

Was verraten moderne Analysen über Wechselwirkungen zwischen **Architektur & Organisation?**

Fallbeispiele aus 10 Jahren Praxiseinsatz

Über Mich

Forschung

- Clone Detection, Architekturanalyse, ...
- PC Mitglied von MSR, ICPC, ICSE, ...



Beratung

- Gründer
- Qualitäts-Bewertung & Qualitäts-Controlling



Gesellschaft für Informatik

- Zum Junior-Fellow ernannt
- Erfahrungsaustausch Forschung <-> Praxis

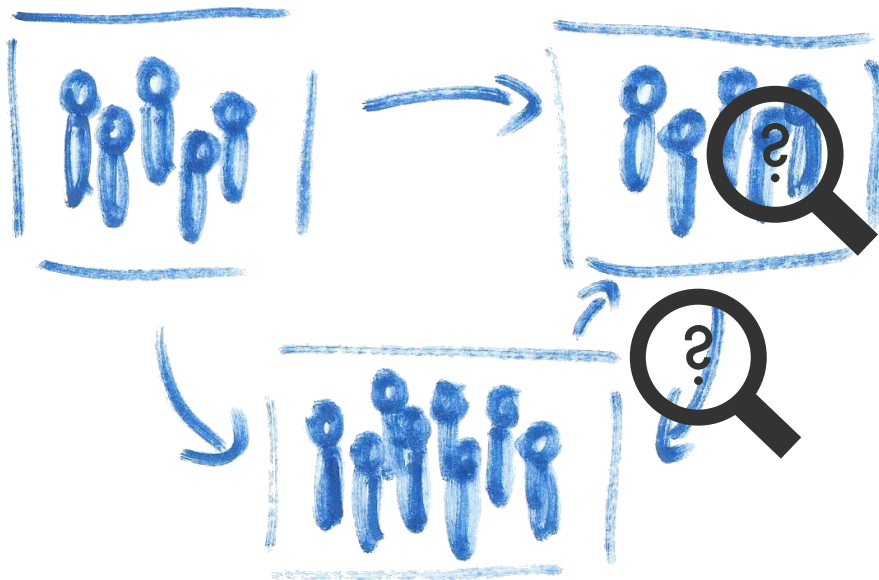


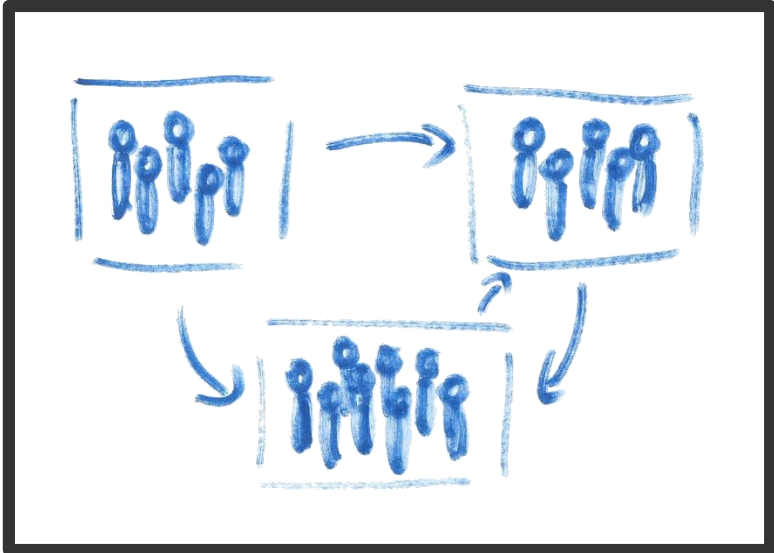
Conway's „Law“

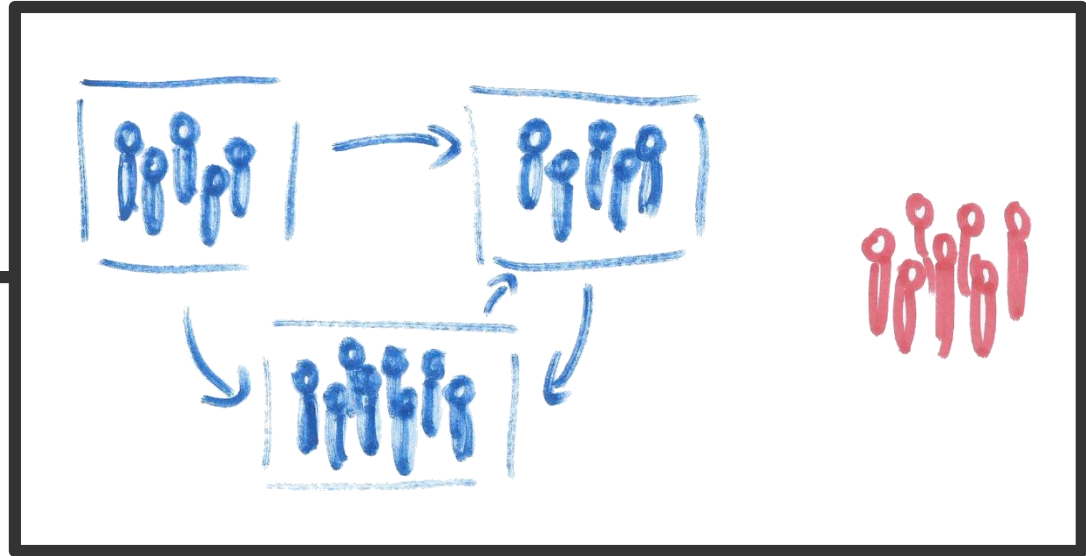
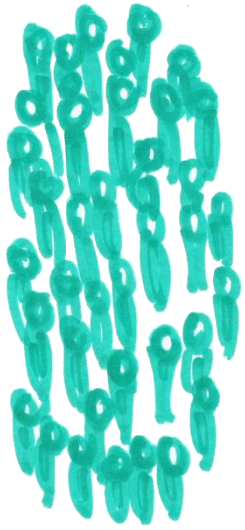
Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.

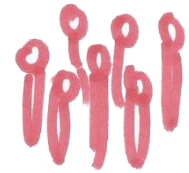
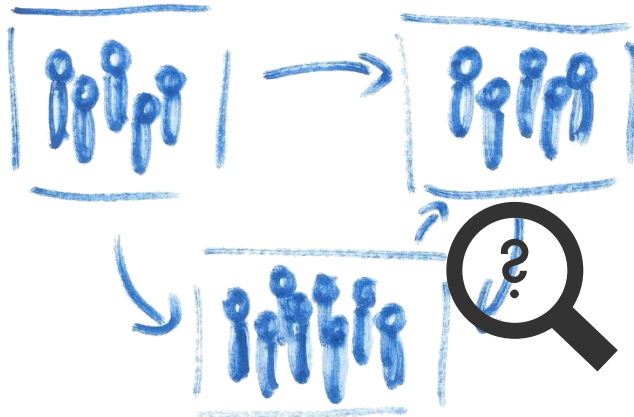
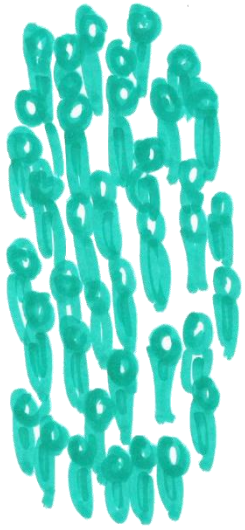
Melvin Conway, 1968

