

Müssen wir wirklich schon wieder alles testen?

Trotz langlaufender Testsuiten Fehler schnell und zuverlässig aufdecken

Dr. Jan Harder
Dr. Nils Göde

CQSE



Praxis

Software-**Audits**

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

 **Teamscale**

CQSE GmbH



Forschung

13+ **Promotionen** in
Software Engineering

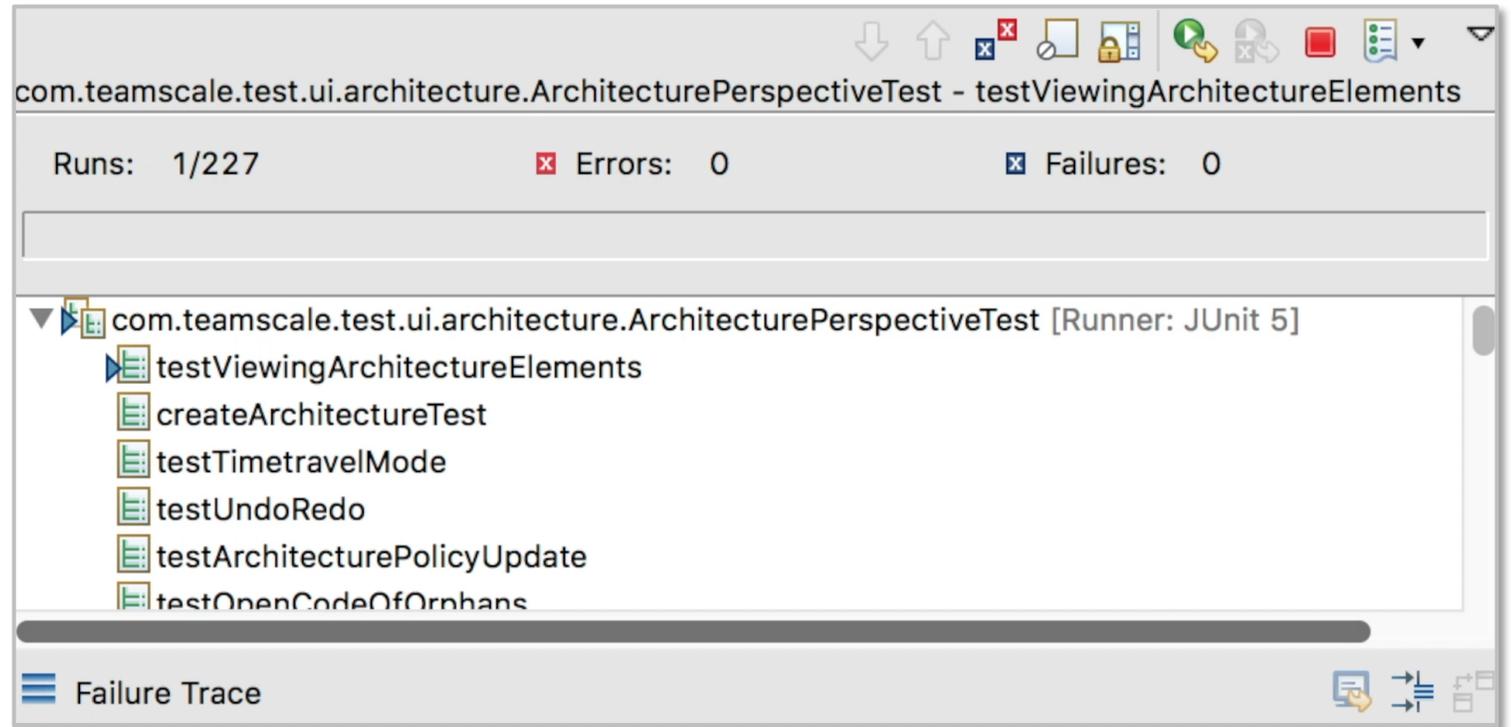
Eigene **Forschung**

Enger Kontakt zu
Universitäten

 Release Early & Often

 Fail Fast

 Instant Feedback



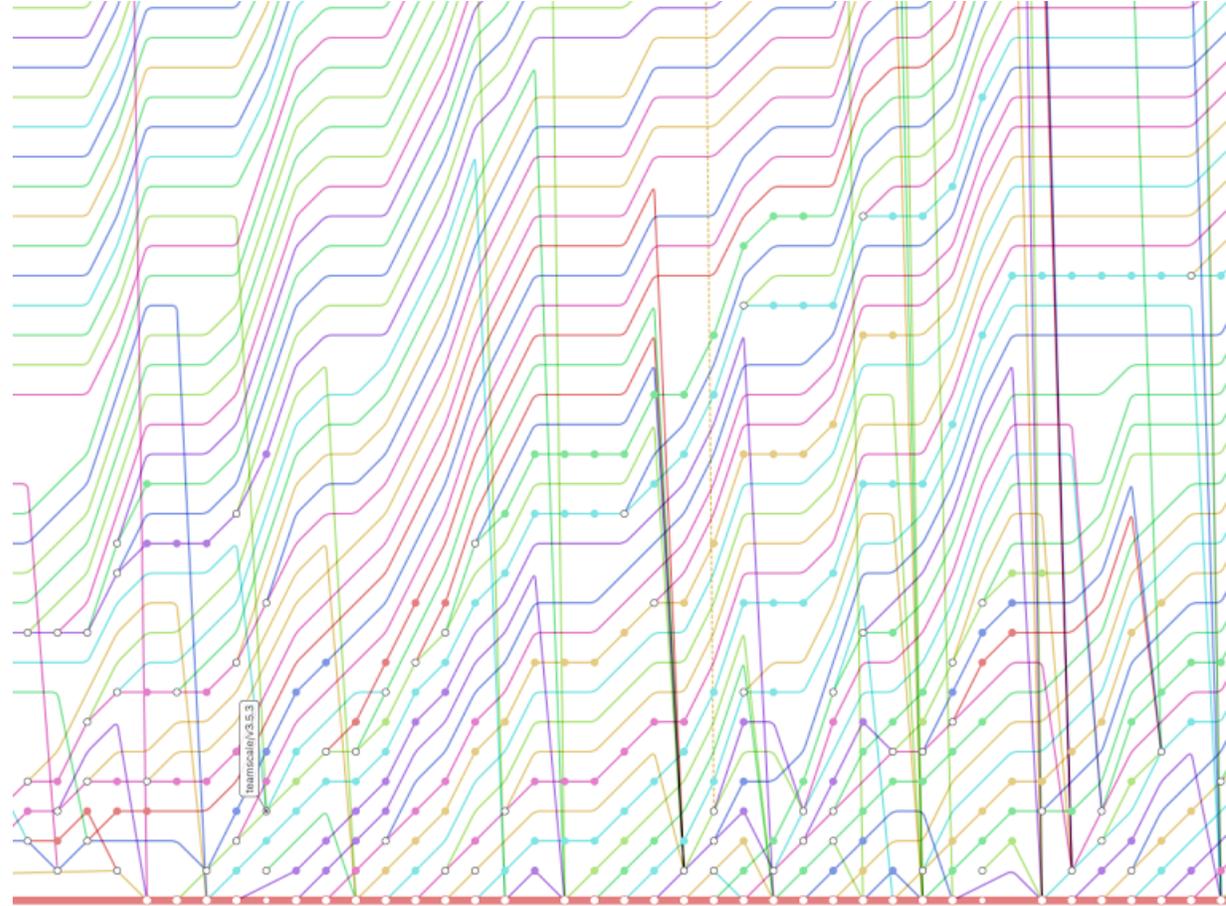
The screenshot shows a test runner window with the following details:

- Test Name: `com.teamscale.test.ui.architecture.ArchitecturePerspectiveTest - testViewingArchitectureElements`
- Runs: 1/227
- Errors: 0
- Failures: 0
- Test Suite: `com.teamscale.test.ui.architecture.ArchitecturePerspectiveTest [Runner: JUnit 5]`
- Test Methods:
 - `testViewingArchitectureElements` (passed)
 - `createArchitectureTest` (passed)
 - `testTimetravelMode` (passed)
 - `testUndoRedo` (passed)
 - `testArchitecturePolicyUpdate` (passed)
 - `testOpenCodeOfOrphans` (passed)
- Bottom Panel: Failure Trace (empty)

Wie lange laufen Ihre Tests?



Eine Stunde?





Eine Stunde?



Eine Nacht?



Eine Stunde?



Eine Nacht?



Eine Woche?



Eine Stunde?



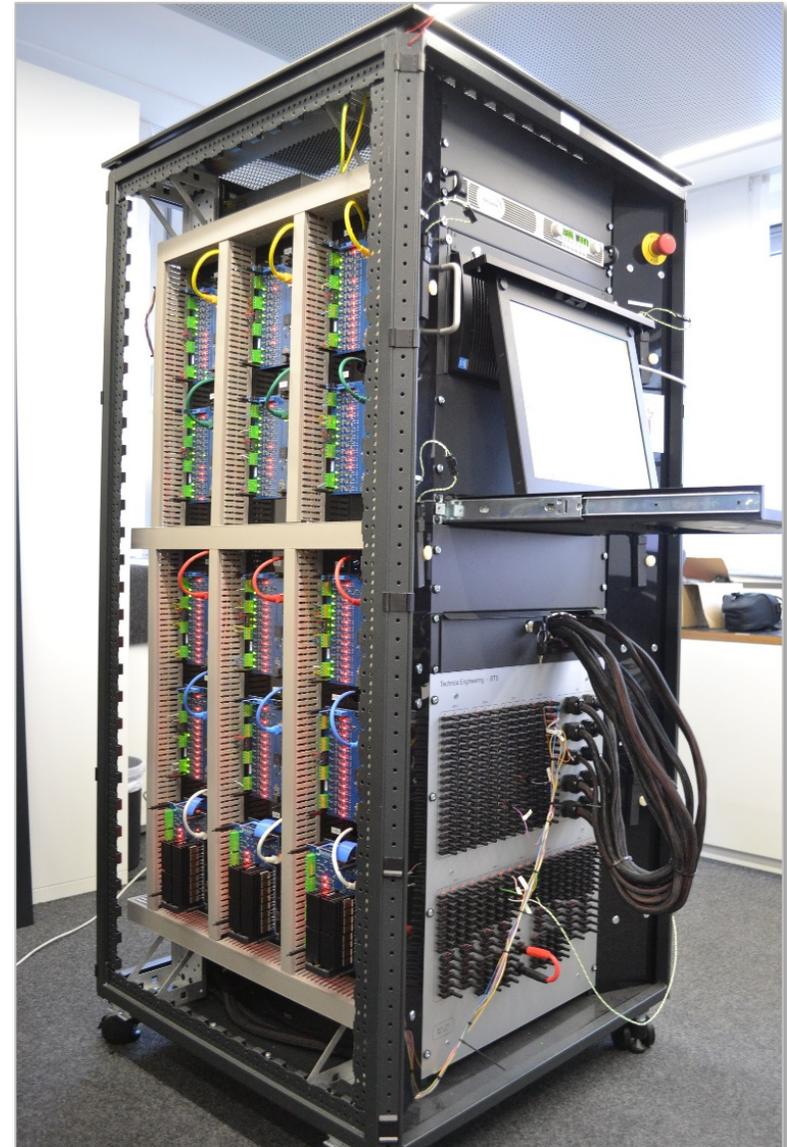
Eine Nacht?



Eine Woche?

