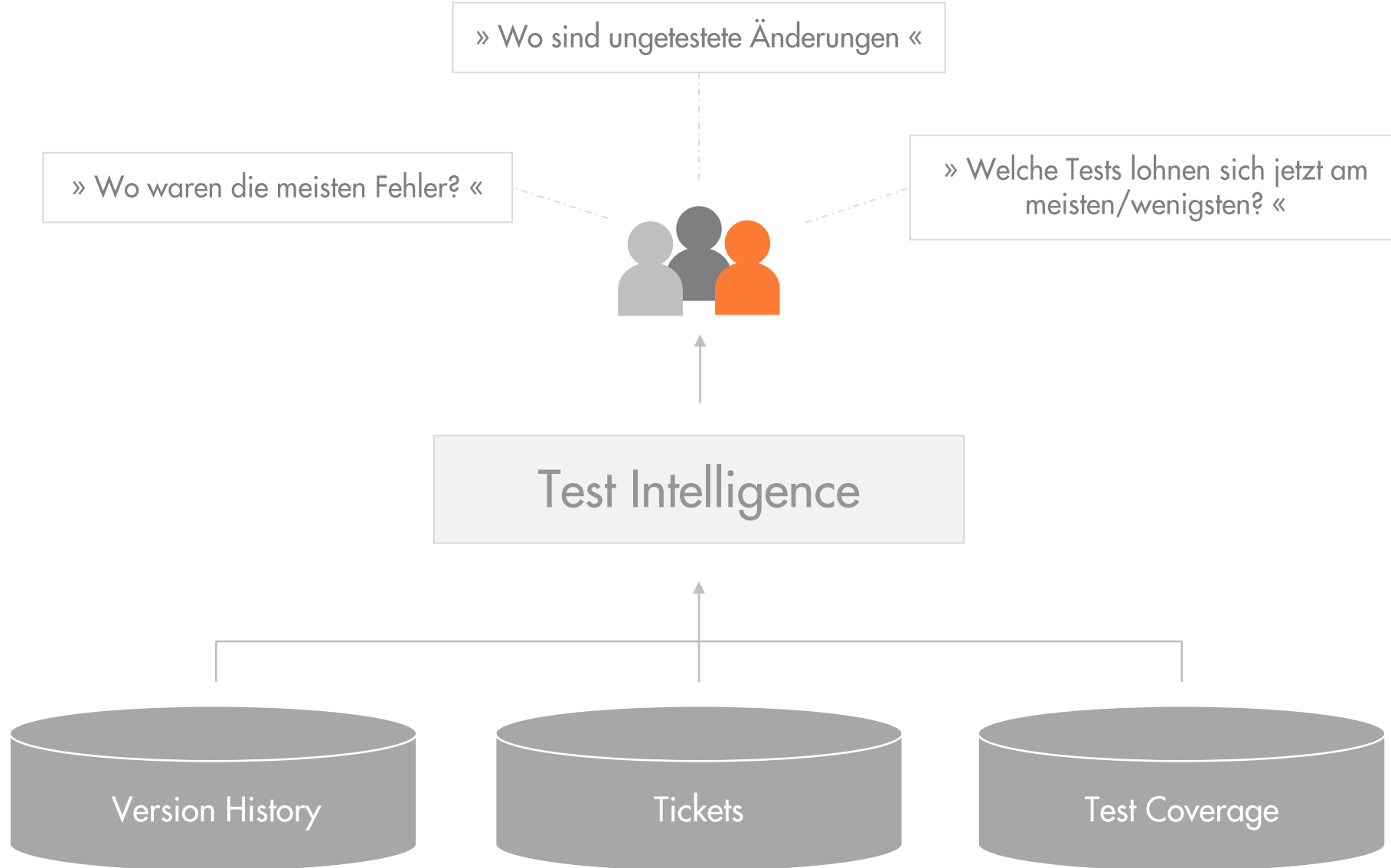


Test Intelligence

für historisch gewachsene Systeme



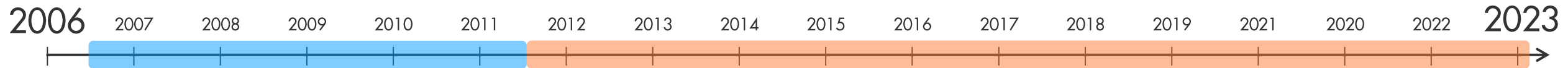


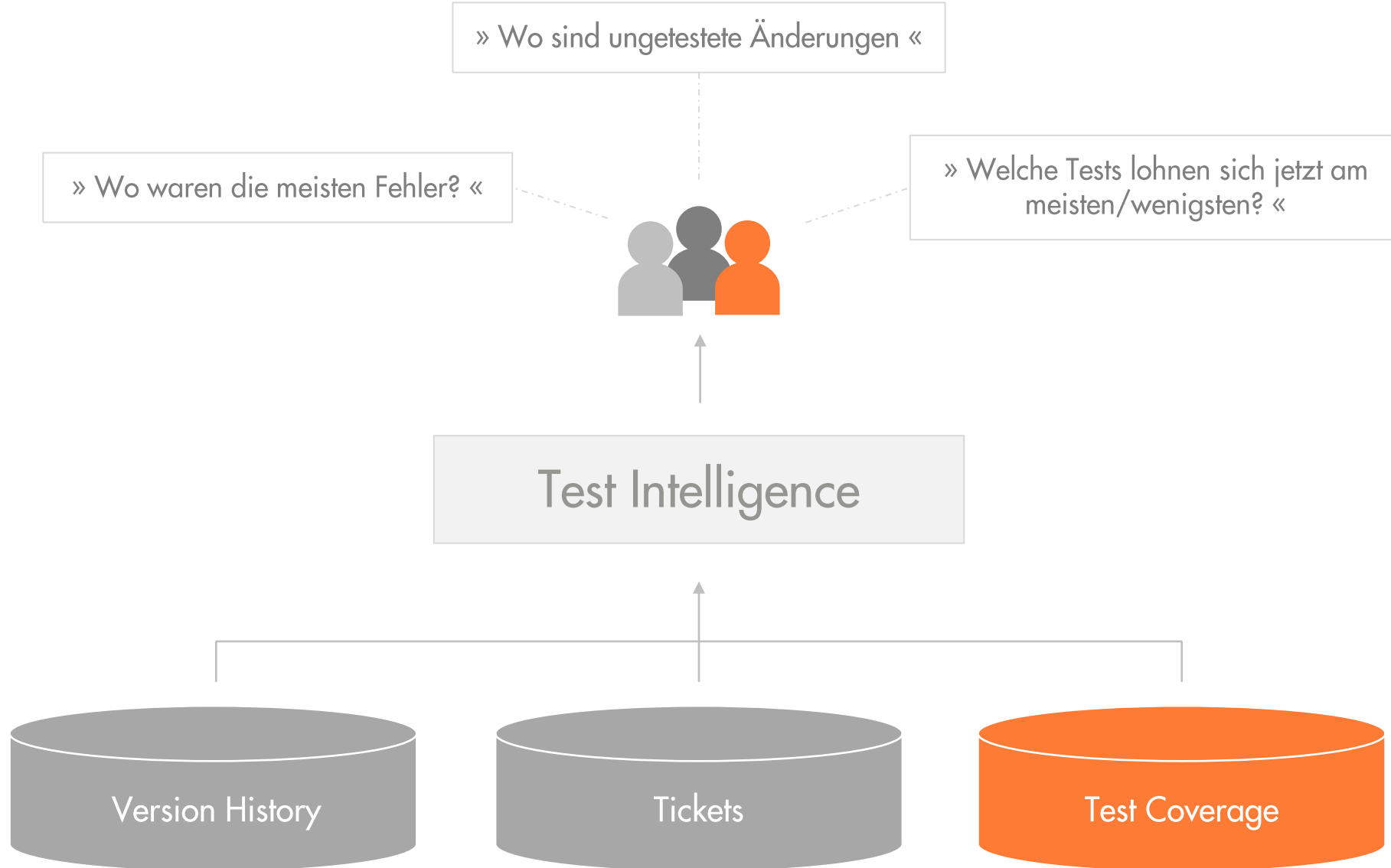


TUM



CQSE





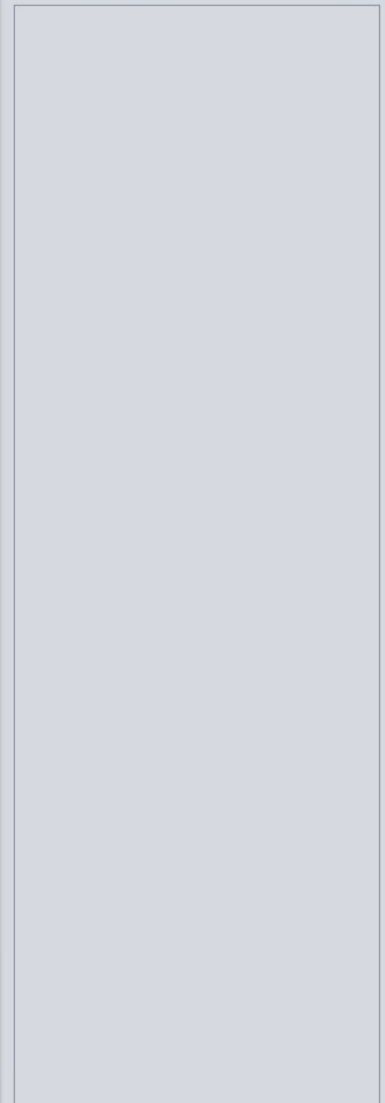
Sample Only the Active Layer/Mask

Untitled1 x Picture1.png x



Layers

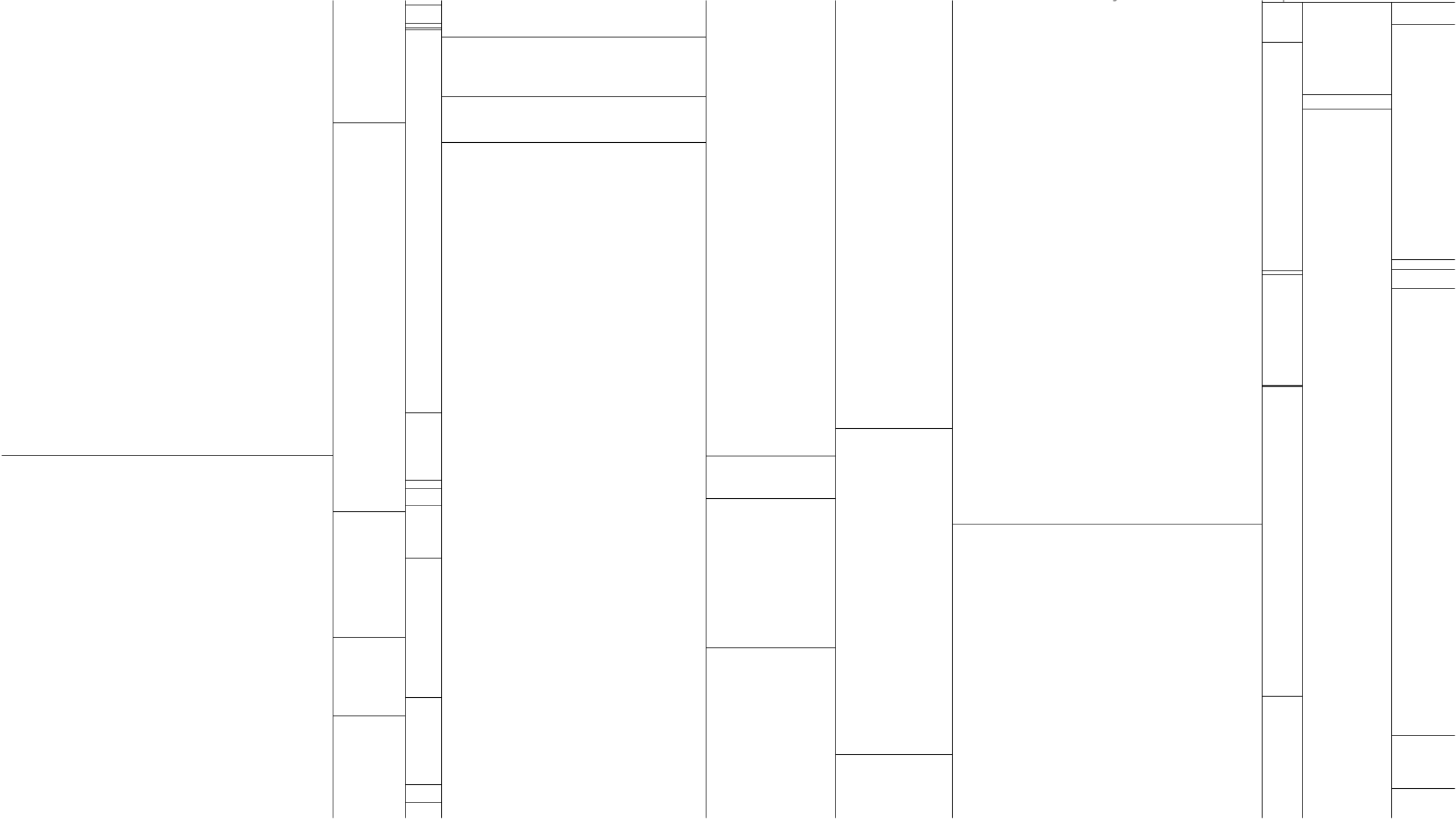
Opacity: 100 % Normal

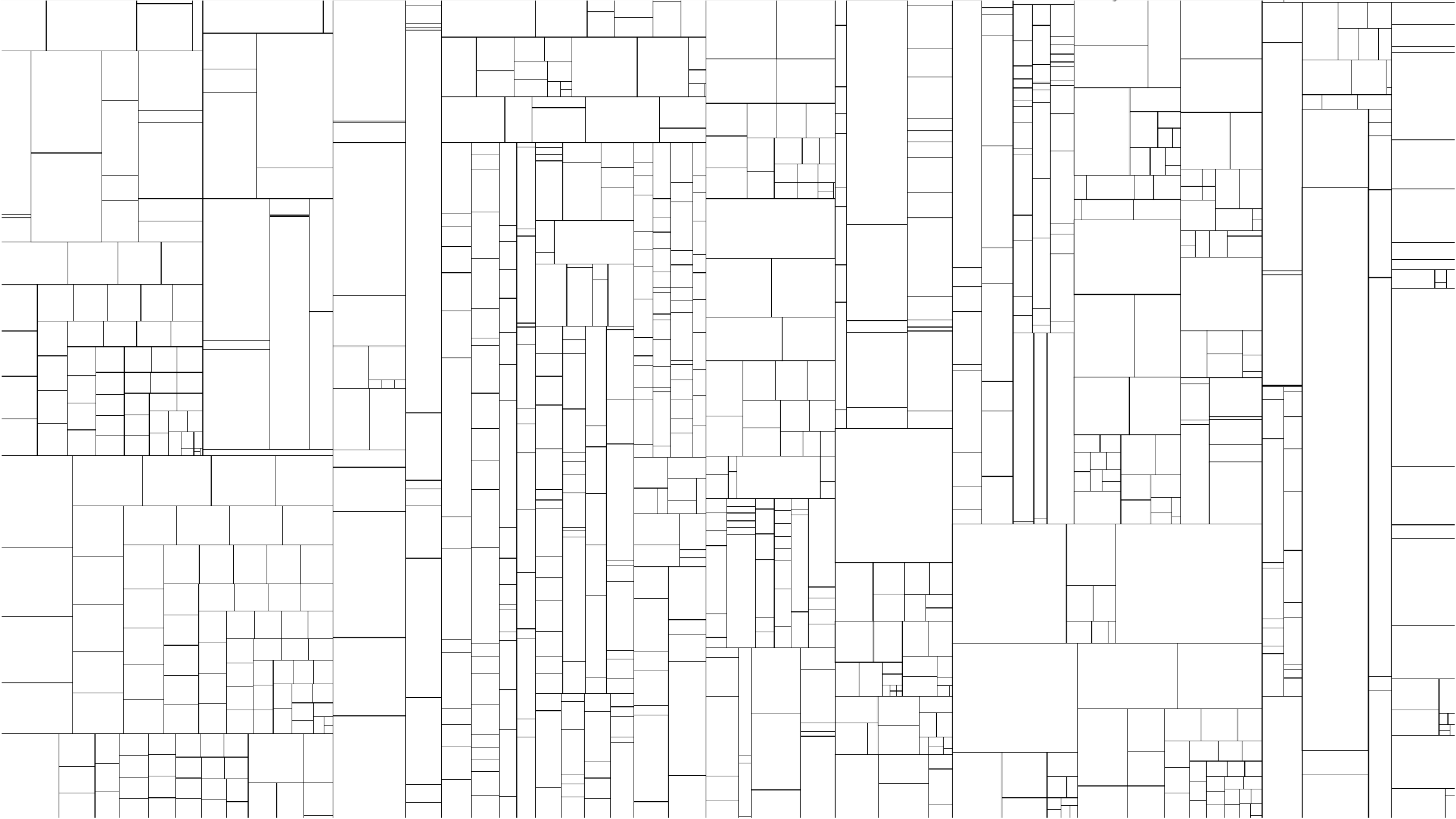


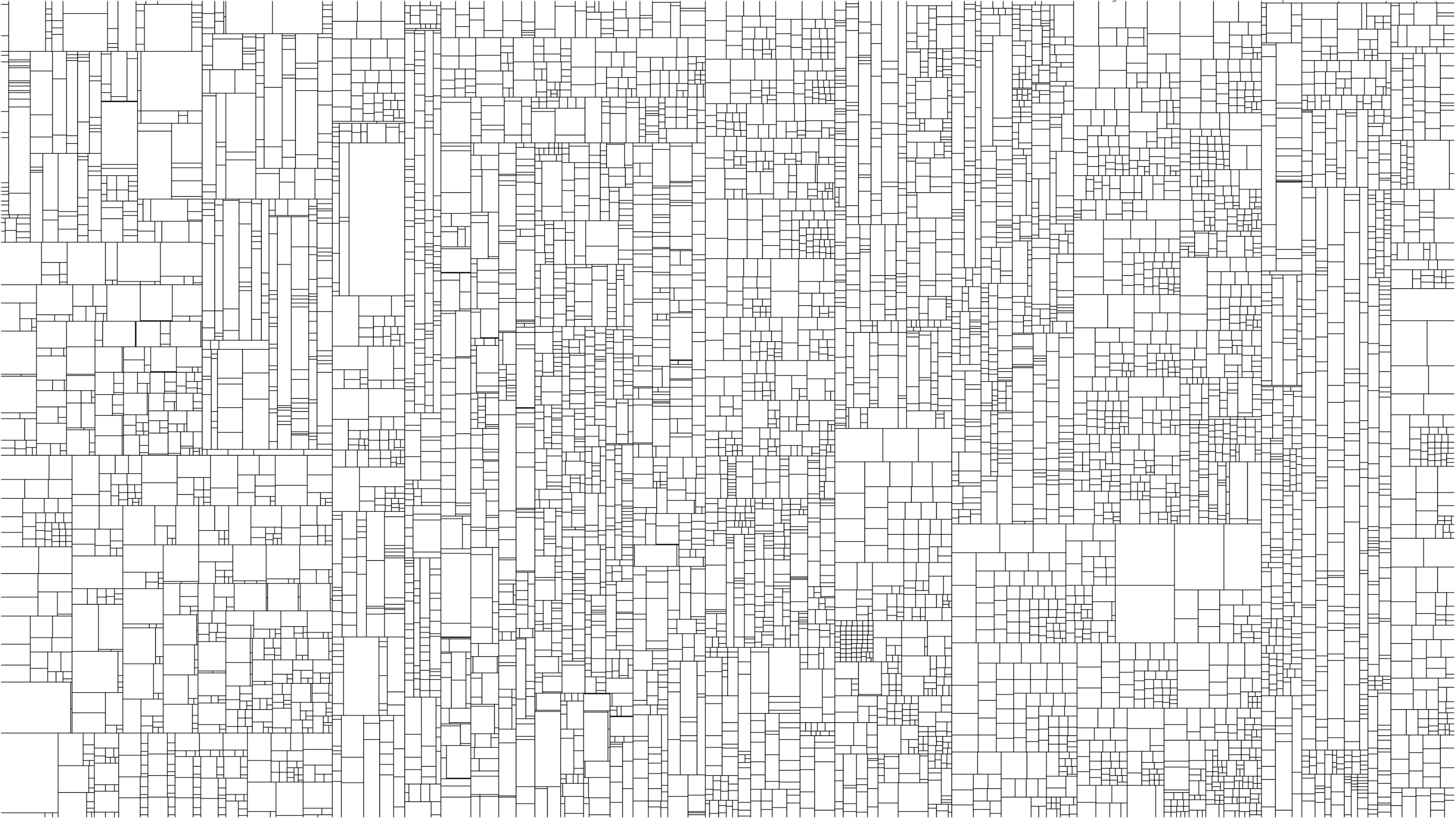
layer 1

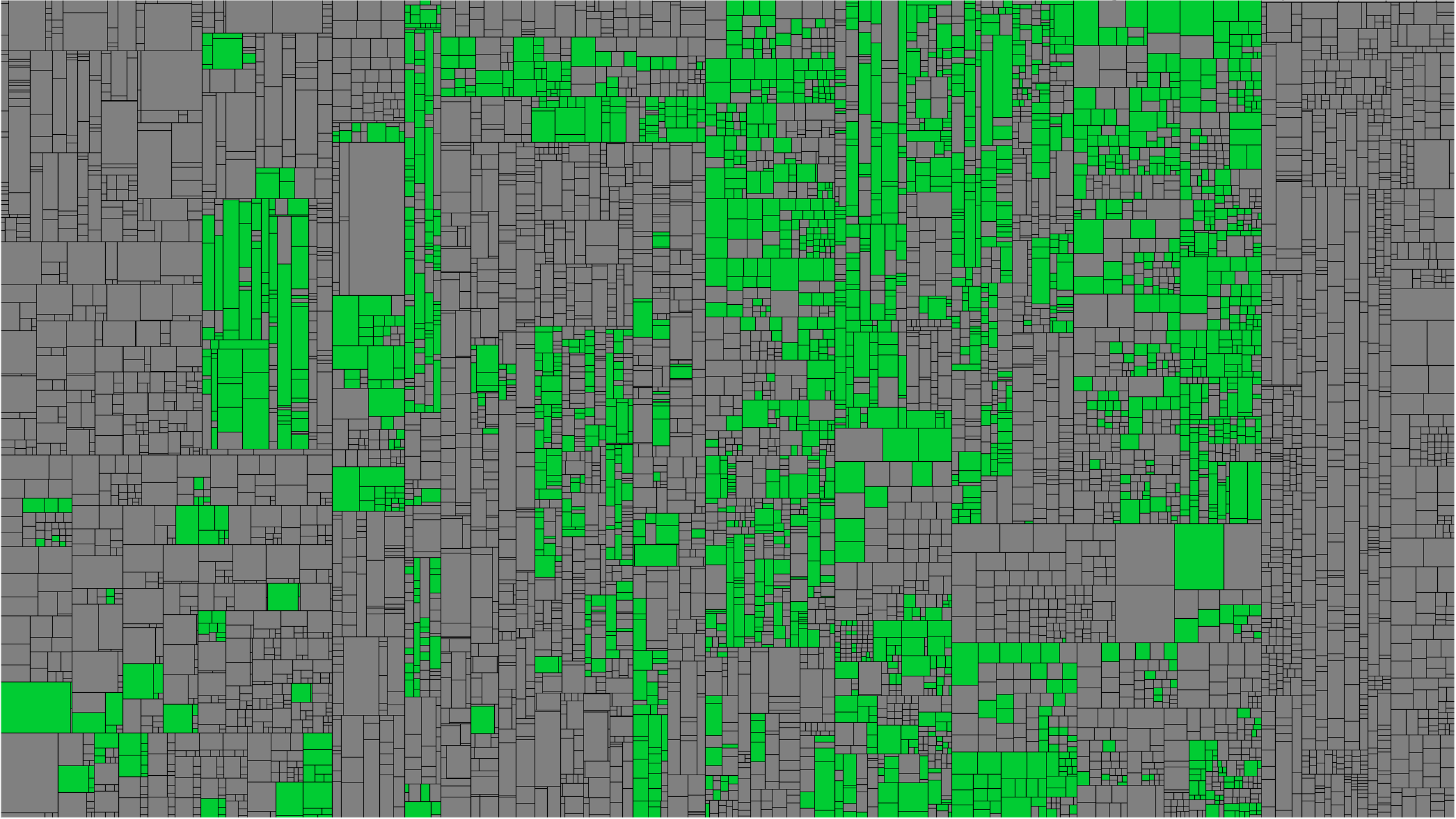
+ - [] [] []

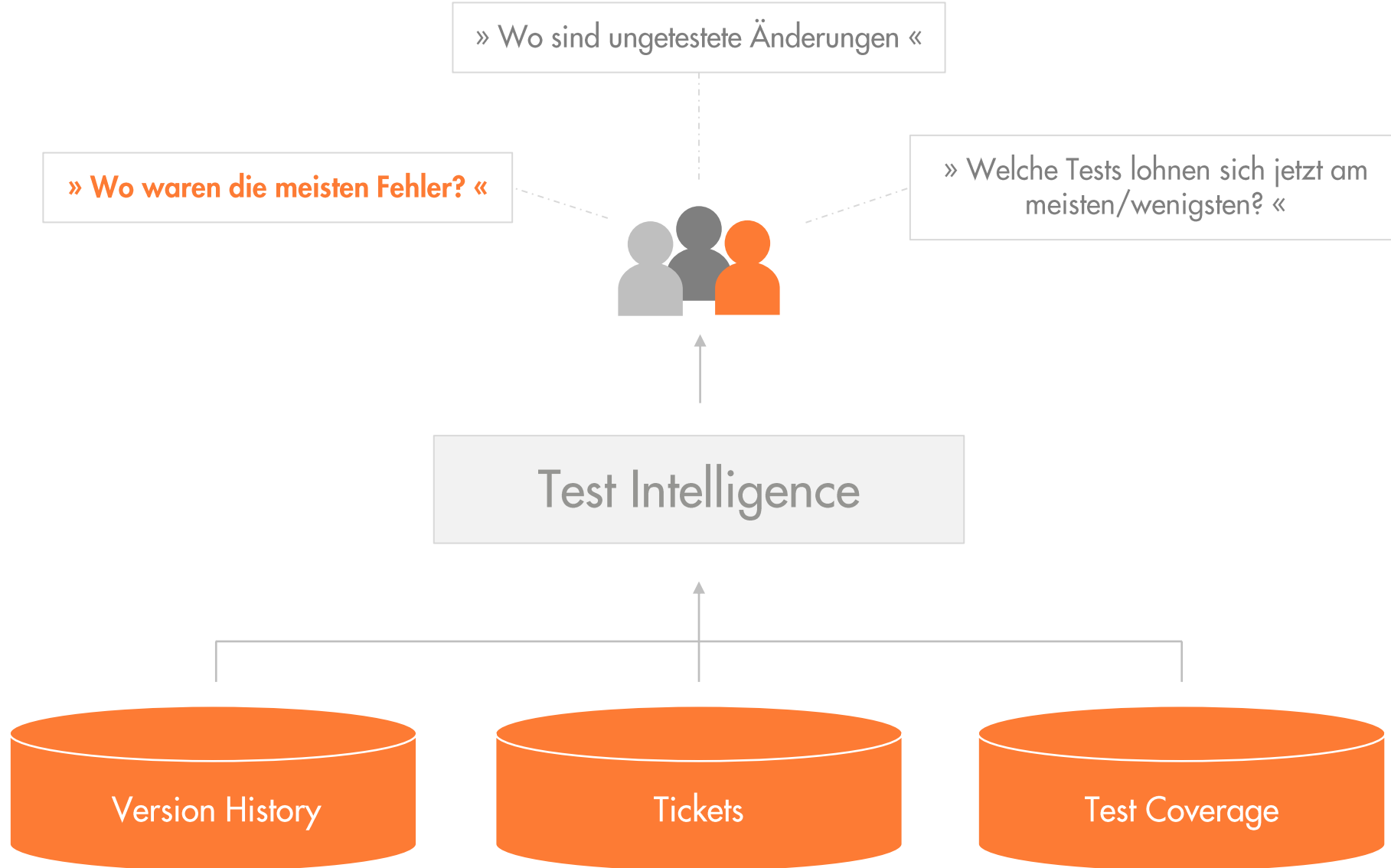
```
114
115i     private static void createAndShowGUI(String[] args) {
116         assert calledOnEDT() : threadInfo();
117
118         Messages.setMsgHandler(new GUIMessageHandler());
119
120         // GlobalKeyboardWatch.showEventsSlowerThan(100, TimeUnit.MILLISECONDS);
121
122         Theme theme = Themes.DEFAULT;
123         // if a LaF was set from the command line, then don't override it
124         if (System.getProperty("swing.defaultlaf") == null) {
125             theme = AppPreferences.loadTheme();
126             Themes.install(theme, false, true);
127         }
128
129         int uiFontSize = AppPreferences.loadUIFontSize();
130         String uiFontType = AppPreferences.loadUIFontType();
131
132         Font defaultFont = UIManager.getFont("defaultFont");
133         if (defaultFont != null) { // if null, we don't know how to set the font
134             if (uiFontSize != 0 || !uiFontType.isEmpty()) {
135                 Font newFont;
