

Risikobasiertes Testen während der Entwicklung

ESE Kongress, 8. Dezember 2022



Dr. Tobias Röhm



Freelancer

TUM

CQSE



2004 - 2010

2010 - 2015

2015 - Heute



Praxis

Software **Audits**

Einführung von
kontinuierlichen **Qualitäts-**
und **Test-Prozessen**

 **Teamscale**

CQSE GmbH



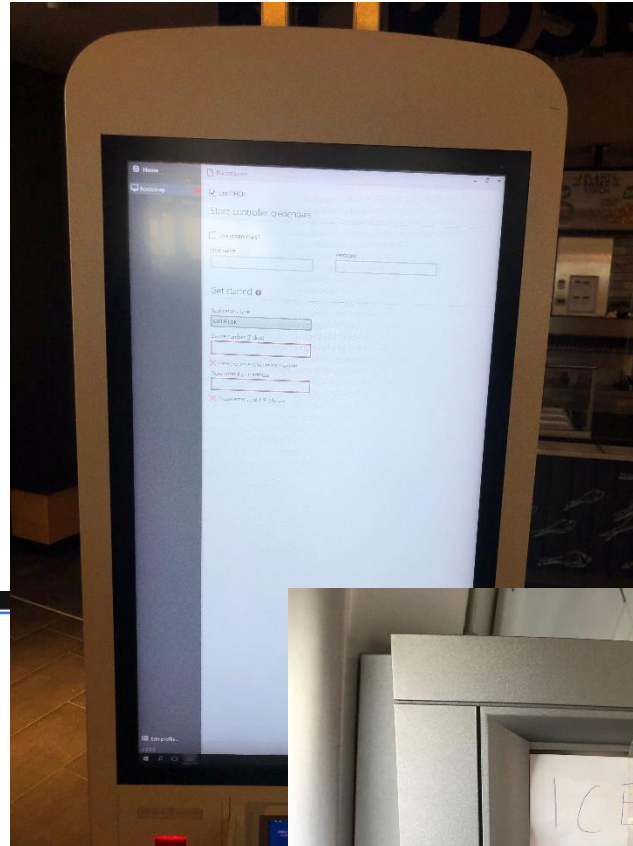
Forschung

16+ **Promotionen** in
Software Engineering

Eigene **Forschung**

Enger Kontakt zu
Universitäten

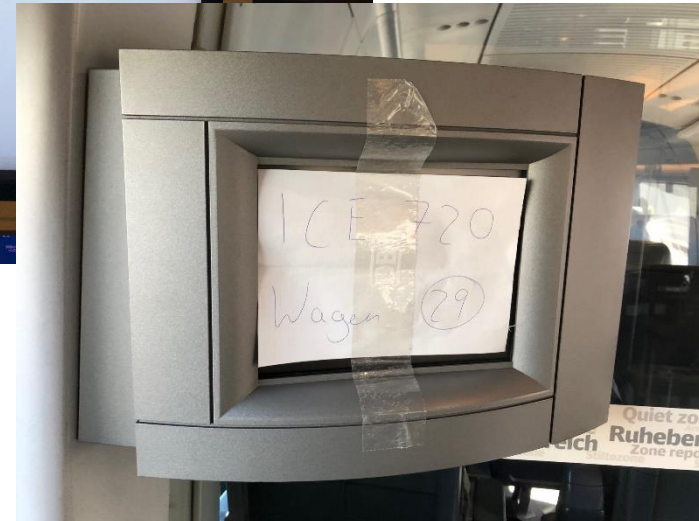
Softwarefehler im Alltag ...



**error-
dialog.generic.header**

error-dialog.generic.body

fatal-error.button-label



Wenn Software nicht scheitern darf ...



Wenn Software nicht scheitern darf ...



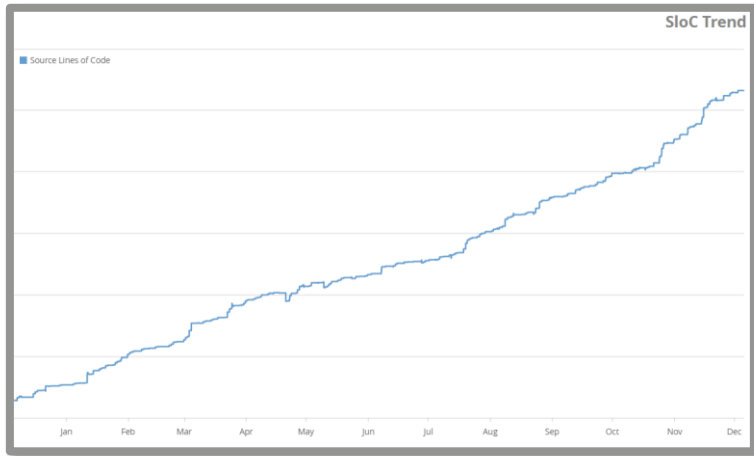
Herausforderungen für Testteams

Release in 2020	
DEV	QA

Release in 2022	
DEV	QA

Herausforderung
»Wachsende Software«

Herausforderung
»Wenig(er) Zeit«



Herausforderung
»Limitierte Ressourcen«

→ **Klassische Testphasen verlieren an Bedeutung, entwicklungsbegleitende Tests gewinnen an Bedeutung**

Risikobasiertes Testen während der Entwicklung

Ziele:

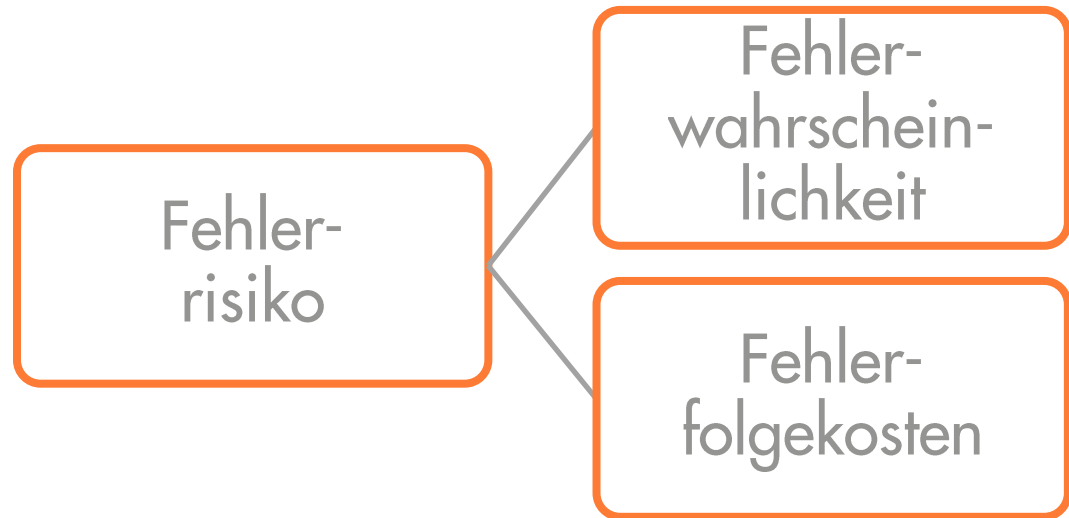
- Qualitätszustand nah an Releasefähigkeit halten
- Limitierte Testressourcen effizient einsetzen
- Fehlerrisiken zu jedem Entwicklungszeitpunkt niedrig halten

Fehler-
risiko

Fehler-
wahr-
schein-
lichkeit

Fehler-
folgekosten

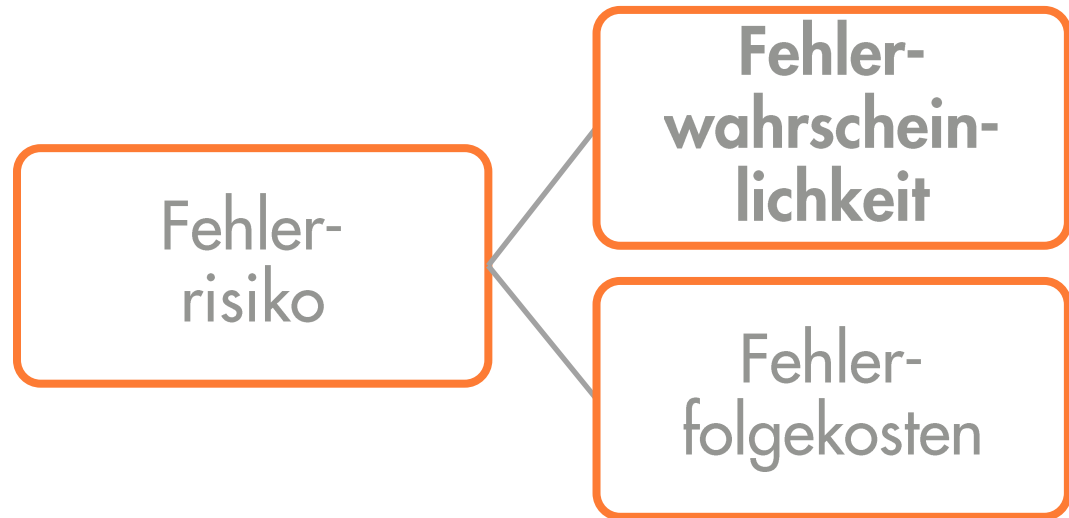
Agenda



Test-Gap-Analyse zur Identifikation von Testlücken
Test-Impact-Analyse zur Beschleunigung von Testsuites

Agiles Requirements Tracing zur Identifikation
von kritischen, ungetesteten Anforderungen

Agenda



Test-Gap-Analyse zur Identifikation von Test Gaps
Test-Impact-Analyse zur Beschleunigung von Testsuites

Agiles Requirements Tracing zur Identifikation von kritischen, ungetesteten Anforderungen

Wo ist die Fehlerwahrscheinlichkeit am höchsten?

Studie: C# System @ Munich Re

Wo ist die Fehlerwahrscheinlichkeit am höchsten?

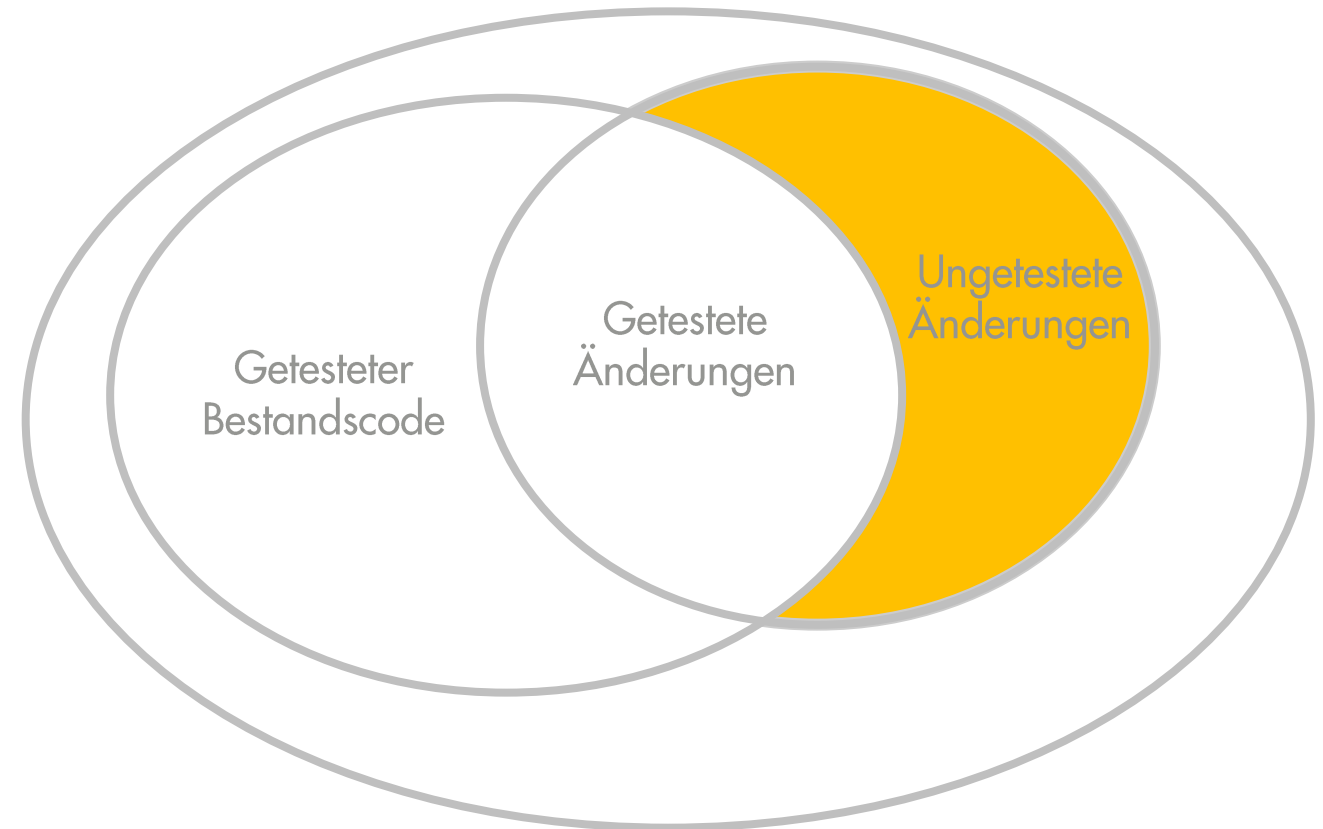
Studie: C# System @ Munich Re

Release A:

15% Code neu/geändert,
>50% ungetestet

Release B:

15% Code neu/geändert,
>60% ungetestet



Feldfehlerwahrscheinlichkeit 5x höher für ungetestete Änderungen!