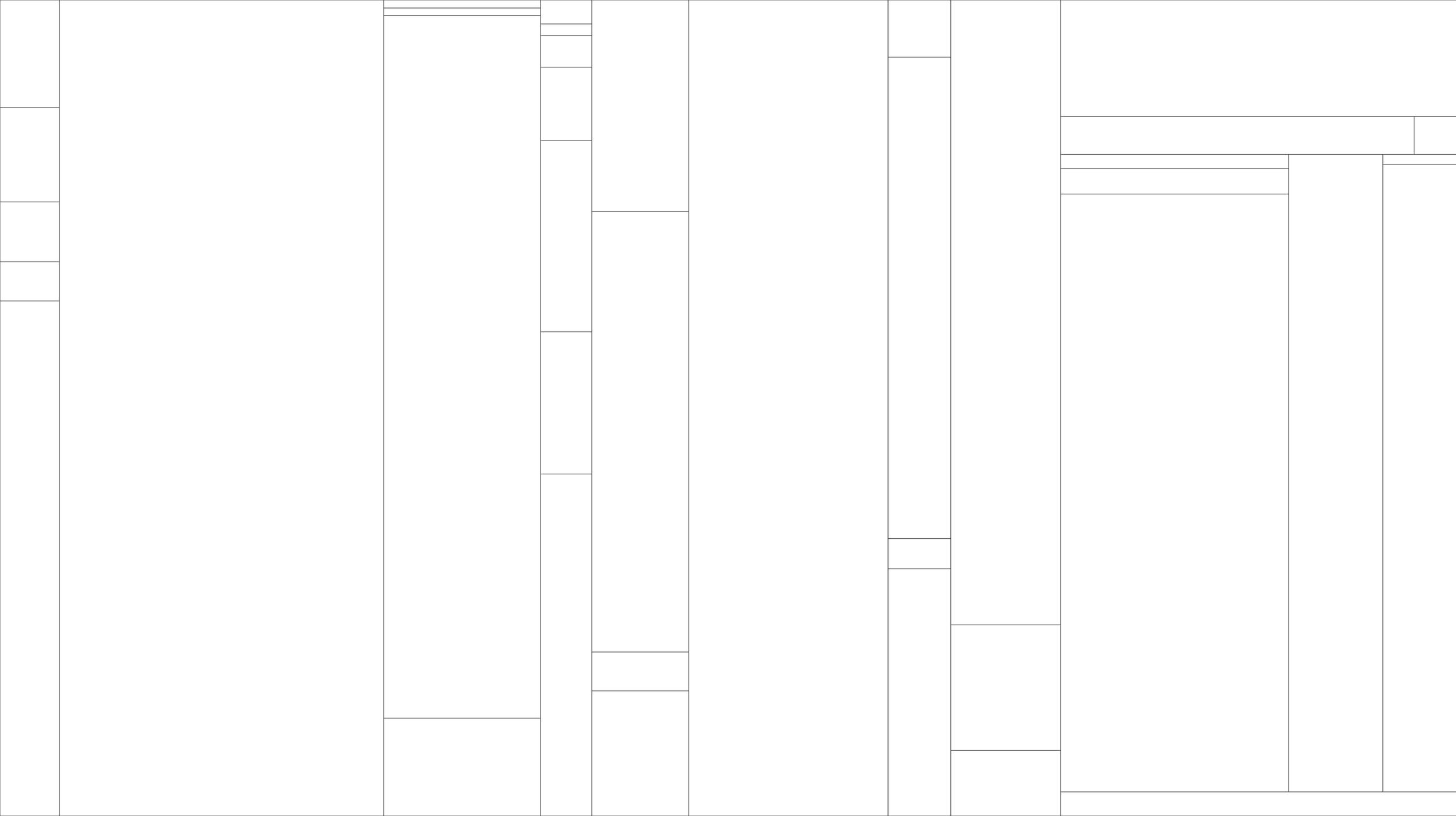


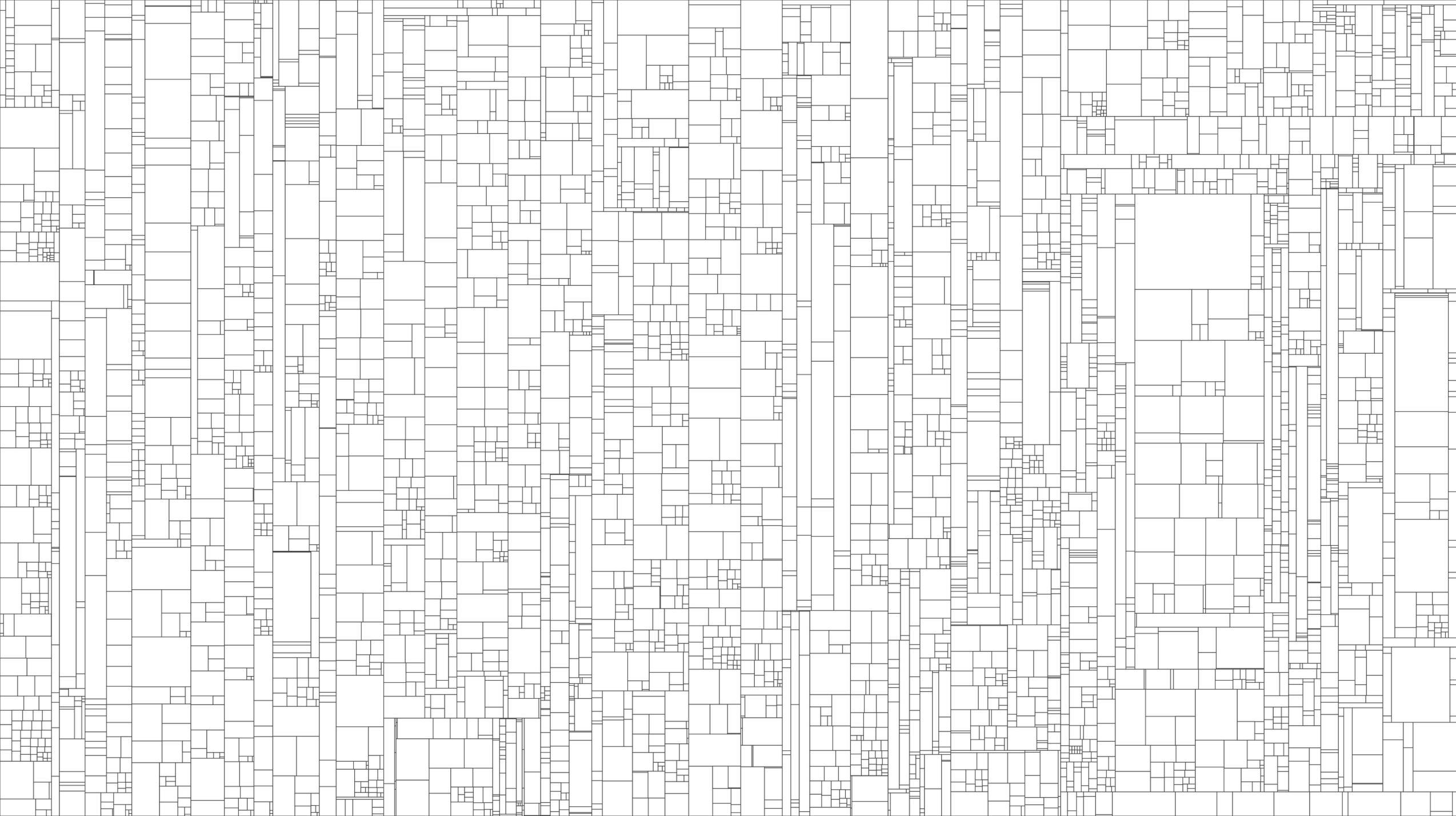
Einen Monat?

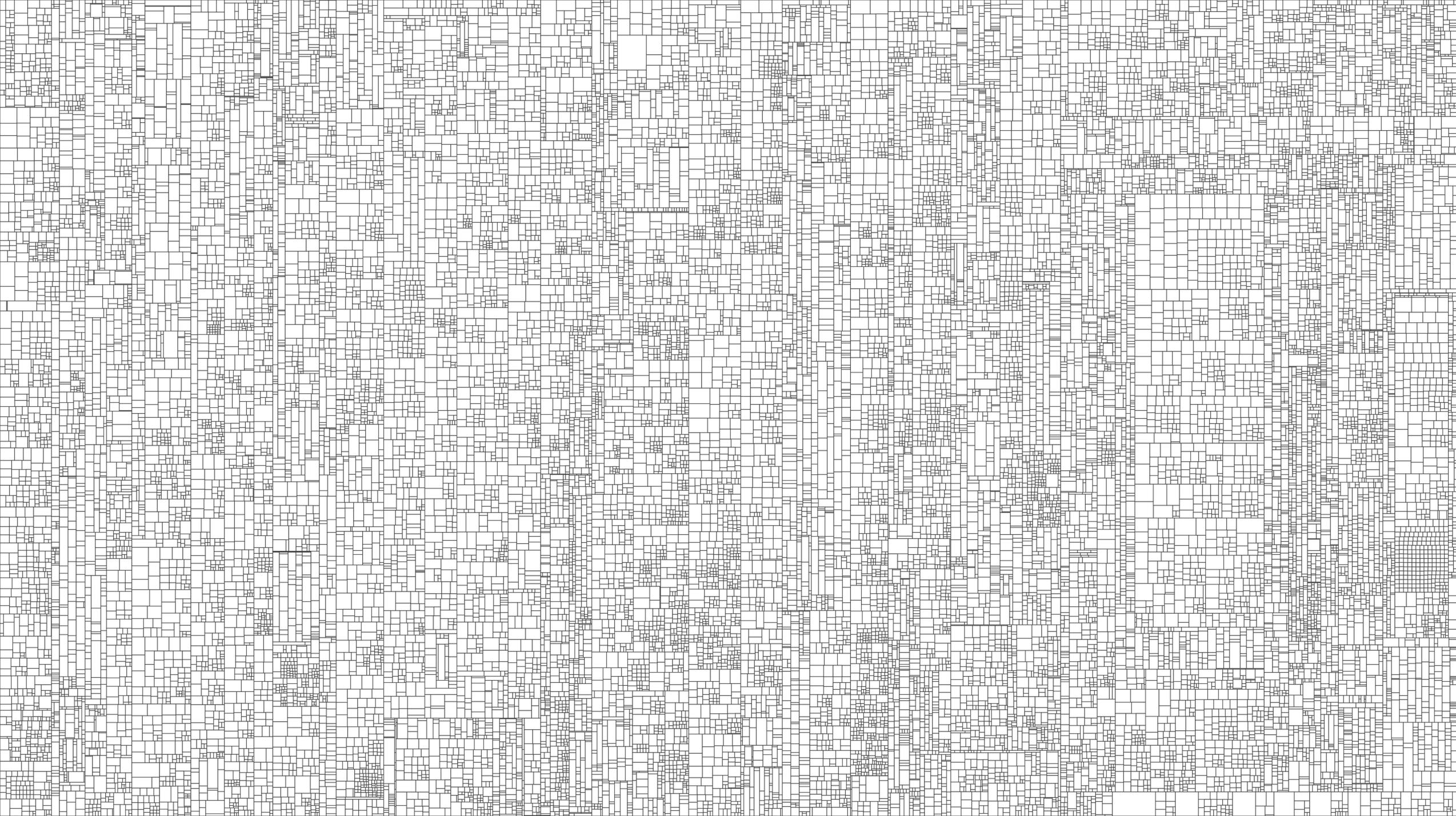


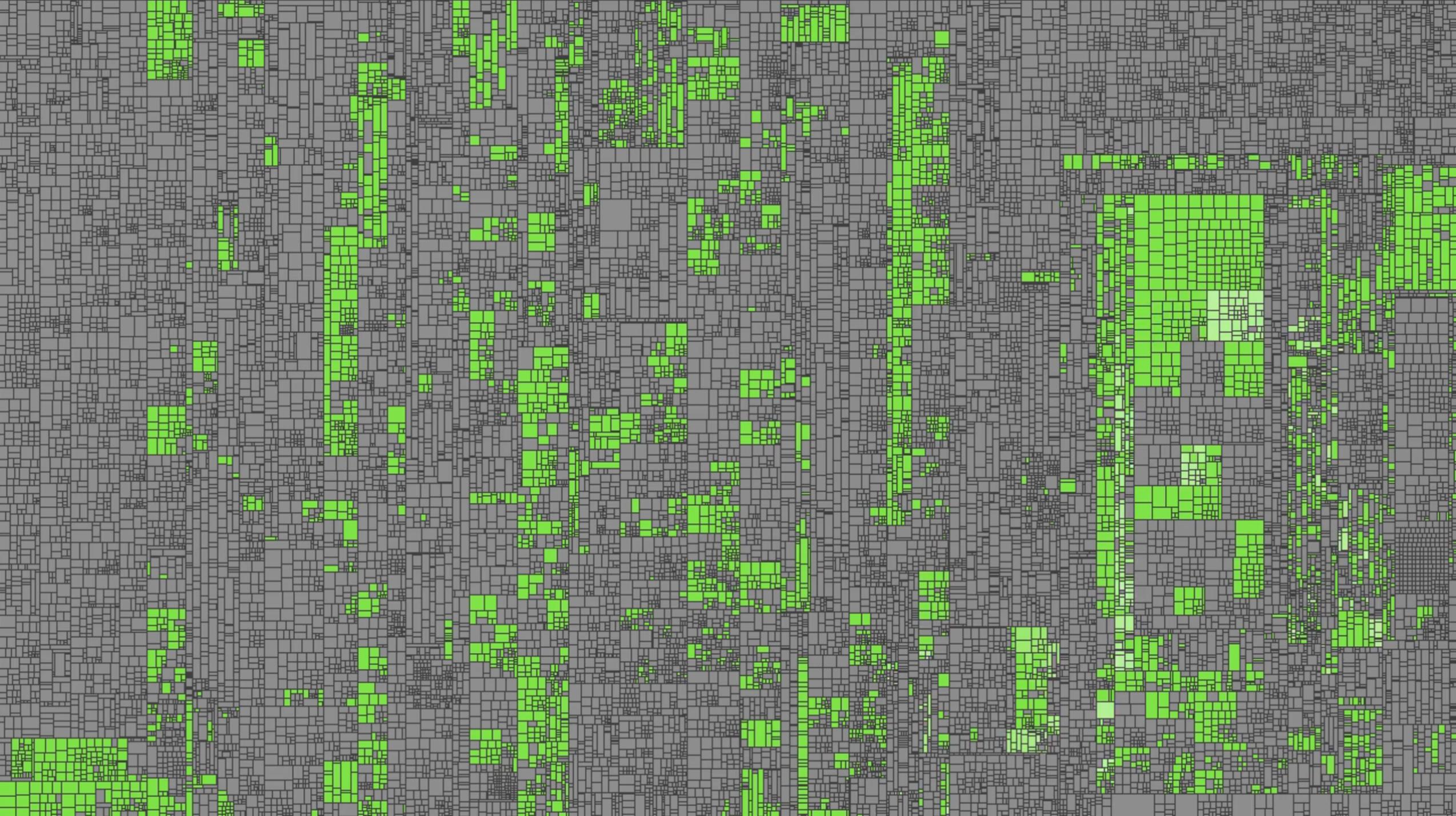
<https://www.technica-engineering.de/produkte/bis-body-electronic-test-system>