136                 if (!uiFontType.isEmpty()) {
137                     newFont = new Font(uiFontType, Font.PLAIN, uiFontSize);
138                 } else {
139                     newFont = defaultFont.deriveFont((float) uiFontSize);
140                 }
141
142                 FontUIResource fontUIResource = new FontUIResource(newFont);
143                 UIManager.put("defaultFont", fontUIResource);
144
145                 if (theme.isNimbus()) {
146                     UIManager.getLookAndFeel().getDefaults().put("defaultFont", fontUIResource);
147                 }
148             }
149         }
150
151         var pw = PixelitorWindow.get();
152         Dialogs.setMainWindowInitialized(true);
153
154         // Just to make 100% sure that at the end of GUI
155         // initialization the focus is not grabbed by
156         // a textfield and the keyboard shortcuts work properly
157         FgBgColors.getGUI().requestFocus();
158
159         TipsOfTheDay.showTips(pw, false);
160
161         MouseZoomMethod.load();
162         PanMethod.load();
163
164         // The IO-intensive preloading of fonts is scheduled
165         // to run after all the files have been opened,
166         // and on the same IO thread
167         openCLFilesAsync(args)
168             .exceptionally(throwable -> null) // recover
169             .thenAcceptAsync(v -> afterStartTestActions(), onEDT)
170             .thenRunAsync(Utils::preloadFontNames, onIOThread)
171             .exceptionally(Messages::showExceptionOnEDT);
172     }
173
```