Änderungen

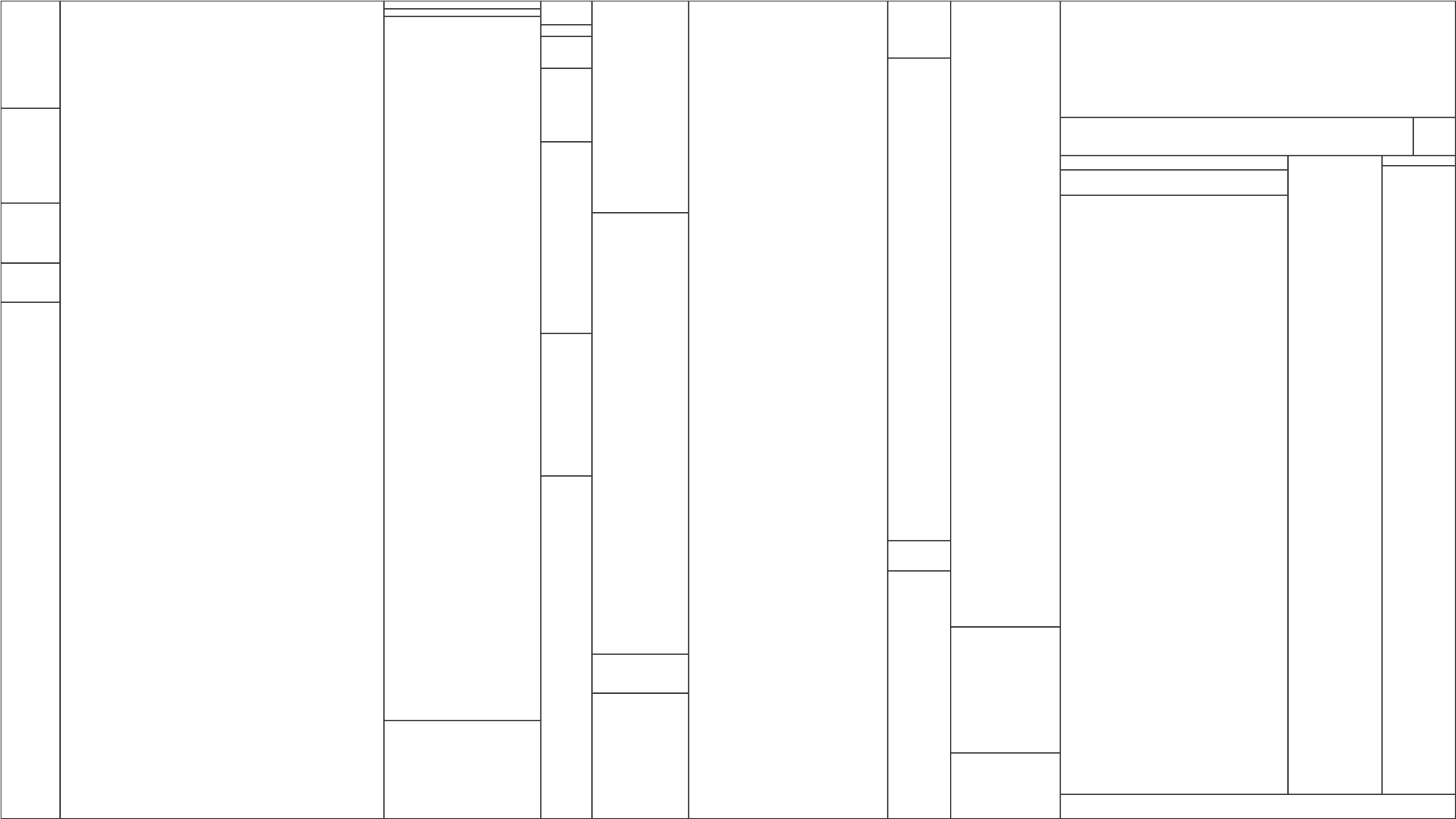


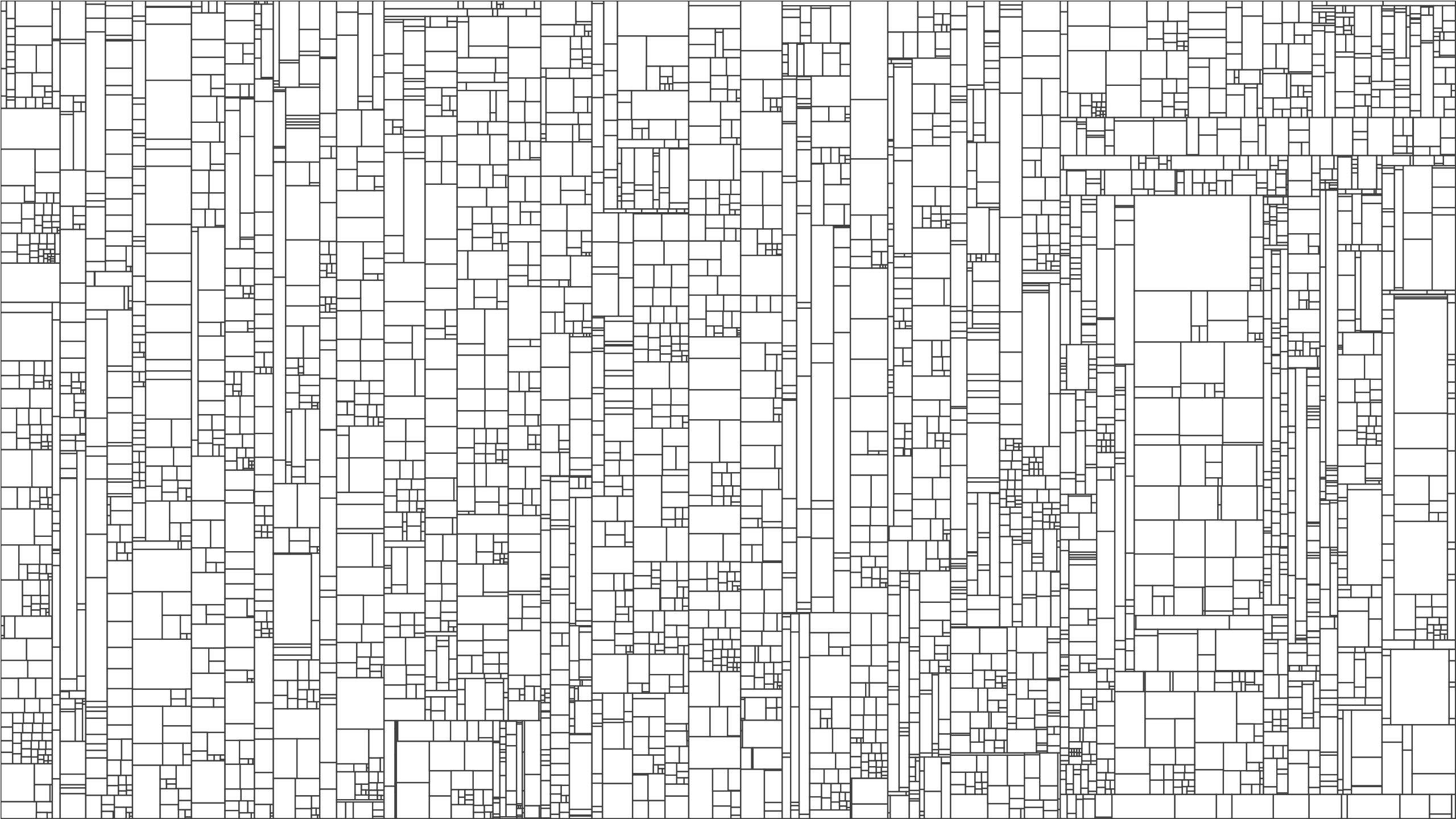
Test-Gap-Analyse

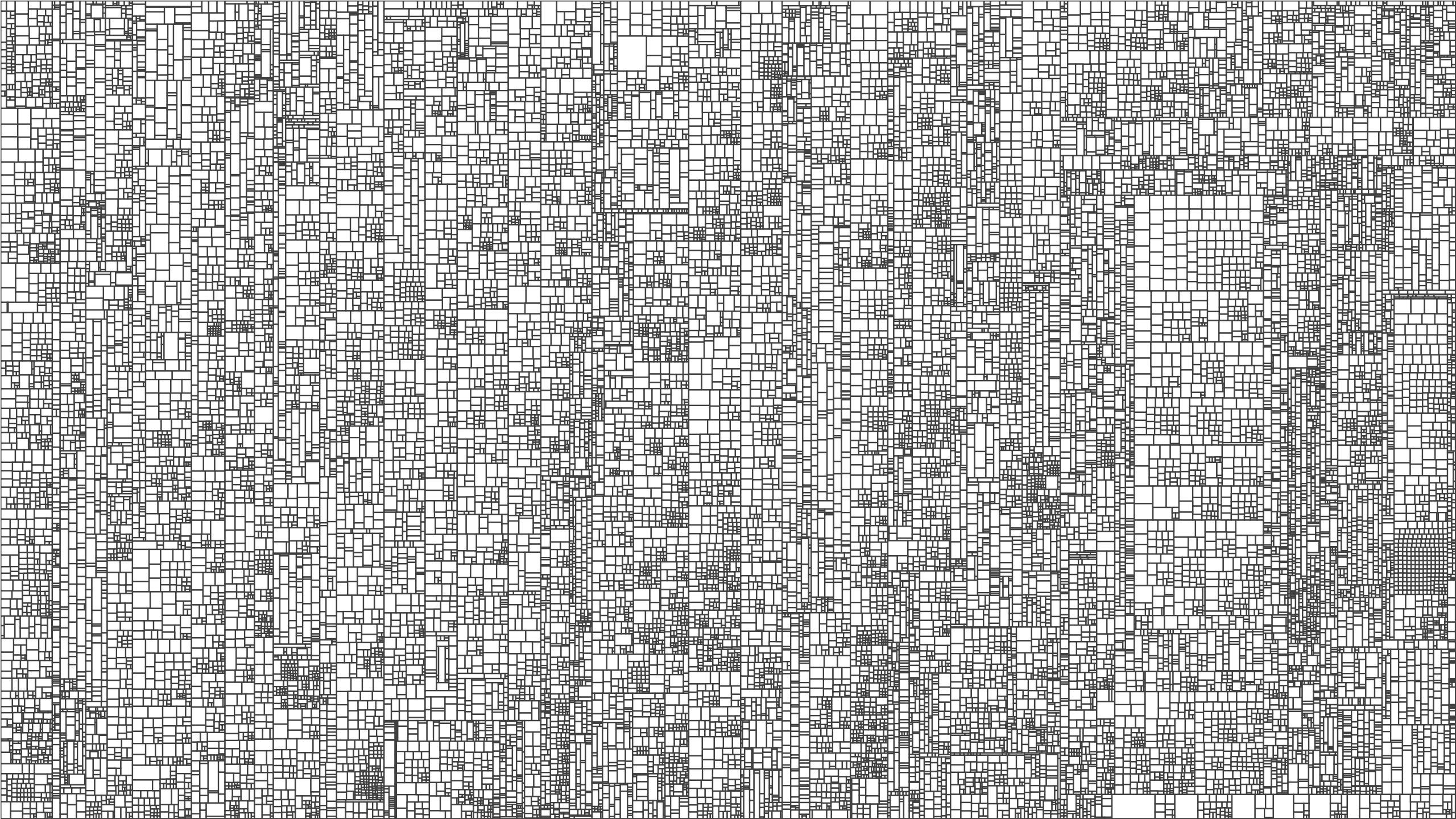
Ausführung



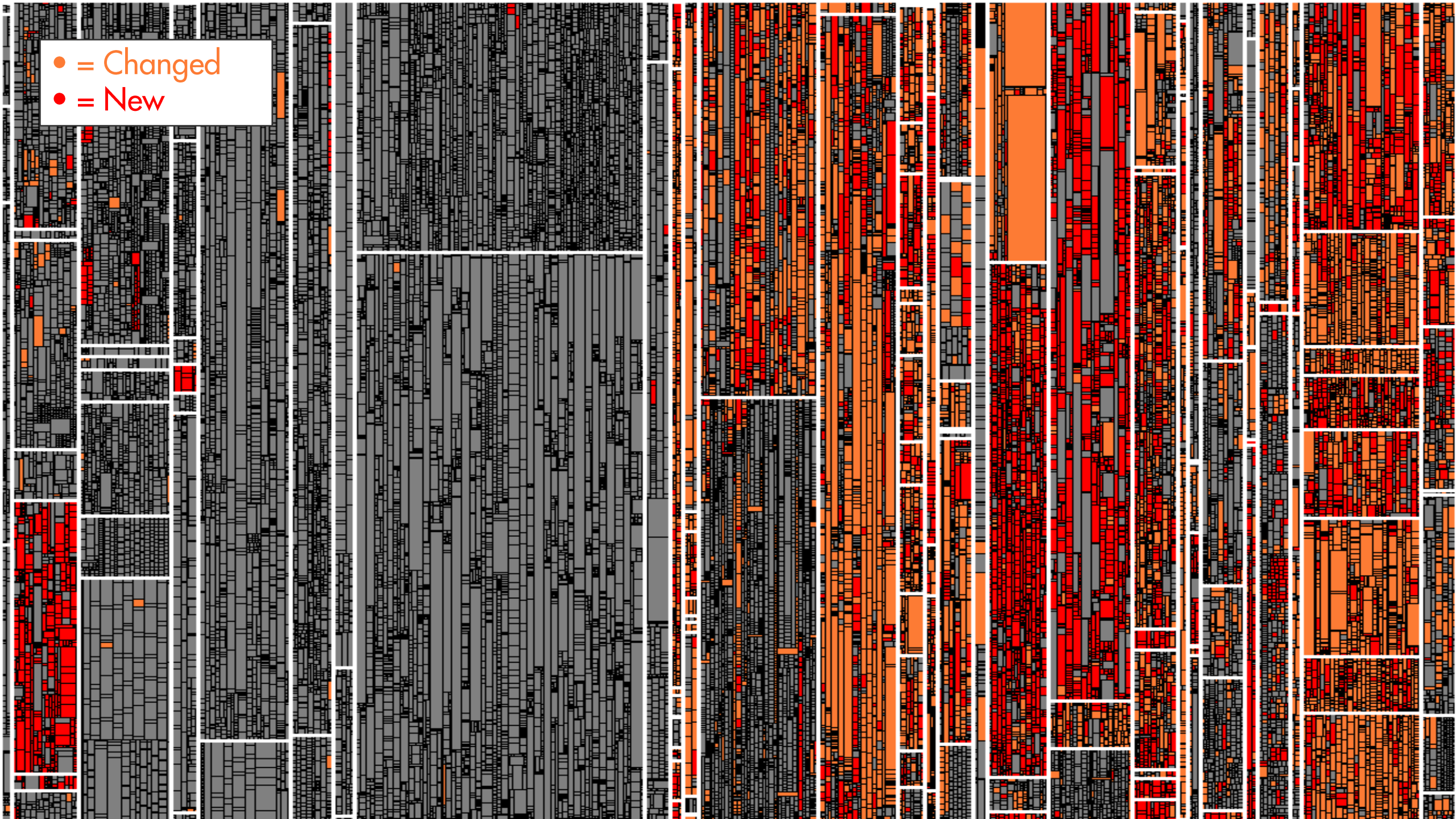
Ungetestete
Änderungen





