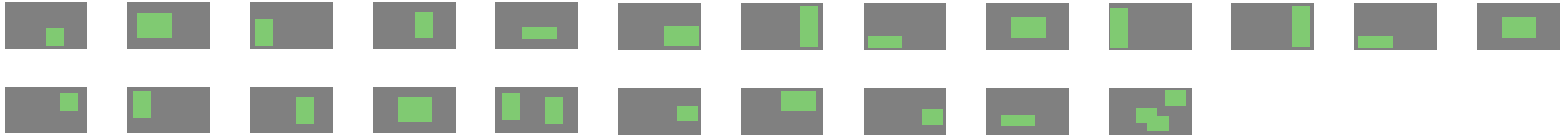
Schritt 1: Selektion betroffener Testfälle



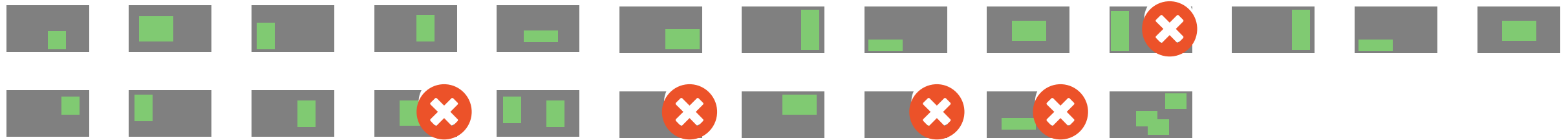
Schritt 1: Selektion betroffener Testfälle



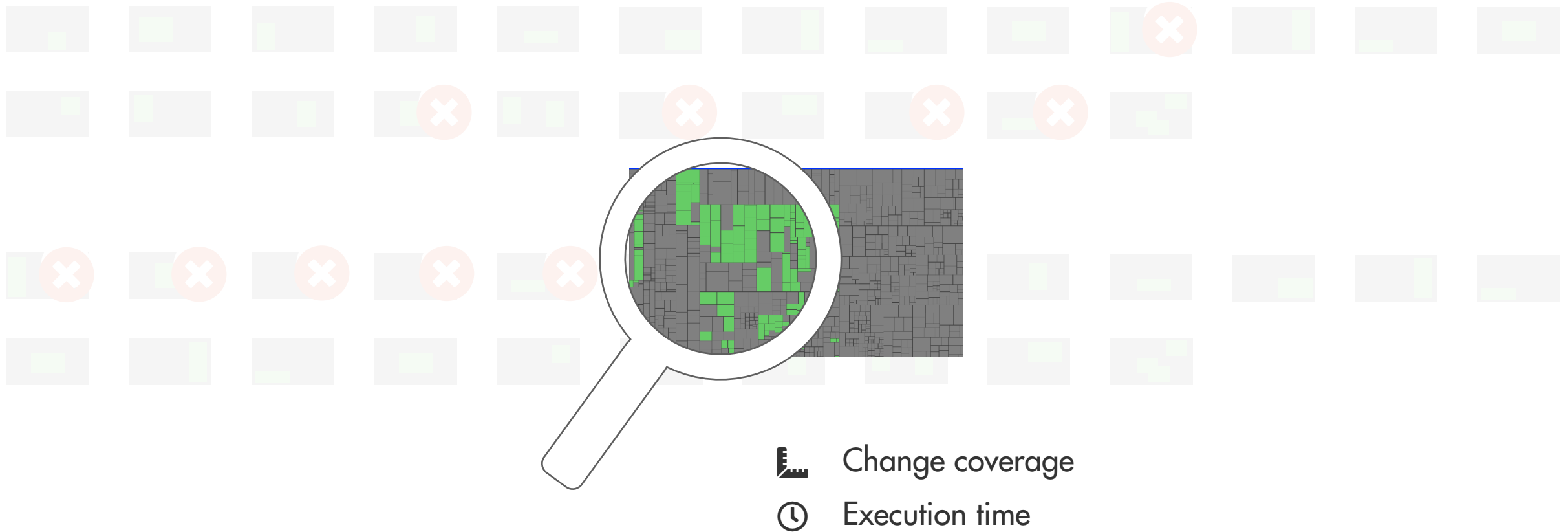
Schritt 1: Selektion betroffener Testfälle



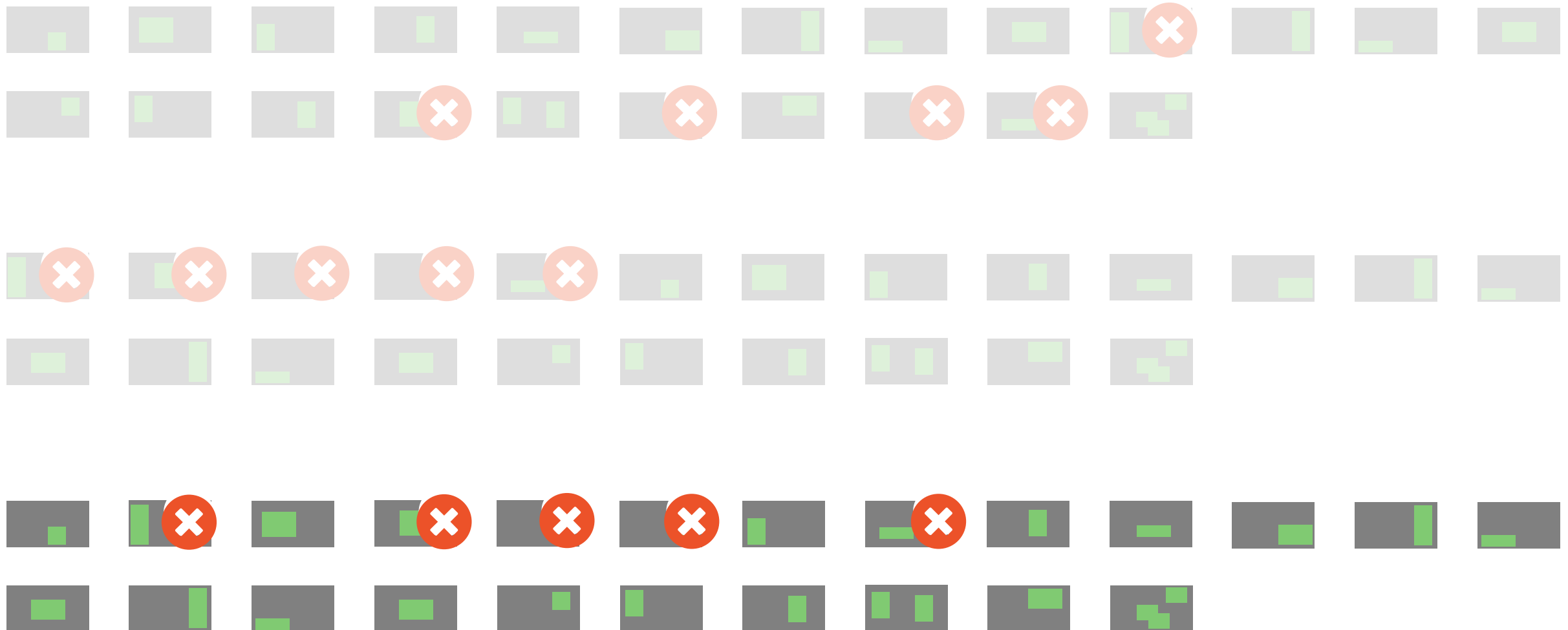
Schritt 2: Priorisierung selektierter Testfälle