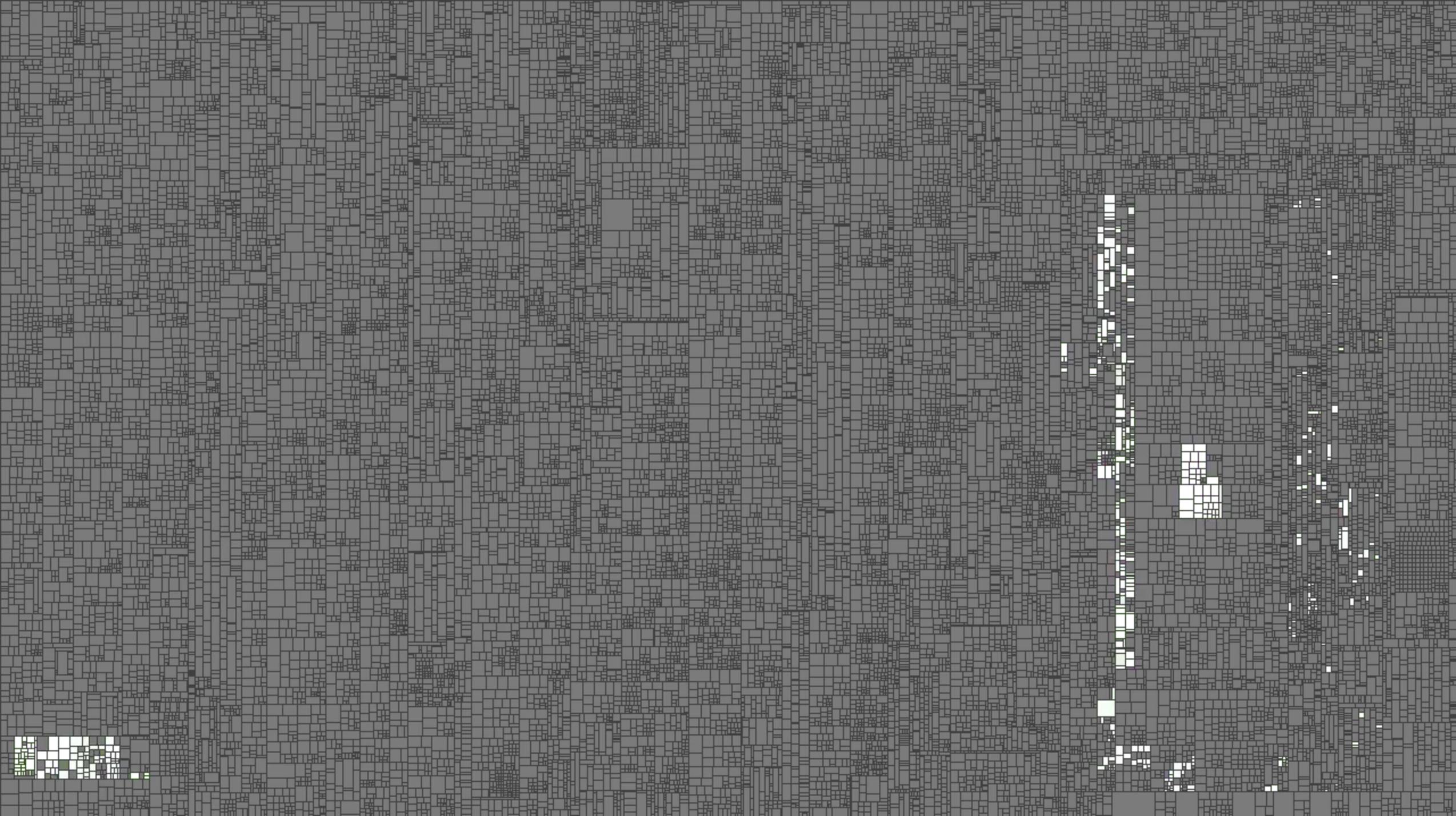
Kann man das besser machen?

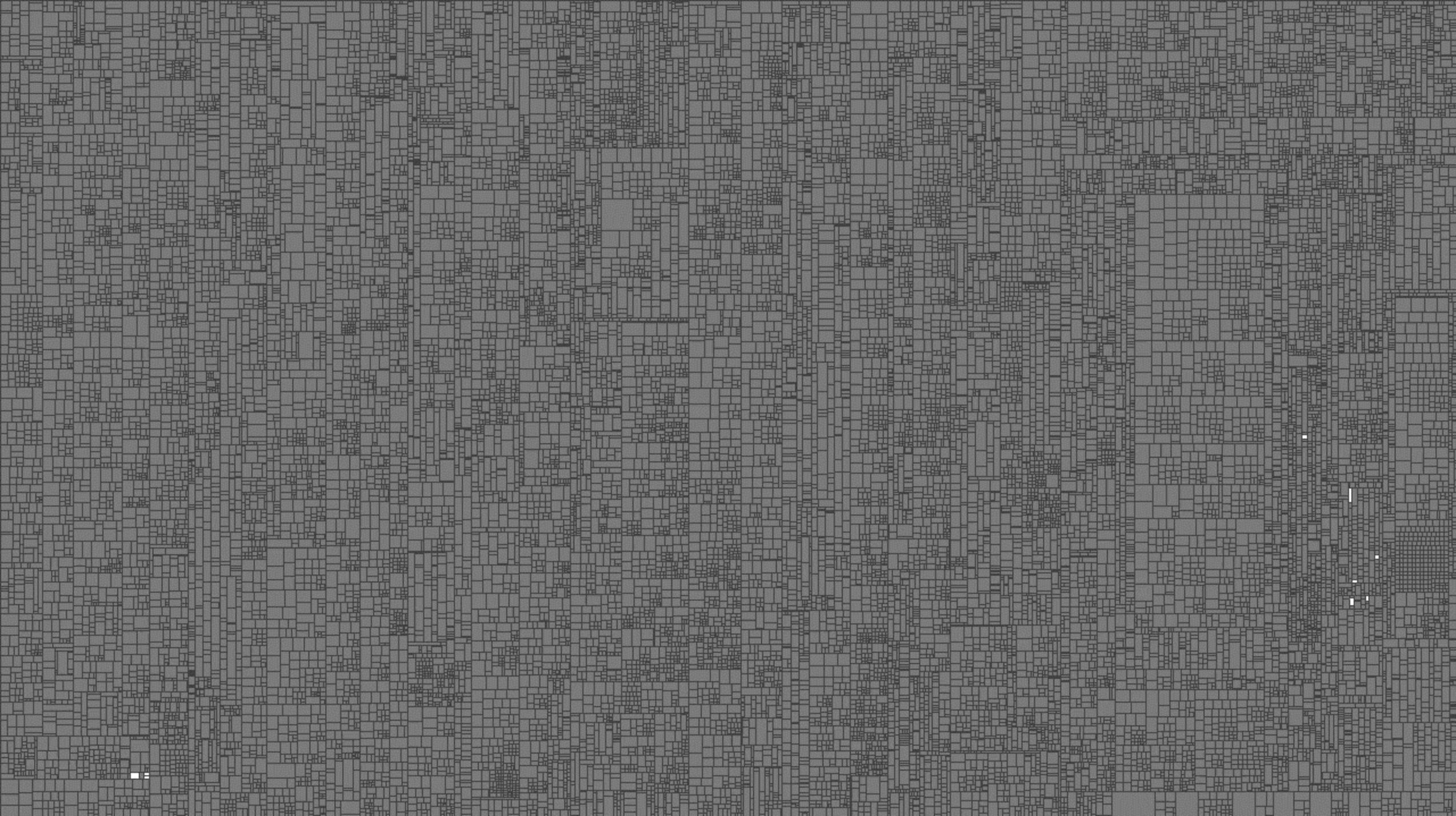














Teamscale / TS-10829

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

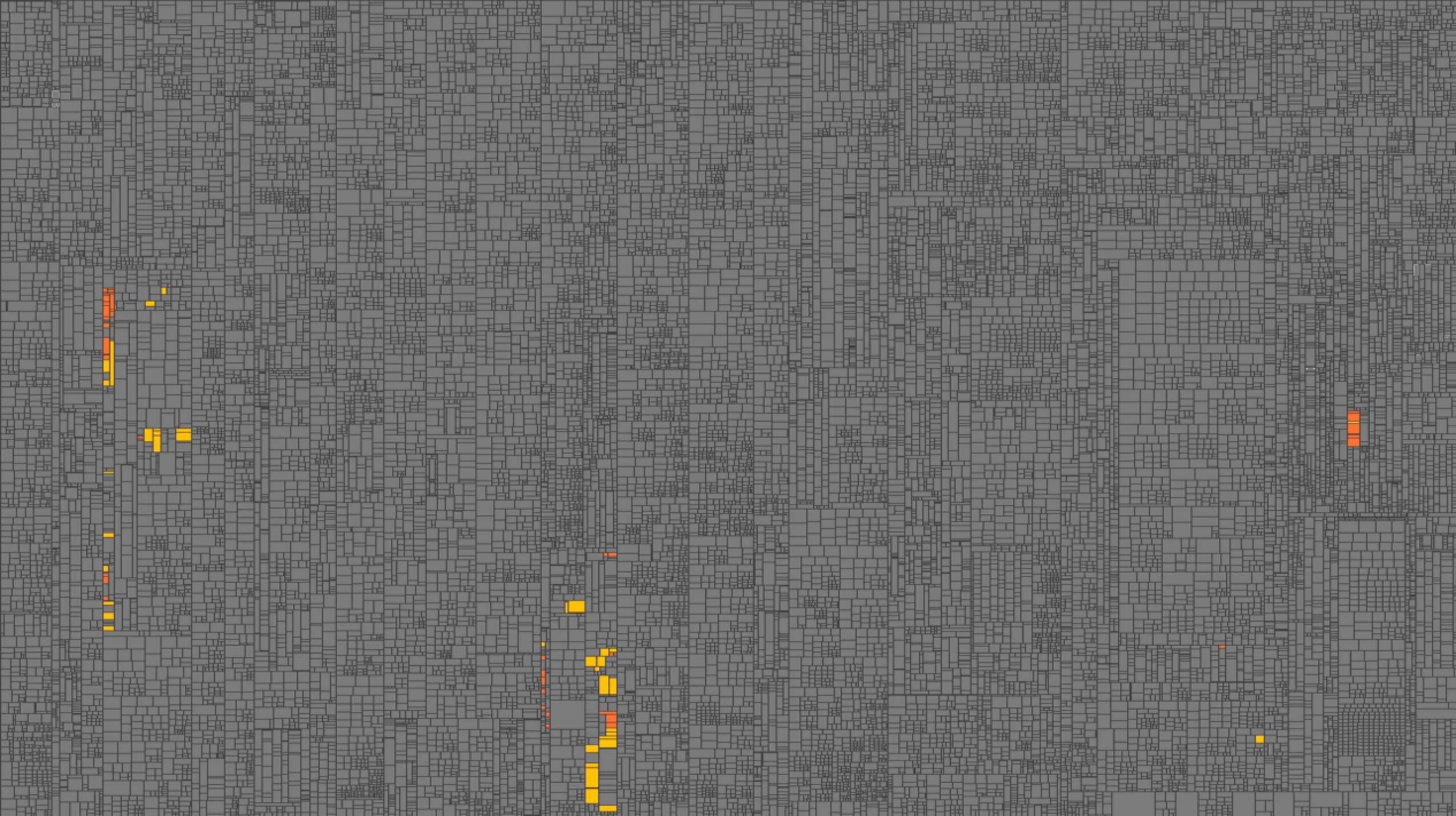
Created: 24/Nov/16 8:35 AM
Updated: 4 days ago
Resolved: 08/May/18 12:42 PM