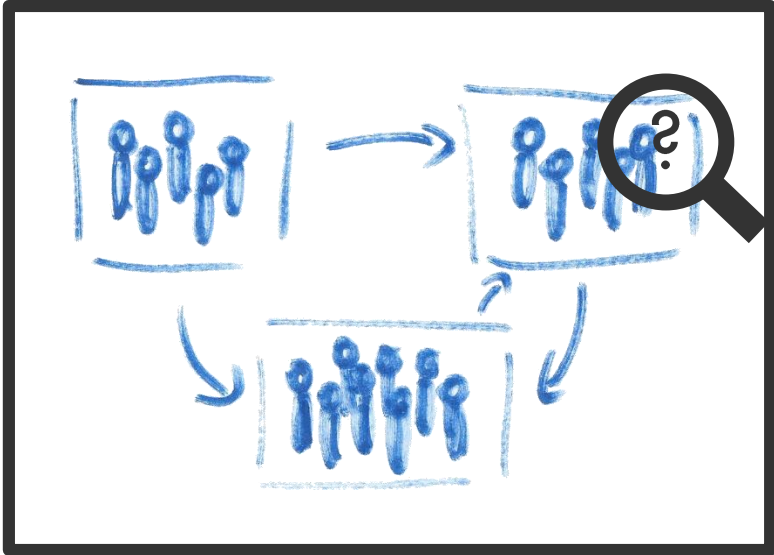






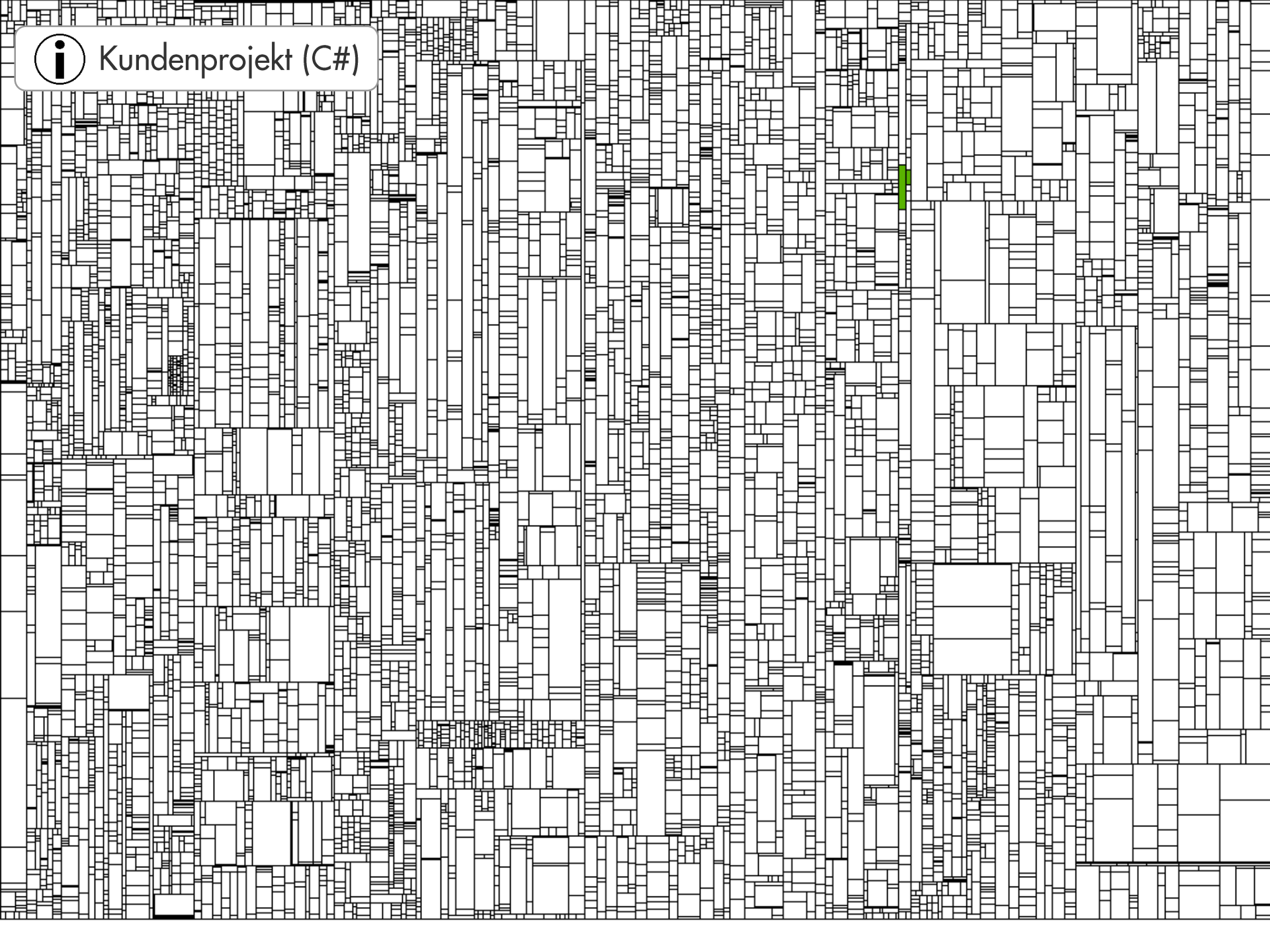








Kundenprojekt (C#)





Entwickler

feilkas
stemplinger
plachot
sahinagic
bader
malinskyi
ribeiro
svejda
hodale
amann
kanis
junker
steidl
streitel
beller
hauptmab
deissenb
hummelb
heineman
poehlman
juergens

Sporadische Committer

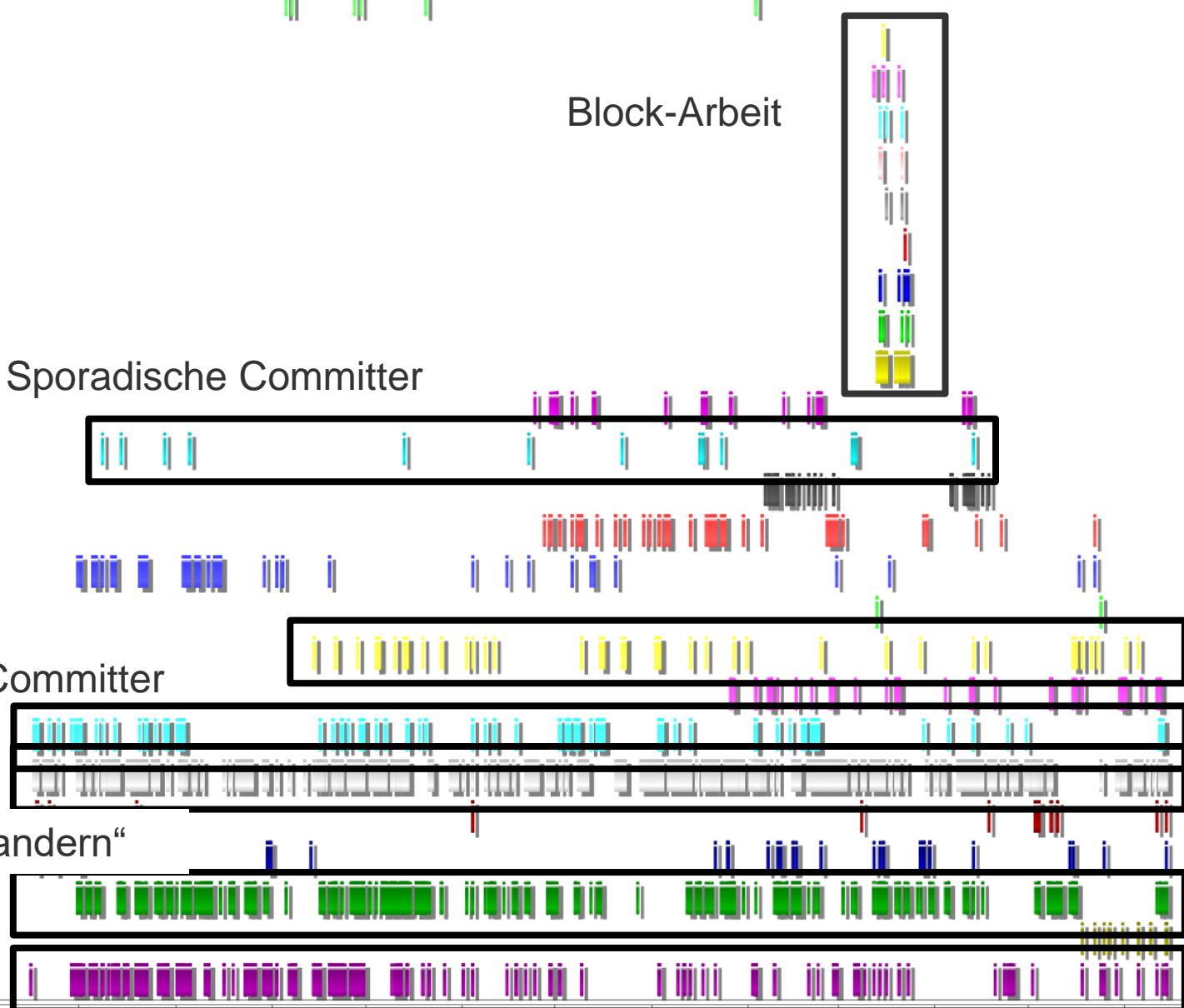
Block-Arbeit

Haupt-Committer

„Abwandern“

Jun-2011 Jul-2011 Aug-2011 Sep-2011 Okt-2011 Nov-2011 Dez-2011 Jan-2012 Feb-2012 Mrz-2012 Apr-2012 Mai-2012 Jun-2012 Jul-2012