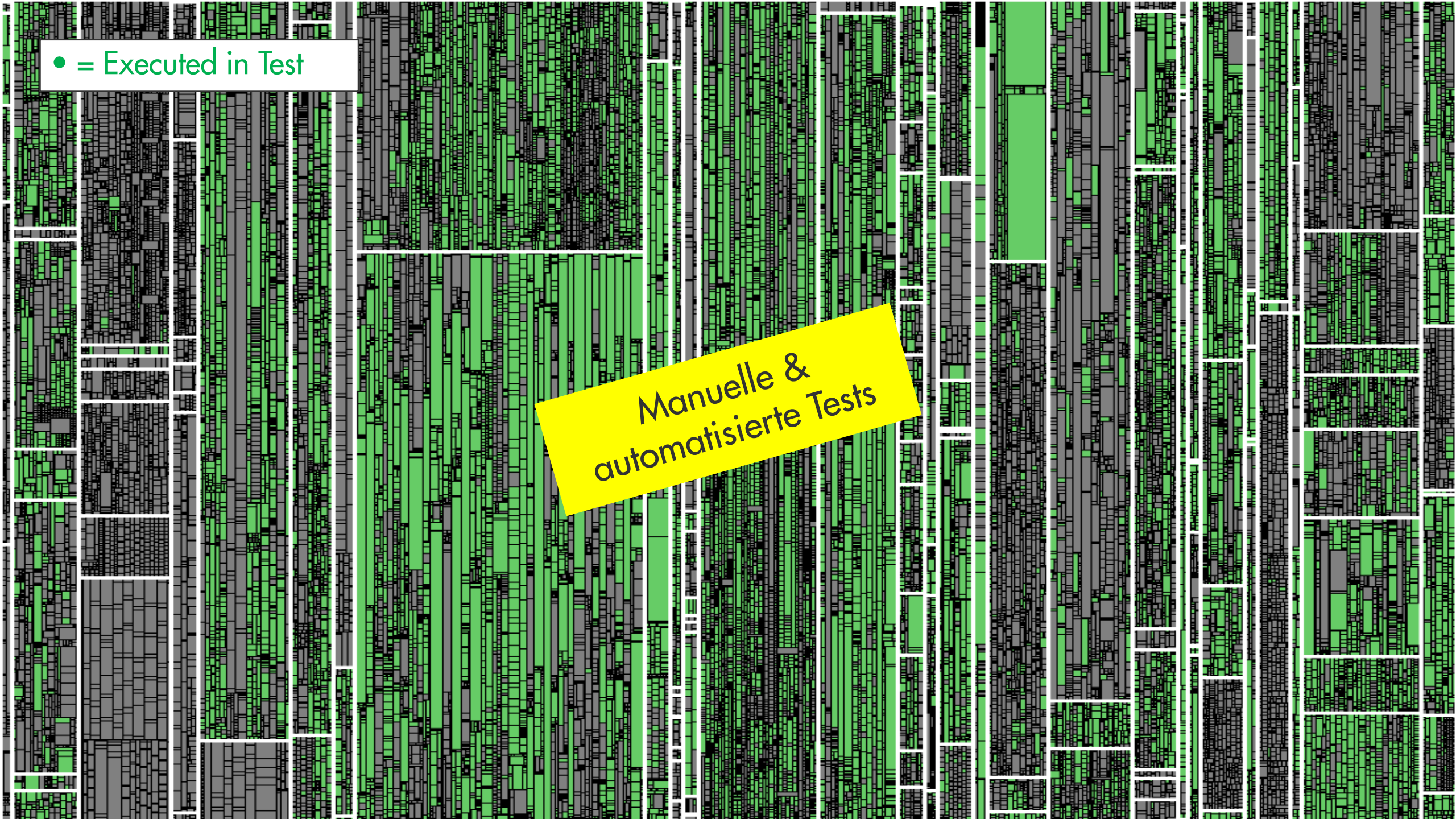


- = Changed
- = New

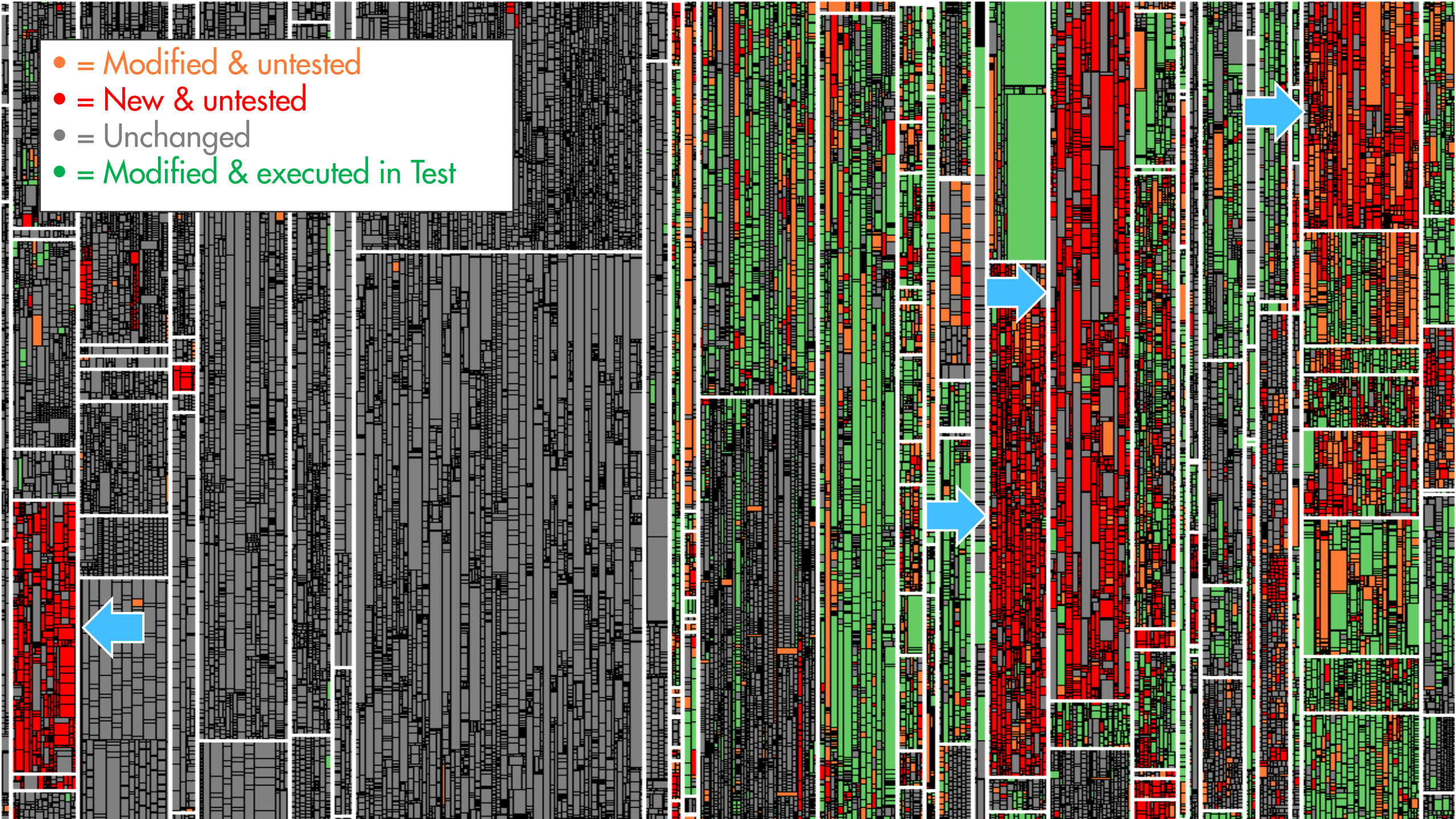


● = Executed in Test

Manuelle &
automatisierte Tests



- = Modified & untested
- = New & untested
- = Unchanged
- = Modified & executed in Test