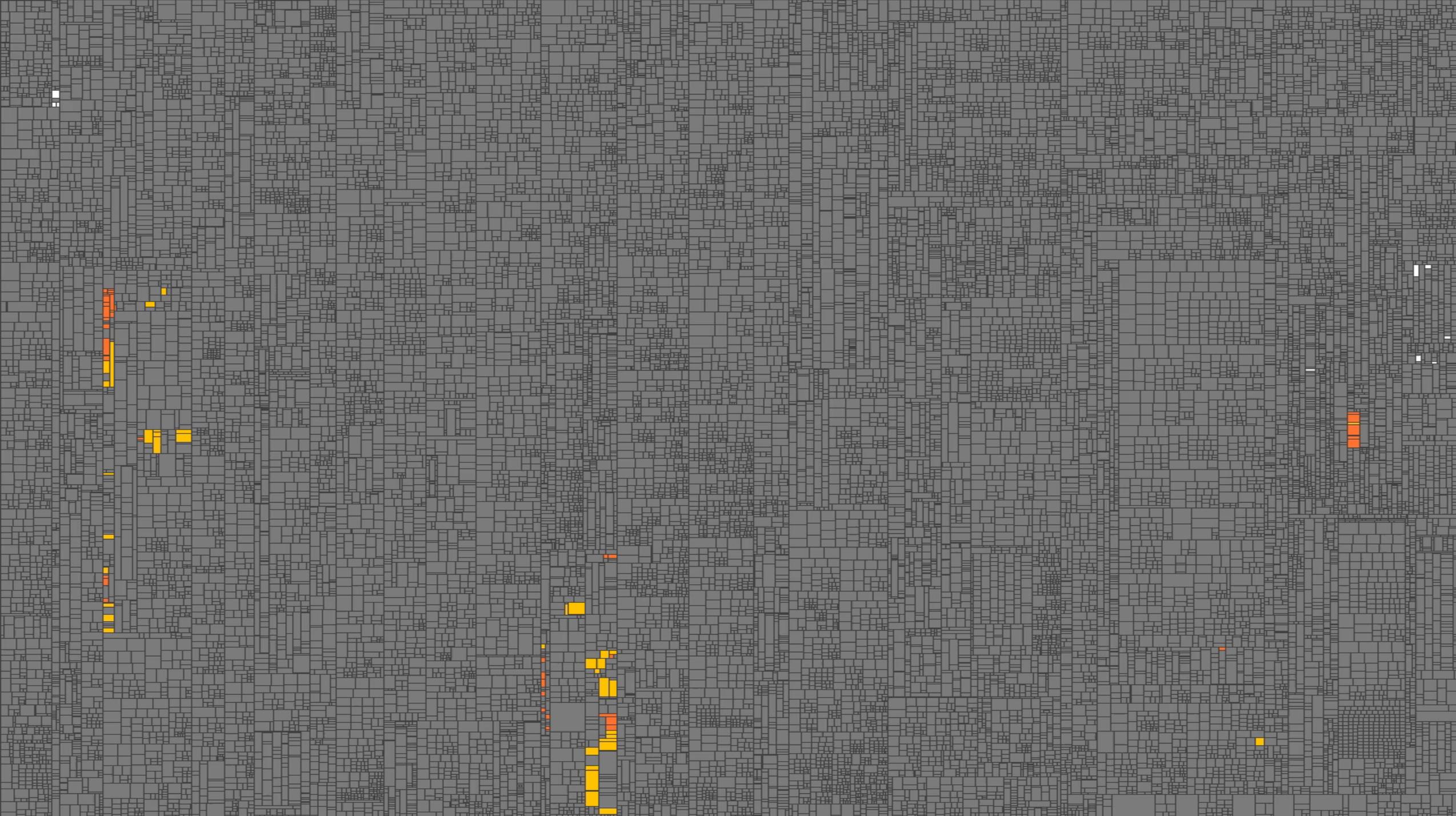
Time Tracking

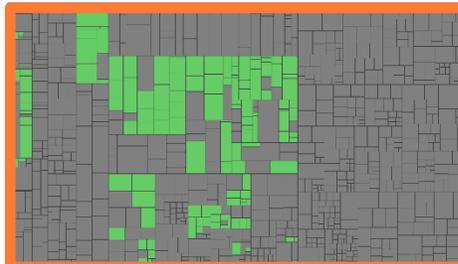
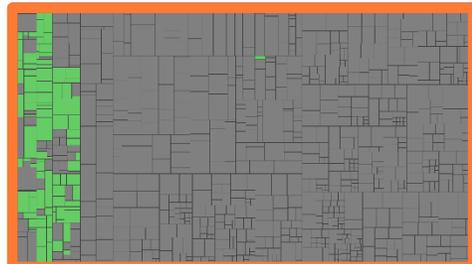
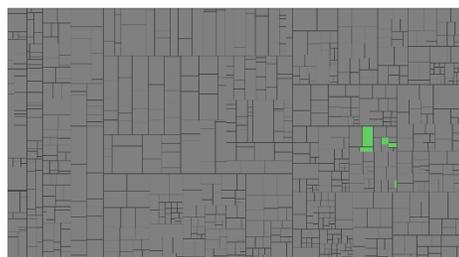
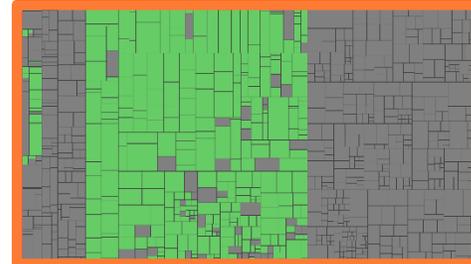
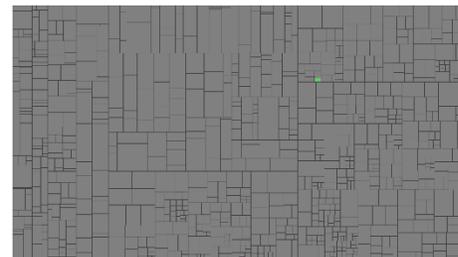
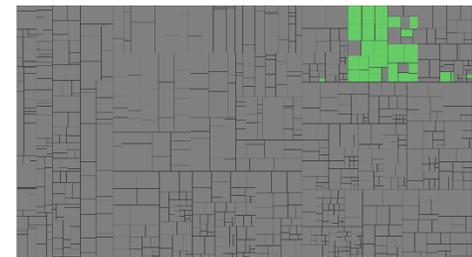
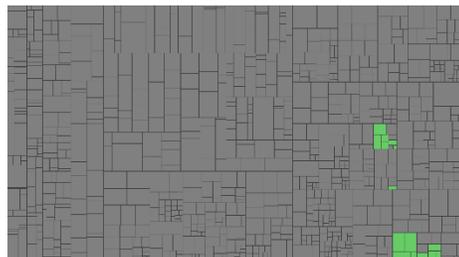
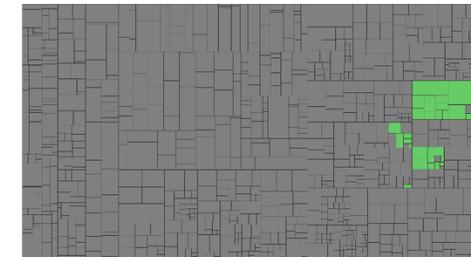
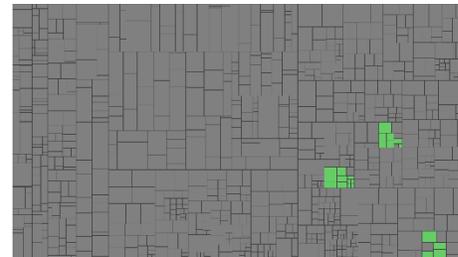
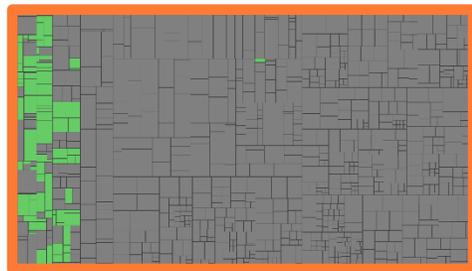
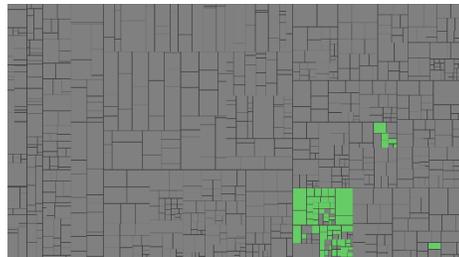
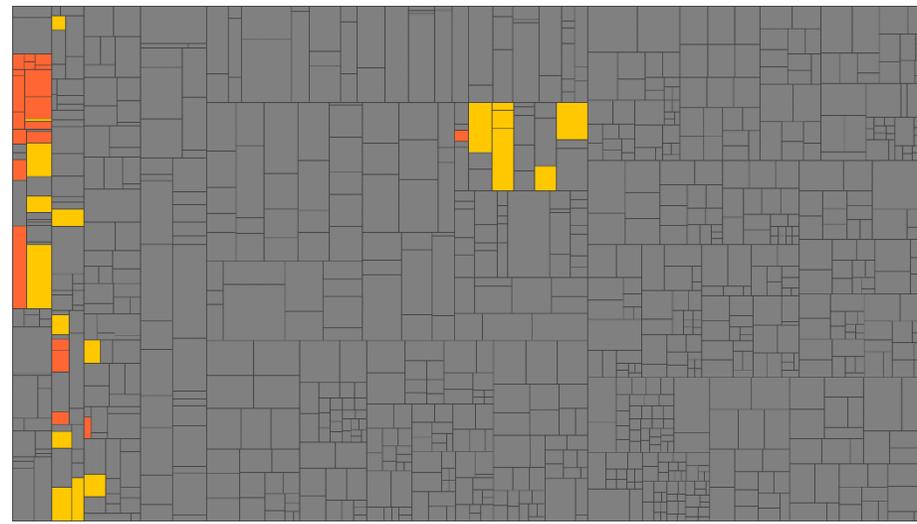
Estimated: Not Specified
Remaining: 0m
Logged: 4d 1h 57m

Agile

[View on Board](#)



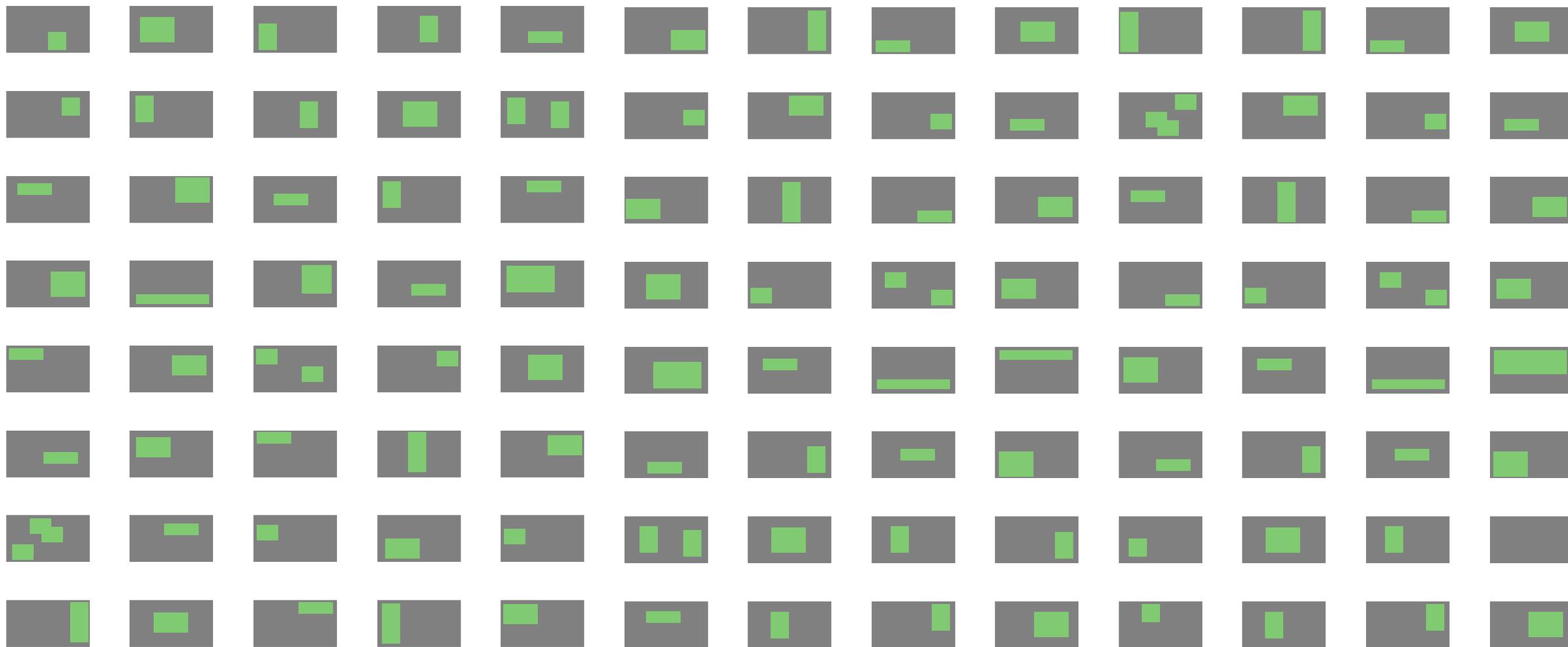




... 4000+ Testfälle
ohne Überdeckung

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

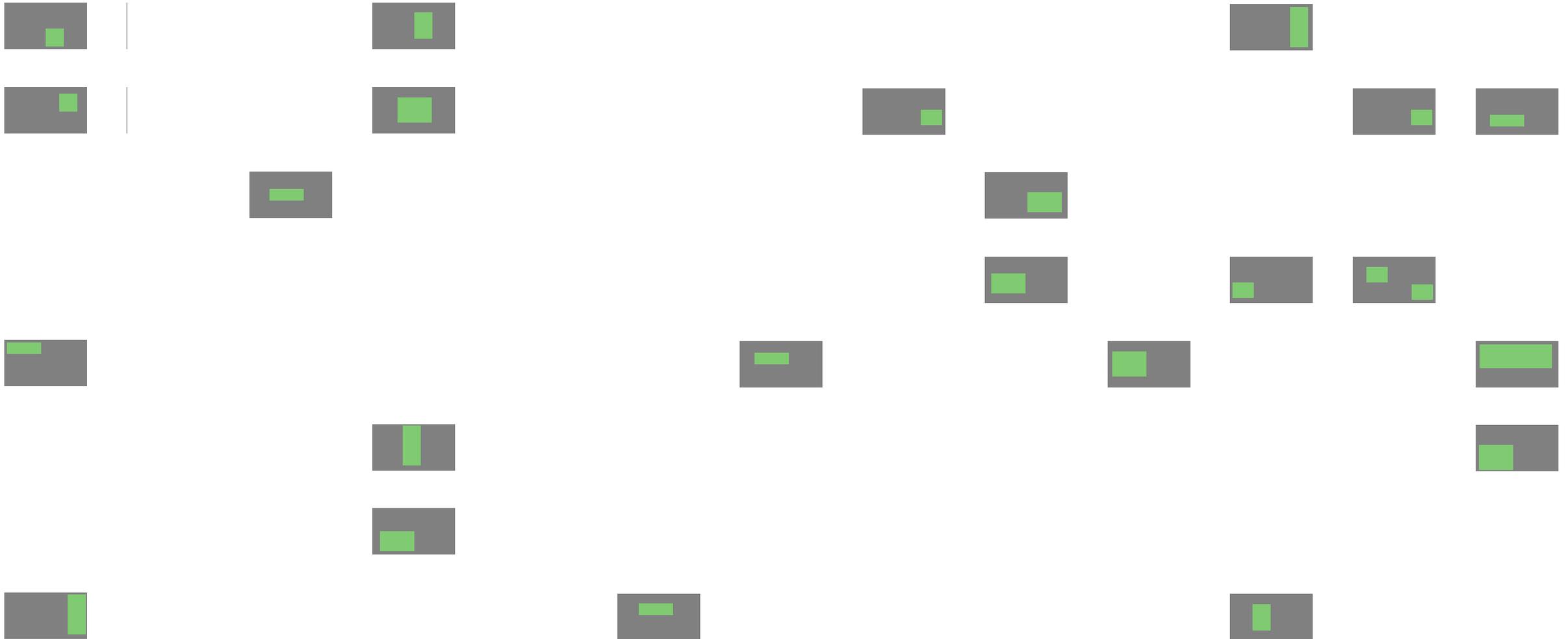
Test-Impact-Analyse



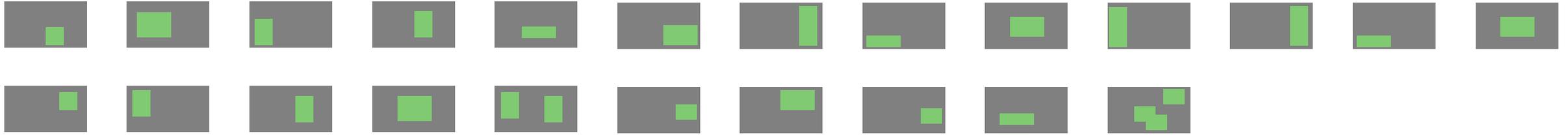
Schritt 1: Selektion betroffener Testfälle