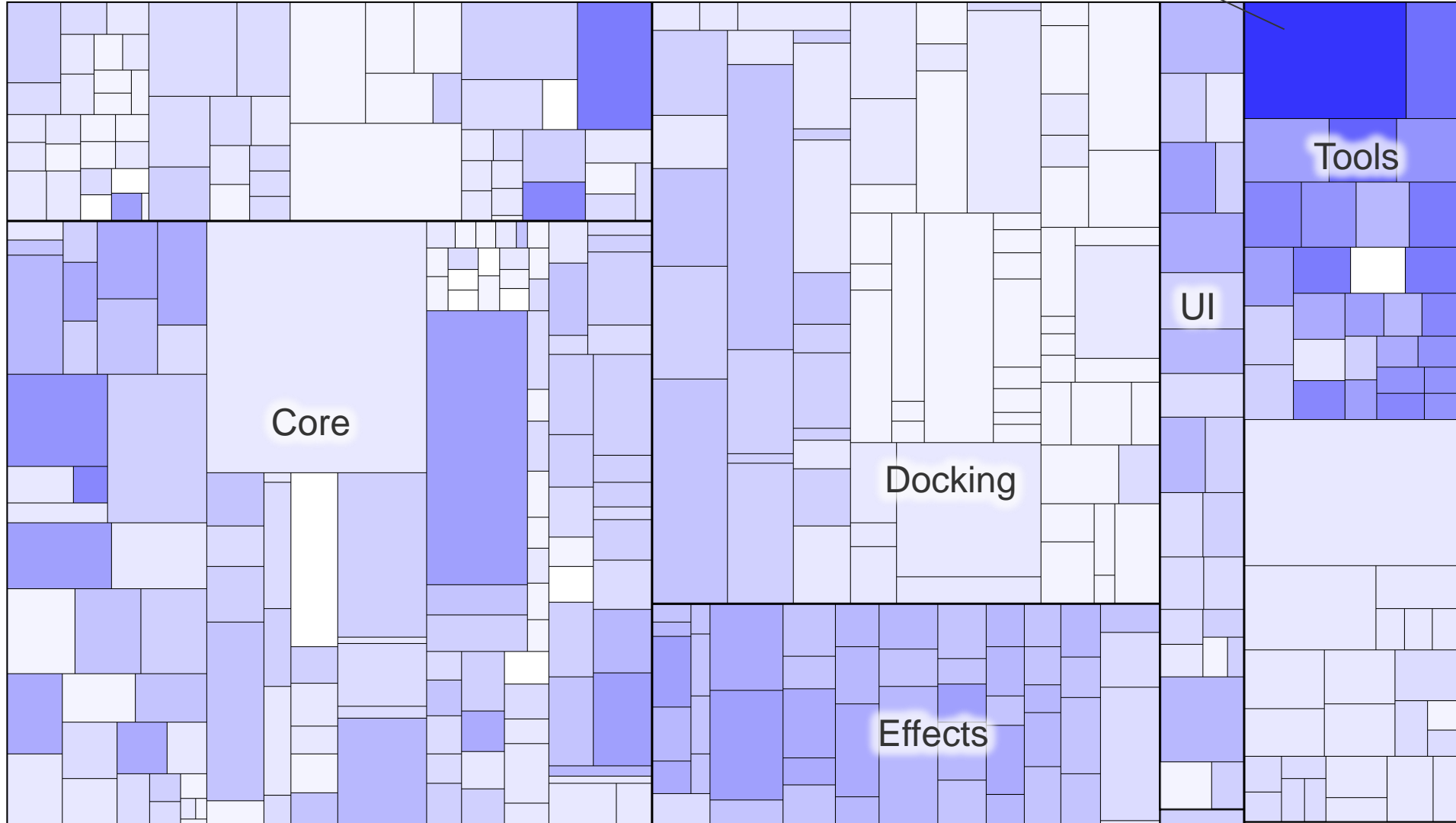
Zeit

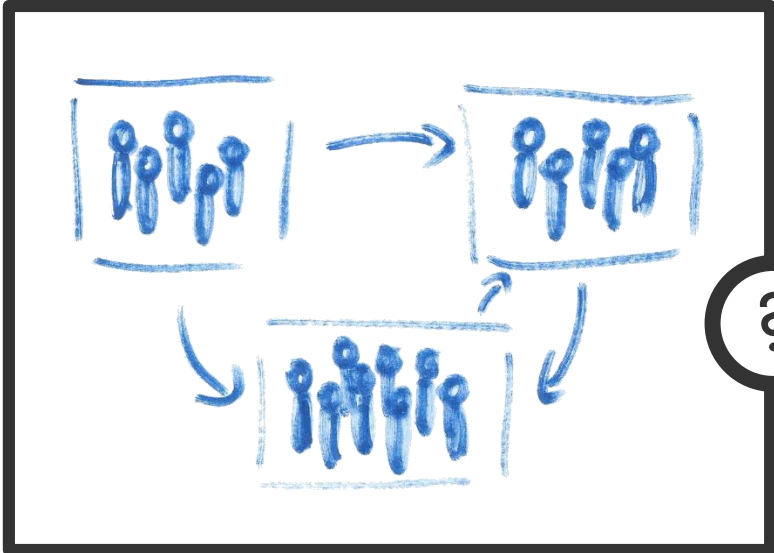