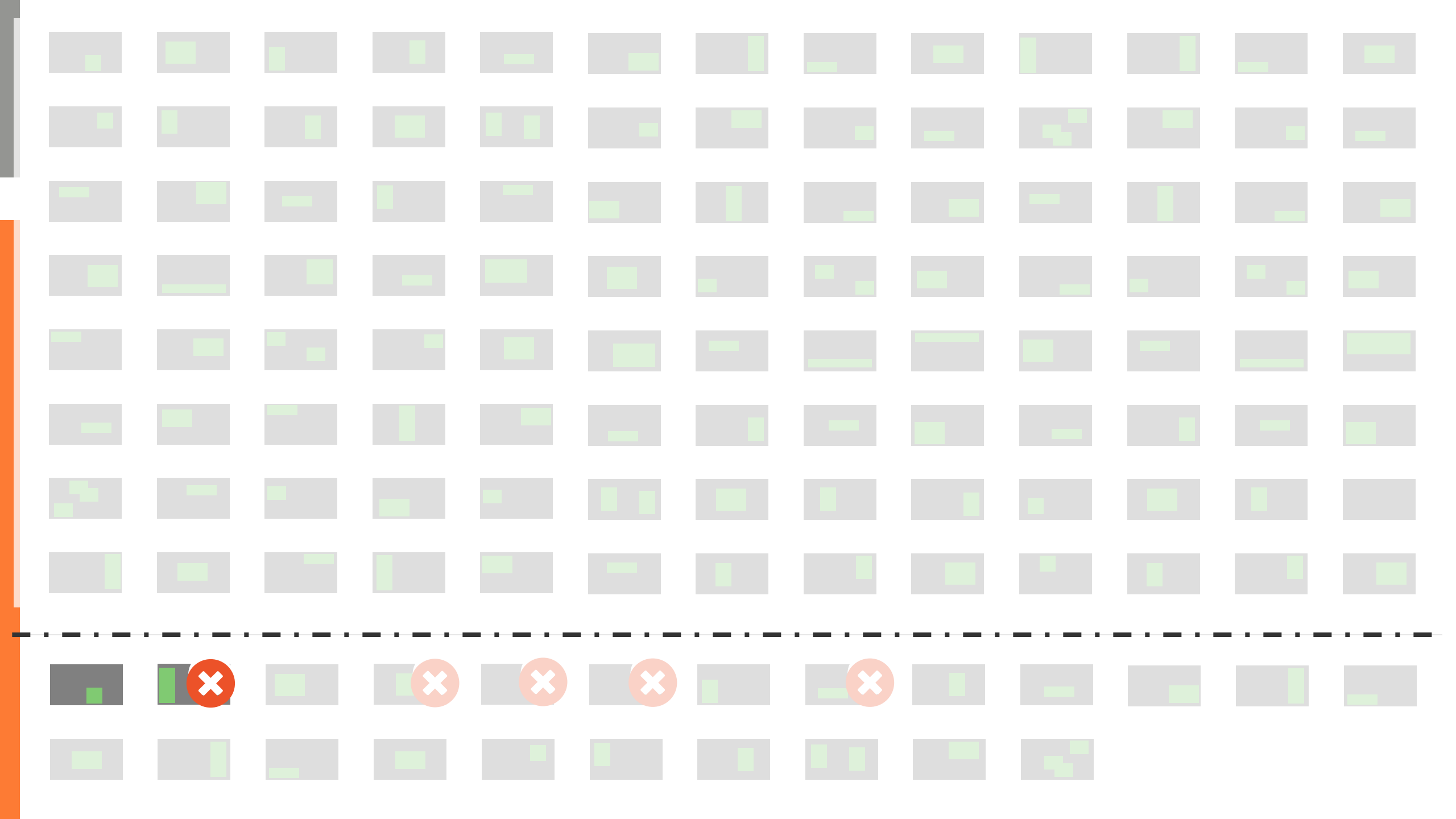


Schritt 2: Priorisierung selektierter Testfälle

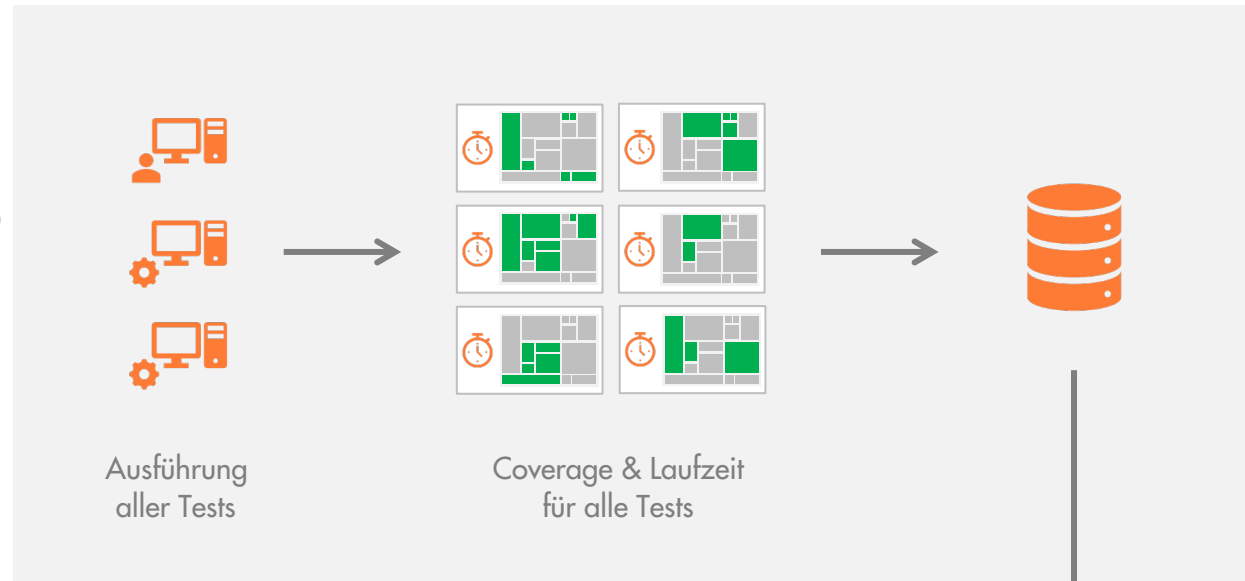


Schritt 2: Priorisierung selektierter Testfälle





Initiale Aufzeichnung aller Tests



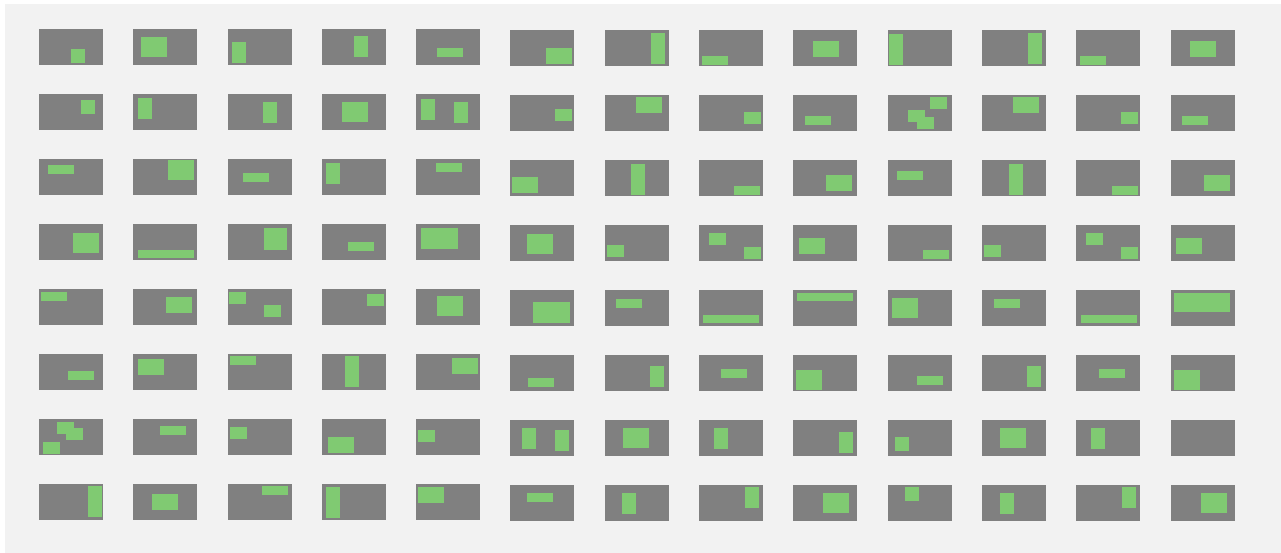