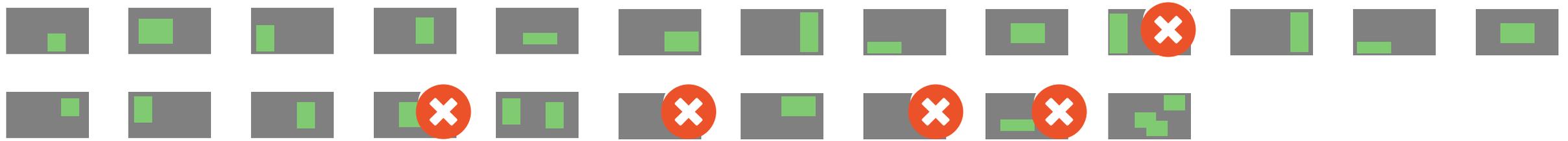
Schritt 1: Selektion betroffener Testfälle



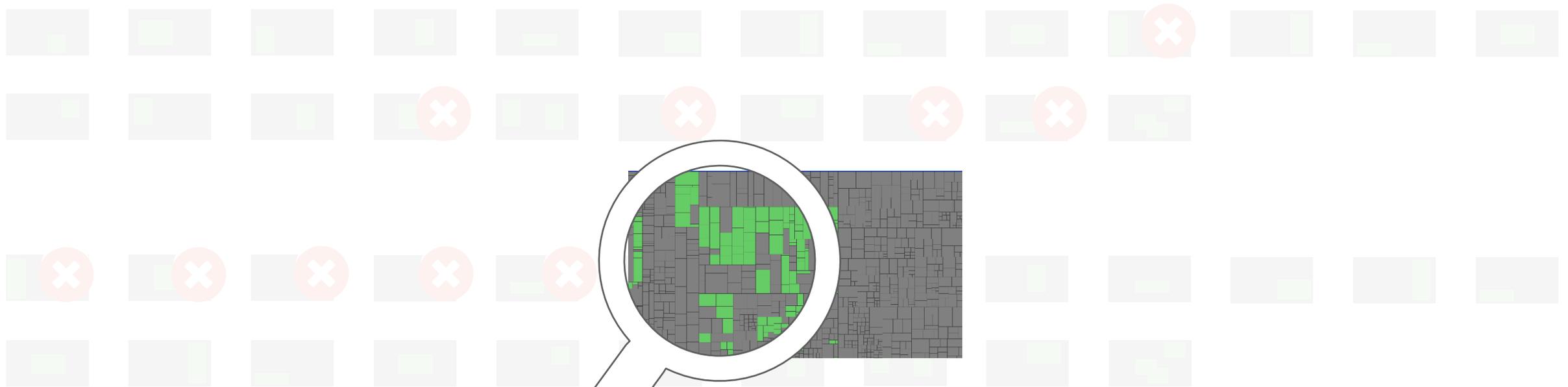
Schritt 1: Selektion betroffener Testfälle