Änderungen



Test-Gap-Analyse

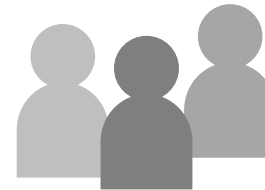


Ungetestete
Änderungen

Ausführung



Testfälle



Entwickler, Tester, Testmanager




Test-Gap-Analyse: Fazit

- Test-Gap-Analyse identifiziert Test Gaps (= geänderter, ungetesteter Code) mit hoher Fehlerwahrscheinlichkeit.
- Fokussierung von Testaktivitäten auf diese Codebereiche
- Senkung der (Feld-) Fehlerwahrscheinlichkeit & effiziente Nutzung von Testressourcen

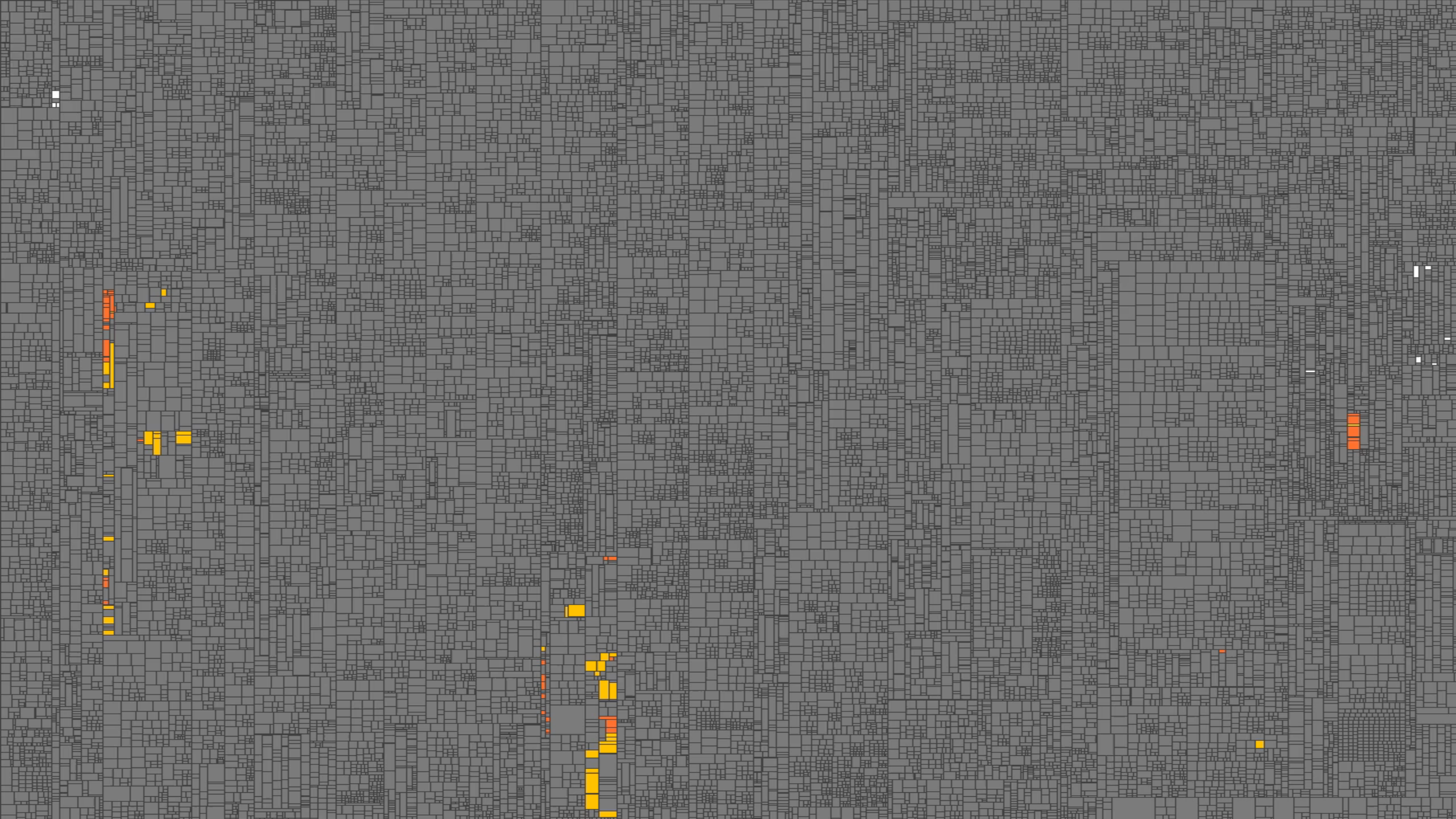
Test-Impact-Analyse: Problemstellung

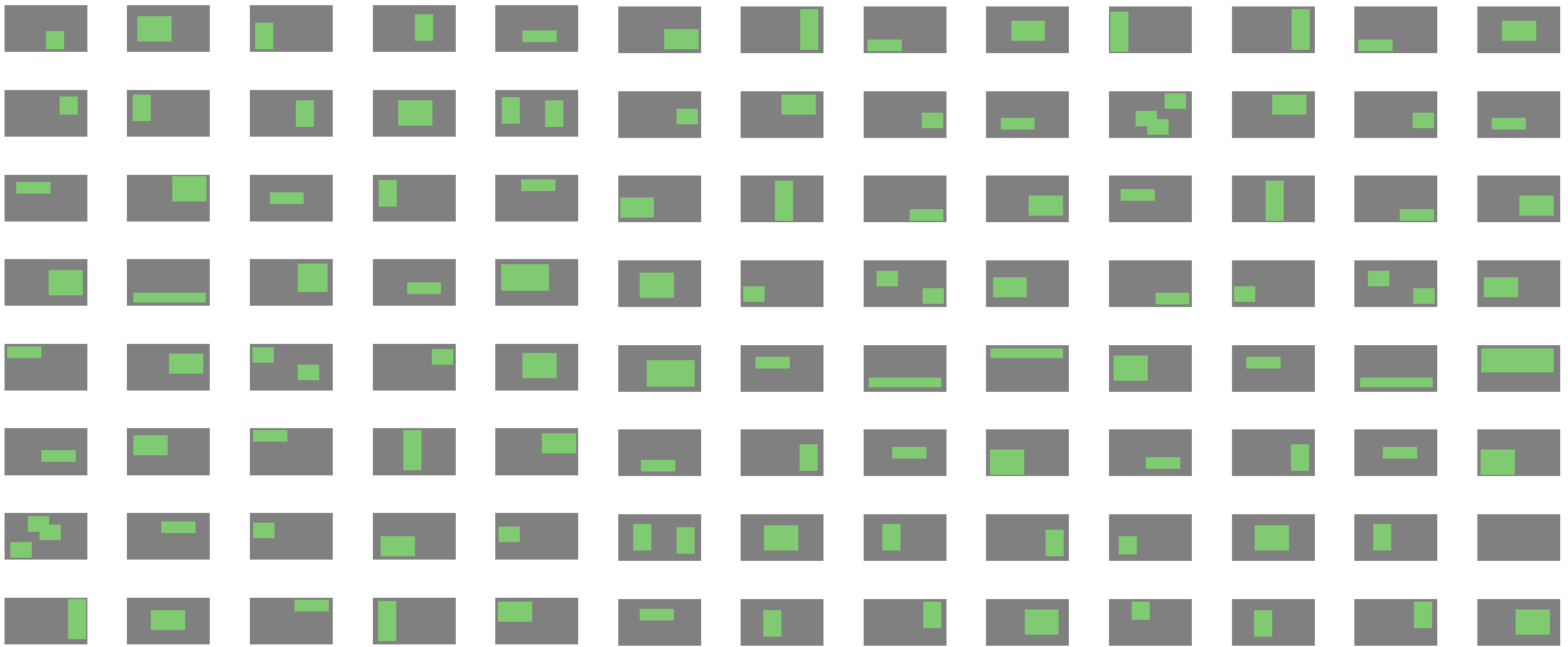


<https://www.technica-engineering.de/produkte/bts-body-electronic-test-system/>



Langlaufende Testsuites
beschleunigen
→ schnelles
Testfeedback

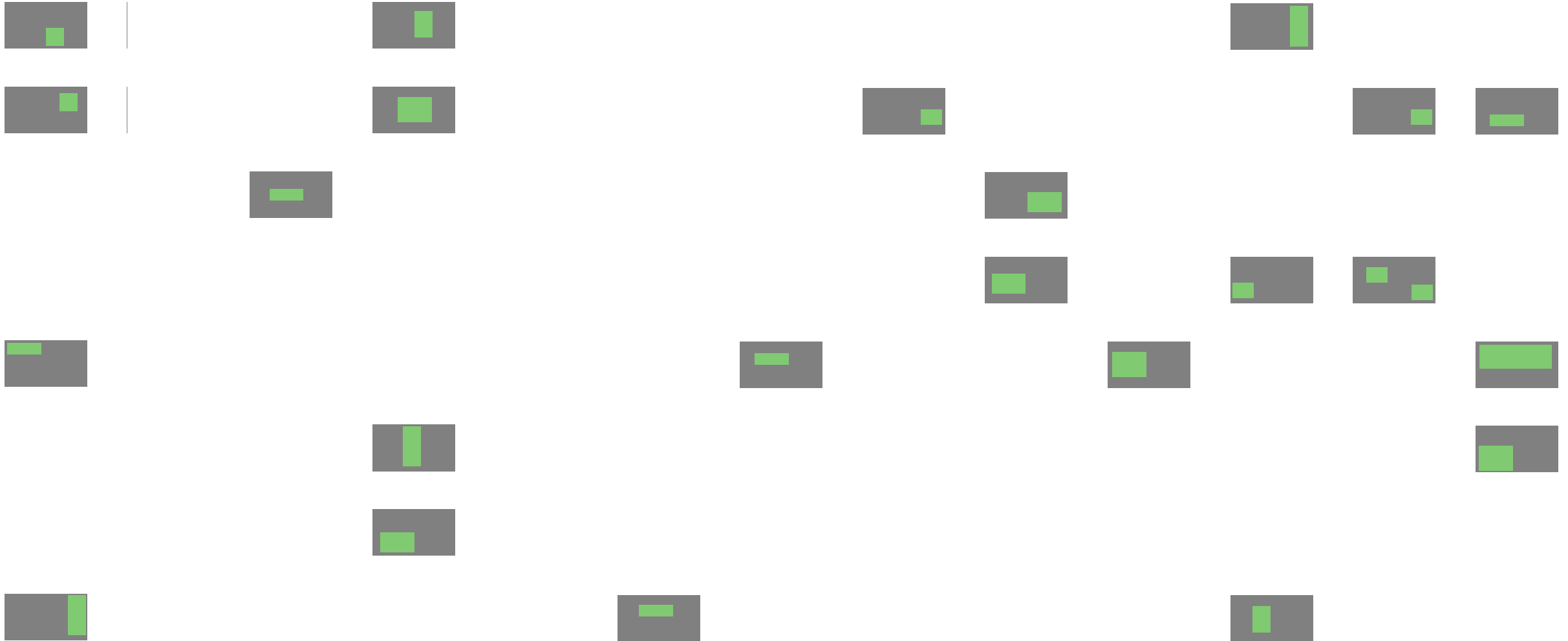




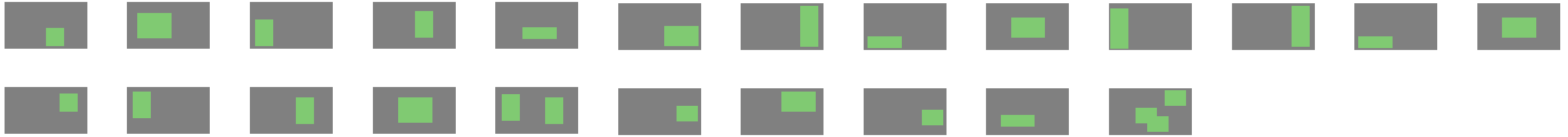
Selektion betroffener Testfälle

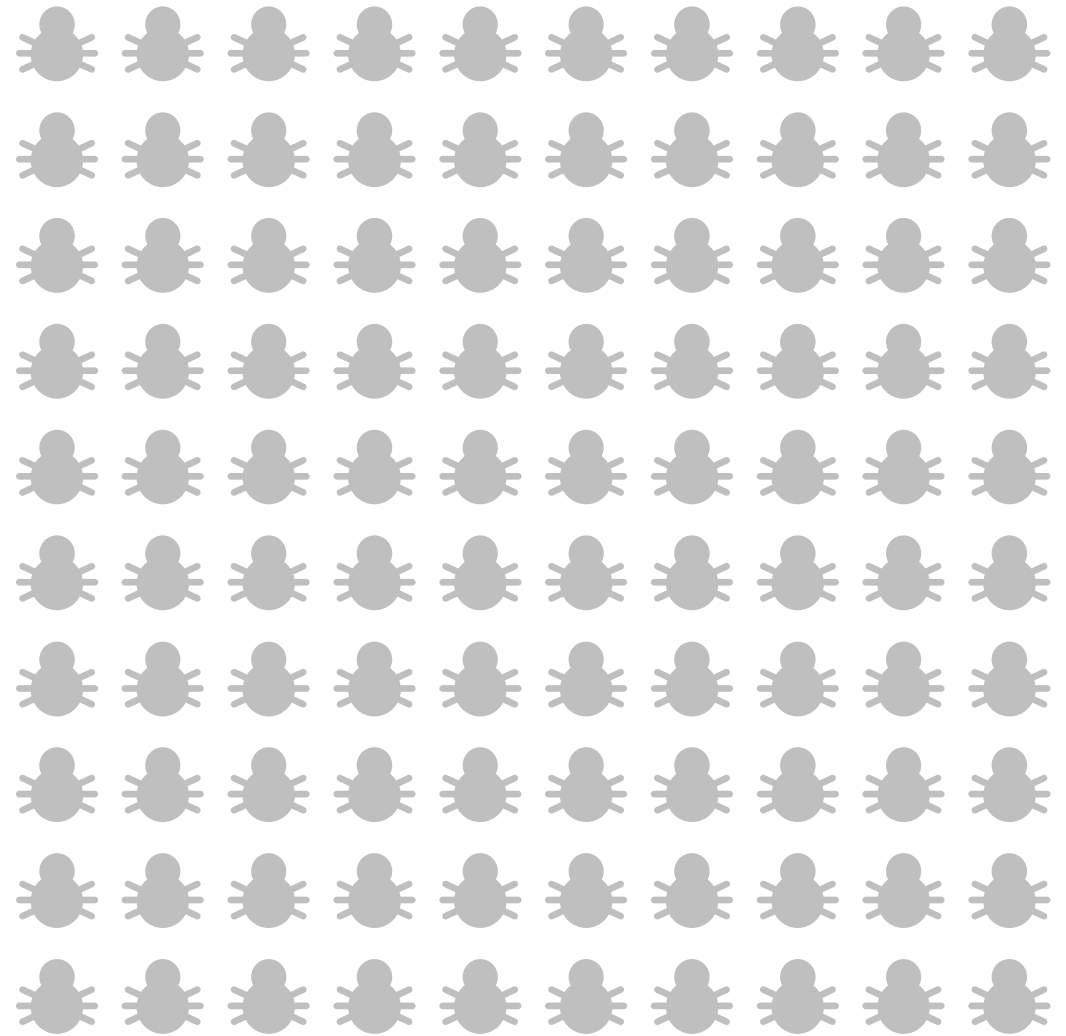
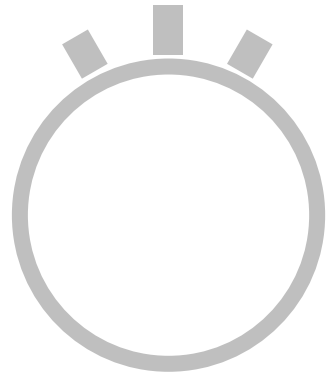


Selektion betroffener Testfälle



Selektion betroffener Testfälle

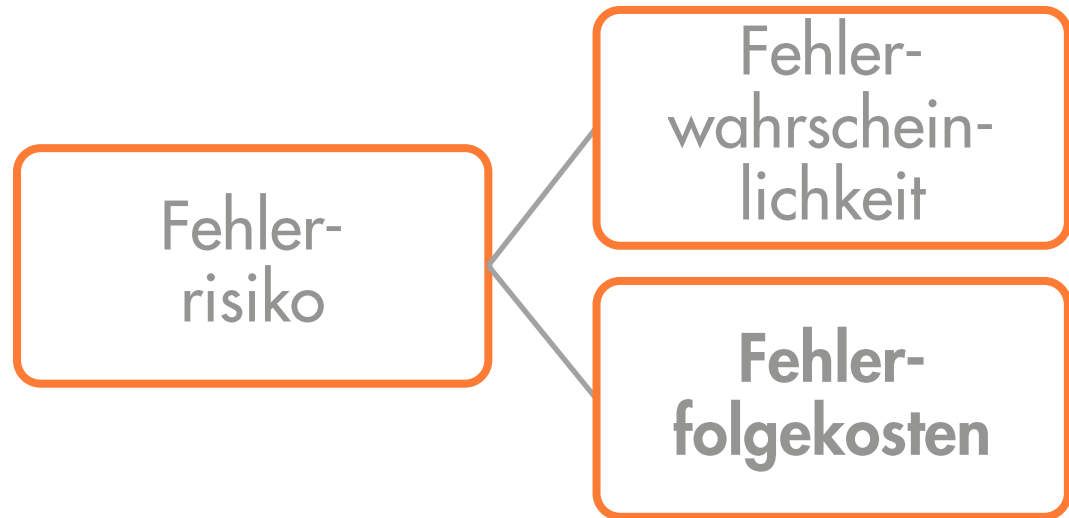




Test-Impact-Analyse: Fazit

- Die Test-Impact-Analyse beschleunigt langlaufende Testsuites.
- Diese können häufig während der Entwicklung ausgeführt werden.
- Schnelles Testfeedback und Reaktionsmöglichkeit