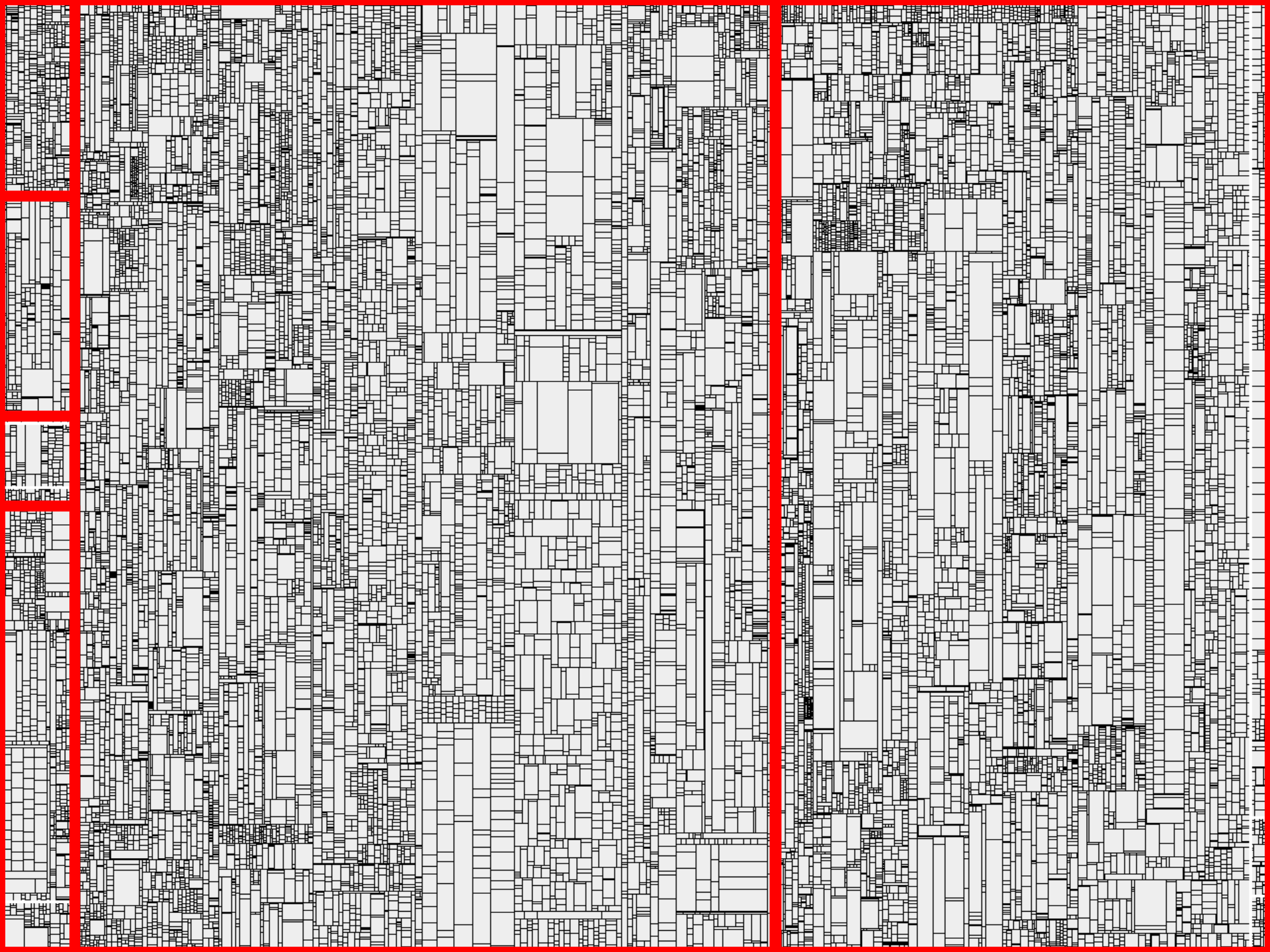


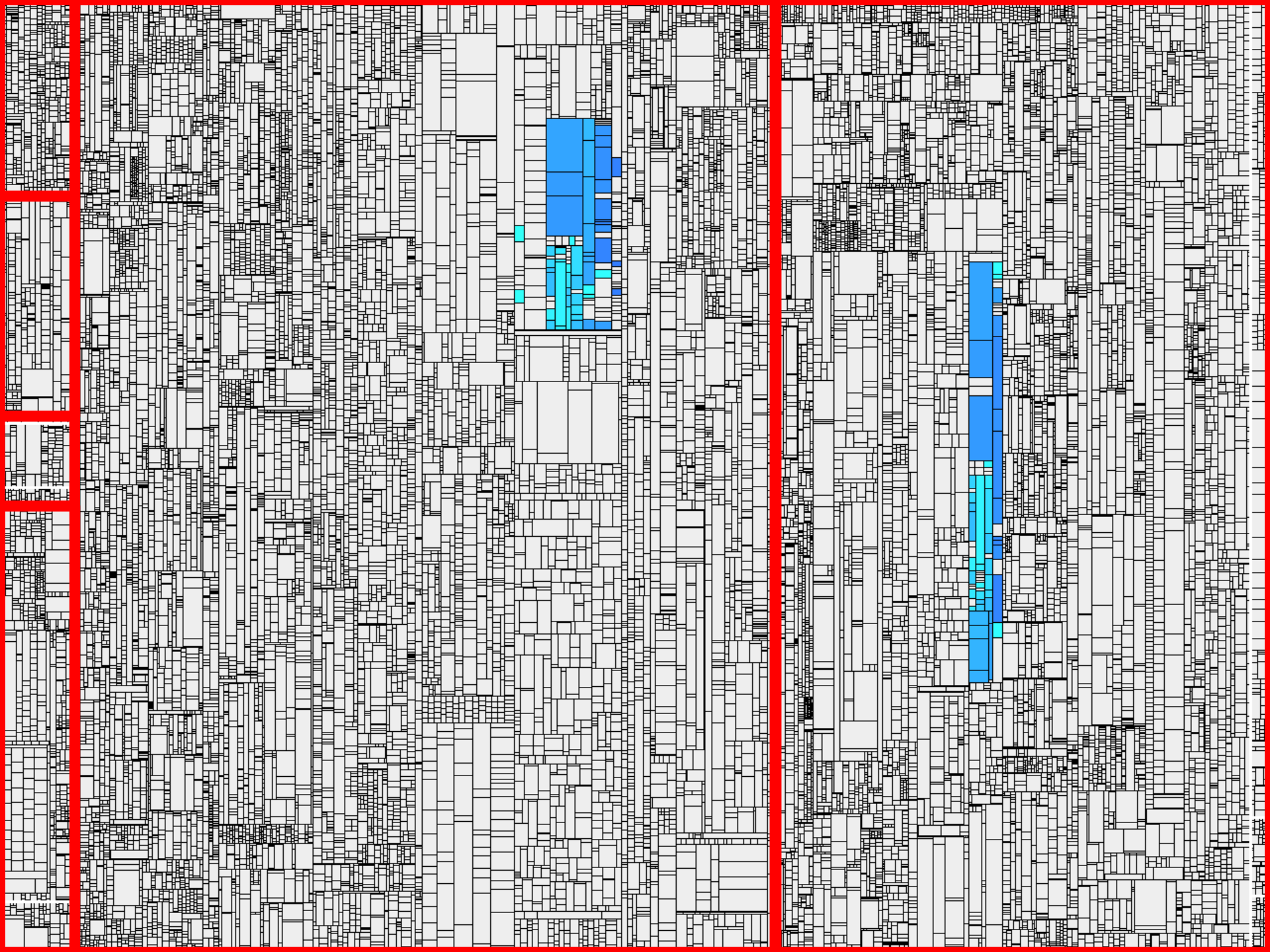
TextTool.cs (17 Entwickler)