Testausführung nach Änderungen



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



Trotzdem: Regelmäßig **alle Tests** laufen lassen



Neue Tests immer ausführen



Geänderte Tests immer ausführen

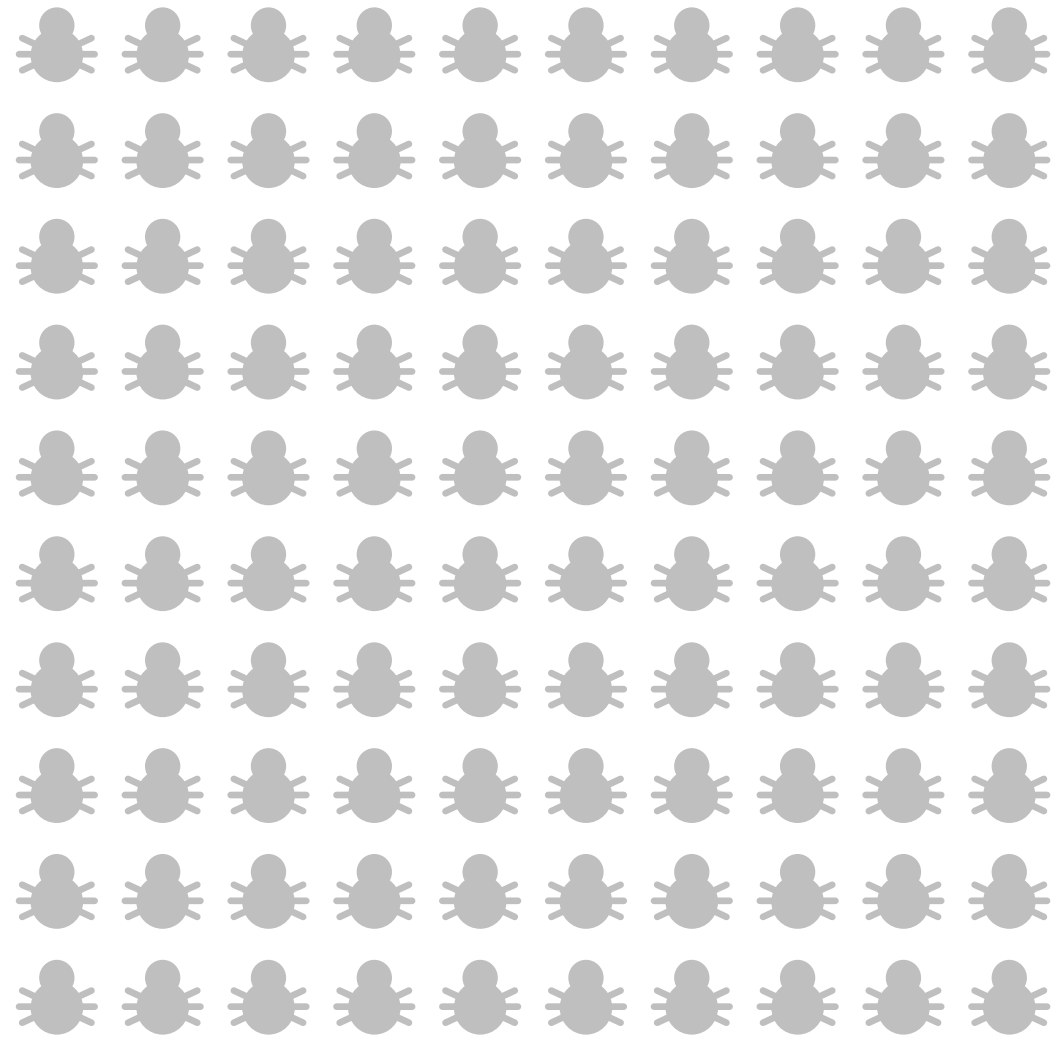
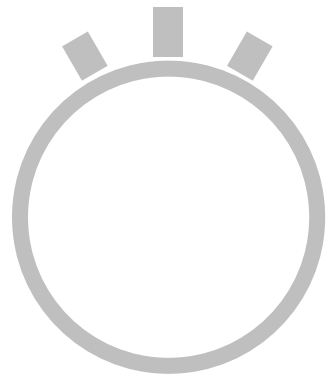