- Senkung der (Feld-) Fehlerwahrscheinlichkeit & effiziente Nutzung von Testressourcen

Agenda



Test-Gap-Analyse zur Identifikation von Testlücken
Test-Impact-Analyse zur Beschleunigung von Testsuites

Agiles Requirements Tracing zur Identifikation von kritischen, ungetesteten Anforderungen

Verifikationsmatrix

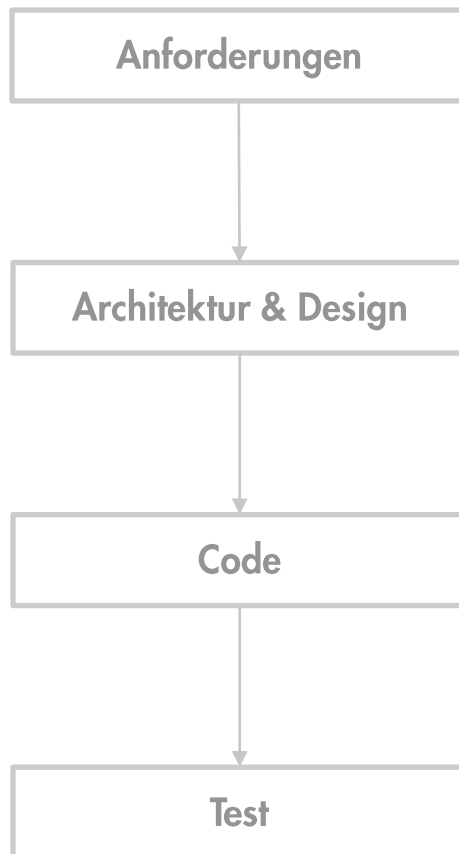
(Kritische) Anforderungen

Requirement Identifiers	Reqs Tested	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 UC	REQ1 TECH	REQ1 TECH	REQ1 TECH
		1.1	1.2	1.3	2.1	2.2	2.3.1	2.3.2	2.3.3	2.4	3.1	3.2	1.1	1.2	1.3	
Test Cases	321	3	2	3	1	1	1	1	1	1	2	3	1	1	1	
Tested Implicitly	77															
1.1.1	1	x														
1.1.2	2		x	x												
1.1.3	2	x											x			
1.1.4	1			x												
1.1.5	2	x												x		
1.1.6	1		x													
1.1.7	1			x												
1.2.1	2				x											
1.2.2	2					x										
1.2.3	2															
1.3.1	1															
1.3.2	1															
1.3.3	1															
1.3.4	1															
1.3.5	1															
etc....																
5.6.2	1														x	

Testfälle

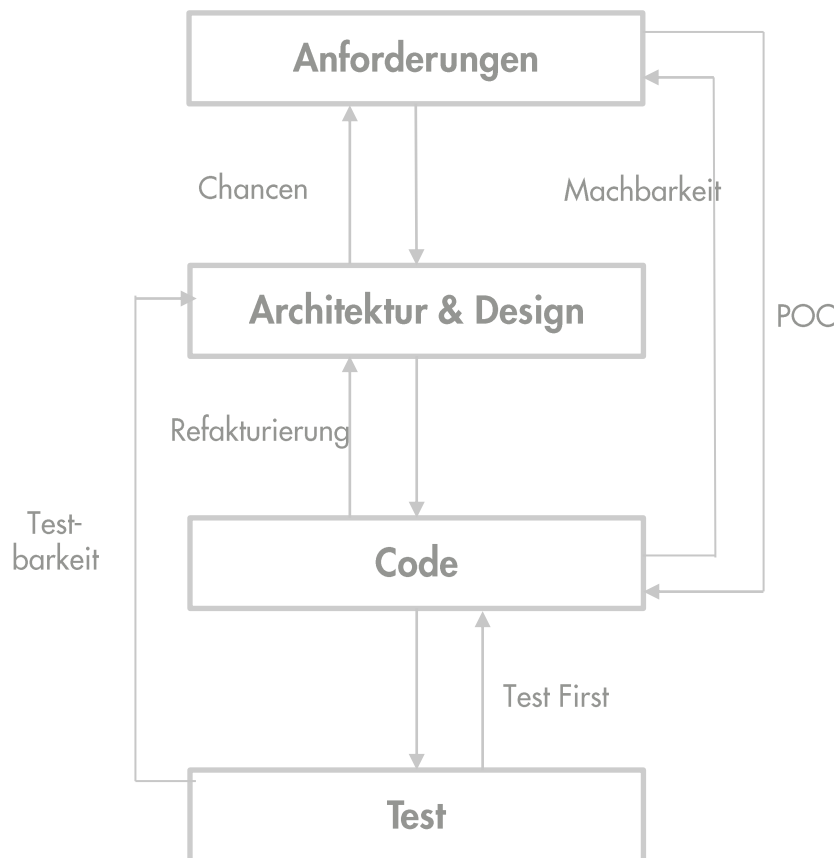
- Medizintechnik: IEC 62304, FDA GAMP, CFR 803.18
- Automotive: ISO 26262
- Luft- & Raumfahrt: DO-178C/ED-12C
- ...