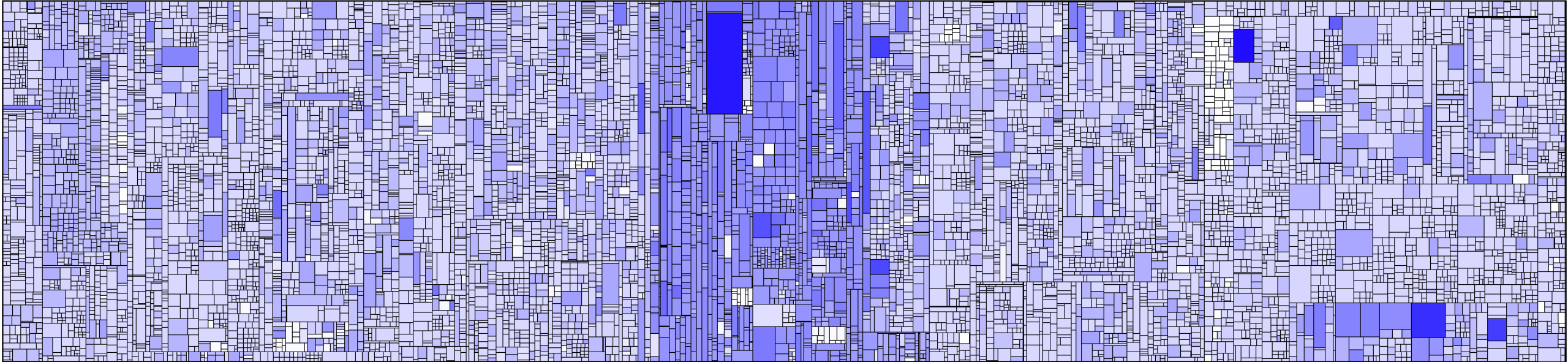


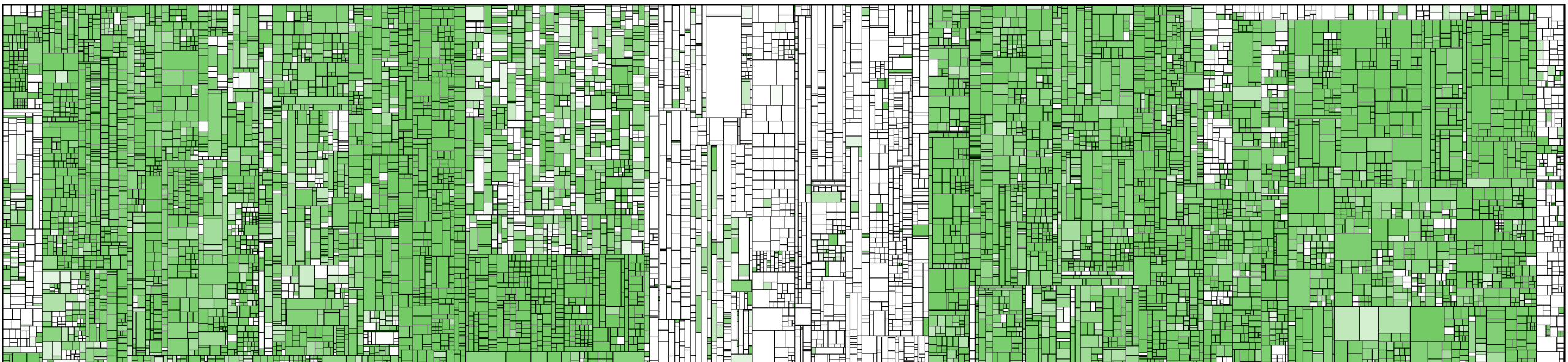


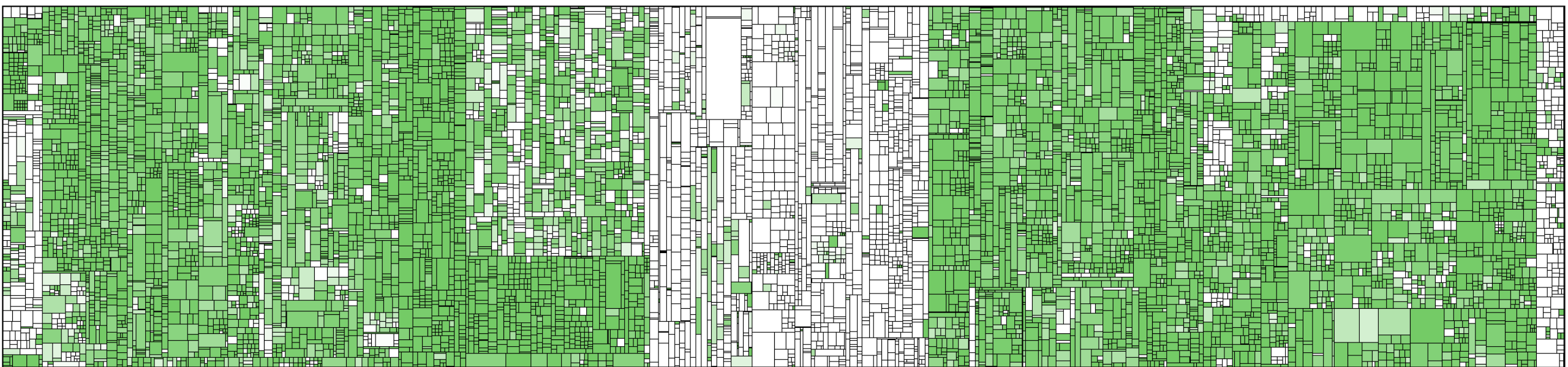
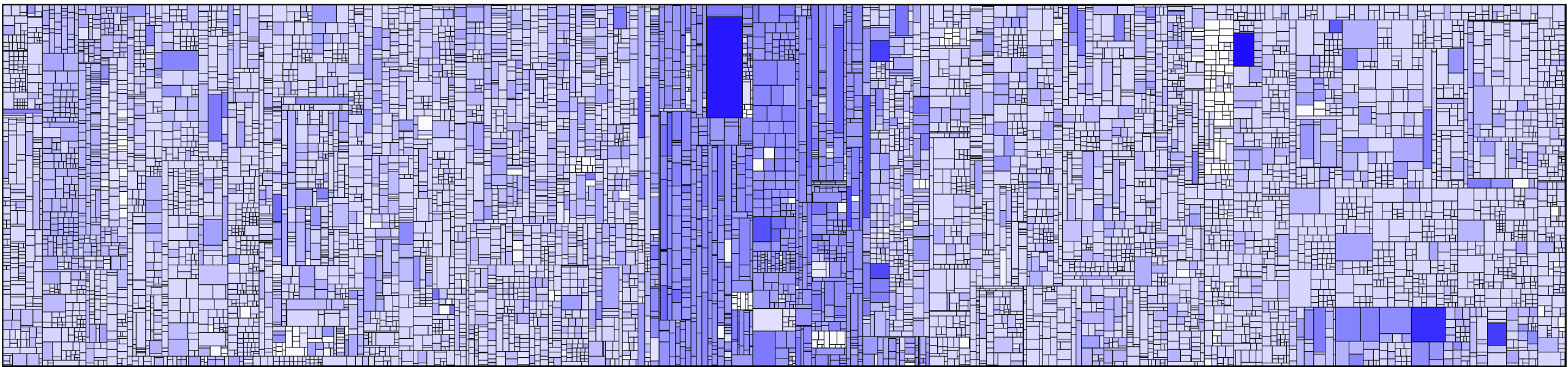