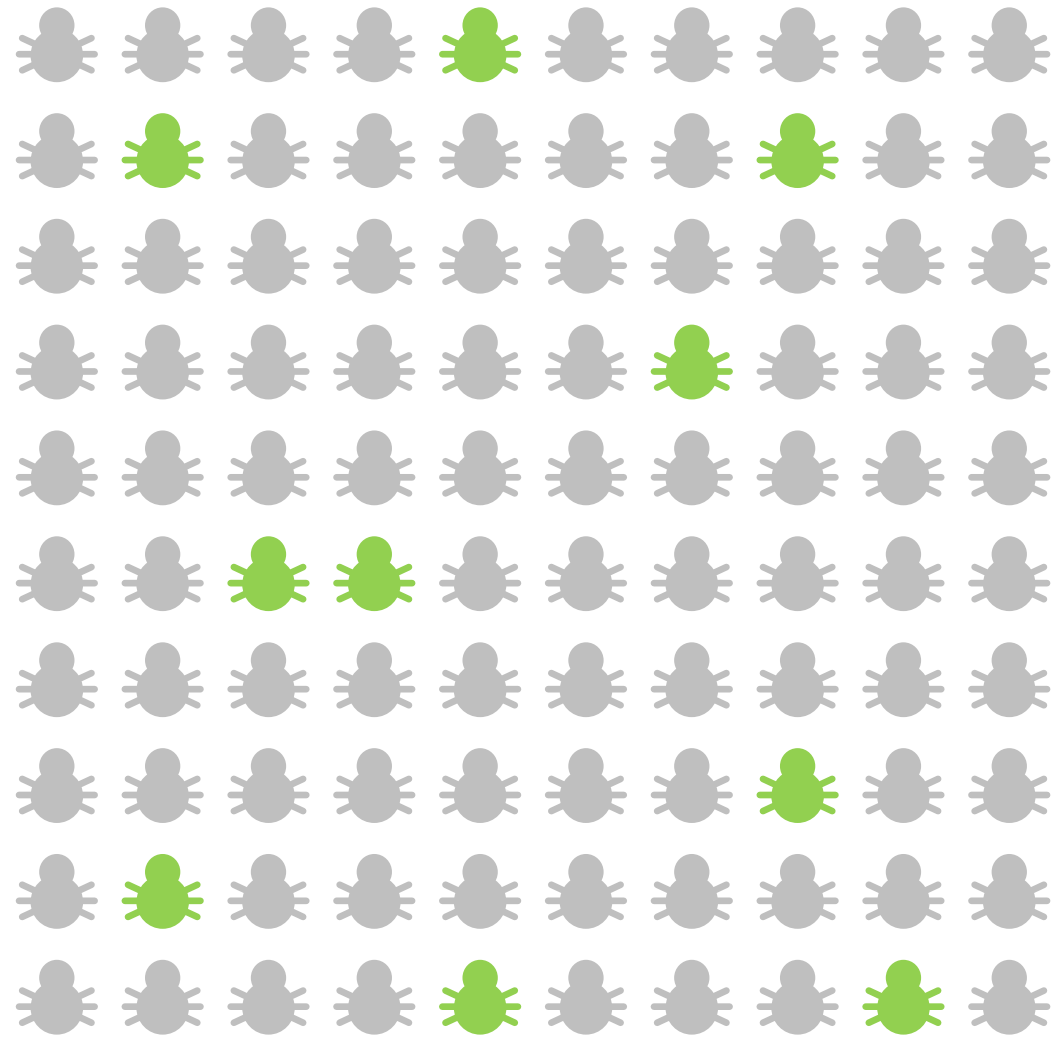
... wenn sich Daten ändern

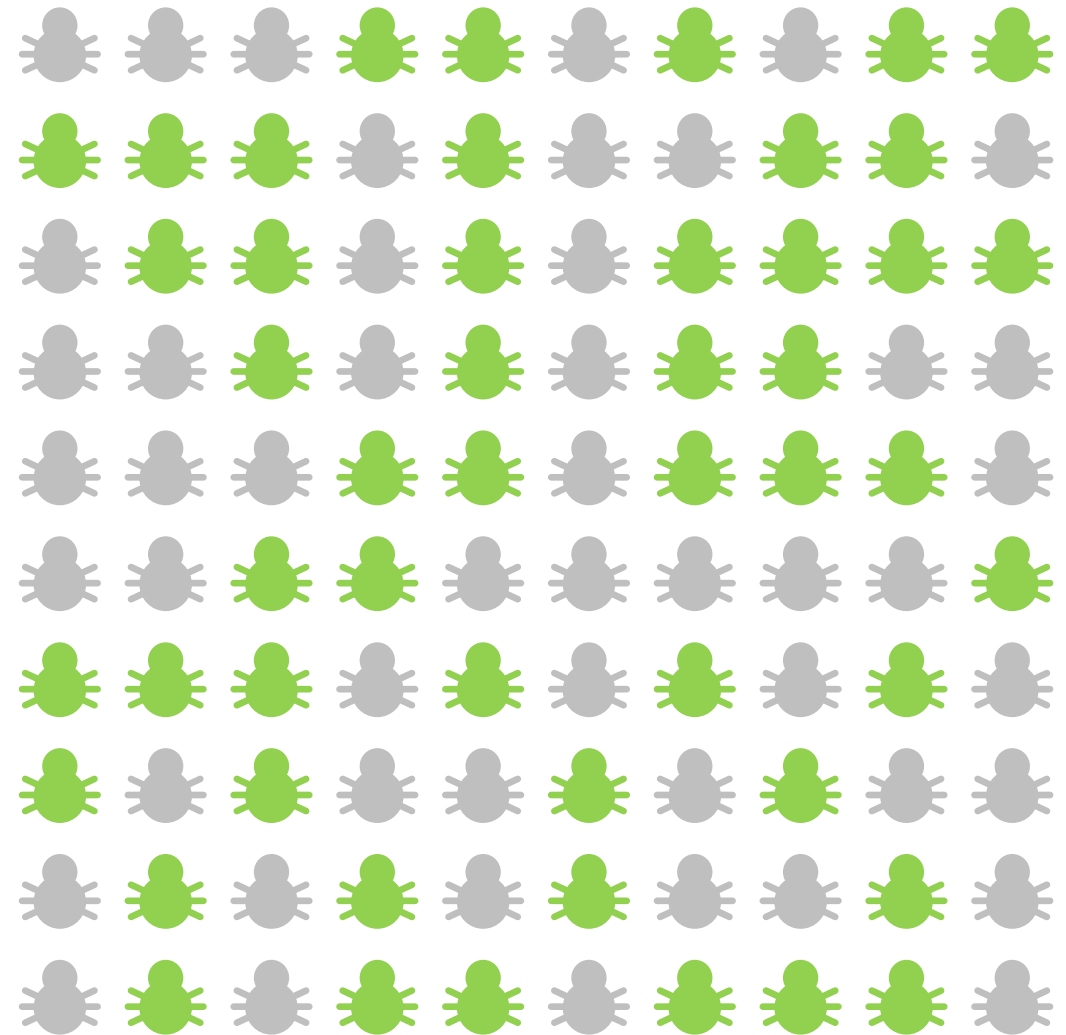


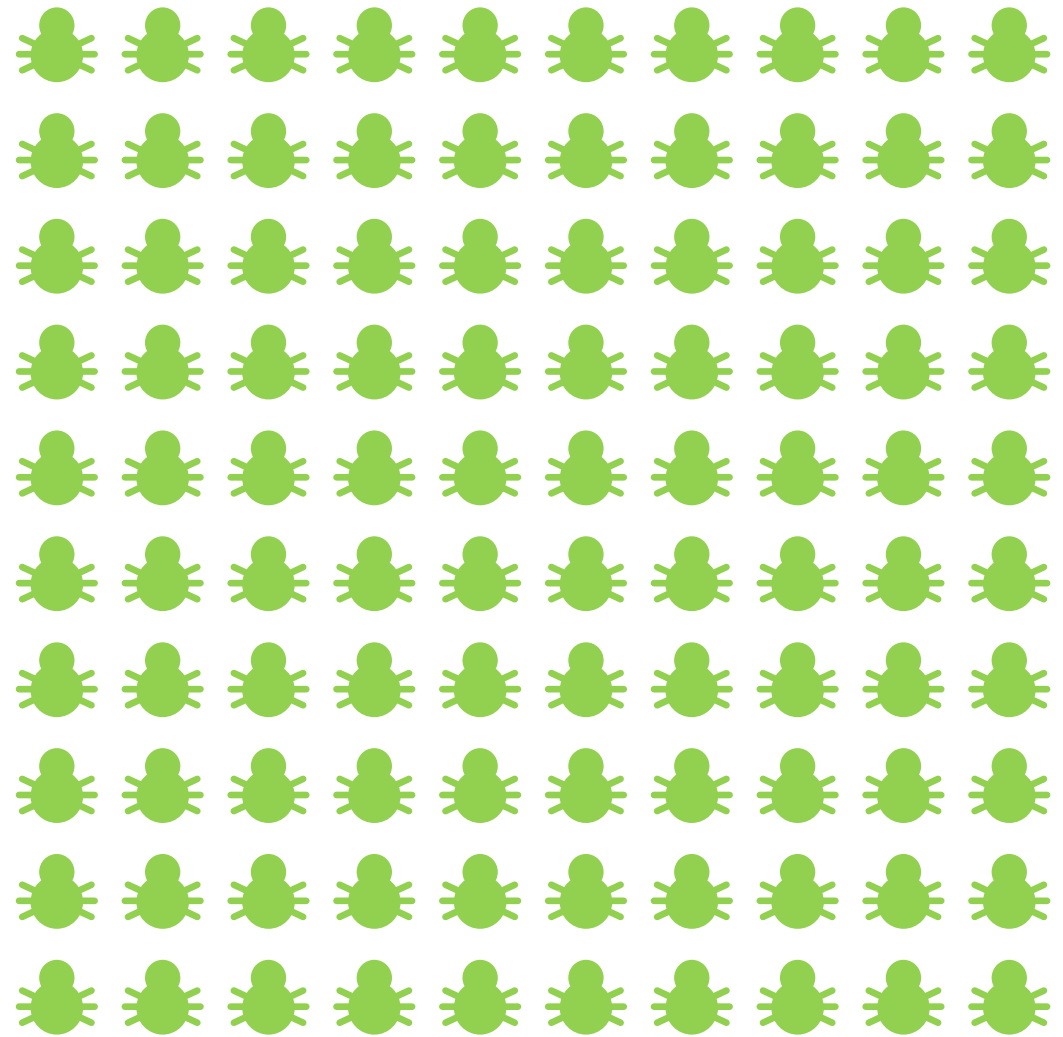
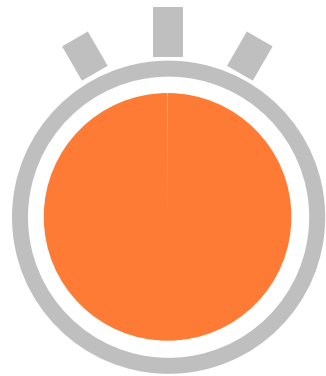
... wenn sich die Konfiguration ändert











Wieviel Zeit kann mit selektierten Tests eingespart werden?