Softwareentwicklung: Perspektive der meisten Standards



Tracing-Informationen aufzeichnen
während die Entwicklung sich vorwärts bewegt!

Softwareentwicklung in innovativen Projekten: Die Realität

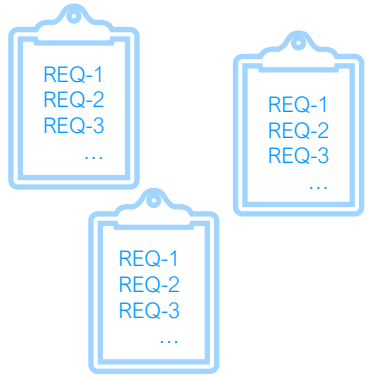


Softwareentwicklung ist eine Lernaktivität!

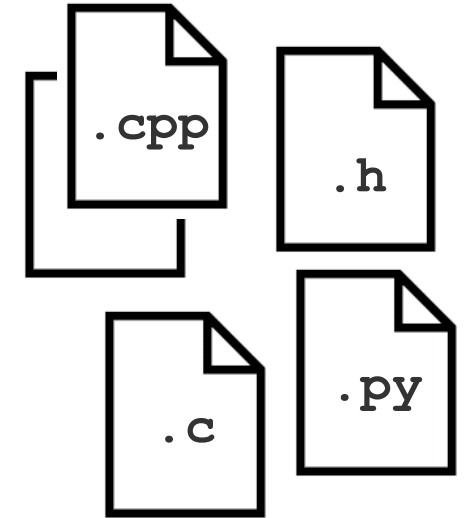
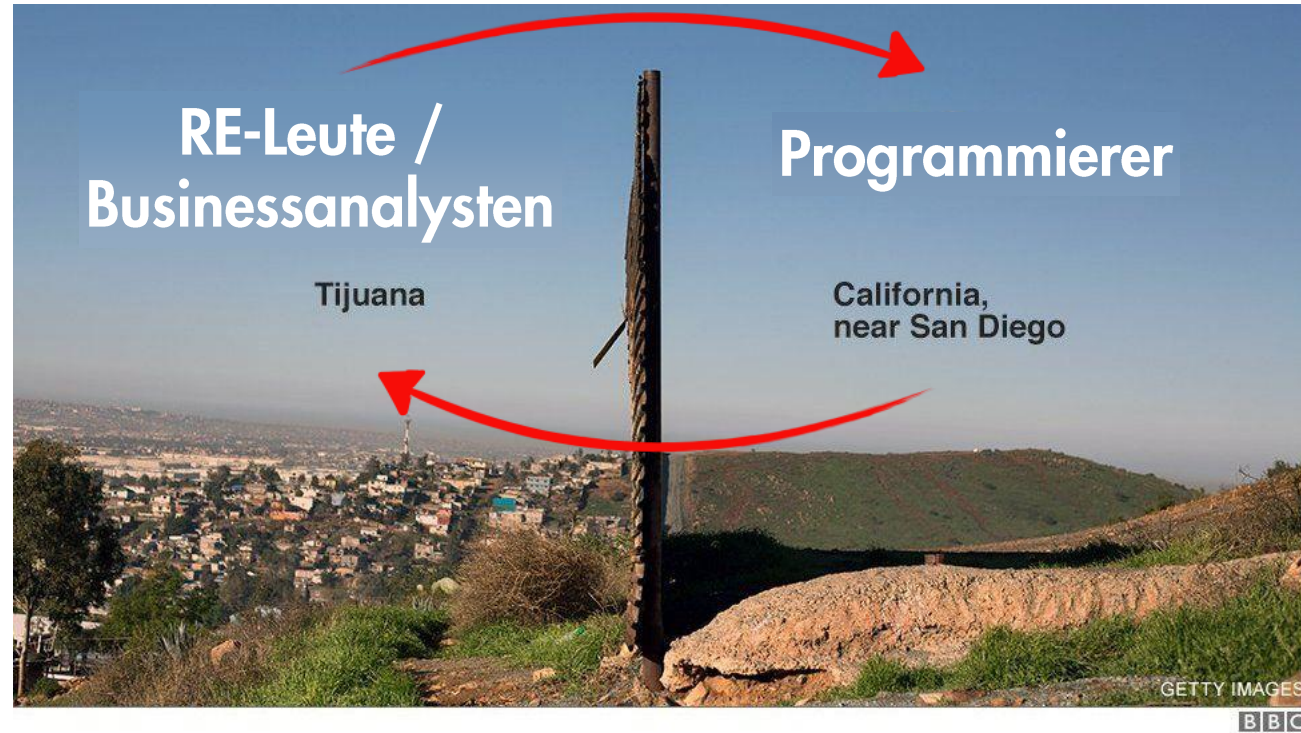
→ Anforderungen und Tests ändern sich häufig und teilweise unabhängig voneinander



Wie können wir Anforderungen und Tests synchron halten?



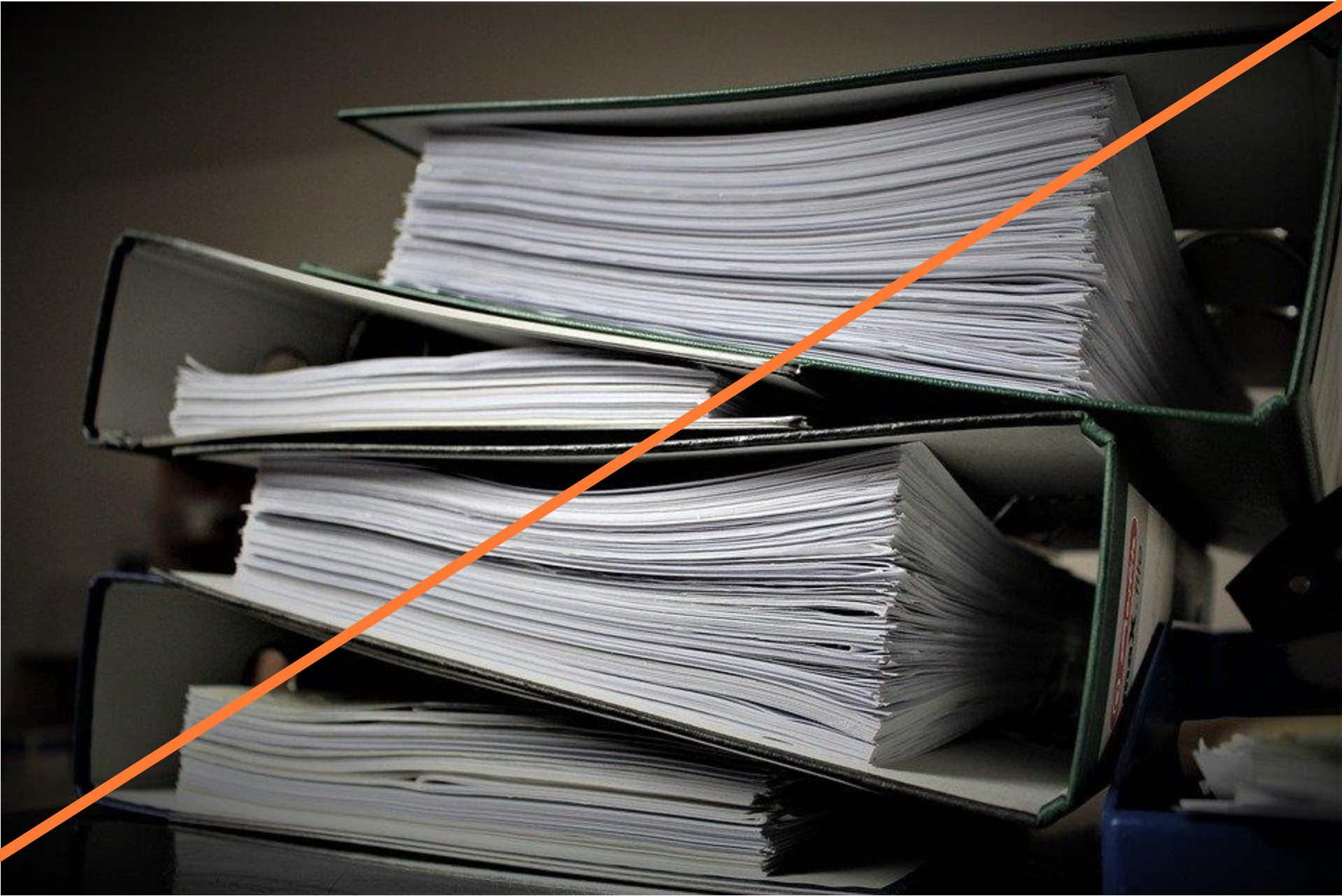
Doors/Polarion,
Text/UML



Git,
Jira,
IDEs

- „Mauer“ führt zu inkonsistenten und veralteten Anforderungen und Spezifikationen
- Dedizierte Prozesse und QS-Aktivitäten sind notwendig um das zu verhindern





Verifikationsmatrix: Datenquellen

(Kritische) Anforderungen

Requirement Identifiers	Reqs Tested	REQ1 UC 1.1	REQ1 UC 1.2	REQ1 UC 1.3	REQ1 UC 1.4	REQ1 UC 1.5	REQ1 UC 1.6	REQ1 UC 1.7	REQ1 UC 1.8	REQ1 UC 1.9	REQ1 UC 1.10	REQ1 UC 3.2	REQ1 TECH 1.1	REQ1 TECH 1.2	REQ1 TECH 1.3
Test Cases	321	3	2									3	1	1	1
Tested Implicitly	77														
1.1.1	1	x													
1.1.2	2		x	x											
1.1.3	2	x											x		
1.1.4	1				x										
1.1.5	1														
1.1.6	1														
1.1.7	1														
1.1.8	1														
1.1.9	1														
1.1.10	1														
1.1.11	1														
1.1.12	1														
1.1.13	1														
1.1.14	1														
1.1.15	1														
1.1.16	1														
1.1.17	1														
1.1.18	1														
1.1.19	1														
1.1.20	1														
1.1.21	1														
1.1.22	1														
1.1.23	1														
1.1.24	1														
1.1.25	1														
1.1.26	1														
1.1.27	1														
1.1.28	1														
1.1.29	1														
1.1.30	1														
1.1.31	1														
1.1.32	1														
1.1.33	1														
1.1.34	1														
1.1.35	1														
1.1.36	1														
1.1.37	1														
1.1.38	1														
1.1.39	1														
1.1.40	1														
1.1.41	1														
1.1.42	1														
1.1.43	1														
1.1.44	1														
1.1.45	1														
1.1.46	1														
1.1.47	1														
1.1.48	1														
1.1.49	1														
1.1.50	1														
etc....															
5.6.2	1														x

Anforderungsmanagement-Werkzeuge



Testfälle

```

58      /** Tests restoring a backup contains the same data. RE-123, RE-789 */
59      @Test
60      public void testBackupAndRestore() throws StorageException {
61          IStorageSystemProvider inputProvider = createInputProvider();
62
63          File backupFile = new File(tempDir, "snapshotBackup.zip");
64          StorageSnapshotBackupUtils.writeSnapshotBackup(backupFile,
65              StorageLayerTestUtils.createStorageSystemProvider());
66
67          IStorageSystemProvider outputProvider = new StorageSystemProvider() {
68              @Override public void writeSnapshotBackup(File backupFile,
69                  StorageSystemProvider provider) throws StorageException {
70                  StorageLayerTestUtils.createStorageSystemProvider().writeSnapshotBackup(backupFile, provider);
71              }
72          };
73          StorageSnapshotBackupUtils.restoreSnapshotBackup(backupFile, outputProvider);
74          assertEquals("SameData", inputProvider, outputProvider);