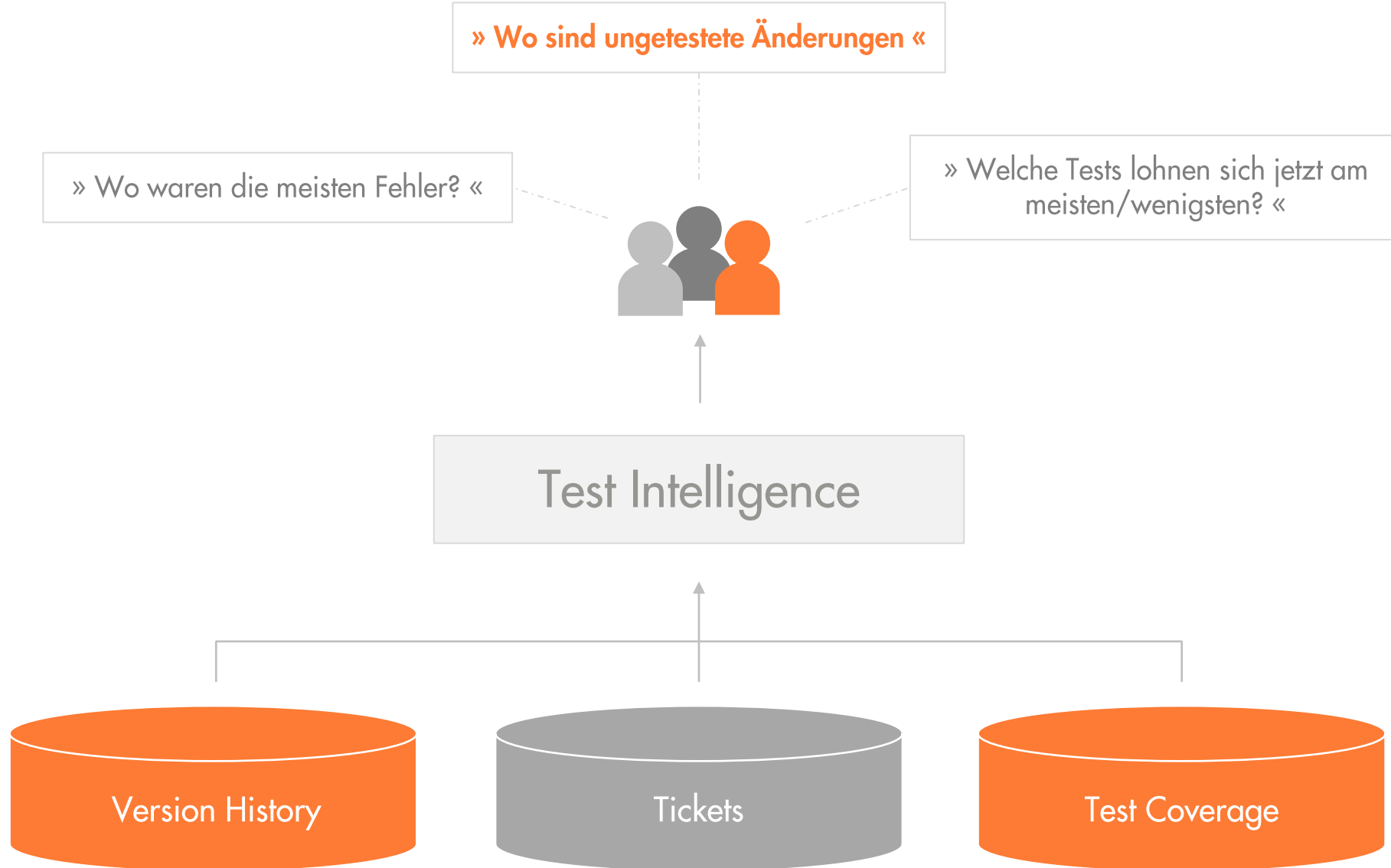




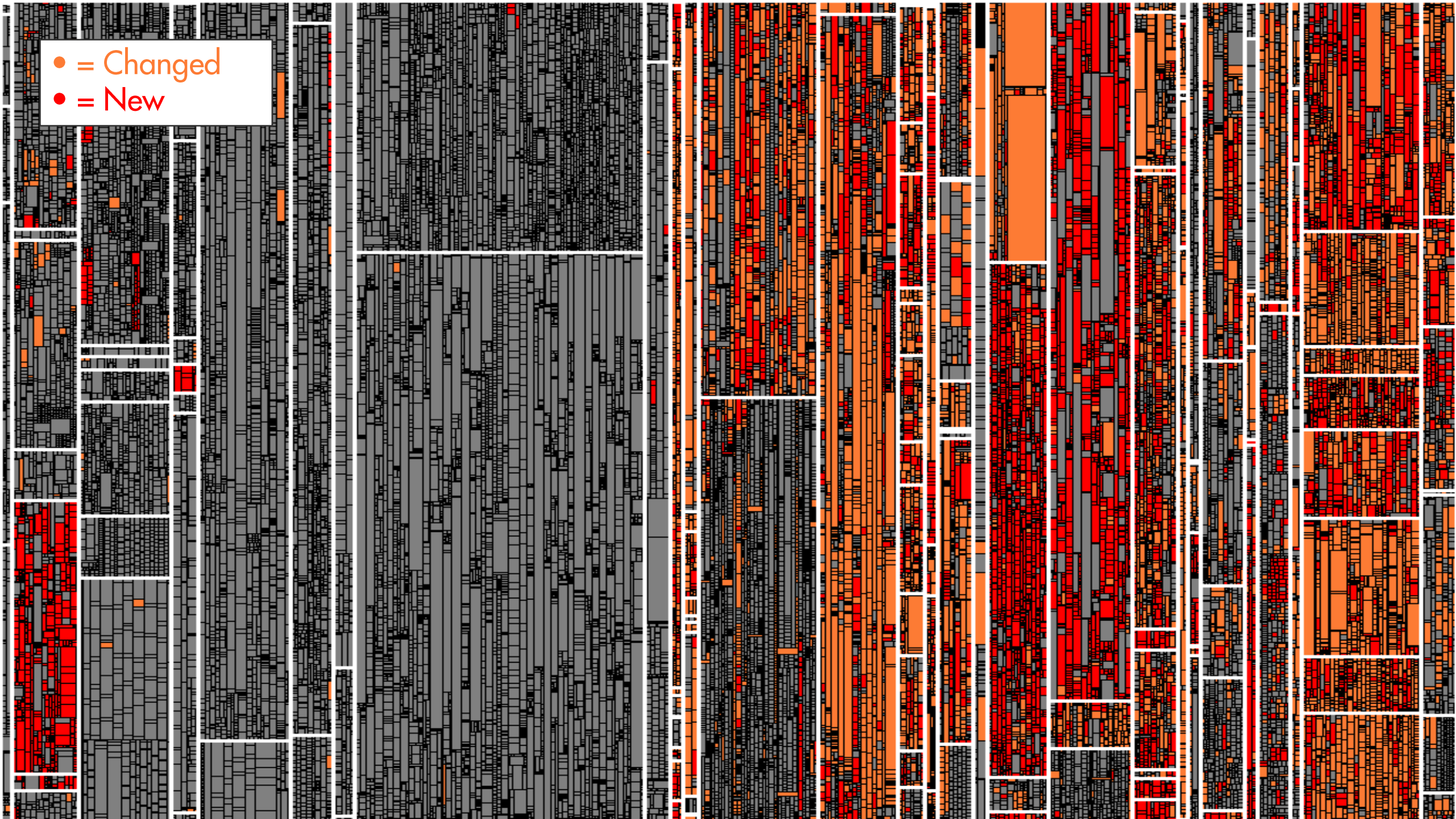






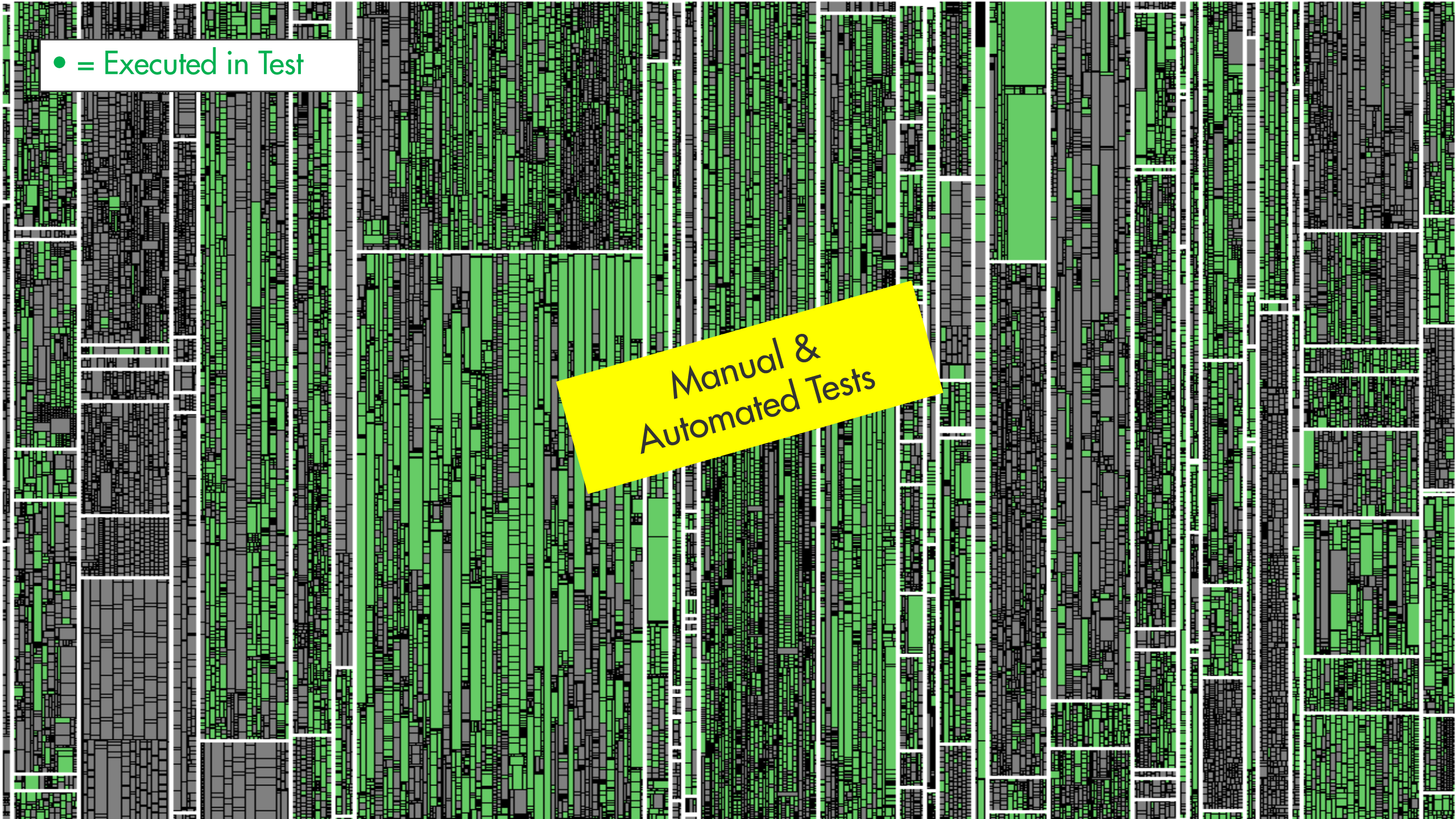


- = Changed
- = New

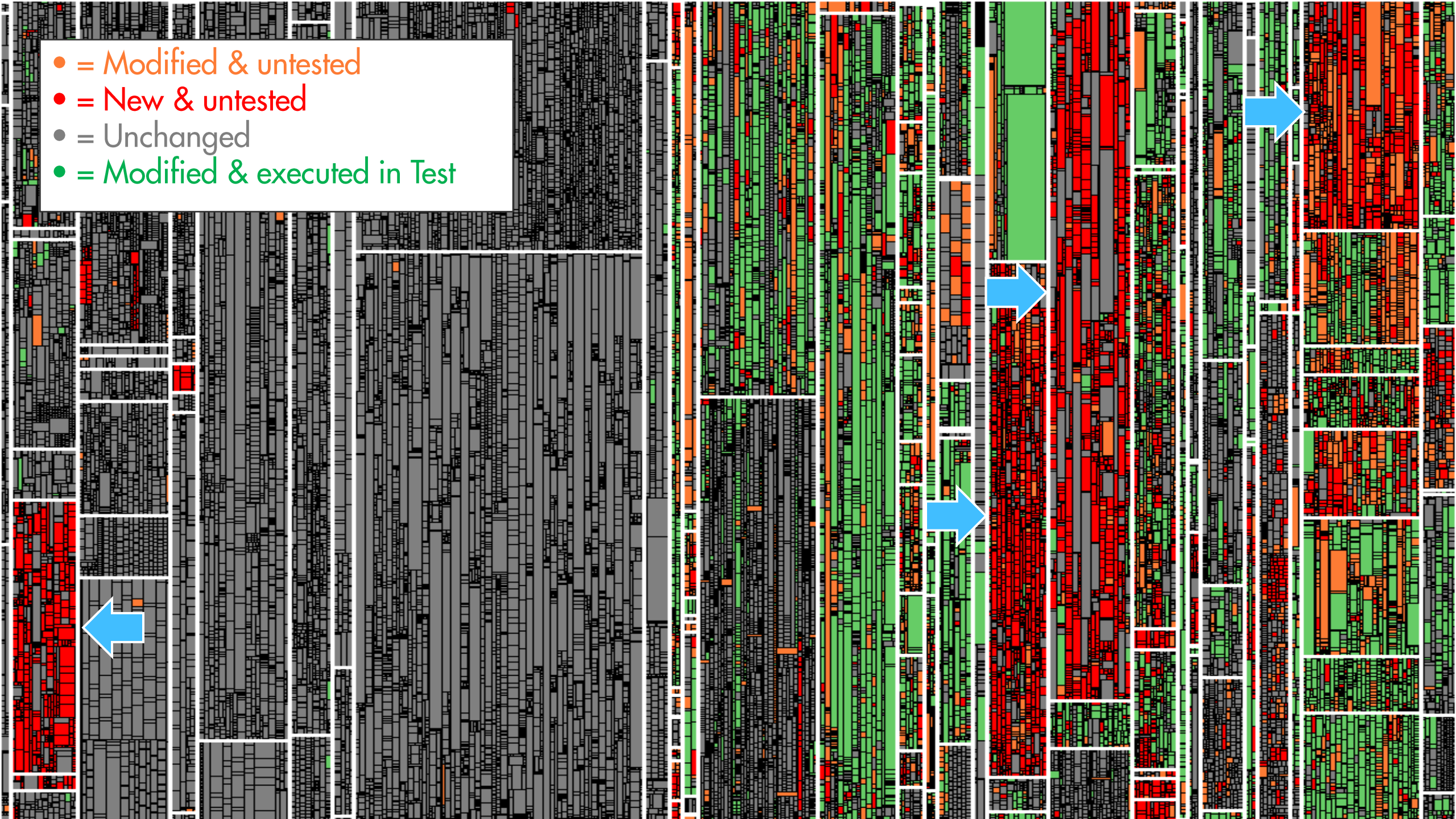



















● = Executed in Test





Manual &
Automated Tests




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




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% 

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

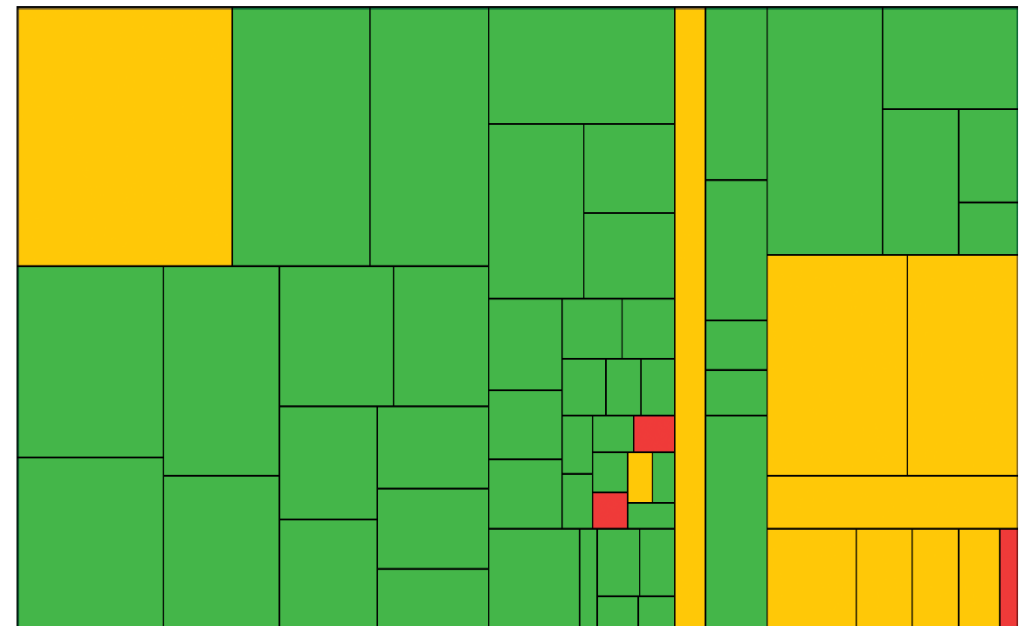
Done Issue TS-14733 - Remove Dataflow blocks

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

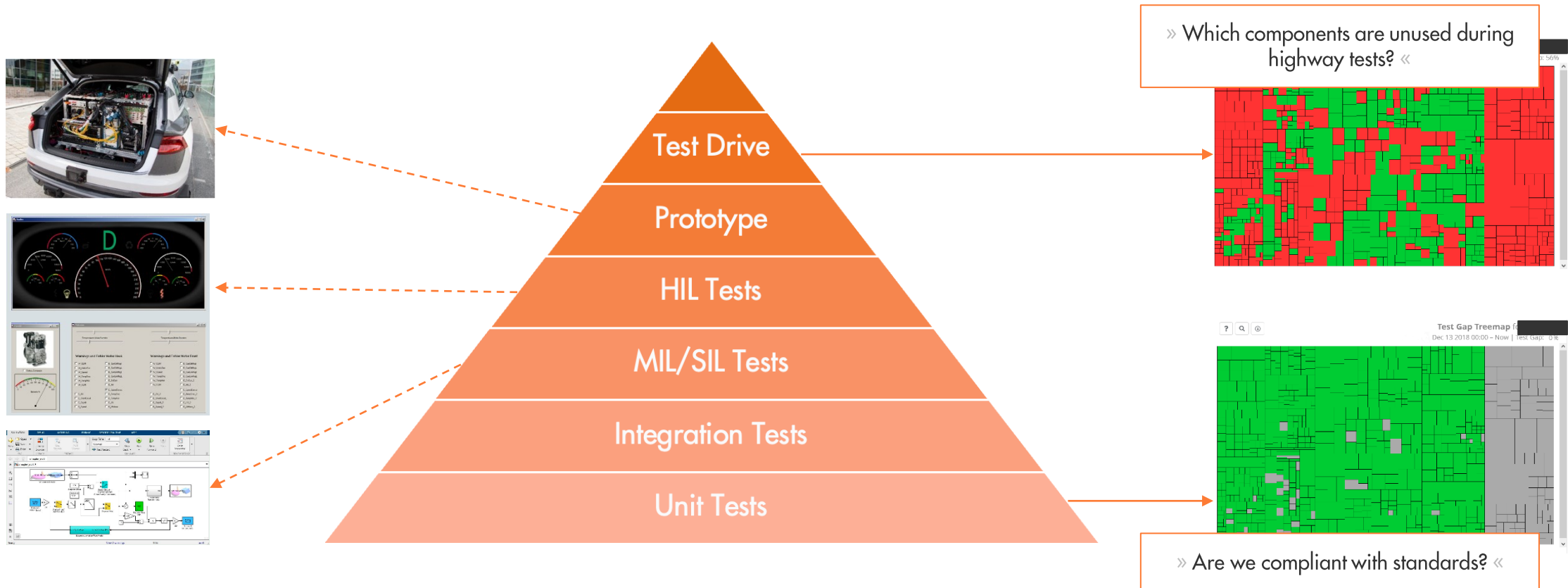
Assignee: 

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		

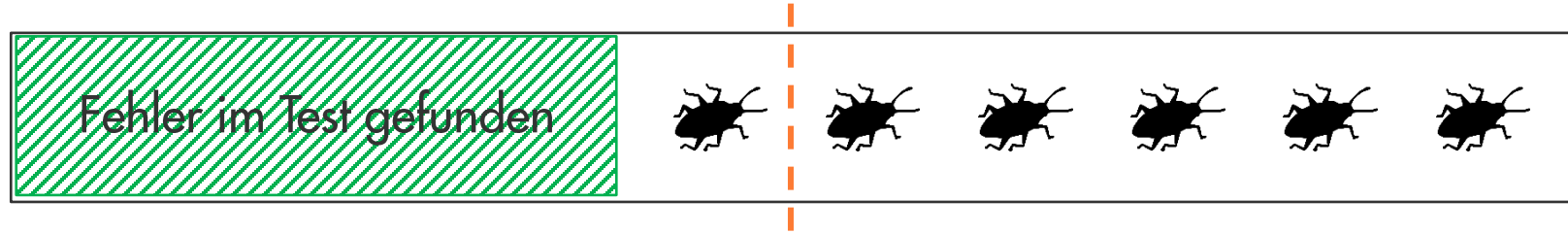
Aug 15 2018 12:37–Now | Test Gap: 22%



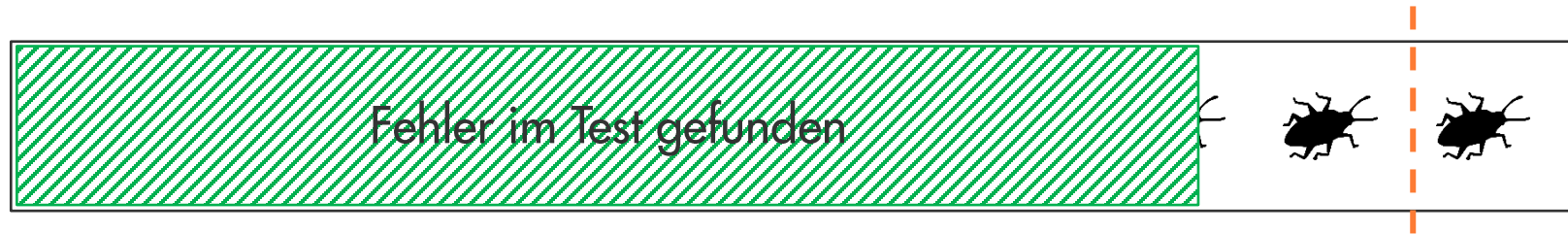
Test Gaps in different Test Environments