System	Alle Tests [s]	Selektierte Tests [s]	Einsparung
Apache Commons Collections	25	1	98 %
Apache Commons Lang	25	3	88 %
Apache Commons Math	160	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	21 %
RxJava	464	171	63 %
Symia Commons Math Parser	1	1	0 %
Teamscale	1249	197	84 %

Im Durchschnitt
64%
Einsparung

Welcher Anteil fehlerhafter Builds wird
in begrenzter Zeit erkannt?

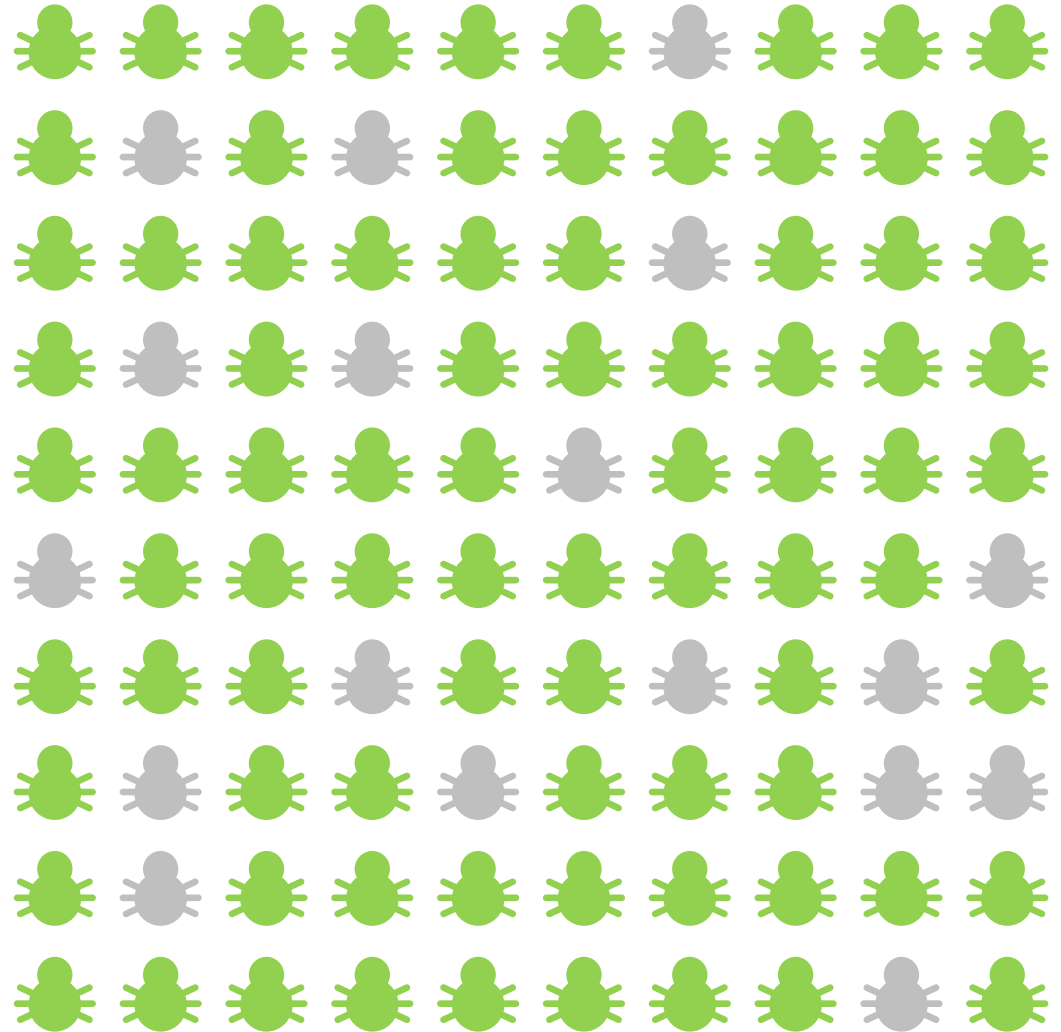
System	1 %	2 %	5 %	10 %
Apache Commons Collections	84 %	97 %	100 %	100 %
Apache Commons Lang	94 %	96 %	98 %	99 %
Apache Commons Math	80 %	86 %	92 %	93 %
Histone Template Engine	88 %	88 %	88 %	88 %
JabRef	96 %	96 %	96 %	96 %
Joda-Time	98 %	99 %	99 %	100 %
Lightweight-Stream-API	61 %	91 %	95 %	97 %
LittleProxy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