```

Testmanagement-Werkzeuge



RE-123, RE-789

Linked Requirements 2

ID	Subject	Status
REQ-123	A backup needs to be a ZIP file	In Development
REQ-789	A backup needs to contain only insensitive data	Rework

Automatische Generierung der Verifikationsmatrix

Test Result Databases



(Kritische) Anforderungen

Testfälle

Show test results from: All

Tests	Specification Items	REQ DP-539	REQ DP-540	TST DP-547	TST DP-548
	Σ	1	0	1	1
data_structures/tests/stack_tests.c/StackTests/Push	1				
data_structures/tests/stack_tests.c/StackTests/Pop	1				
data_structures/tests/stack_tests.c/StackTests/Peek	1				

Legend: Linux (blue), Windows (blue), SKIPPED (yellow), FAILURE (red)





Sync-Nachverfolgung an Pull/ Merge-Requests

Sync-Nachverfolgung

- Identifikation und Nachverfolgung von Beziehungen zwischen Anforderungen und Testfällen
- Nachverfolgung von Änderungen in Testcode
- Checkliste für Entwickler und Reviewer bei Änderungen an verlinktem Testcode

^ Impacted Requirements

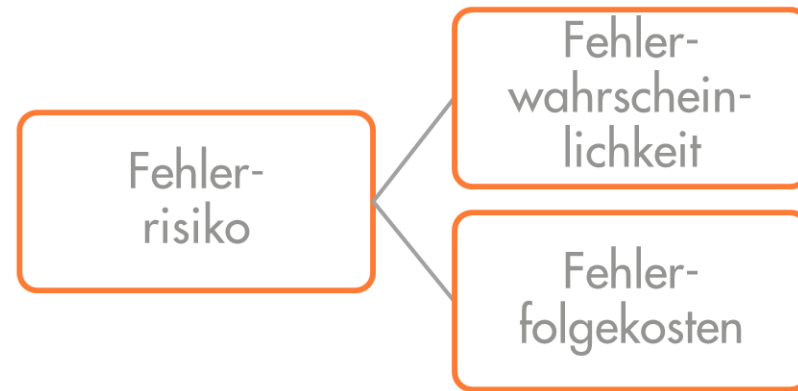
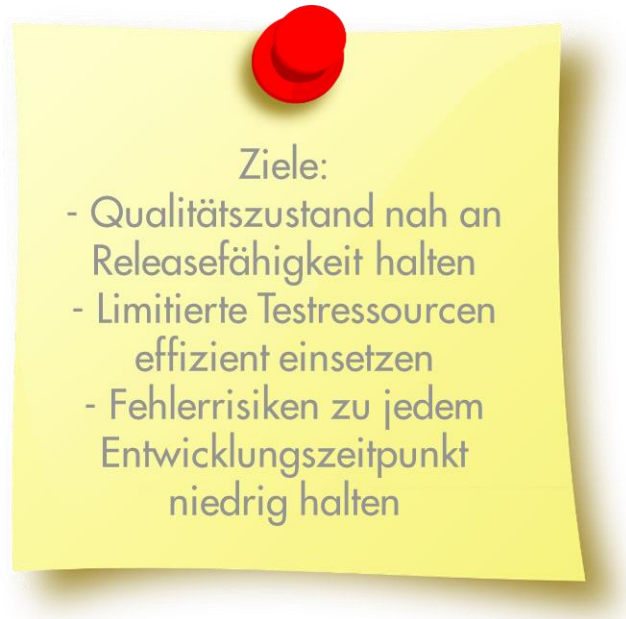
The following 4 requirements are related to the changes in this merge request. Please ensure that these requirements and the referencing code are still in line with one another.

Type	Message	Reference location
	Function declaration StackTests_Peek now references requirement DP-539	data_structures/tests/stack_tests.c: 37
	Function declaration StackTests_Pop has been modified and references requirement DP-547	data_structures/tests/stack_tests.c: 22
	Function declaration StackTests_Push no longer references requirement DP-541	data_structures/tests/stack_tests.c: 11
	Function declaration StackTests_Push no longer references requirement DP-544	data_structures/tests/stack_tests.c: 11

Agiles Requirements Tracing: Fazit

- Generierung einer stets aktuellen Verifikationsmatrix um ungetestete, kritische Anforderungen zu identifizieren und Testabdeckung nachzuweisen
- Generierung einer Checkliste bei Änderungen von Testcode um Anforderungen und Tests synchron zu halten
- Insgesamt: Vermeidung von hohen Fehlerfolgekosten bei Nichteinhaltung von kritischen Anforderungen

Risikobasiertes Testen während der Entwicklung: Fazit



Test-Gap-Analyse zur Identifikation von Testlücken
Test-Impact-Analyse zur Beschleunigung von Testsuites

Agiles Requirements Tracing zur Identifikation von kritischen, ungetesteten Anforderungen

- Fehlerrisiken durch änderungsgetriebenes Testen (Whitebox) und anforderungsgetriebenes Testen (Blackbox) schon während der Entwicklung effizient managen
- Entwicklungs- und Testdaten strukturiert erheben und auswerten

Kontakt



Interesse an Folien?

→ http://cqse.eu/ese_2022_vortrag bzw. QR-Code

Interesse an Testanalysen-/methodiken?

Interesse an Beratungs- und/ oder Entwicklungsjob?

Dr. Tobias Röhm

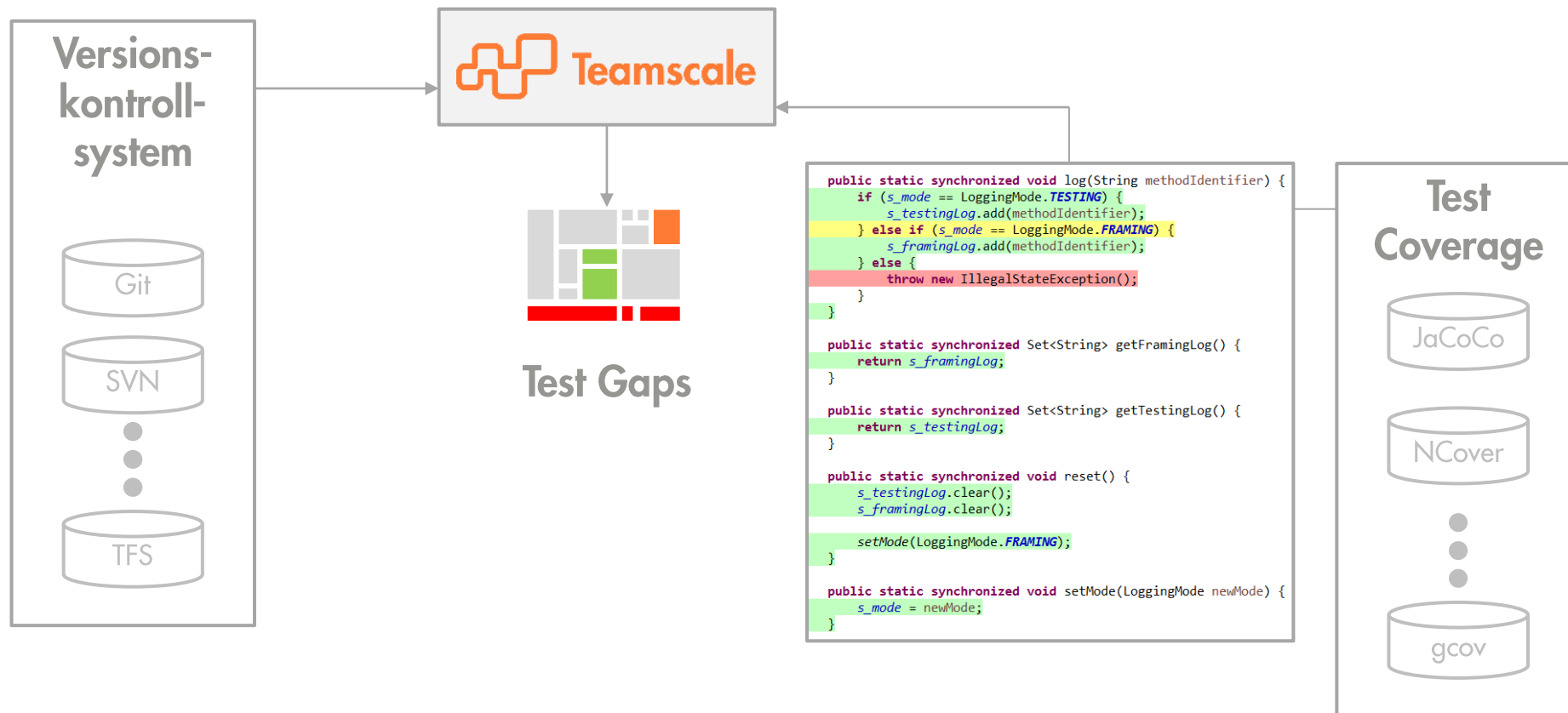
@langelot · roehm@cqse.eu · +49 159 04330842

Danke an die Kollegen – insbesondere Martin Feilkas, Elmar Jürgens, Dennis Pagano, Roman Haas, Raphael Nömmer und Fabian Streitl - für ihre Beiträge!

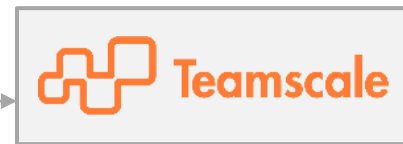
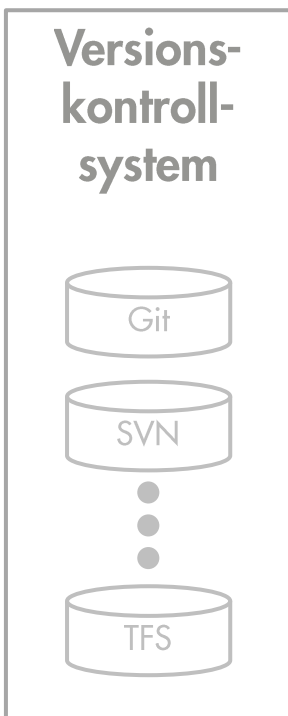
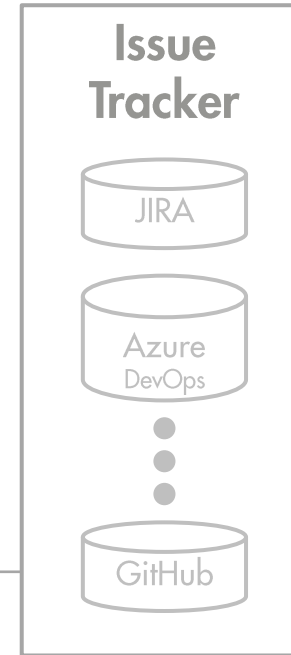


BACKUP

Test-Gap-Analyse: Analyseschema



CR#9838: Added TODO	26.07.16 16:38
CR#9838: Adjust naming	26.07.16 15:33
CR#9533: RED	26.07.16 15:13
CR#9533: GREEN	26.07.16 15:12
CR#10181: Added new finding for deprecated classes, methods and fields	26.07.16 14:43
CR#10037: Moved ReviewMetricsSynchronizer to Crucible package and made some improvements to its internal structure	26.07.16 14:31
CR#10037: Updated aggregation strategy of open reviews so each review is only counted once, even over multiple files	26.07.16 13:04
CR#10203: Fixed "field could be made final" for Java interfaces	26.07.16 12:16
CR#10200: Rename pathRestriction -> subPath (1)	26.07.16 11:35
CR#10172: Removed unwanted colons from headers in the commit view of the activity perspective	26.07.16 11:20
CR#9838: Fix: only one color of a threshold is specified in a corridor	26.07.16 11:14
CR#0: Fix findings	26.07.16 11:01
CR#9838: minor improvement	26.07.16 10:56
CR#10199: Mail notifications do now support starTLS	26.07.16 10:52
CR#9533: working on developer feedback	26.07.16 09:50
CR#9838: Amend last commit	26.07.16 09:38
CR#9838: minor refactoring	26.07.16 09:05
CR#9838: Fix NPE	26.07.16 09:01