$\% \text{Restfehler} = 60\%$

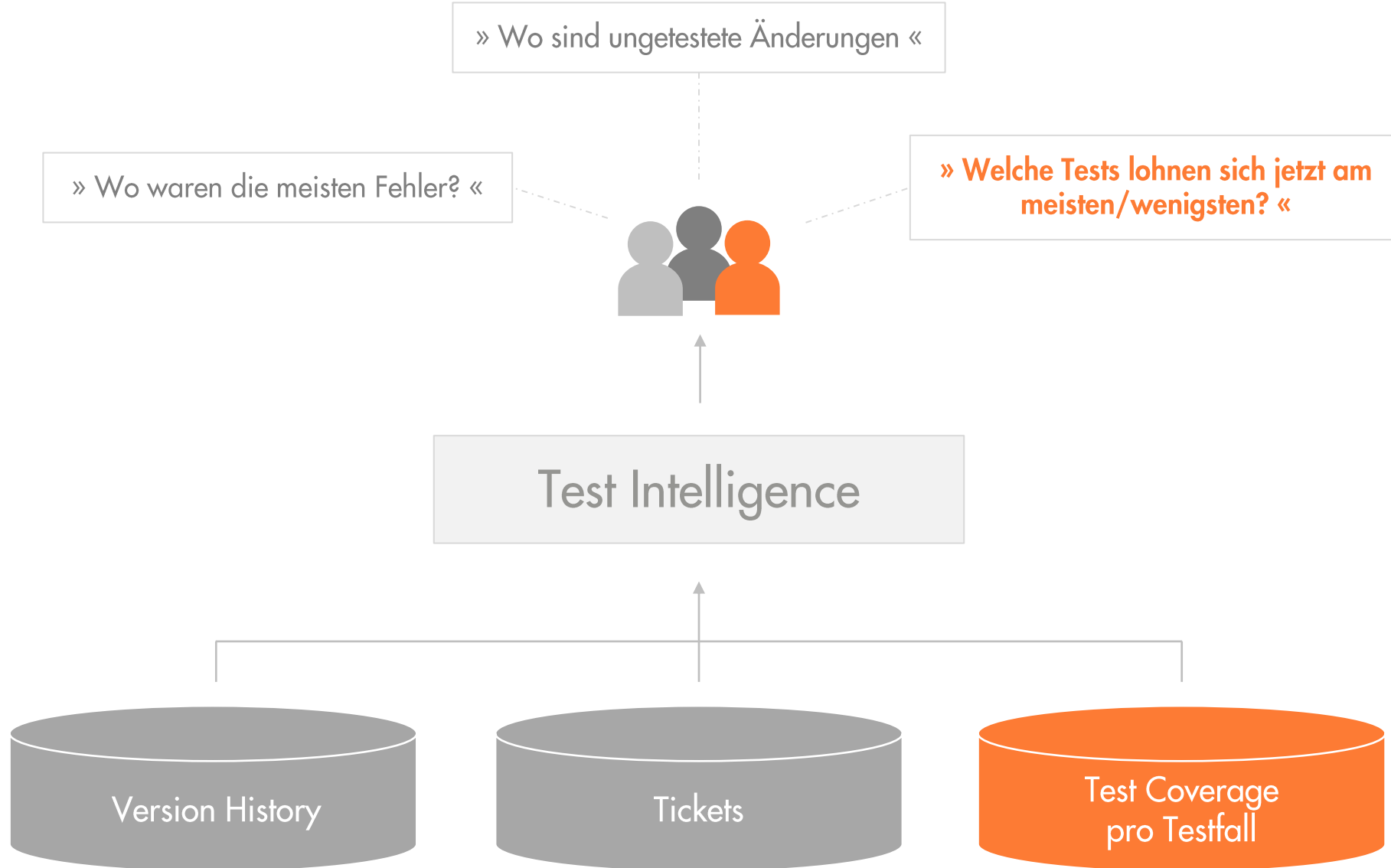


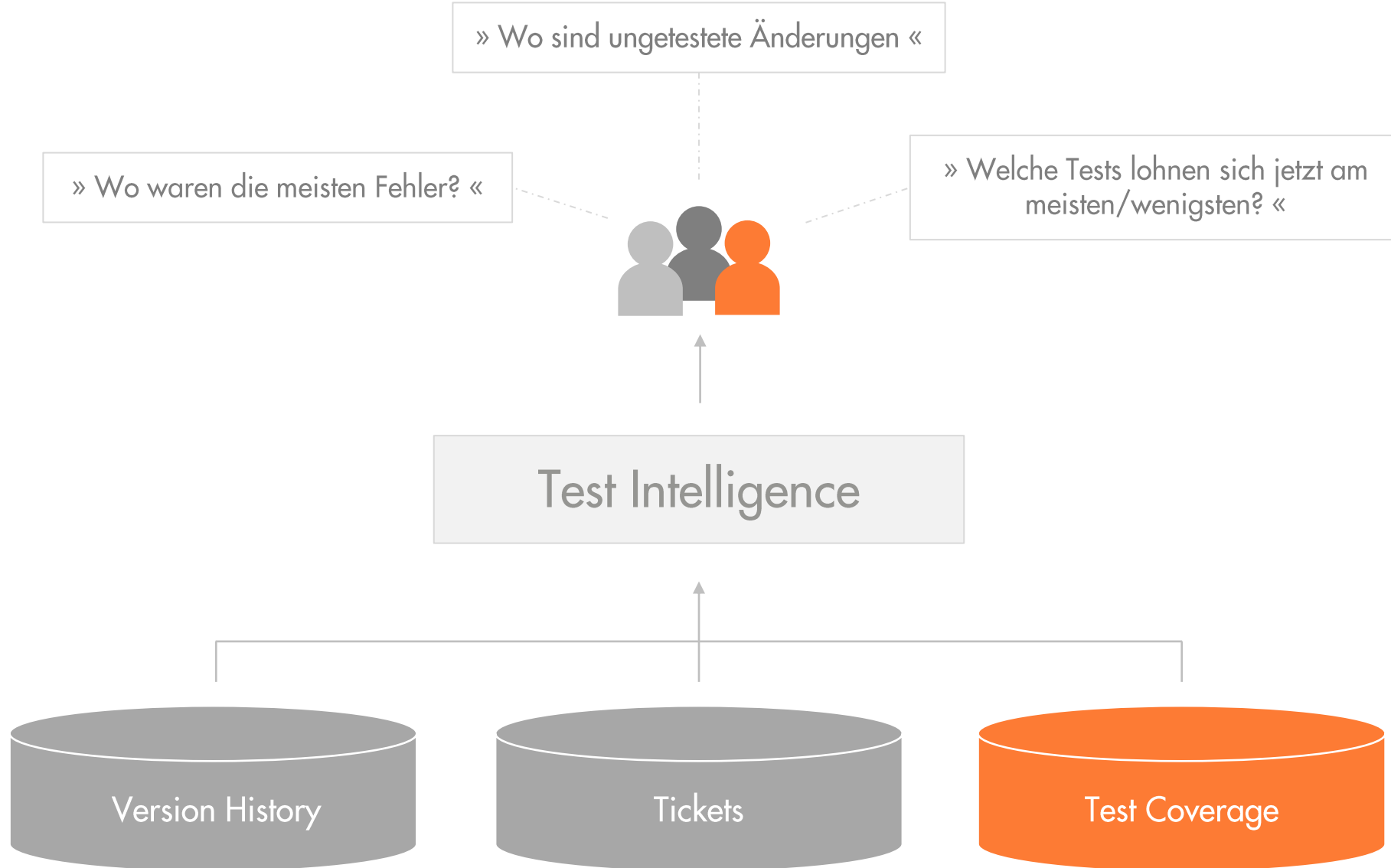
$\% \text{Restfehler} = 28\%$

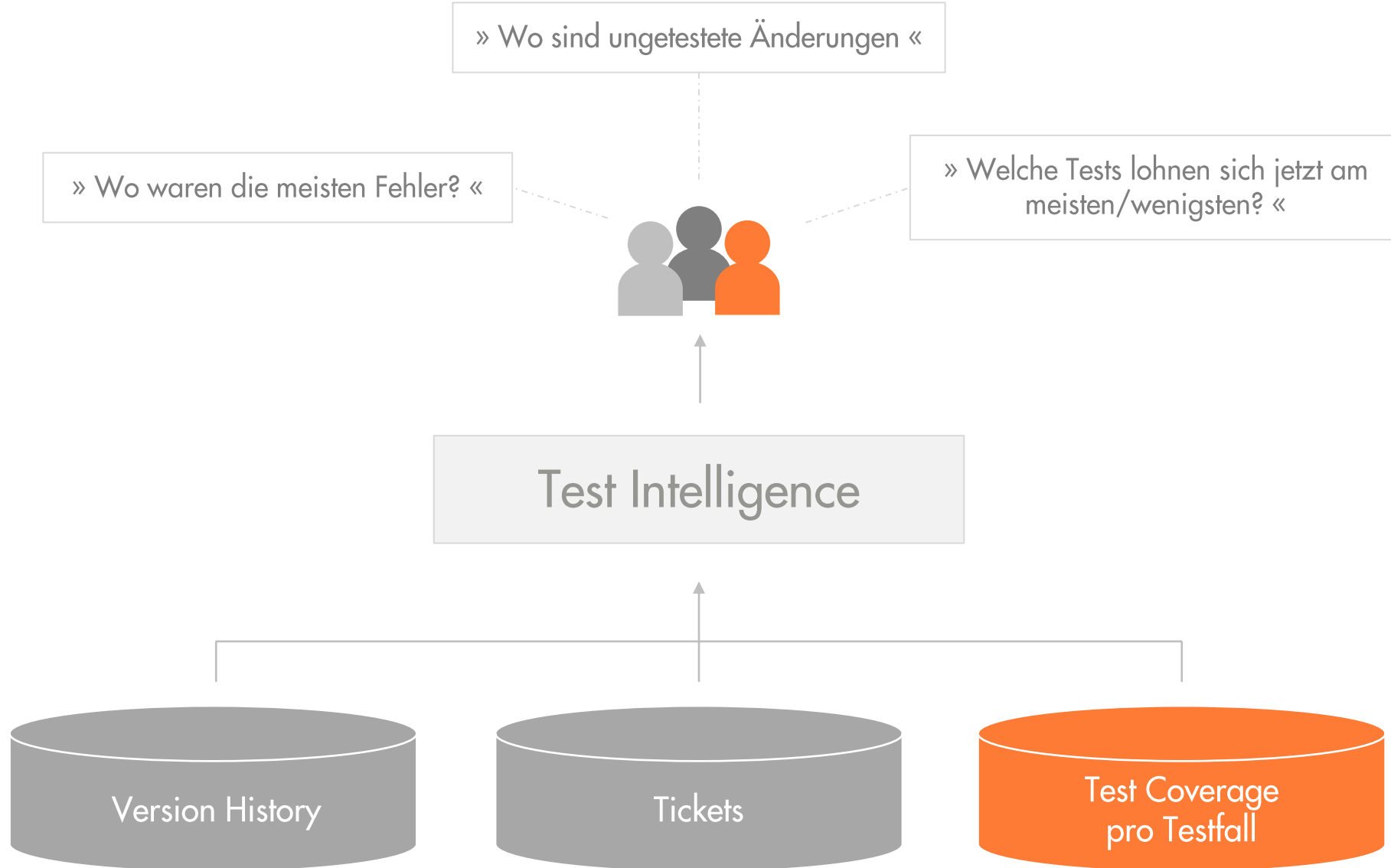


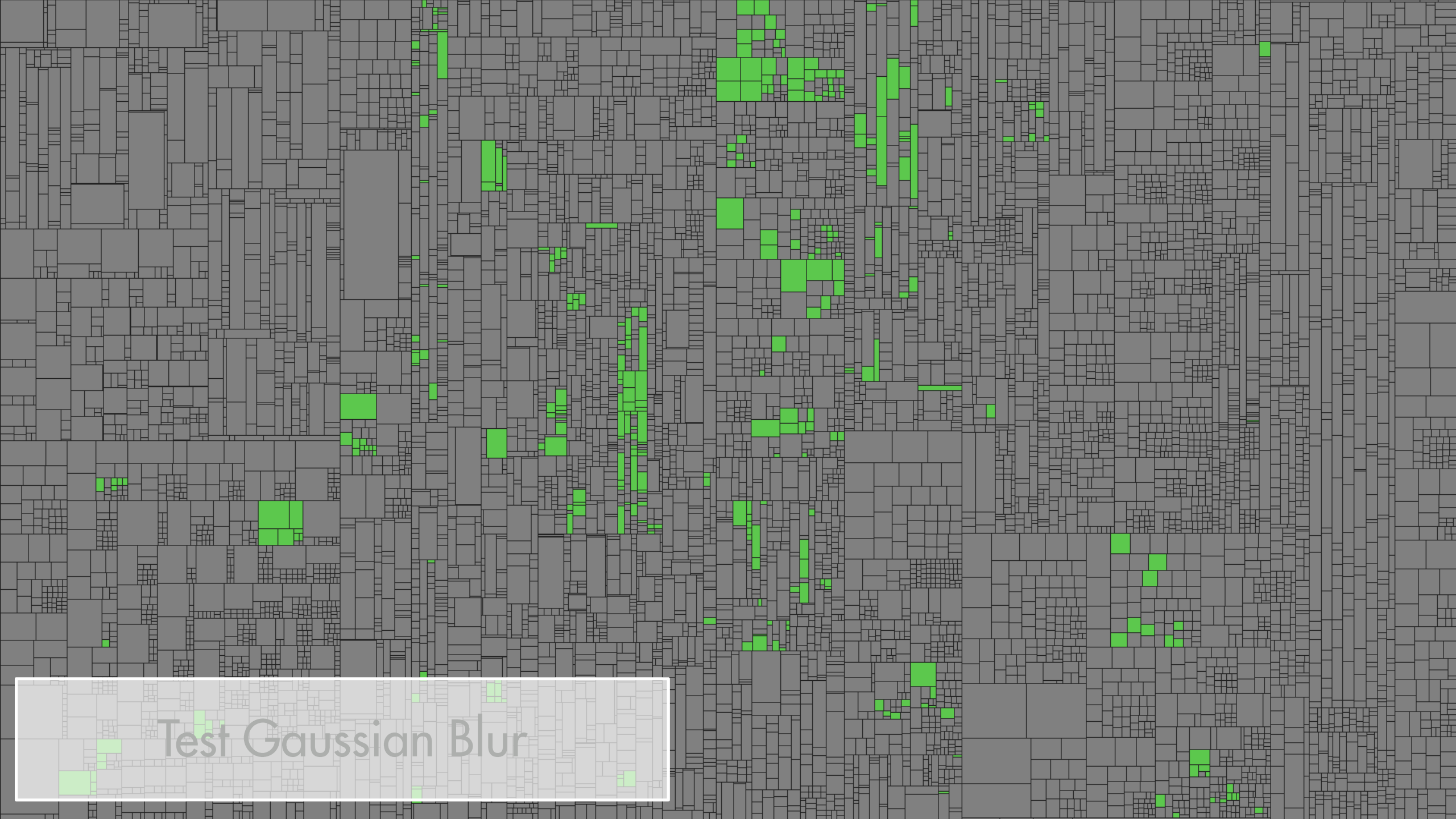
Reduzierte Feldfehler = **50%**

Test-Gap-Analyse reduziert Feldfehler in den Applikationen der Munich Re um 1/2

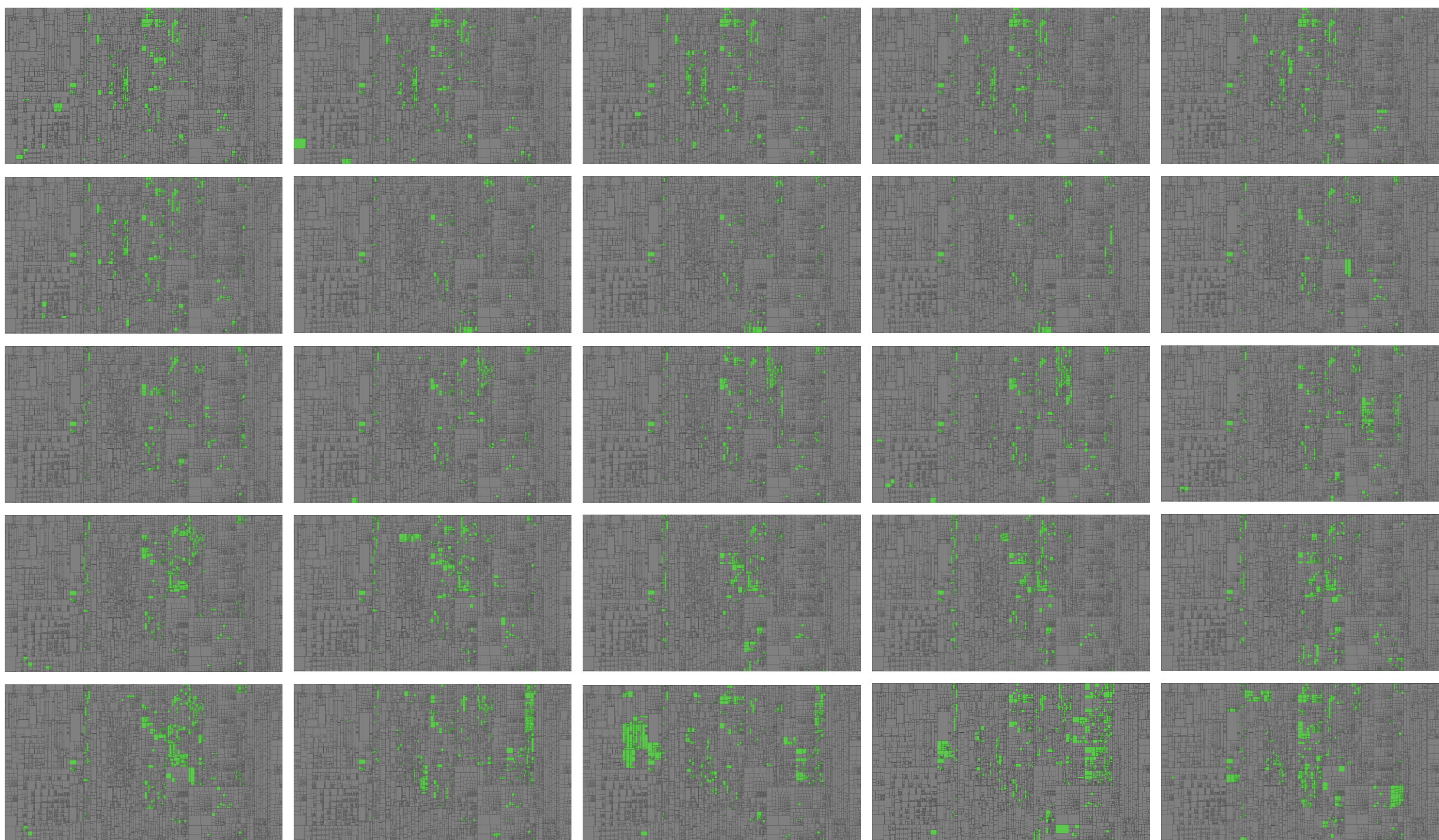


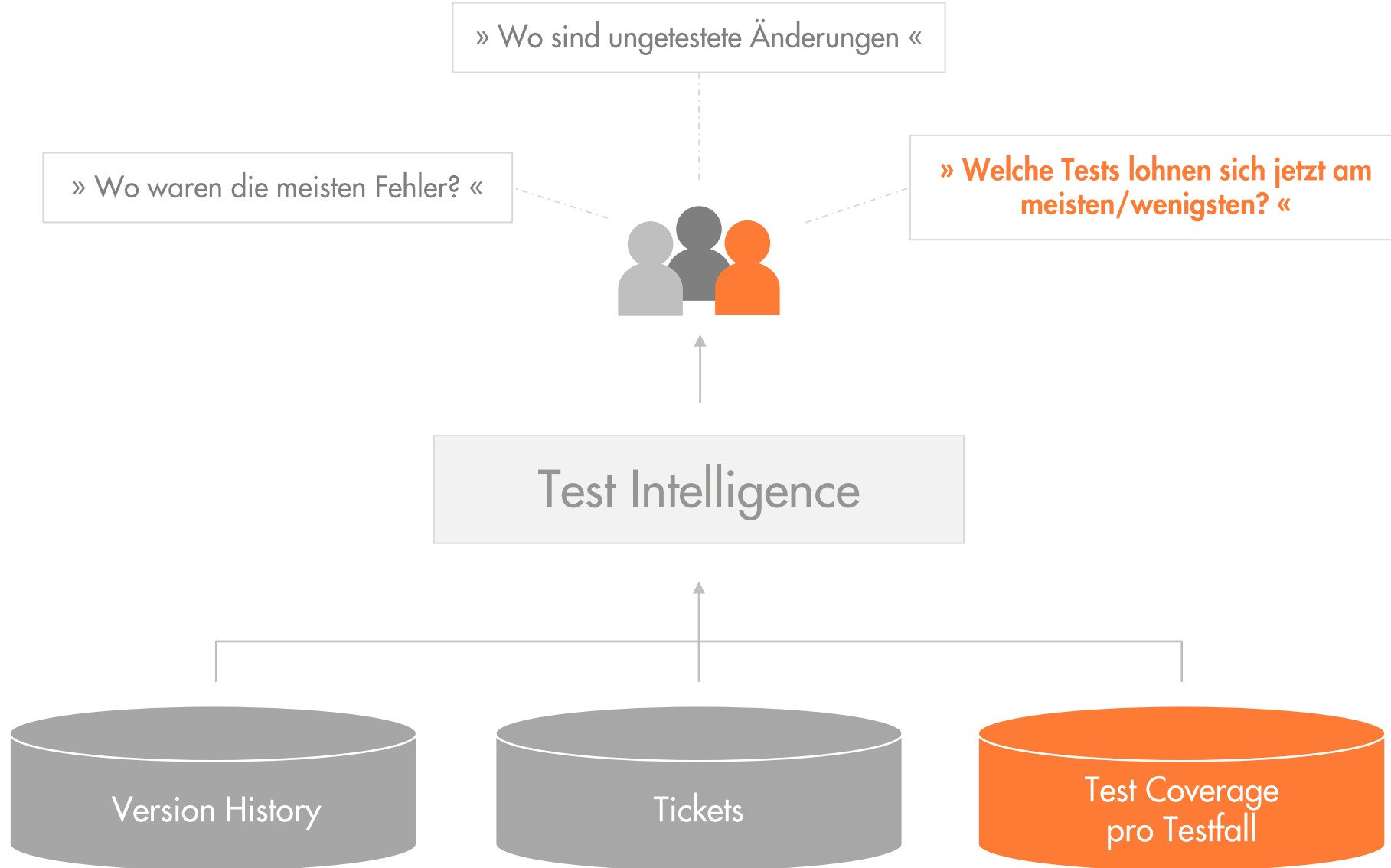