Schritt 2: Priorisierung selektierter Testfälle

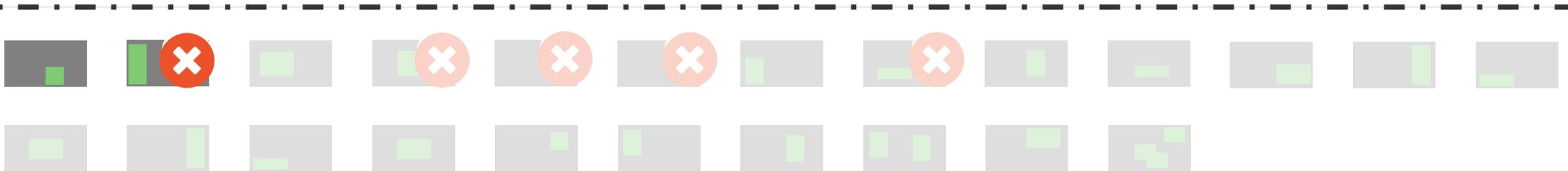


Schritt 2: Priorisierung selektierter Testfälle

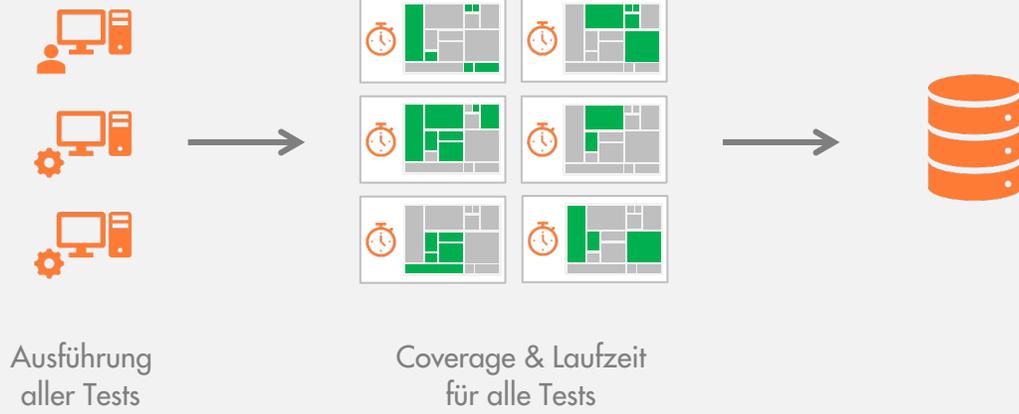


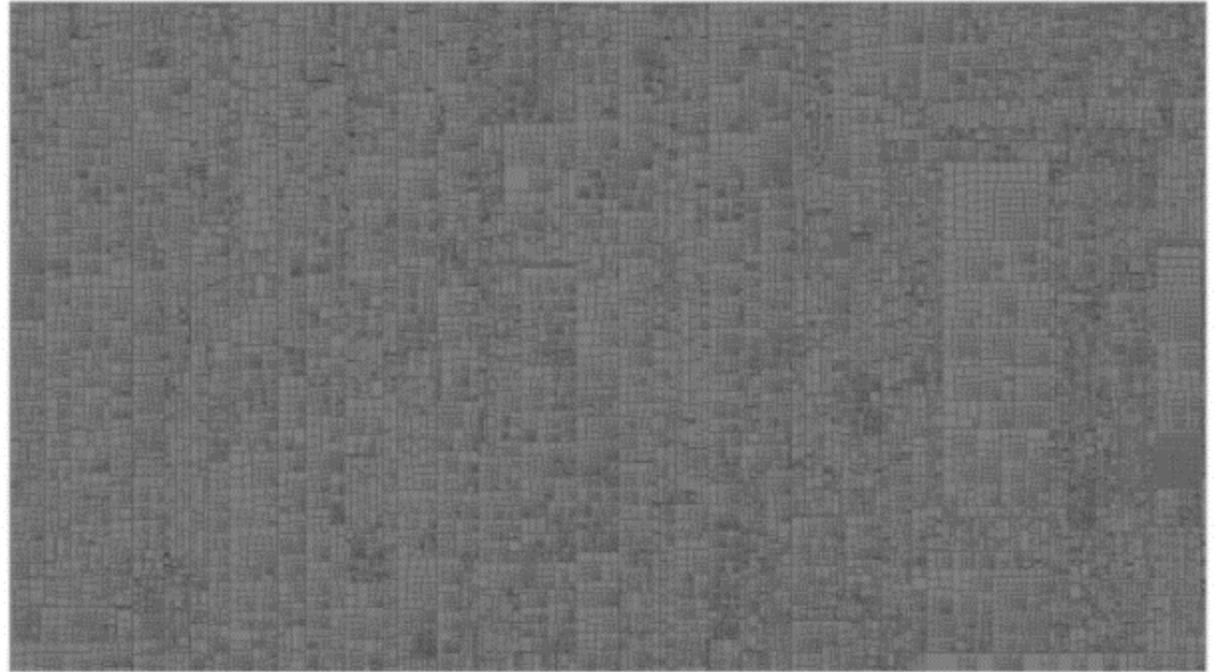
 Change coverage

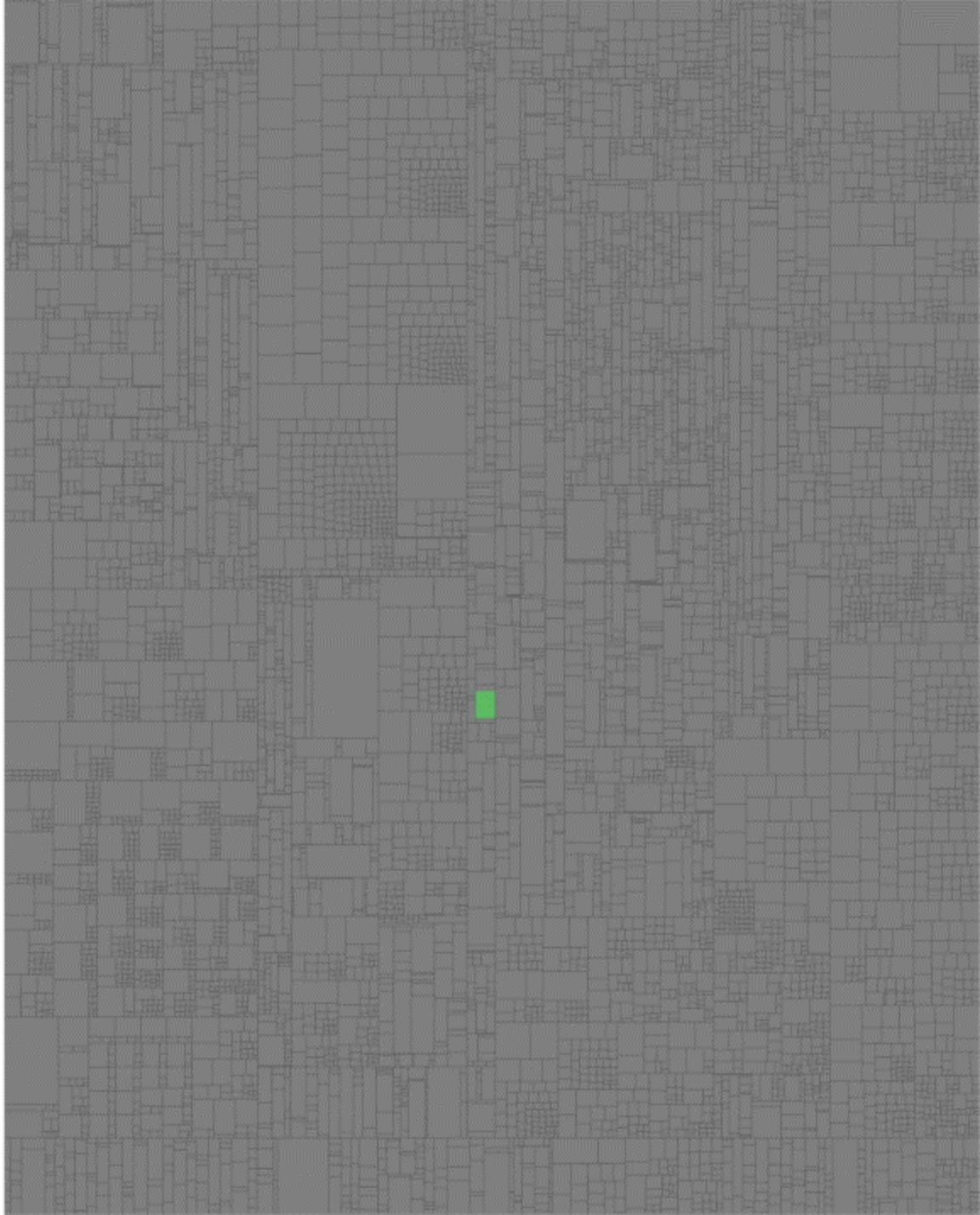
 Execution time



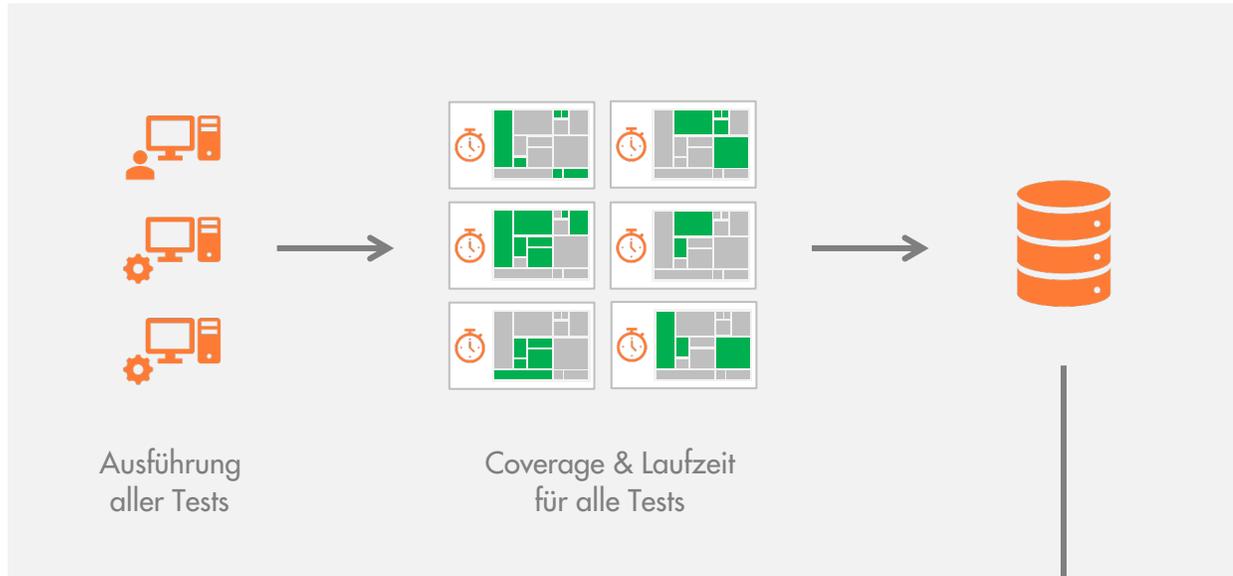
Initiale Aufzeichnung aller Tests







Initiale Aufzeichnung aller Tests



Testausführung nach Änderungen



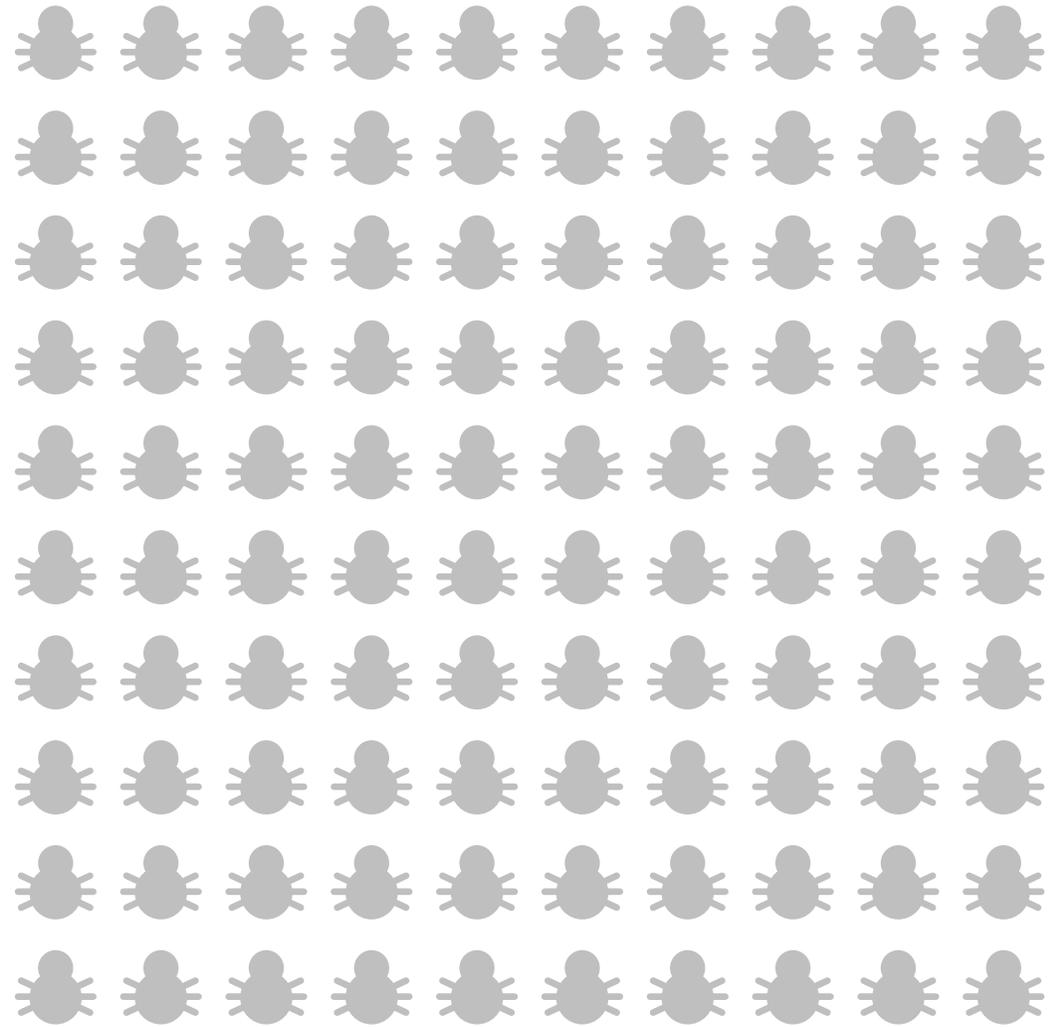
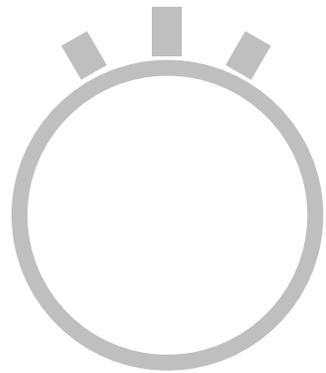
Grenzen der Test-Impact-Analyse

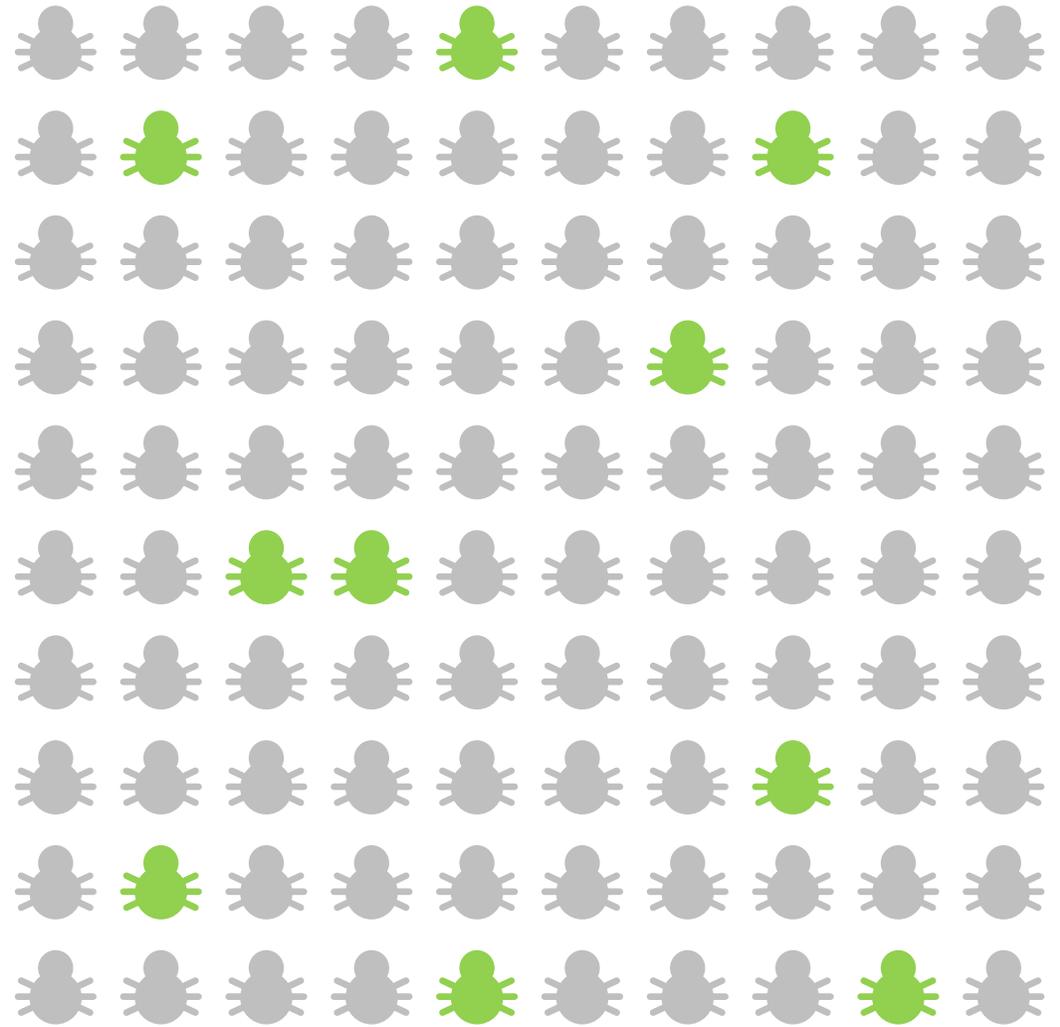
- Manche Änderungen betreffen alle Tests
 - Änderungen von Konfigurationsinformation
 - Änderungen an Testdaten
- Wenn sich die Tests selbst ändern, müssen sie erneut ausgeführt werden
- Es gibt keine Garantie, dass die selektierten Tests alle Fehler finden

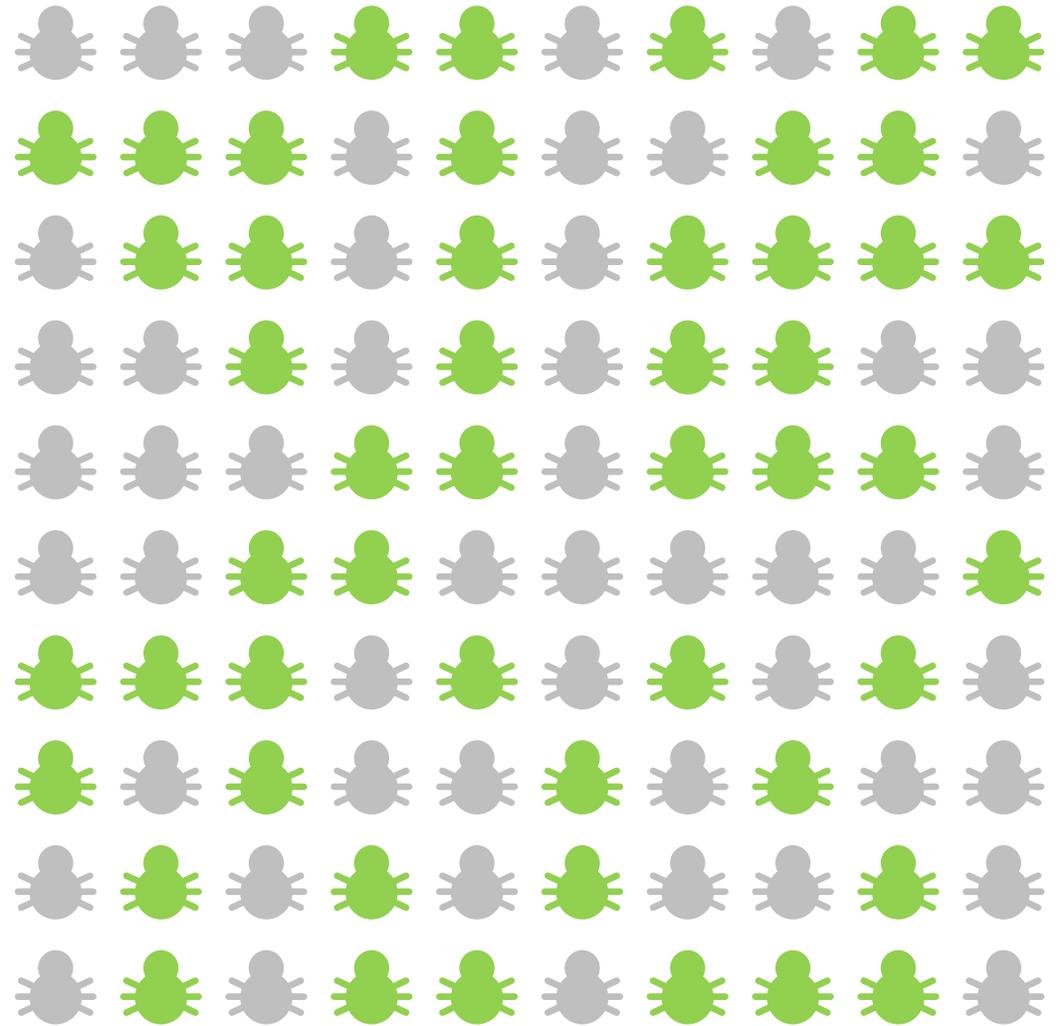
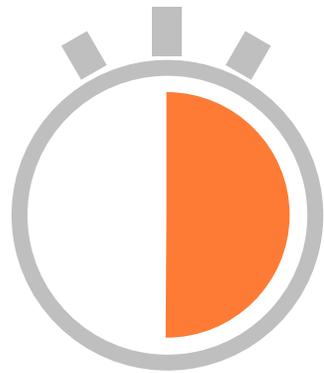
⇒ In regelmäßigen Abständen alle Tests ausführen