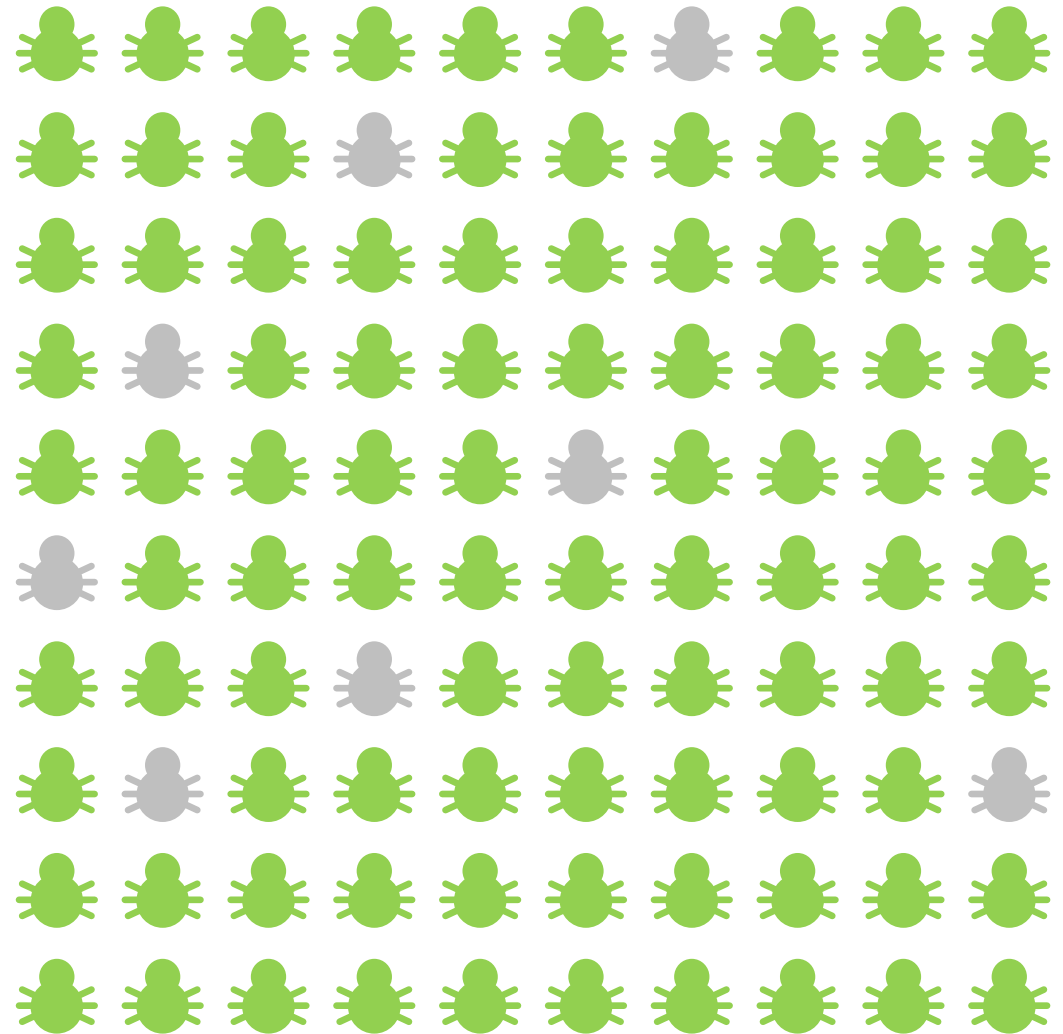


1 %



80 %

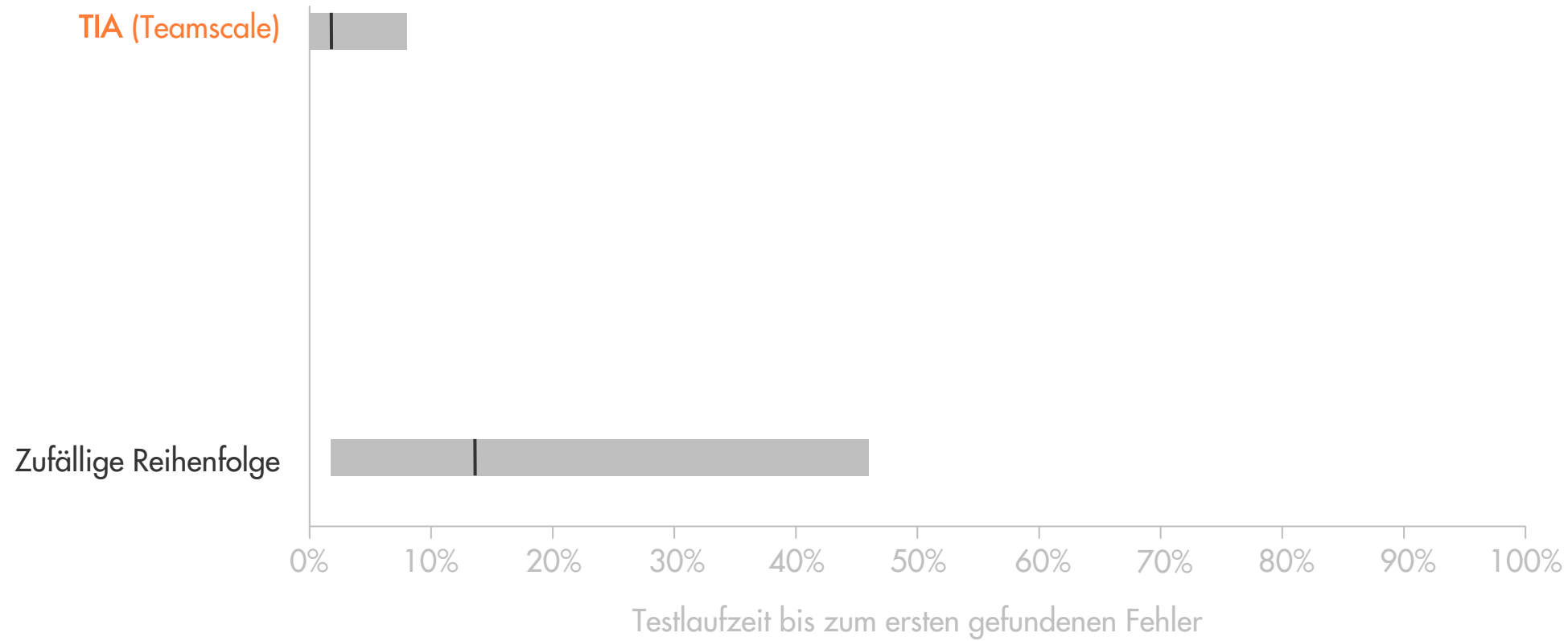




Gilt das auch für echte Fehler in Systemen?



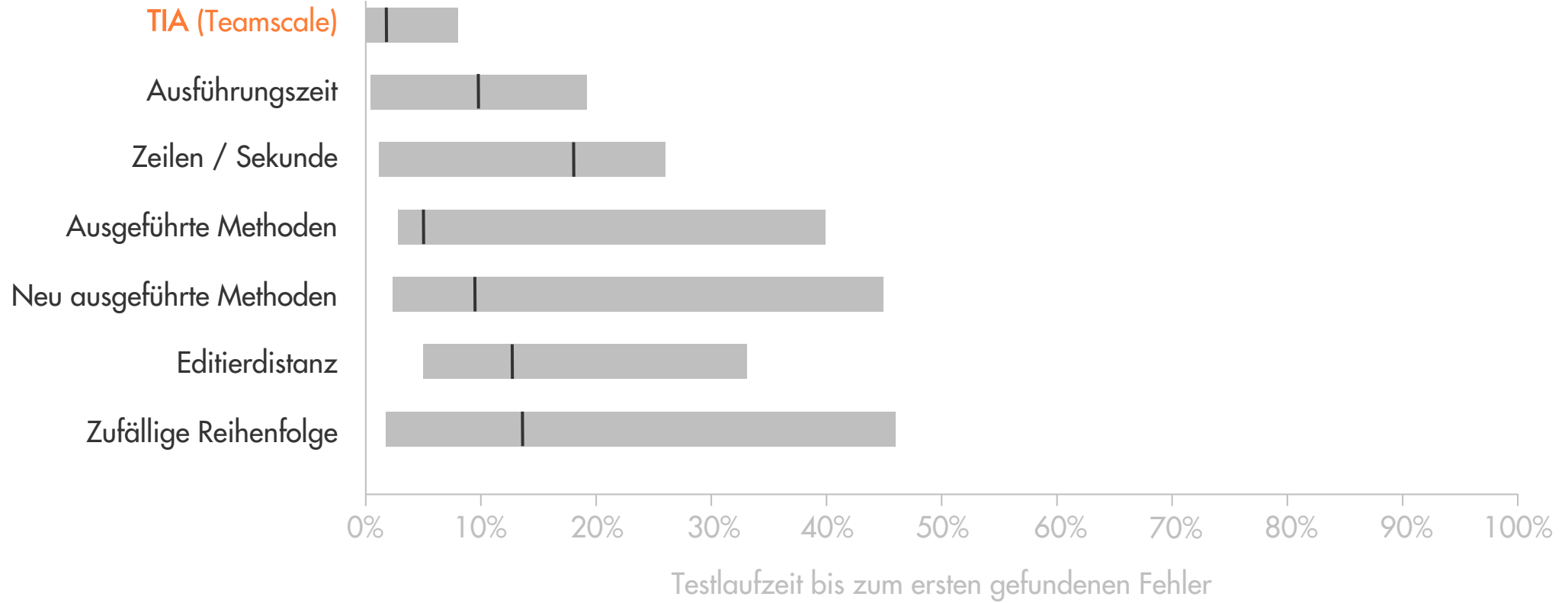
100+



Gibt es nicht einfachere Verfahren?