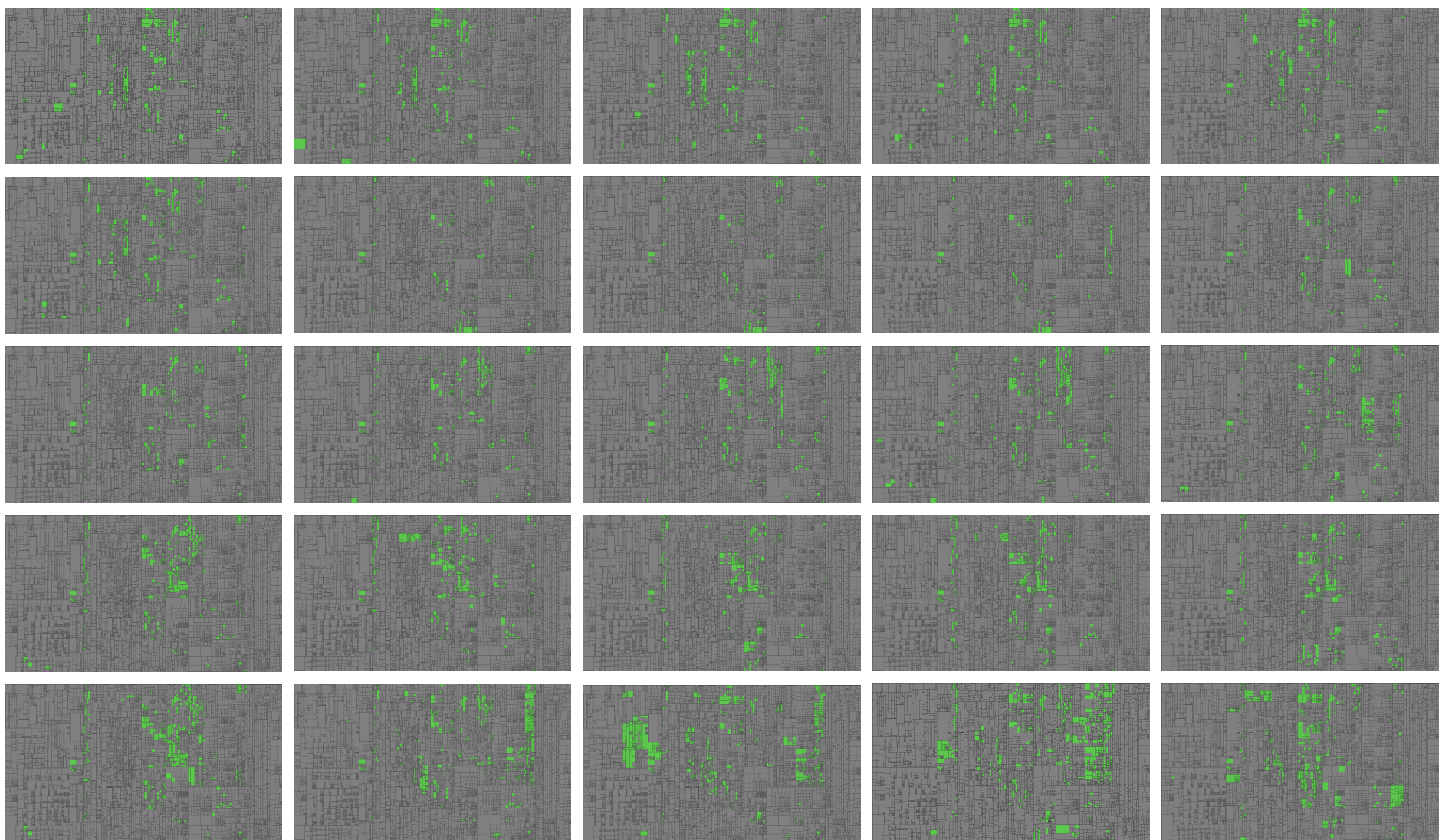


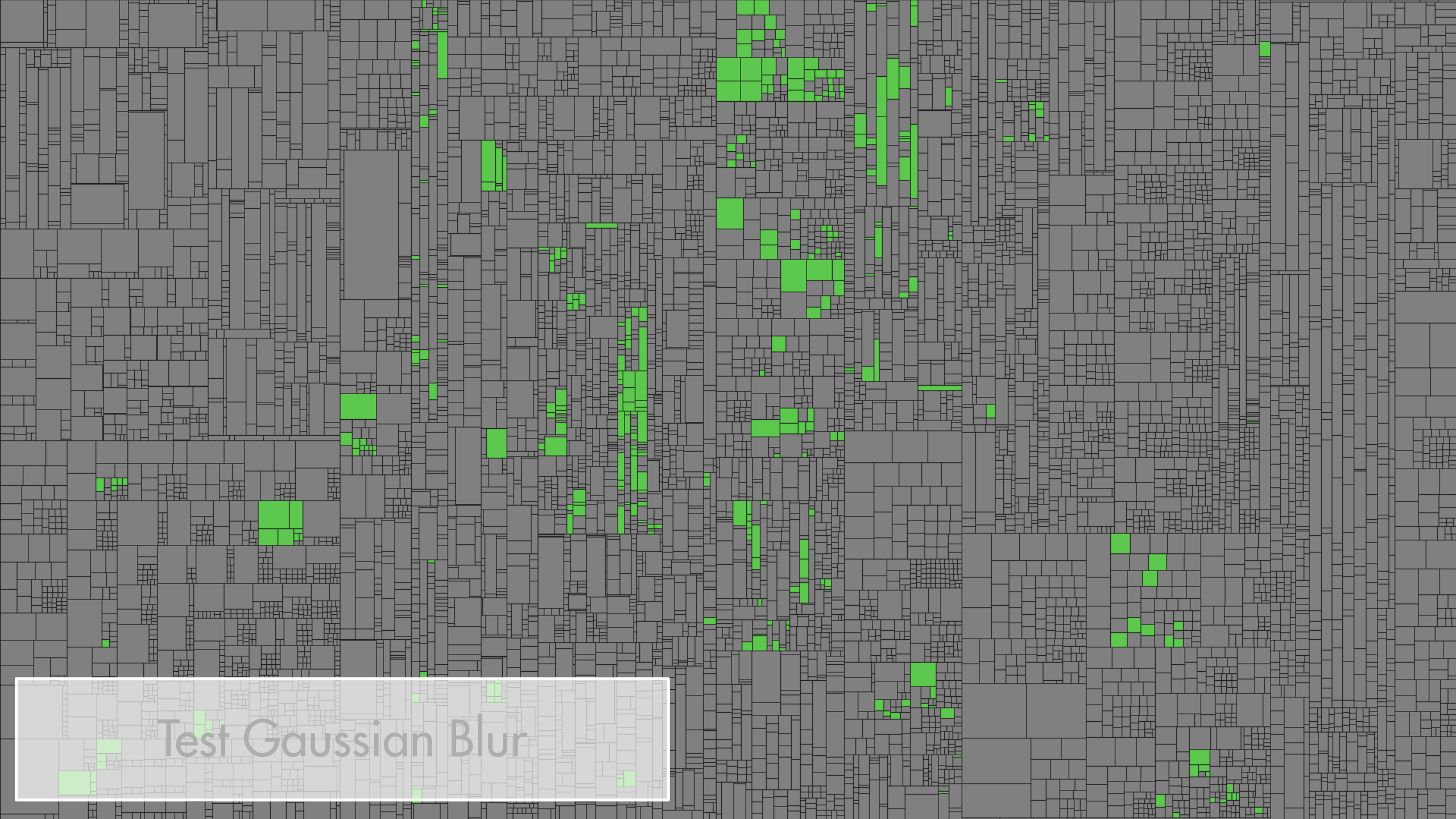


Test Gaussian Blur





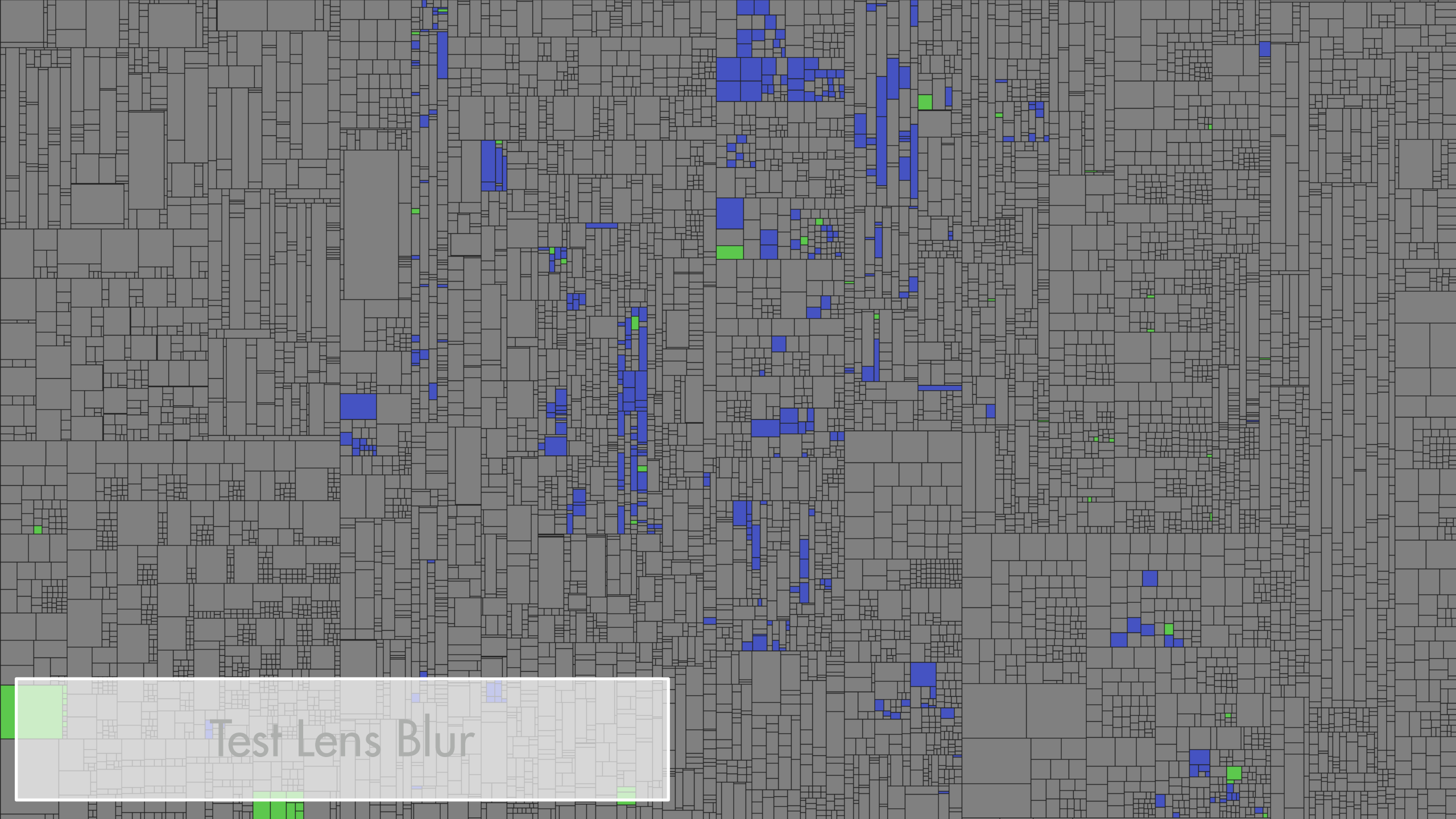




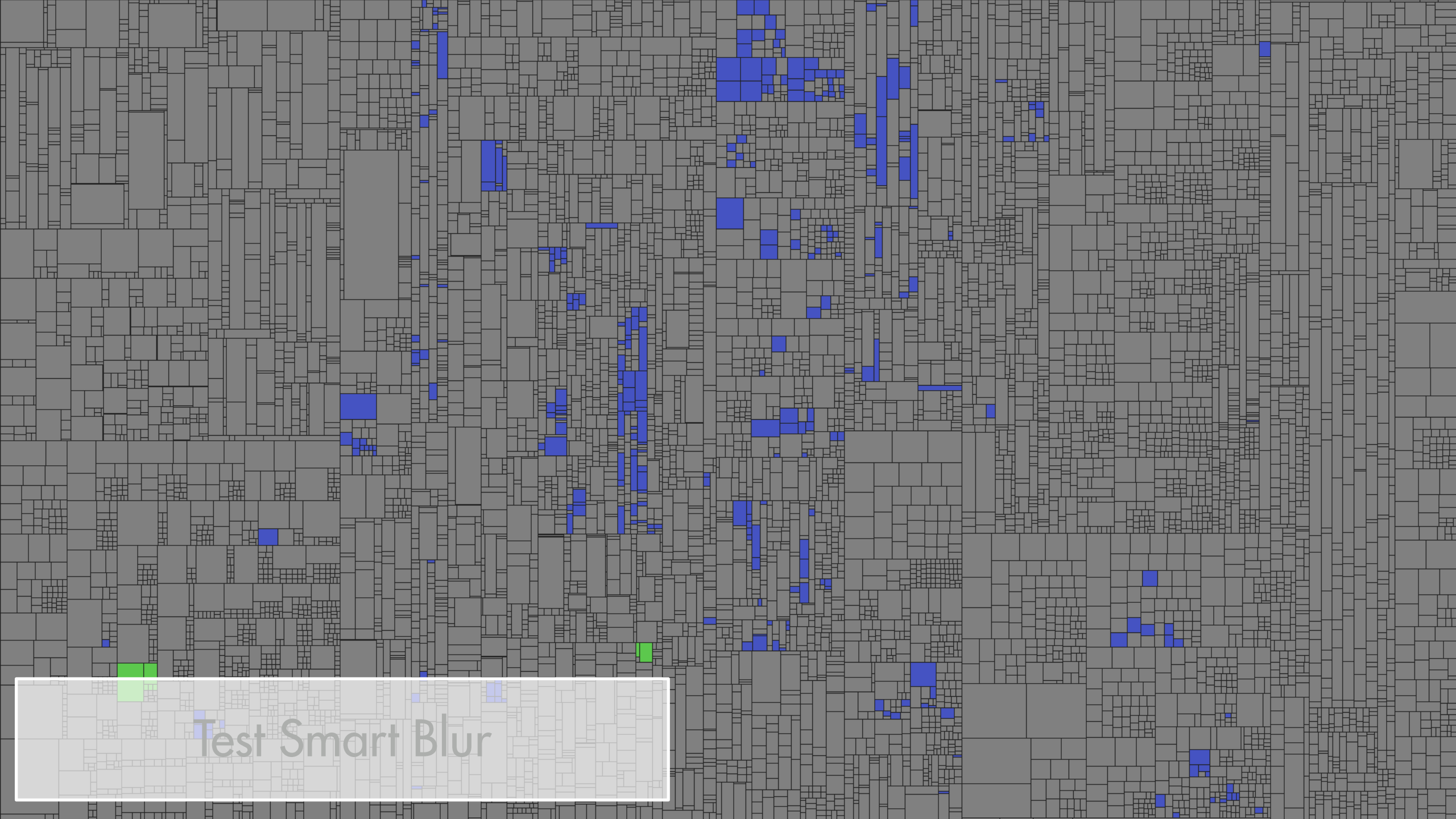
Test Gaussian Blur



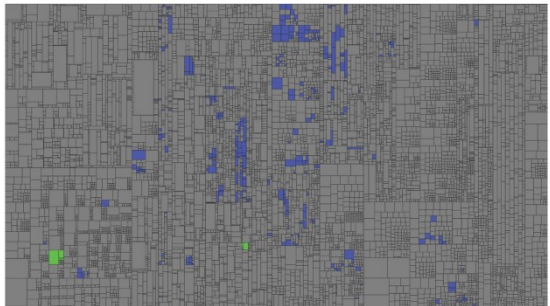
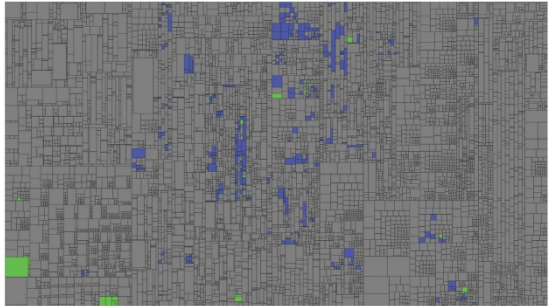
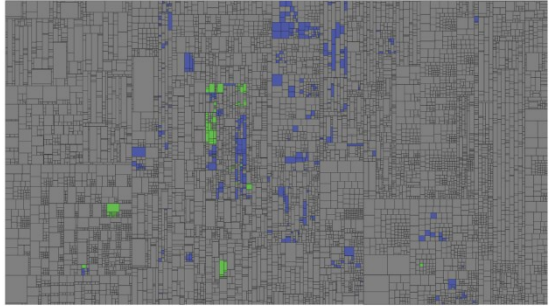
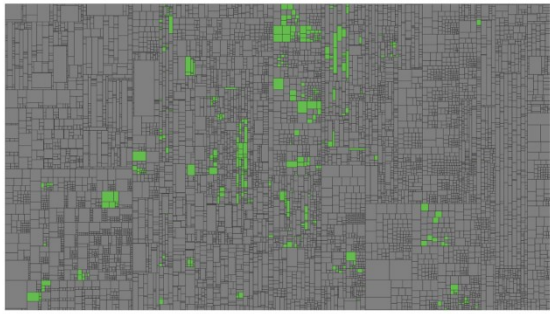
Test Motion Blur