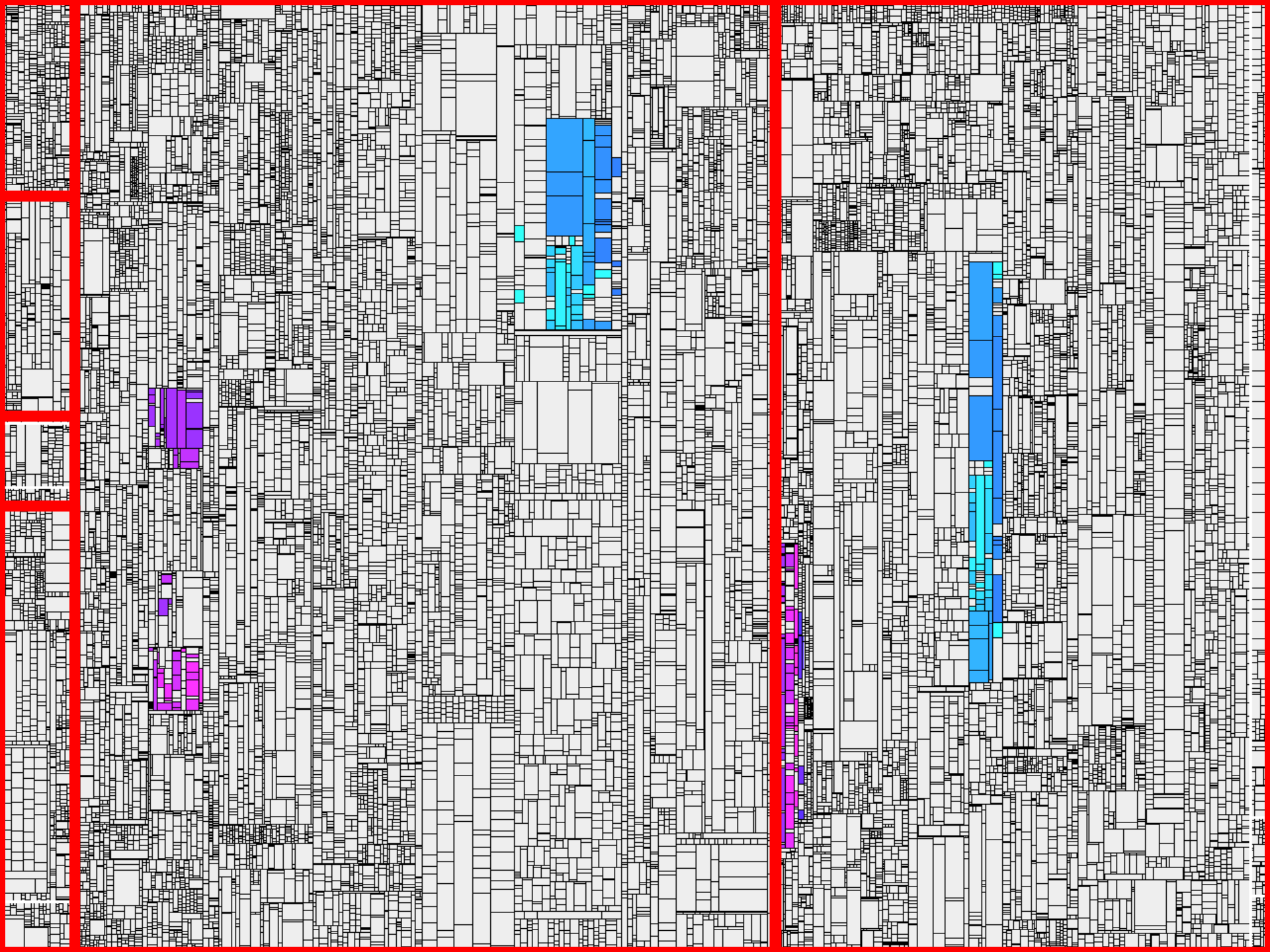
Ownership Distribution for [pinta](#)

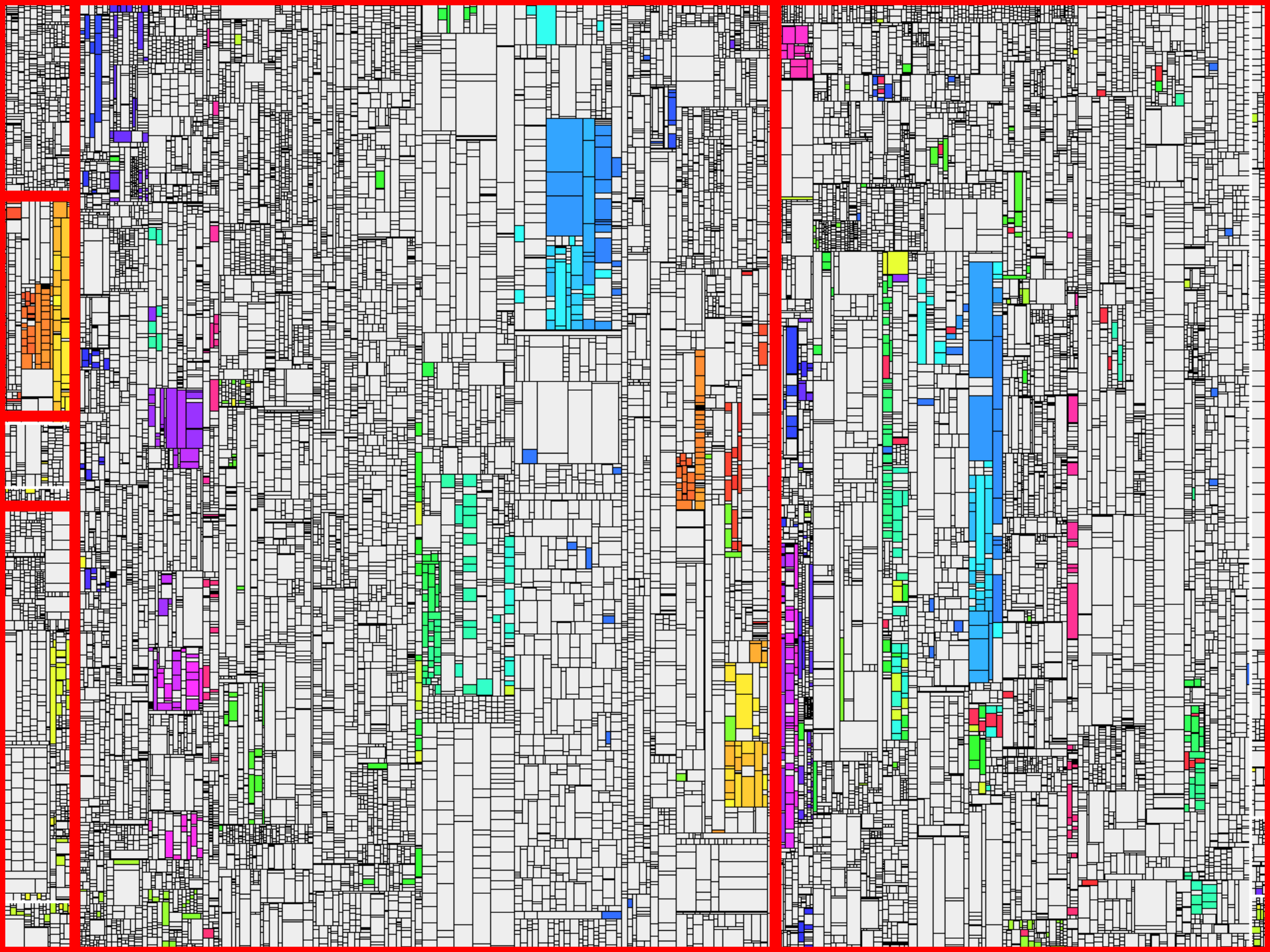


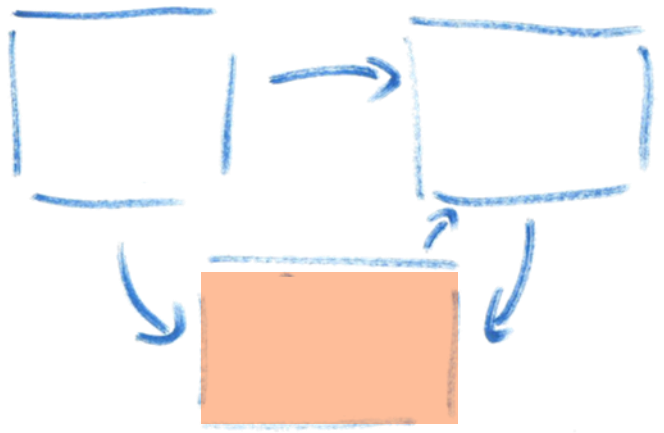
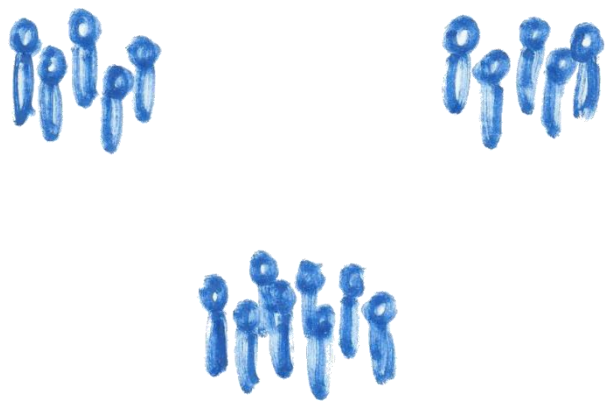