100+



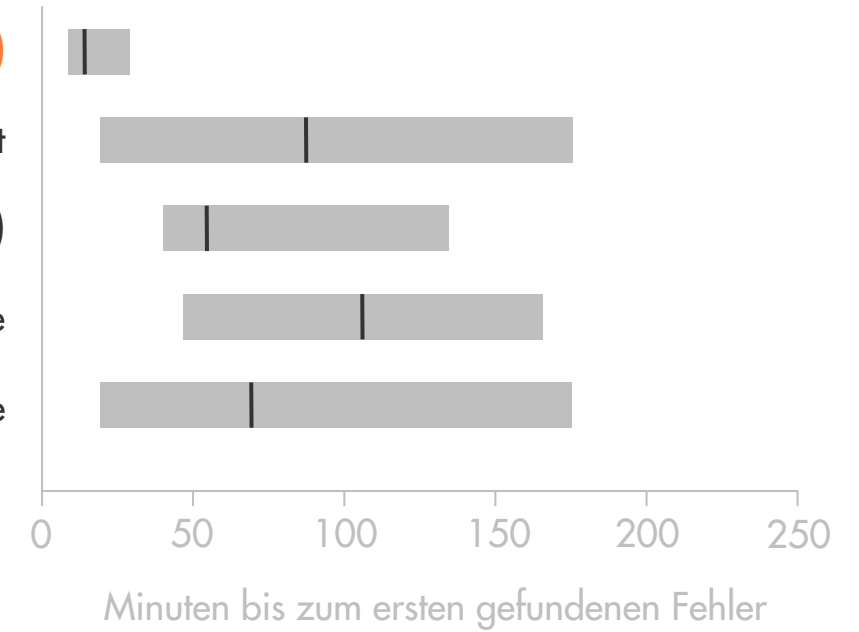
HIL-Test





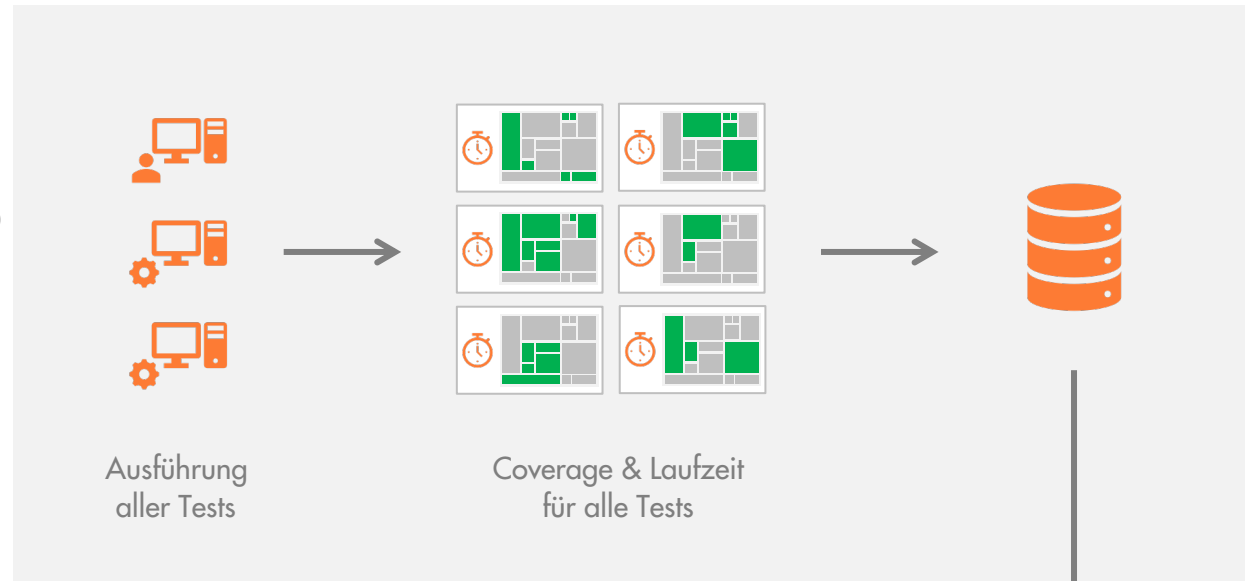
TIA (Teamscale)

Ausführungszeit
Ausführungszeit (absteigend)
Methoden / Sekunde
Zufällige Reihenfolge



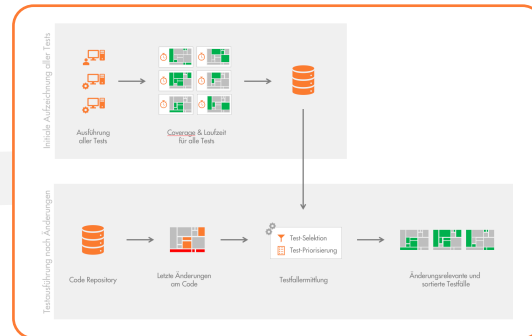
Test  ntelligence

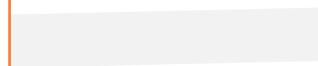
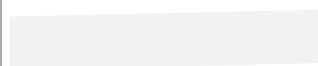
Initiale Aufzeichnung aller Tests