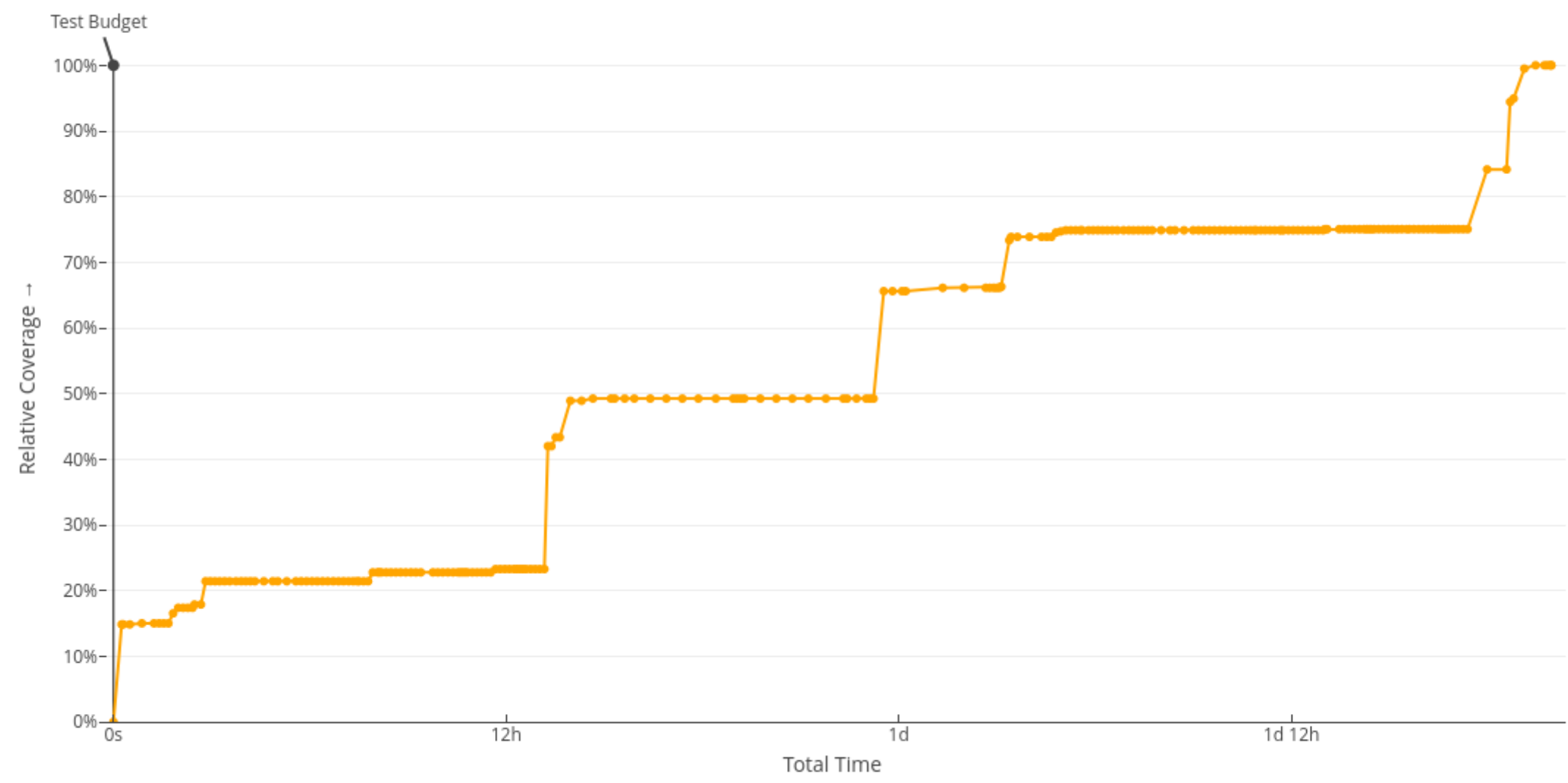
Test Lens Blur




Test Smart Blur



Coverage over Time ?



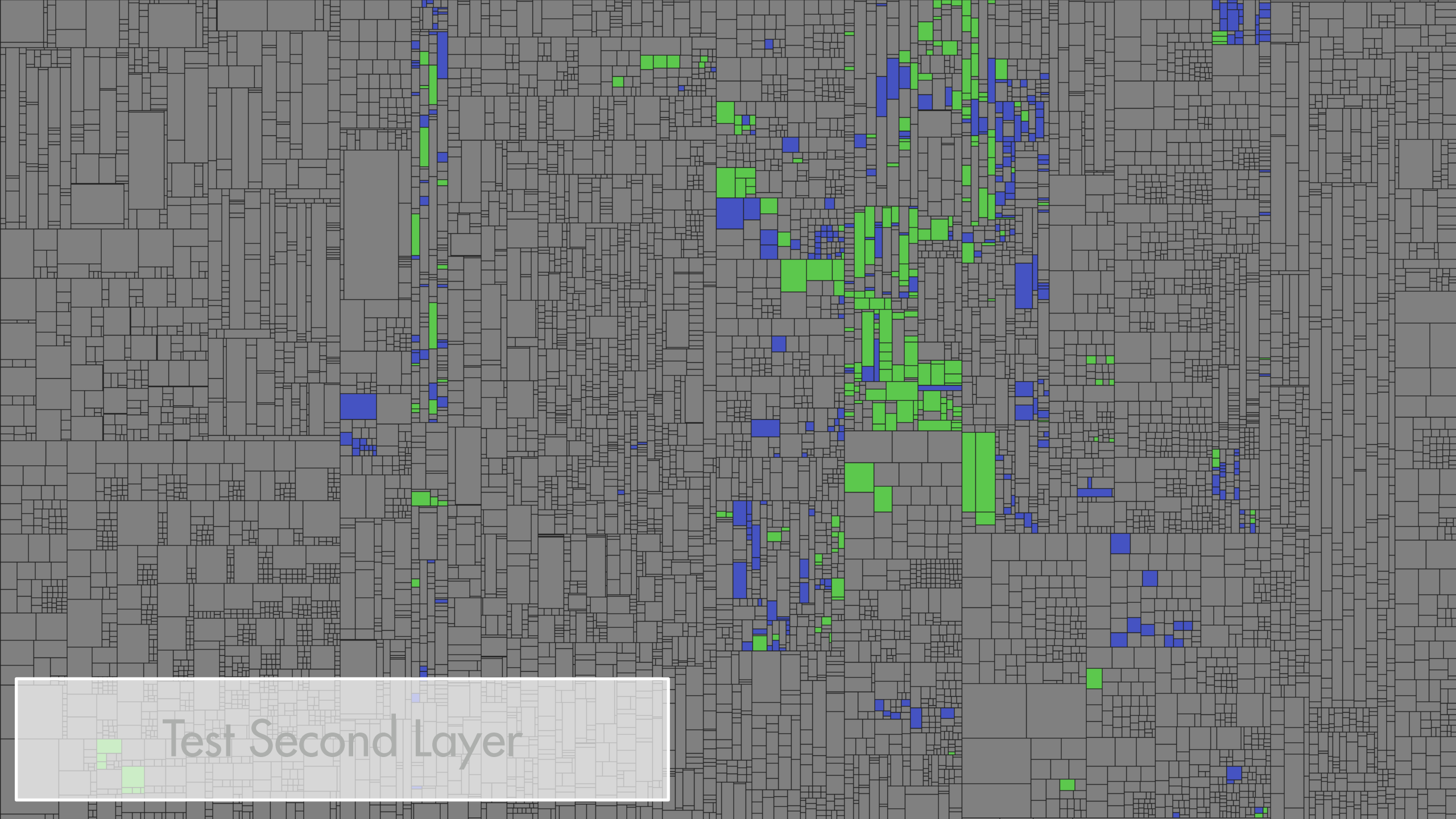
Results for Test Query & Budget Restriction
Relative Coverage: 0%, Selected Tests: 0 out of 236 (0%)



Test Create and Modify
Selection



Test Change View Settings

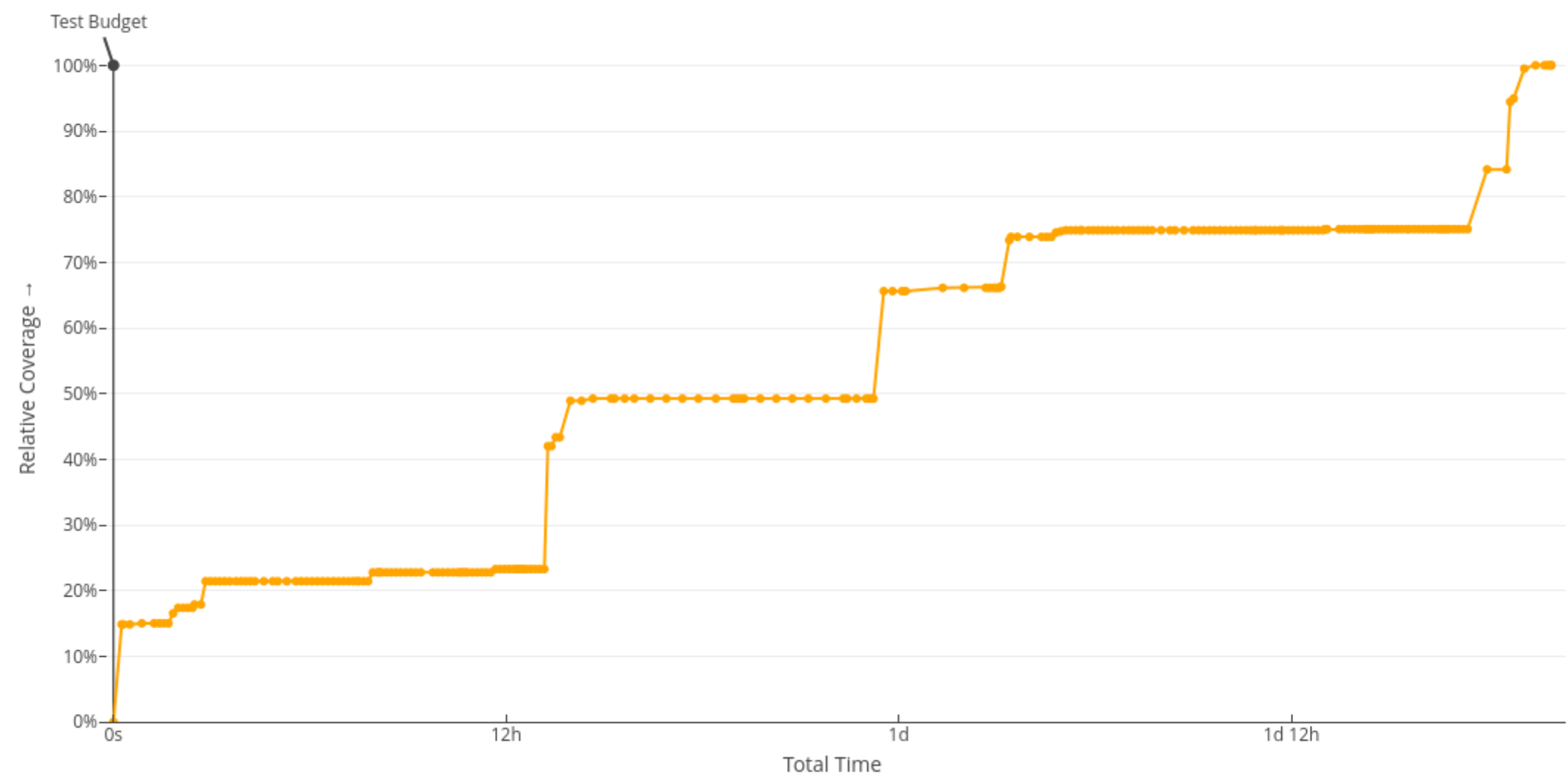


Test Second Layer

The image consists of a dense, overlapping grid of gray rectangles of various sizes. Scattered throughout this grid are several smaller rectangles in blue and green. The blue rectangles are more numerous and appear in various orientations and sizes. The green rectangles are fewer in number and often appear in small clusters or as single blocks. The overall effect is a complex, textured pattern of gray with occasional color accents.

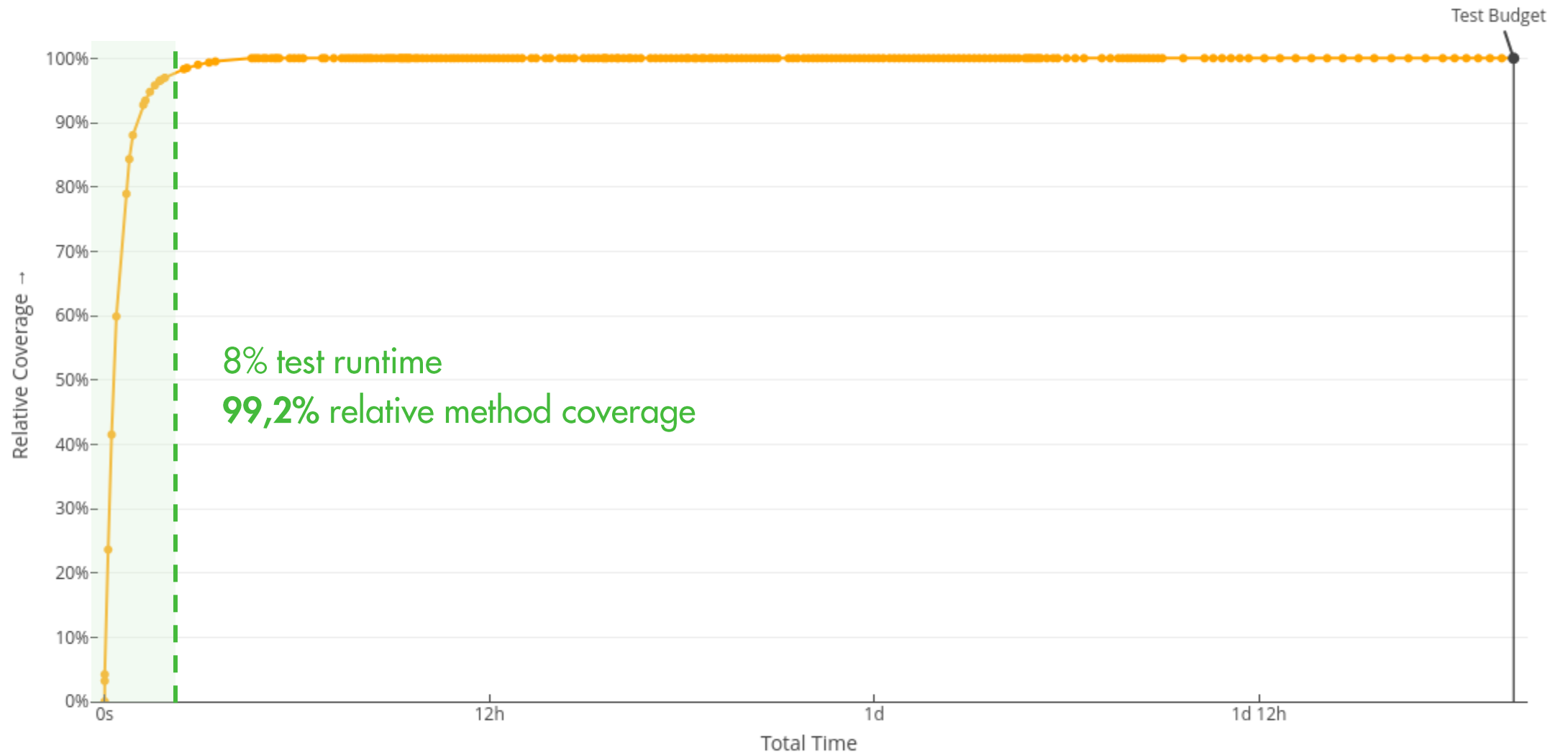
Test Save Image

Coverage over Time ?