Issue Test Gaps

```

public static synchronized void log(String methodIdentifier) {
    if (s_mode == LoggingMode.TESTING) {
        s_testingLog.add(methodIdentifier);
    } else if (s_mode == LoggingMode.FRAMING) {
        s_framingLog.add(methodIdentifier);
    } else {
        throw new IllegalStateException();
    }
}

public static synchronized Set<String> getFramingLog() {
    return s_framingLog;
}

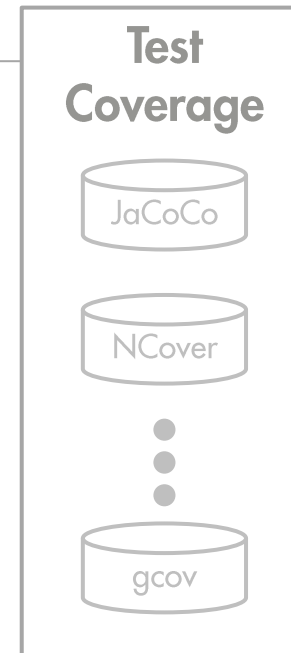
public static synchronized Set<String> getTestingLog() {
    return s_testingLog;
}





public static synchronized void reset() {
    s_testingLog.clear();
    s_framingLog.clear();
}

setMode(LoggingMode.FRAMING);


public static synchronized void setMode(LoggingMode newMode) {
    s_mode = newMode;
}


```



Issue # ▾	Subject		Test Gap
TS-14421	Get rid of TestGapSynchronizer block	Done 	0% 
TS-14733	Remove Dataflow blocks	Done 	22% 

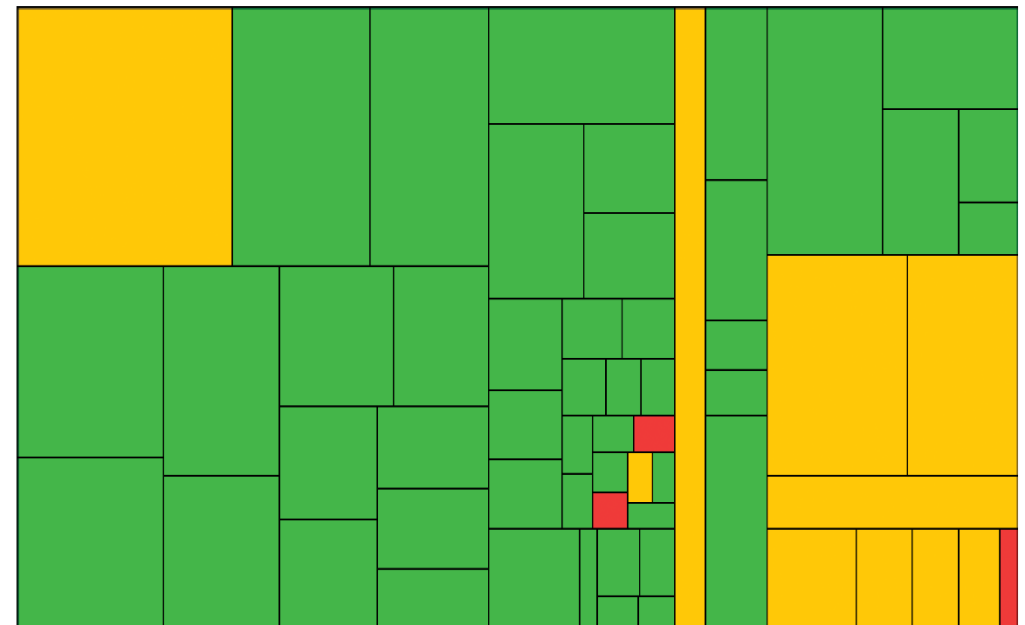
Done Issue TS-14733 - Remove Dataflow blocks


















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

Assignee: 

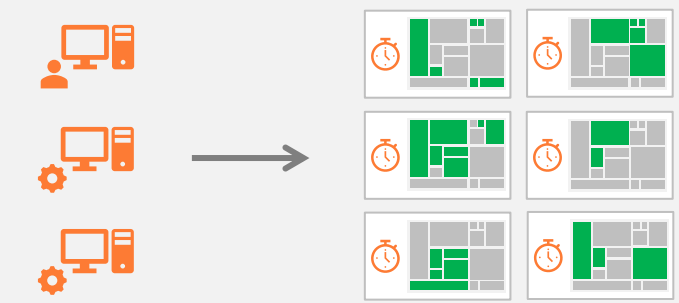
Project	Type	Priority	Resolution	Fix Version
TS	Maintenance	Normal	Green	Teamscale 4.5
Component	Labels	Affected Version	Customer	Customer Issue
Backend	Performance			
Epic Name	Freshdesk URL	Merge Request		
		https://git.cqse.eu/cqse/teamscale/3621		

Aug 15 2018 12:37–Now | Test Gap: 22%



Issue # ▼	Subject	Done		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% 

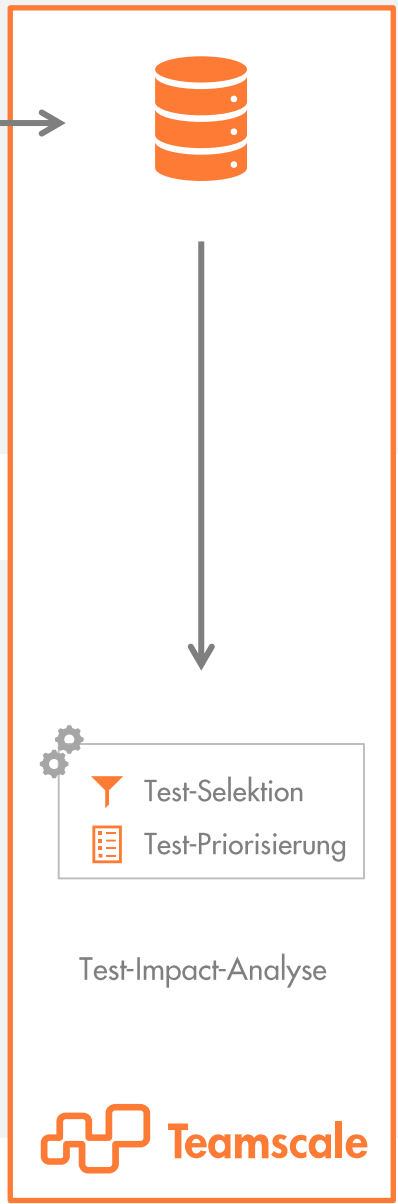
Initiale Aufzeichnung aller Tests



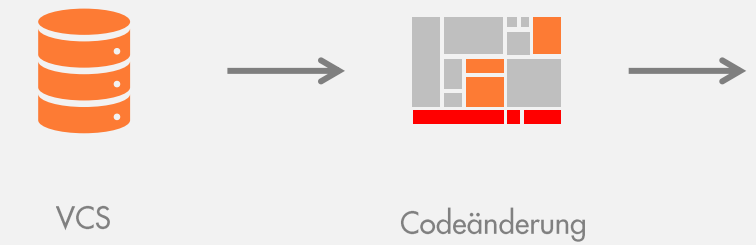
Ausführung aller Tests

Coverage & Laufzeit für alle Tests

Test-Impact-Analyse

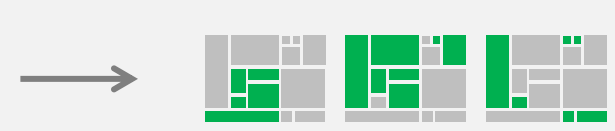


CI Pipeline



VCS

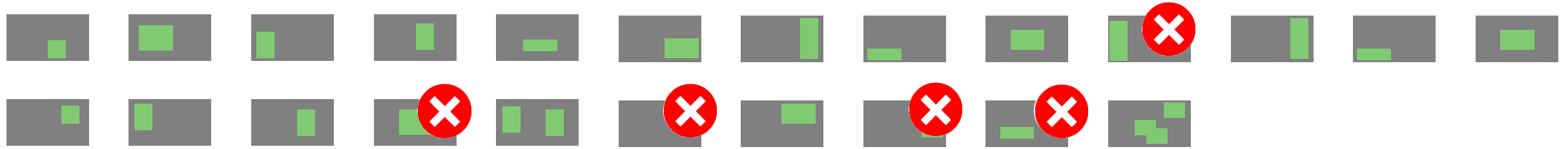
Codeänderung



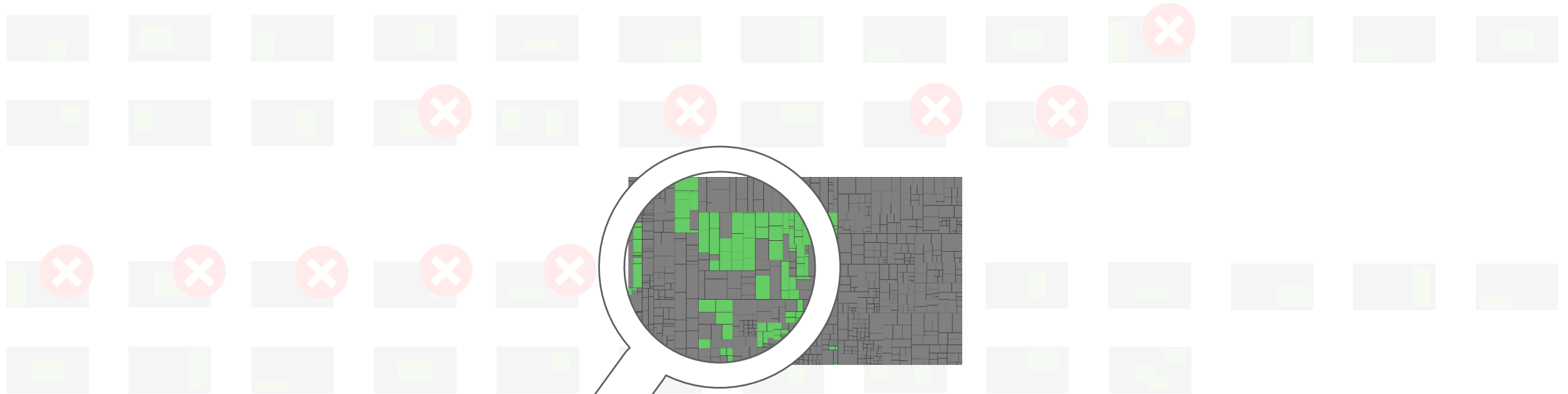
Änderungsrelevante und sortierte Testfälle



Schritt 2: Priorisierung selektierter Testfälle



Schritt 2: Priorisierung selektierter Testfälle



Change coverage


Execution time

Anforderungsmanagement-
Werkzeuge





Polarion JIRA

Versionskontrollsysteme



git ...

Testergebnis-Datenbank



Artifactory

 Teamscale

Anforderungsanalyse

- Lesen von Anforderungen, Eigenschaften und Beziehungen
- Nachverfolgung von Historie und Metriken
- Abfragemöglichkeit

Verifikation

- Tests als „first class citizens“
- Identifikation von Tests im Testcode: cmocka, gtest, custom framework
- Nachverfolgung von Links zwischen Testcode und Anforderungen

Testergebnis-Analyse

- Einlesen von Testergebnissen und Abbildung auf den Testcode
- Nachverfolgung der Historie



	DP-547	DP-541	DP-539	DP-548	
	Σ	1	0	1	1
data_s-urtests/stack_tests.c#StackTests_Pop	1	∅			
data_s-urtests/stack_tests.c#StackTests_Push	1			∅	
data_s-urtests/stack_tests.c#StackTests_Peek	1		∅		

PASSED verify_stack@Feb 12 2021 18:32 20s

FAILURE verify_stack@Feb 11 2021 13:25 20s

SKIPPED verify_stack@Feb 11 2021 13:17 20s



Projectmanager



Qualitätsingenieur



Entwickler,
Reviewer