Testausführung nach Änderungen



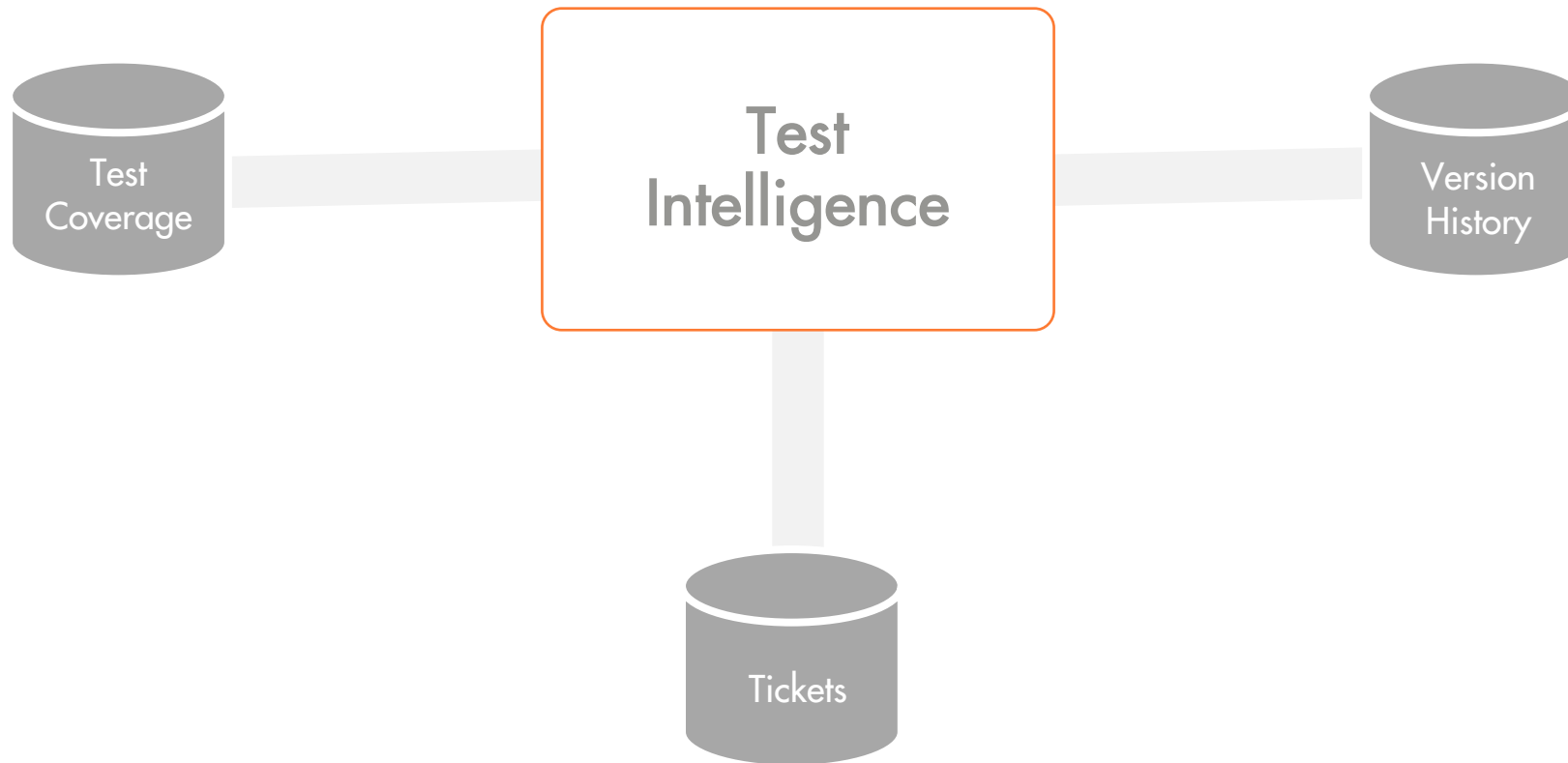




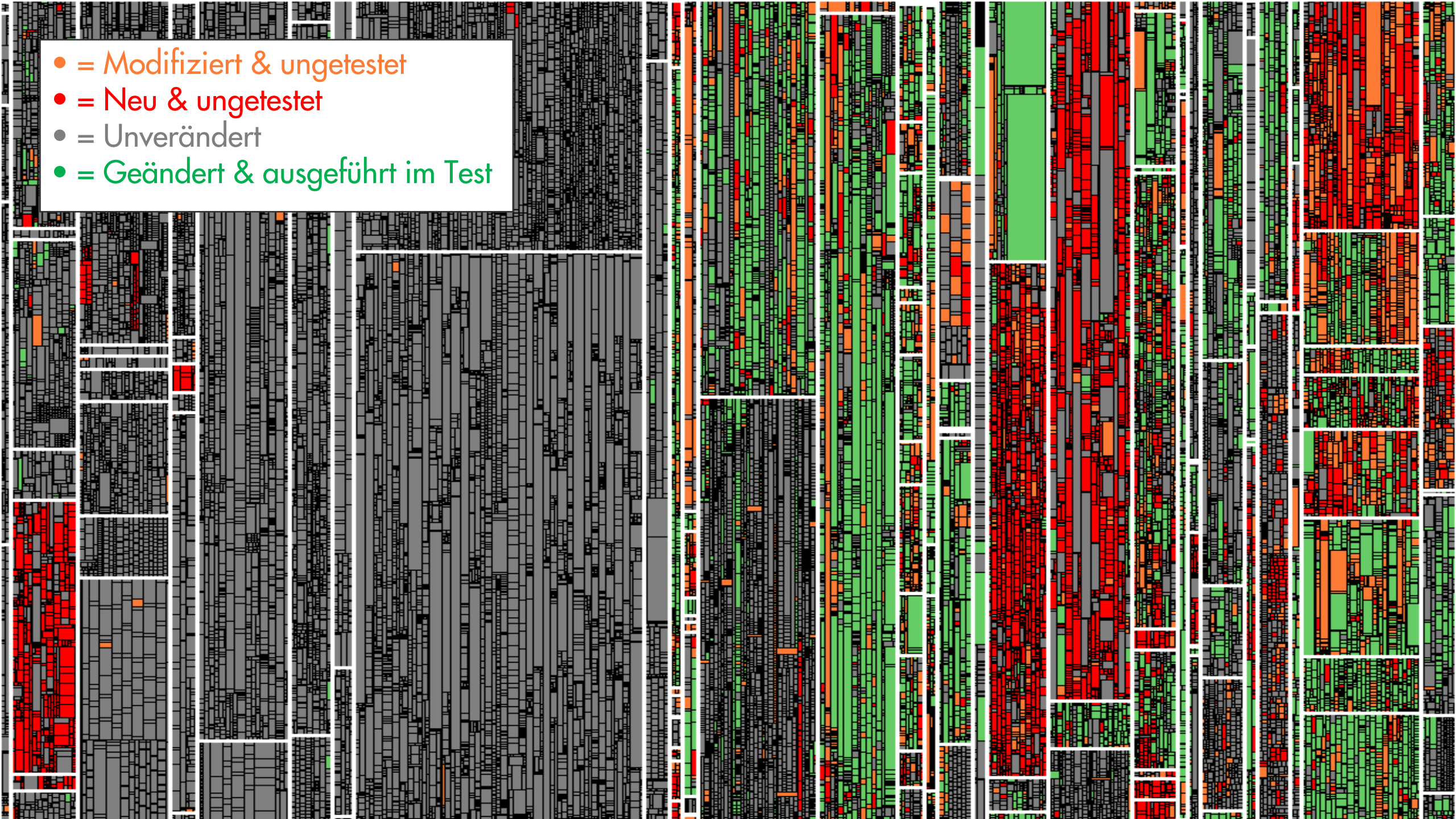


Test
Intelligence





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



Issue # ▼	Subject			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%

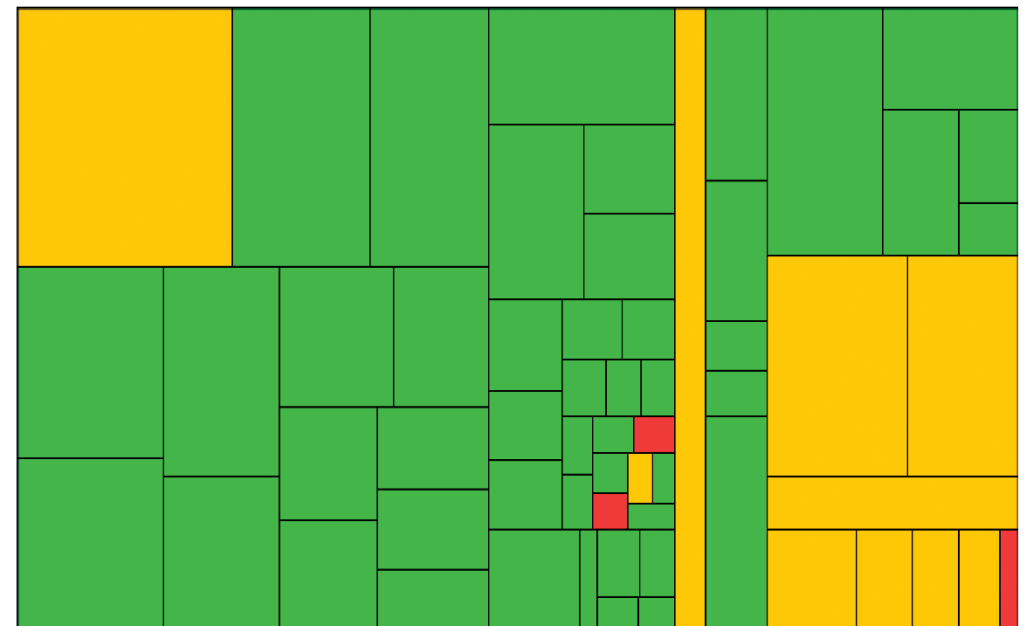
Done **Issue TS-14733 - Remove Dataflow blocks**

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%



Kontakt



Dr. Sven Amann · amann@cqse.eu · +49 172 1860063

CQSE GmbH
Centa-Hafenbrädl-Str. 59
81249 München