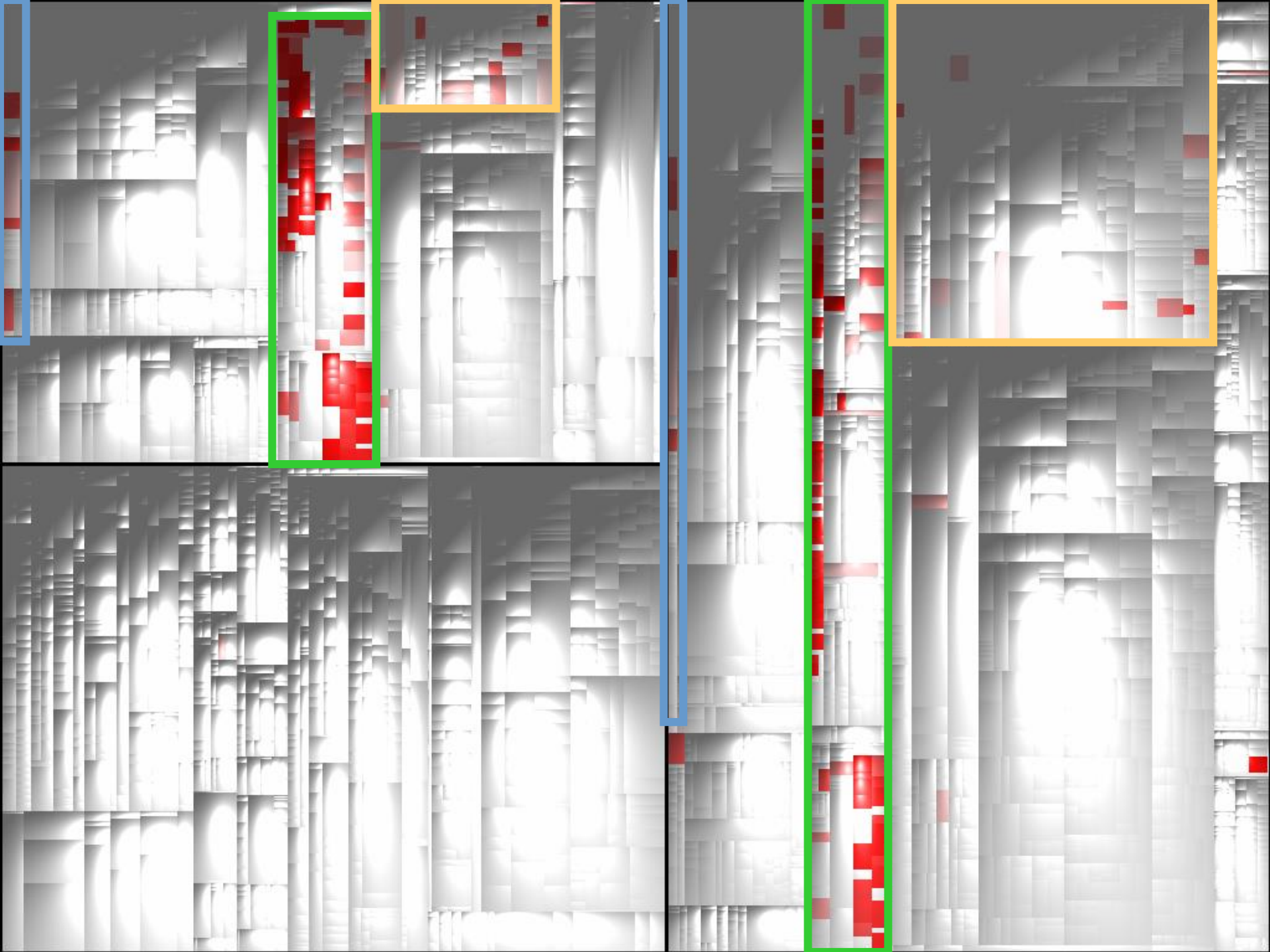


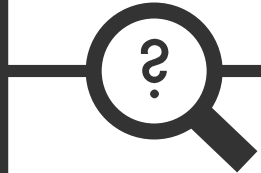
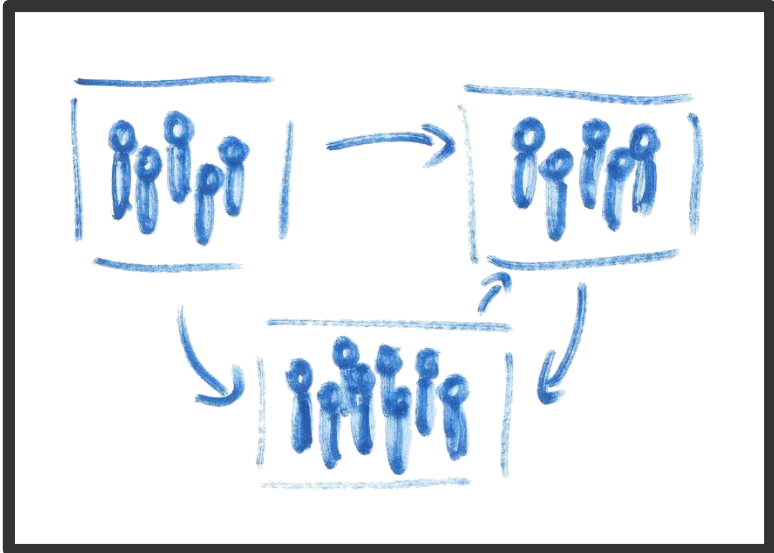
**Mehrere Anwendungen
in einer Abteilung**