Results for Test Query & Budget Restriction
Relative Coverage: 0%, Selected Tests: 0 out of 236 (0%)

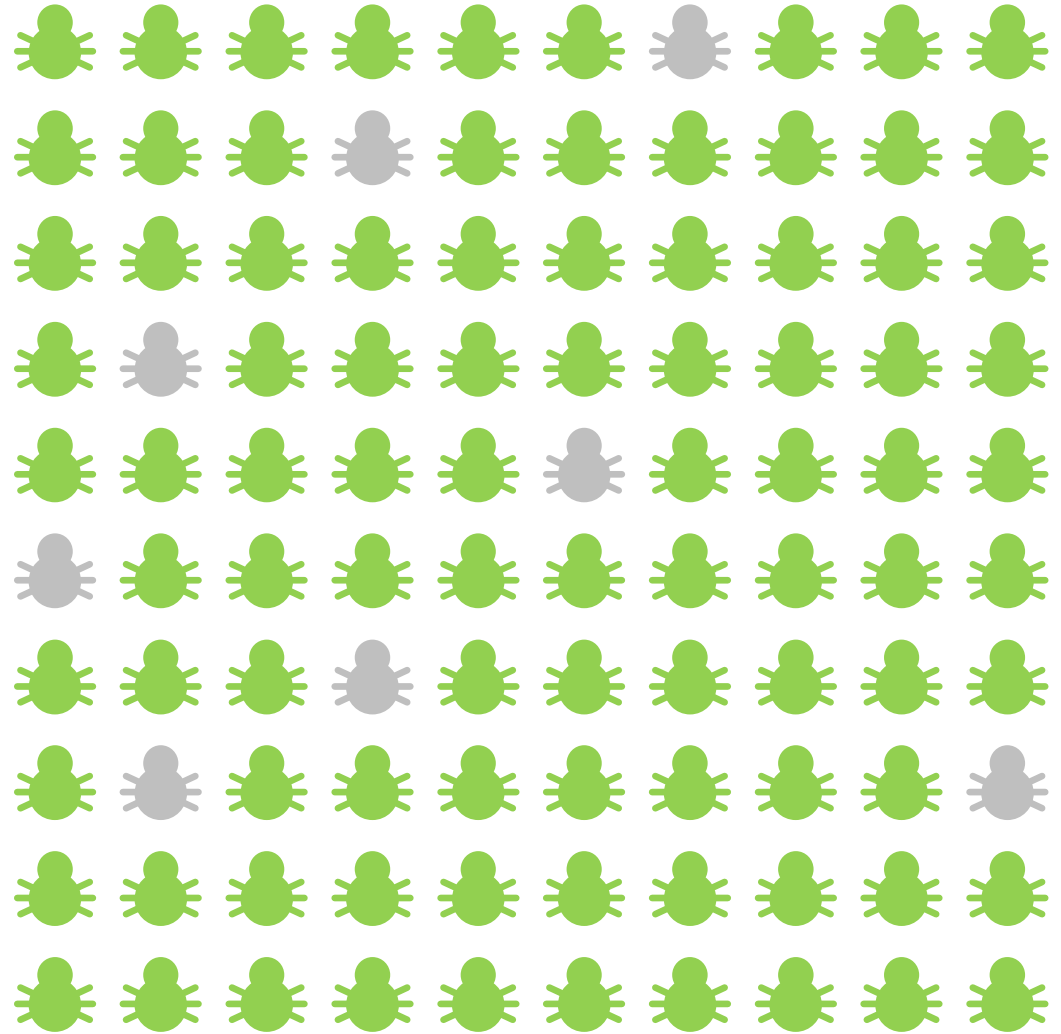
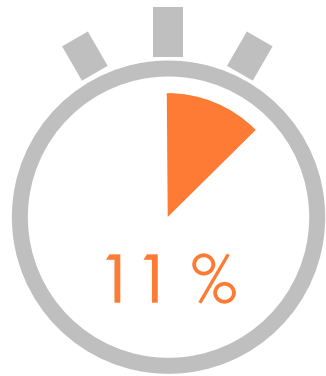
Coverage over Time ?



8% test runtime
99,2% relative method coverage

Results for Test Query & Budget Restriction

Relative Coverage: 100%, Selected Tests: 236 out of 236 (100%)



Pareto-Testliste in der CI

Vorher: **nächtlich** „Re-run all“

Feedback erst spät am nächsten Tag



Tagsüber **1 h Testbudget** für Pareto-Testliste

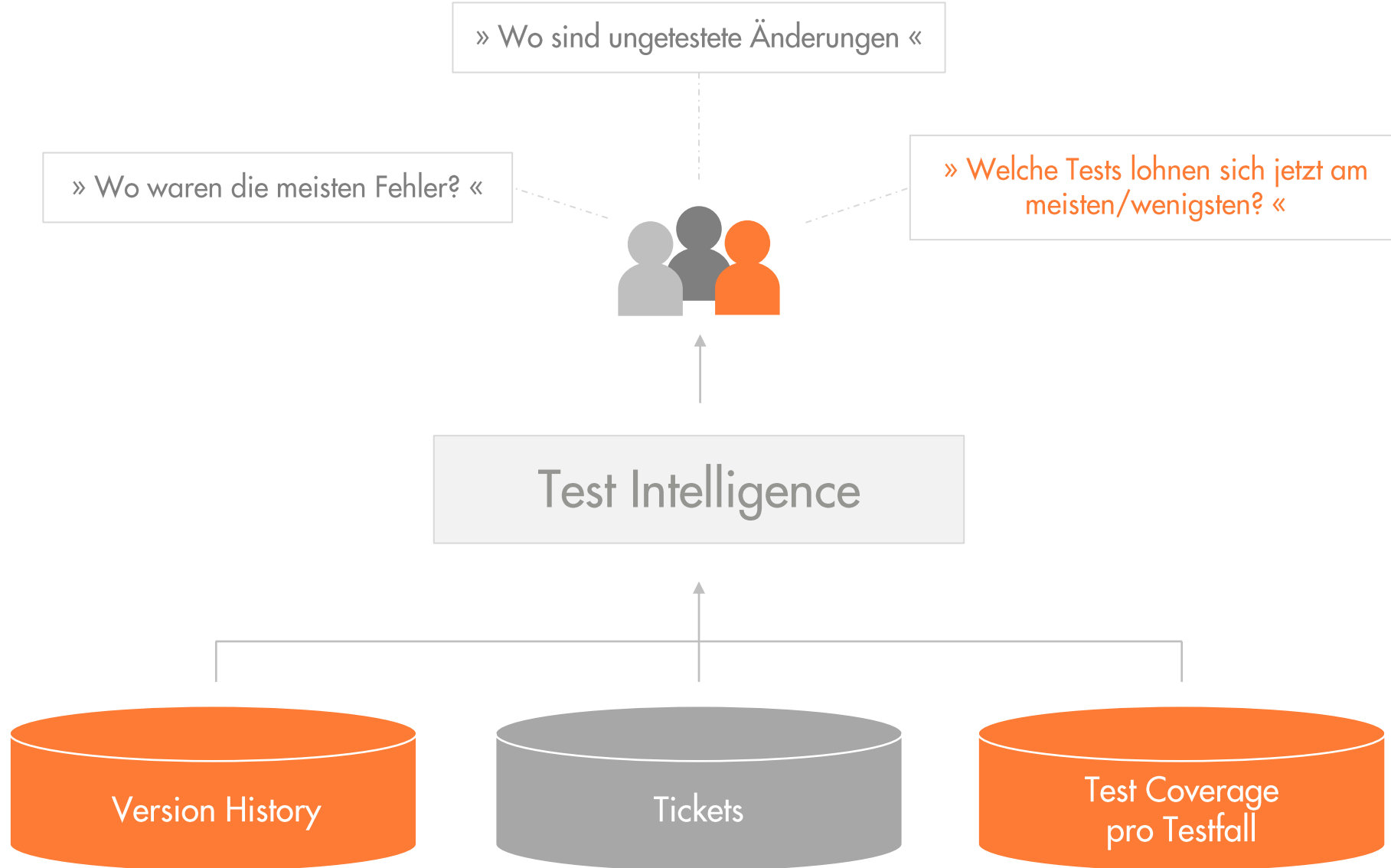


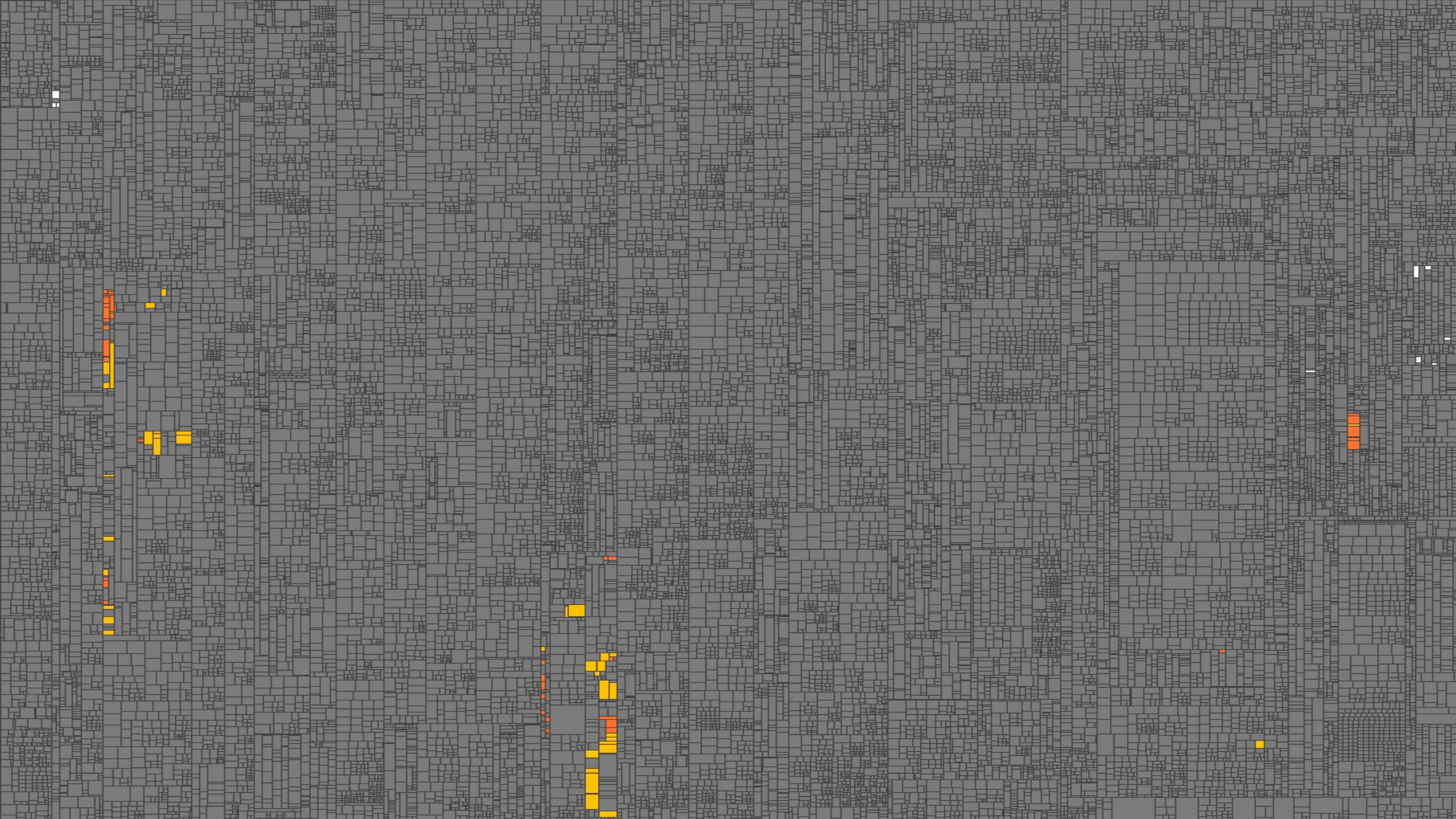
Weiterhin: **nächtlich** „Re-run all“



Mit nur 5% der
Testlaufzeit finden wir
**70% der
fehlgeschlagenen
Commits**









Testlaufzeiten von 3-12 Stunden durch eine Vielzahl von Parameterkombinationen.



Test-Feedback aus der CI innerhalb von 10 Minuten.



Mit TIA

Seit einem Jahre kommt das Test-Feedback immer innerhalb von 10 Minuten, obwohl die Gesamtzahl der Tests im selben Zeitraum um 57% zunahm.

Test-Impact-Analyse

Tests werden für jeden Lauf **passend zu Änderungen** ausgewählt

90% der Fehler in 2% der Zeit

Erfordert kontinuierliche Messung der Coverage und Integration der Test-Auswahl in die CI / Testautomatisierung.

Stärkere Beschleunigung des Feedbacks (bei höherem Aufwand)

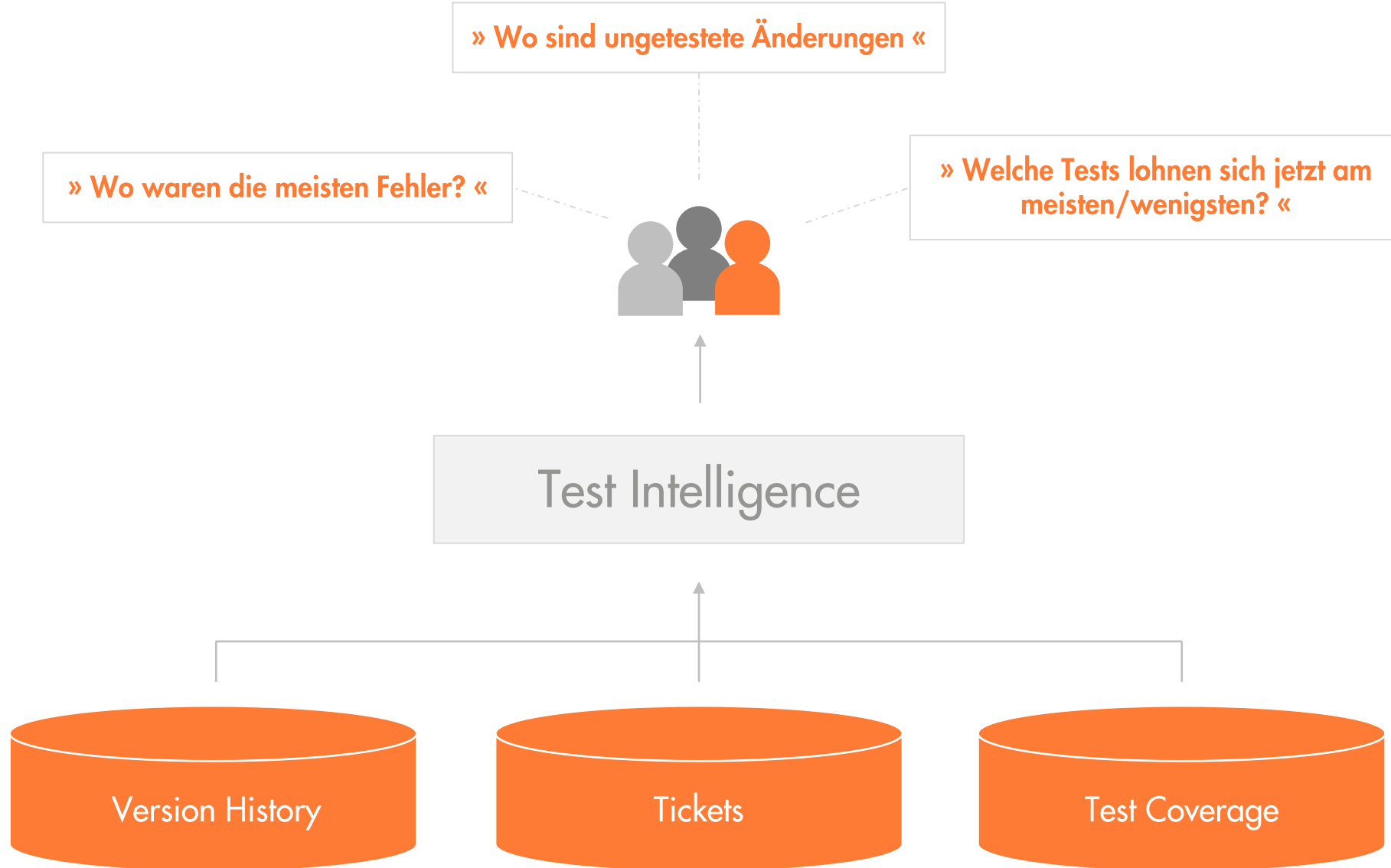
Pareto-Optimierung

Tests werden **unabhängig von Änderungen** ausgewählt

90% der Fehler in 11% der Zeit

Einmalige Messung der Coverage reicht aus.

Viel geringerer Aufwand



Test-Gap-Analyse

Ungetestete Änderungen im Quelltext aufdecken



Für Aufzeichnung anmelden
tmscl.me/tga-244-java



Schnelles Feedback trotz langsamer Tests

Testselektion für historisch gewachsene Test-Suites



Aufzeichnung ansehen
tmscl.me/ts-243-java



Fazit

Historisch gewachsene Test-Suites testen oft gleichzeitig zuviel (wegen redundanten Testfällen) und zuwenig (wegen Test-Gaps).

Die Auswertung der Daten aus unserem eigenen Entwicklungsprozess (Versionshistorie, Test-Coverage und Tickets) ermöglicht uns, mehr Fehler in kürzerer Zeit zu finden.

Ich teile unsere Erfahrung hierzu gerne.



tmscl.me/coffee-elmor