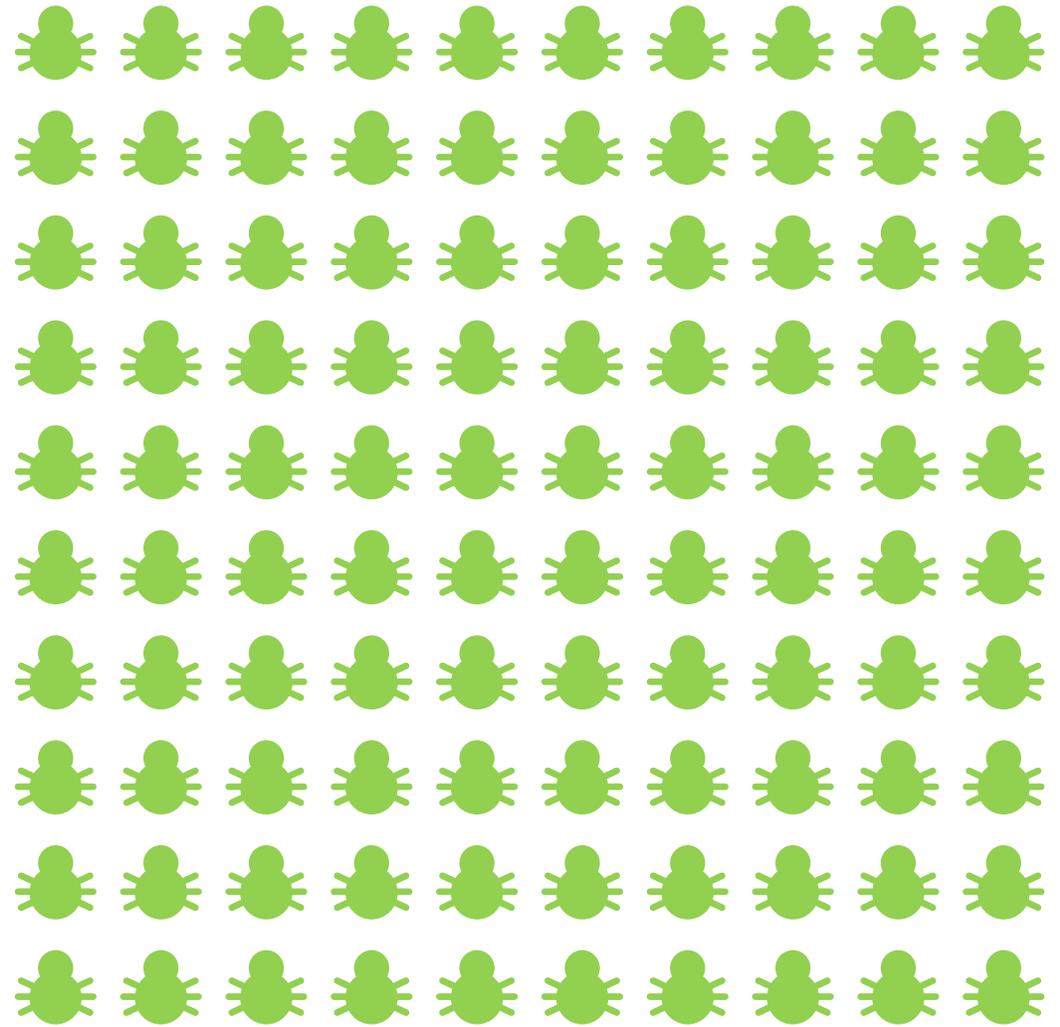
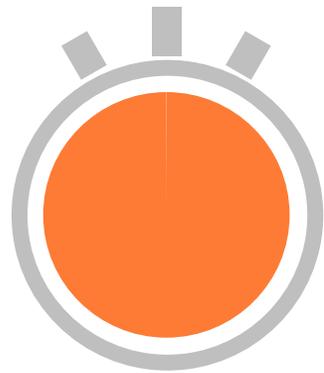
⇒ Hält testfallspezifische Coverage-Daten aktuell

Was bringt die Test-Impact-Analyse?











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	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 %
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 %



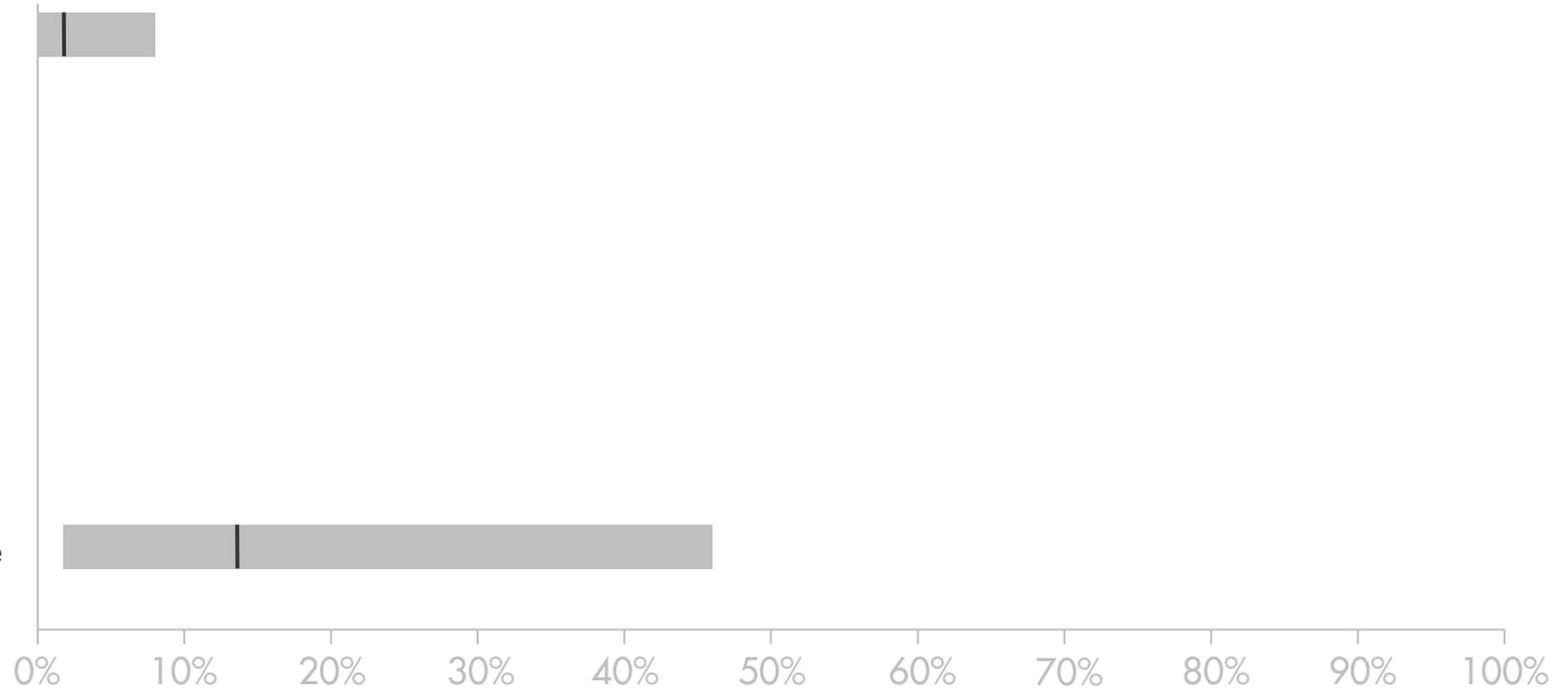
?



400+

TIA (Teamscale)

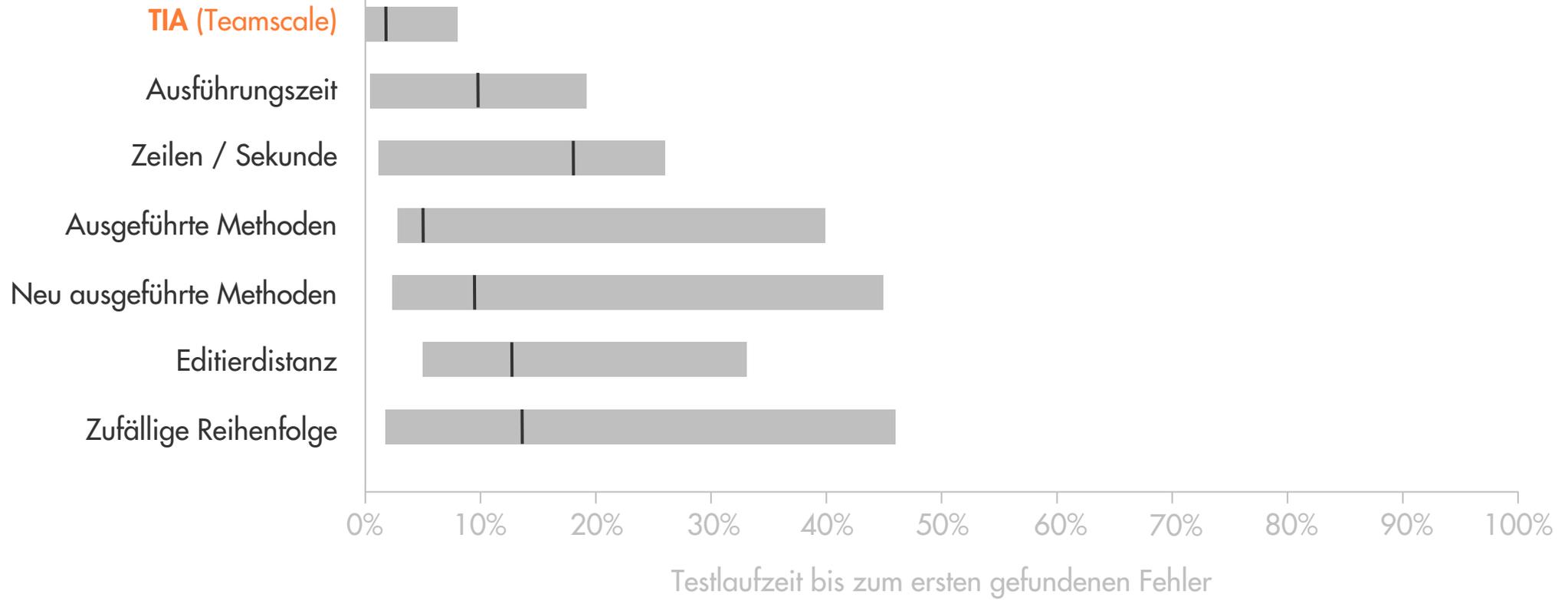
Zufällige Reihenfolge



Testlaufzeit bis zum ersten gefundenen Fehler



400+

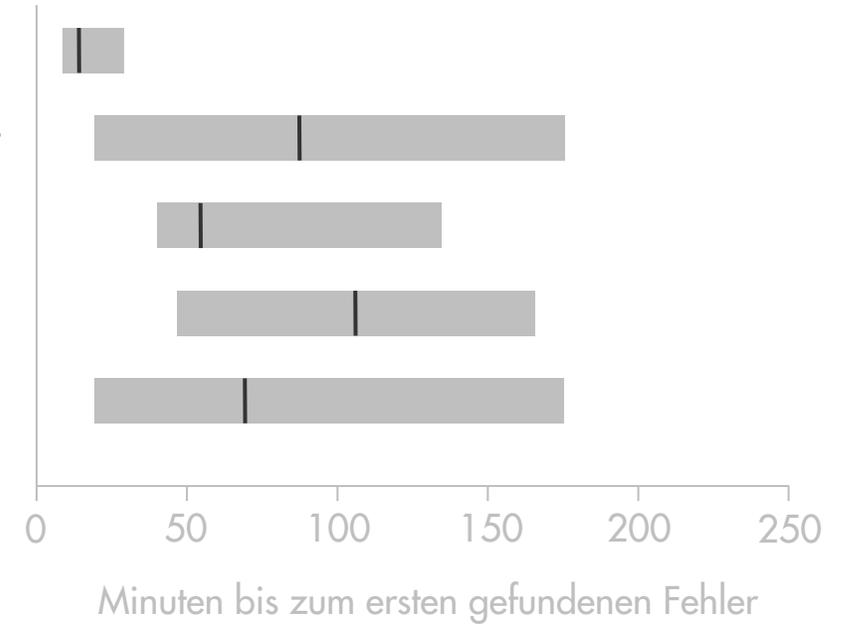




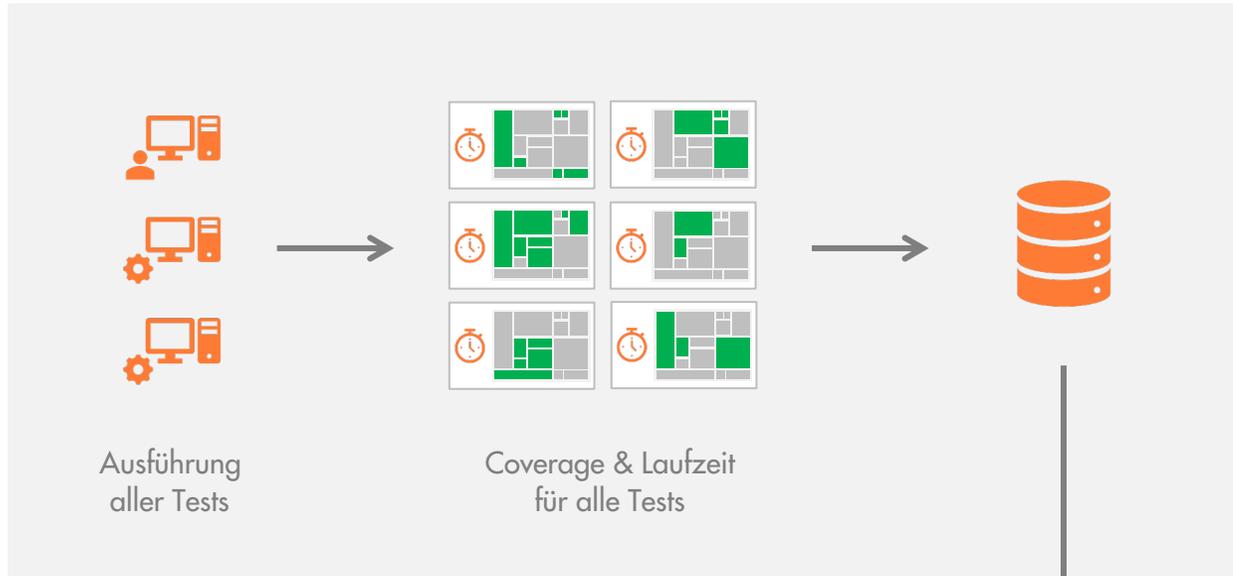


TIA (Teamscale)