375 kloc

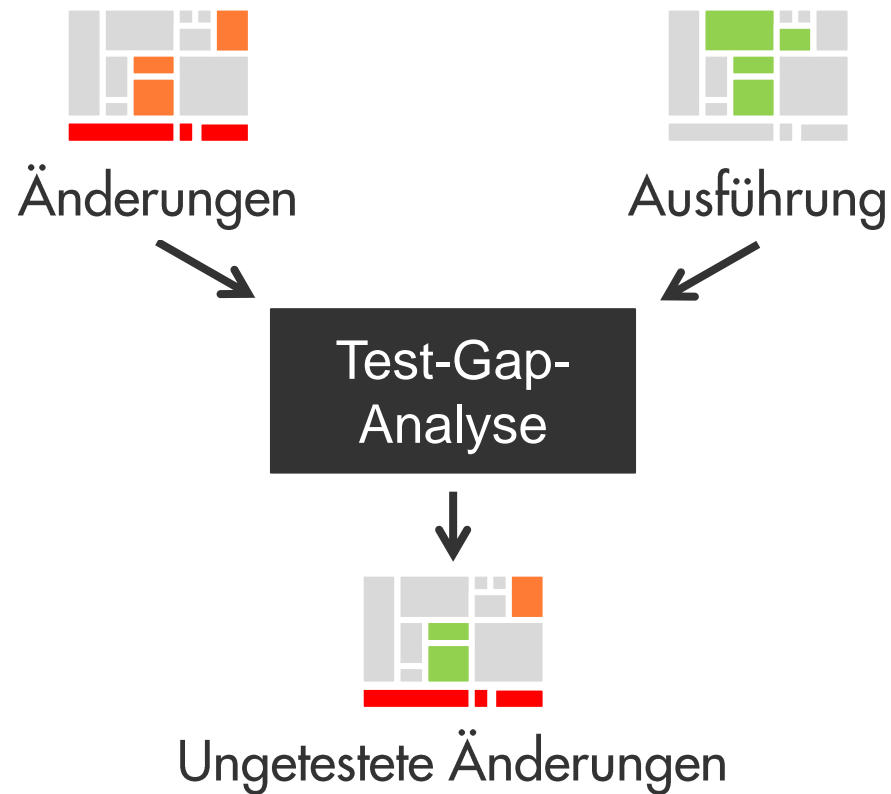
430 kloc

795 kloc





Anatomie Test-Gap-Analyse



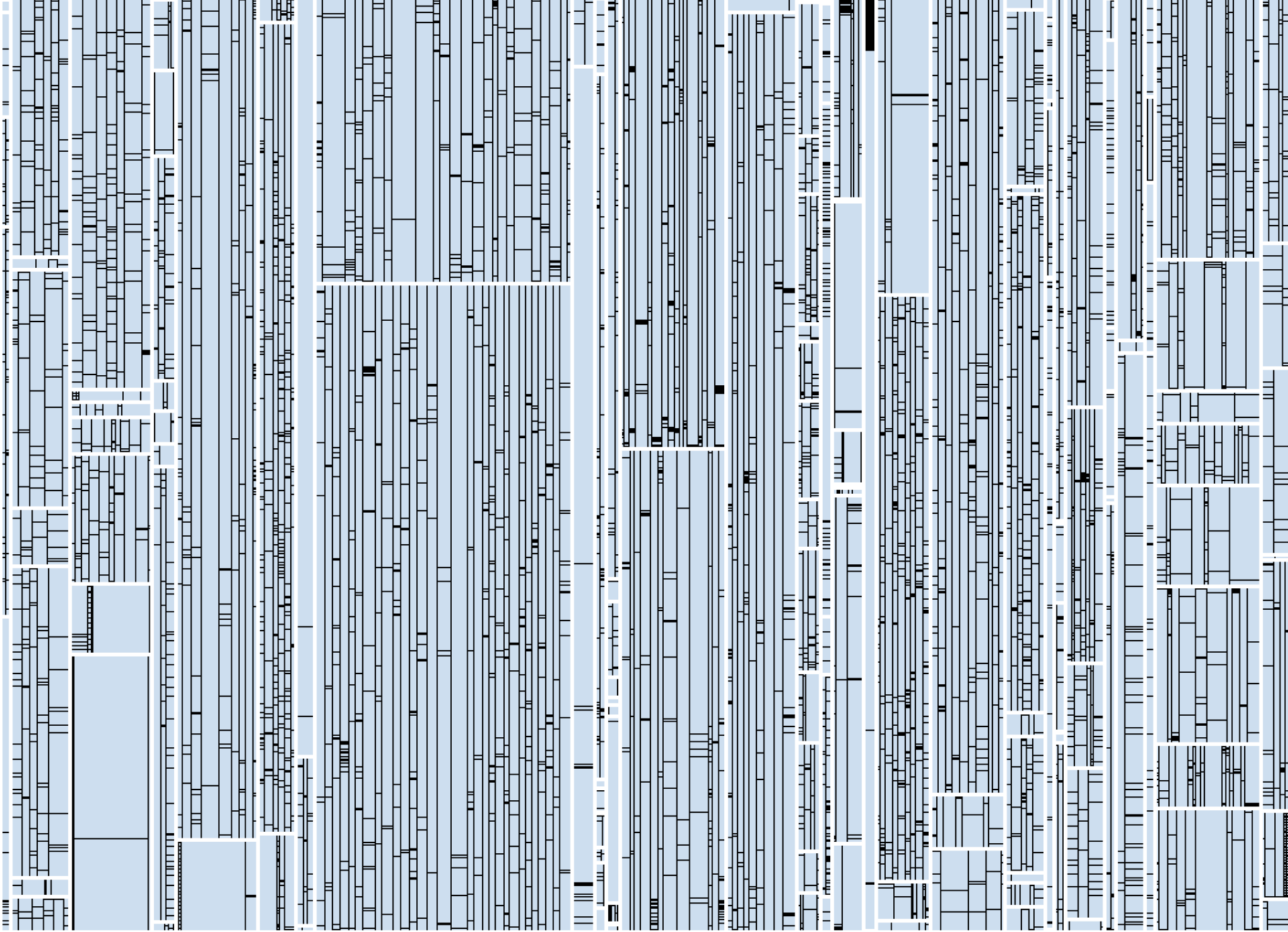
GUI.Dialogs

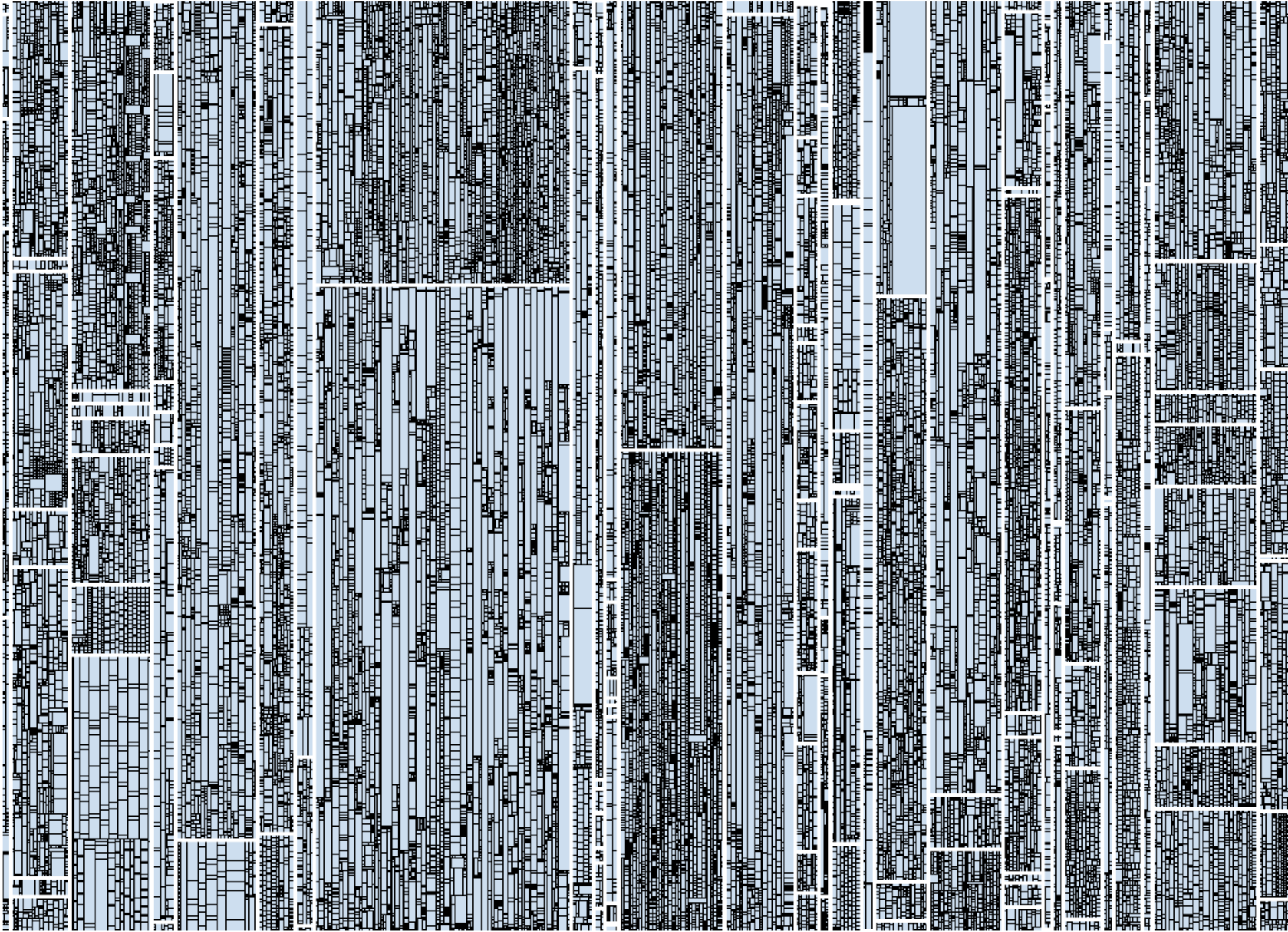