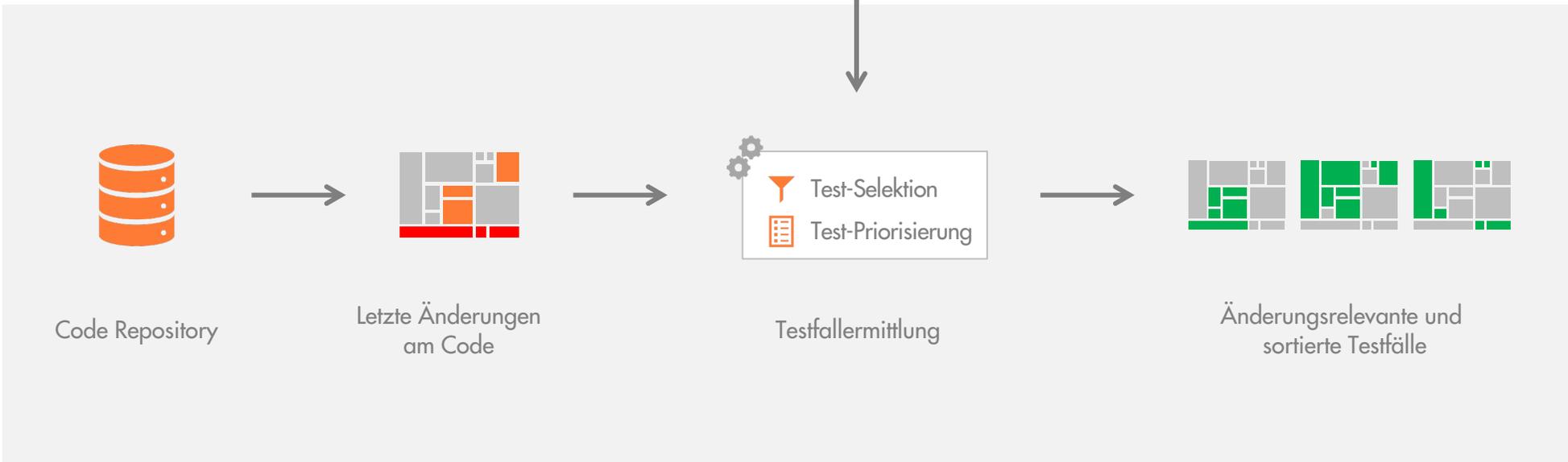
- Ausführungszeit
- Ausführungszeit (absteigend)
- Methoden / Sekunde
- Zufällige Reihenfolge

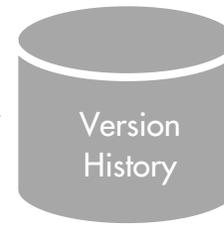
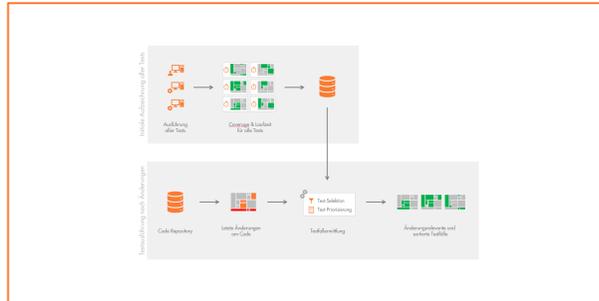


Initiale Aufzeichnung aller Tests

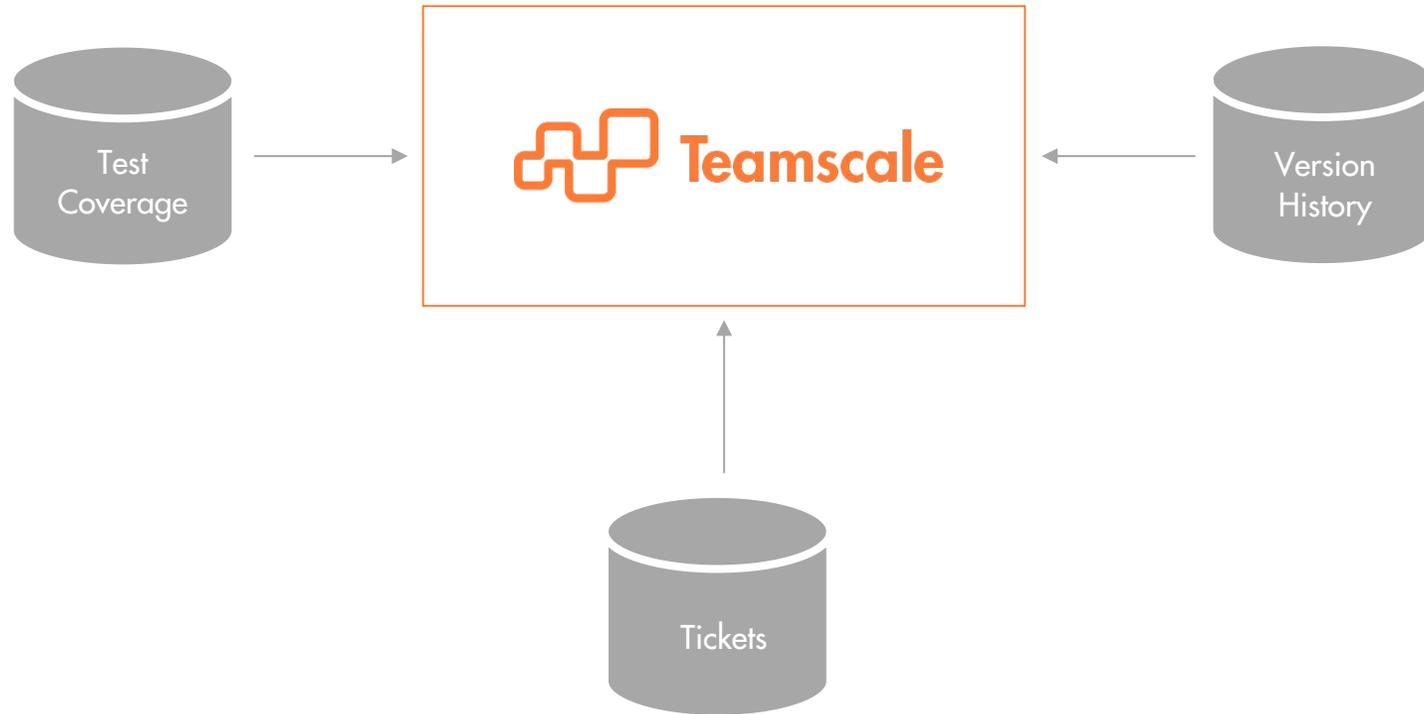


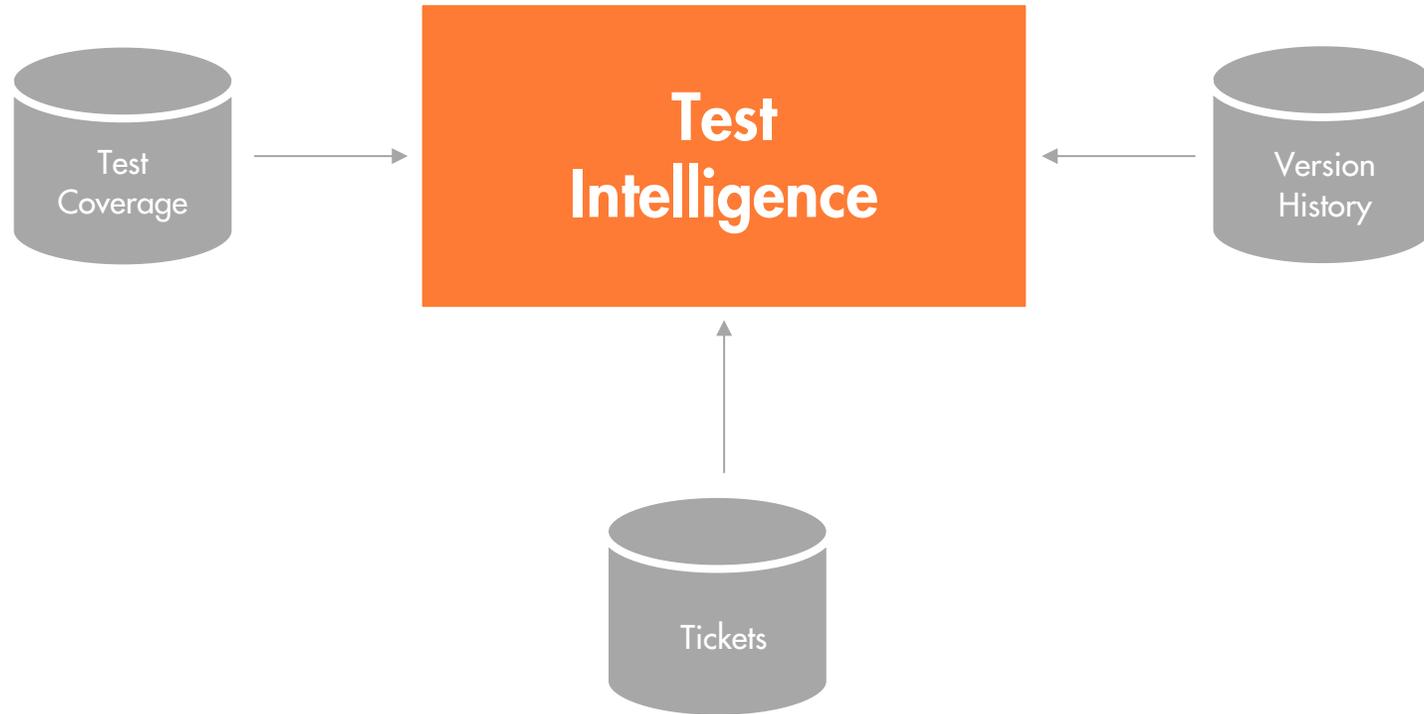
Testausführung nach Änderungen











Jobs

Pipeline	Id ▾	Name	Commit	Status	Runtime	Tests executed	Time saved	TIA enabled	Runtime distribution
83370	696909	ui-tests-common-lang	master (May 10, 2019 1:11:46 PM)	SUCCESS	19m 56s	100% -> master always runs all tests	01m 10s	Yes	
83370	696908	ui-tests	master (May 10, 2019 1:11:46 PM)	FAILED	25m 45s	100% -> master always runs all tests	02m 19s	Yes	
83369	696884	ui-tests-common-lang	cr/14511_findings_churn_blacklisted_findings (May 10, 2019 12:45:01 PM)	SUCCESS	21m 37s	100% -> No Testwise Coverage on branch	- 00m 31s	Yes	
83369	696883	ui-tests	cr/14511_findings_churn_blacklisted_findings (May 10, 2019 12:45:01 PM)	FAILED	26m 24s	100% -> No Testwise Coverage on branch	01m 40s	Yes	
83368	696861	ui-tests-common-lang	cr/18940_alerts_in_delta (May 10, 2019 12:23:51 PM)	SUCCESS	21m 13s	100% -> Teamscale not at HEAD	- 00m 07s	Yes	
83368	696860	ui-tests	cr/18940_alerts_in_delta (May 10, 2019 12:23:51 PM)	SUCCESS	28m 00s	100% -> Teamscale not at HEAD	00m 04s	Yes	
83367	696838	ui-tests-common-lang	cr/18778_add_tasks_slide (May 10, 2019 12:16:17 PM)	SUCCESS	06m 28s	00%	14m 38s	Yes	
83367	696837	ui-tests	cr/18778_add_tasks_slide (May 10, 2019 12:16:16 PM)	SUCCESS	06m 28s	00%	21m 36s	Yes	
83366	696814	ui-tests-common-lang	cr/18985_apply_patterns_when_toggle_on (May 10, 2019 11:40:16 AM)	SUCCESS	03m 00s	00%	18m 06s	Yes	
83366	696813	ui-tests	cr/18985_apply_patterns_when_toggle_on (May 10, 2019 11:40:16 AM)	SUCCESS	04m 34s	02%	23m 30s	Yes	
83365	696790	ui-tests-common-lang	cr/18940_alerts_in_delta (May 10, 2019 11:33:35 AM)	SUCCESS	04m 13s	01%	16m 53s	Yes	
83365	696789	ui-tests	cr/18940_alerts_in_delta (May 10, 2019 11:33:35 AM)	SUCCESS	04m 08s	02%	23m 56s	Yes	



Dr. Jan Harder · harder@cqse.eu · +49 172 1860021

Dr. Nils Göde · goede@cqse.eu · +49 176 10452662

CQSE GmbH
Centa-Hafenbrädl-Straße 59
81249 München