Authentication

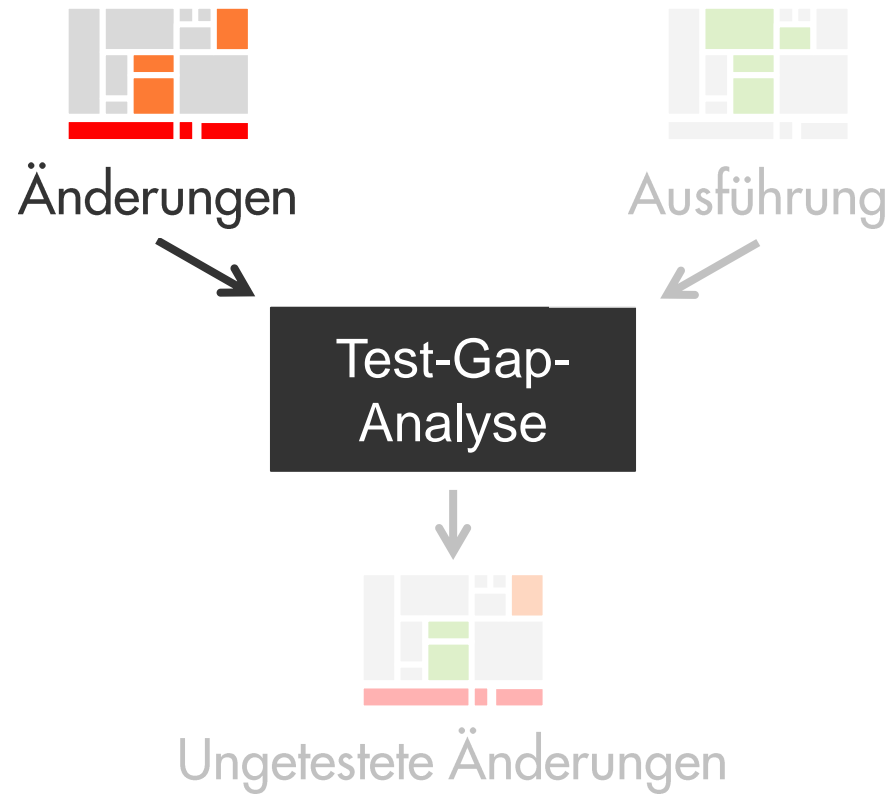
UI Controls

GUI.Base

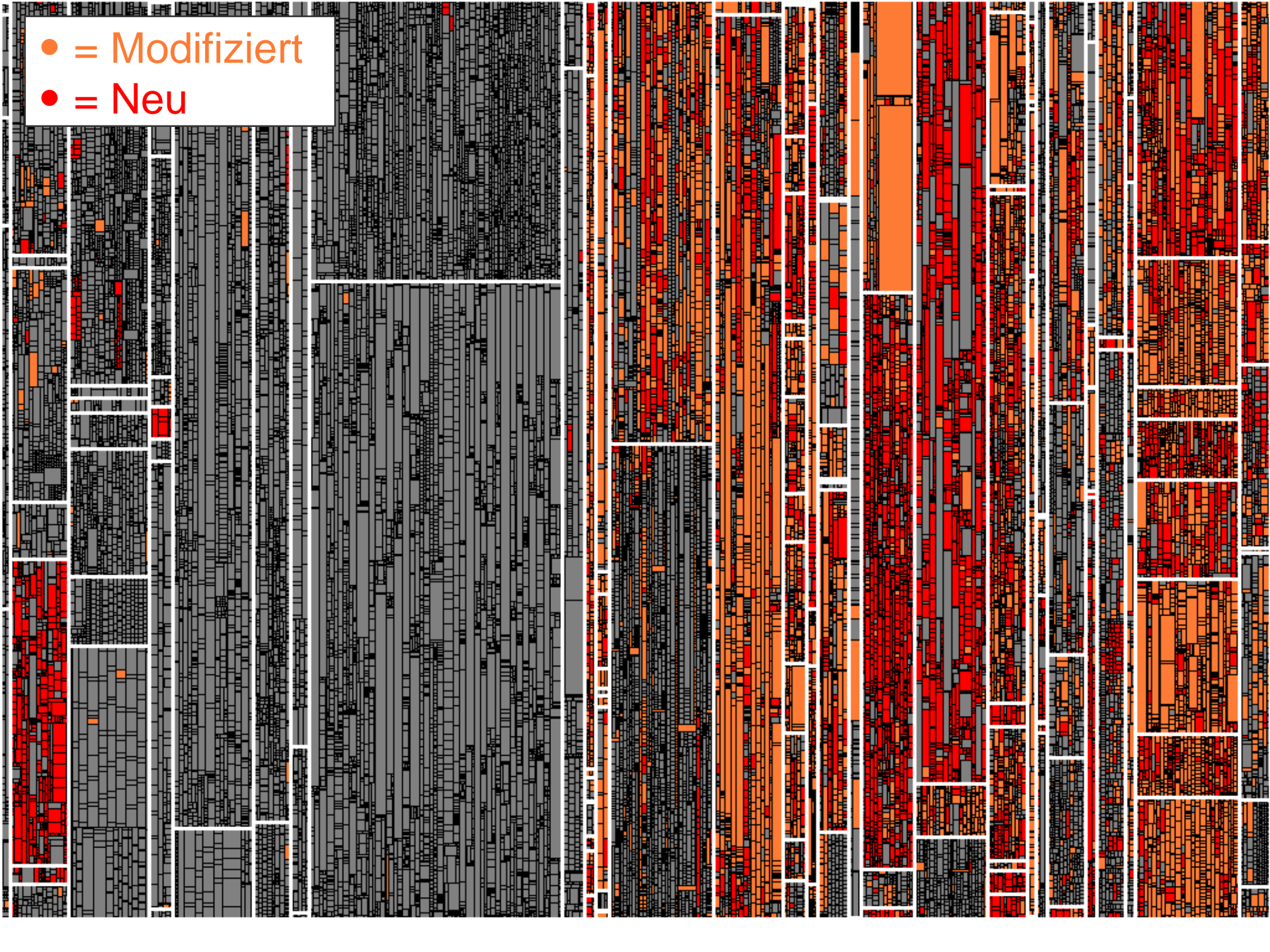
**Data
Validation**

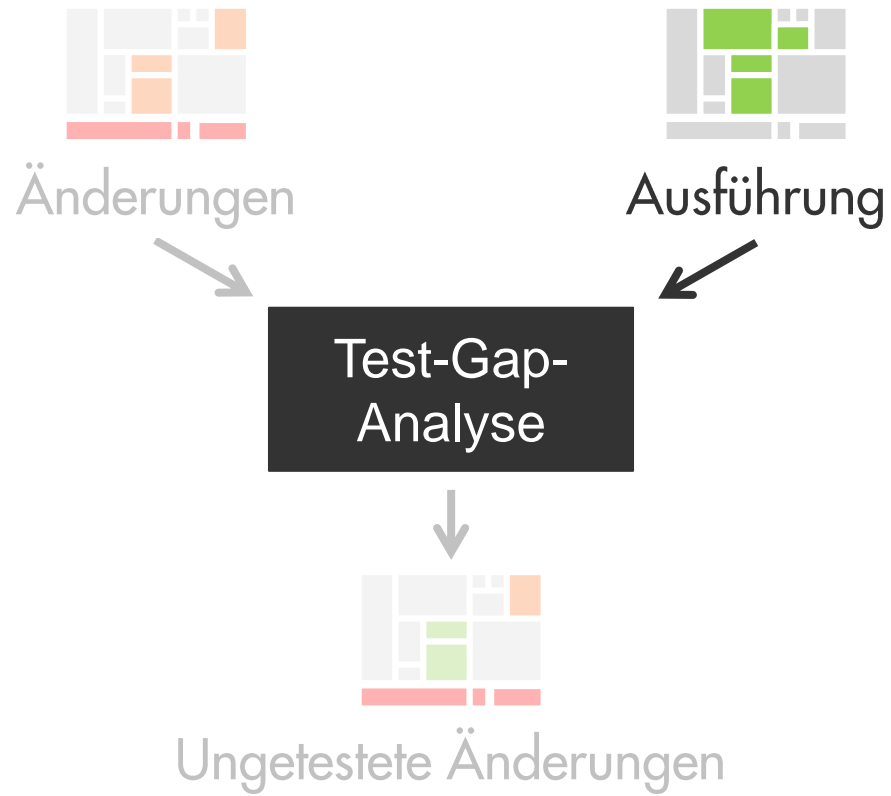




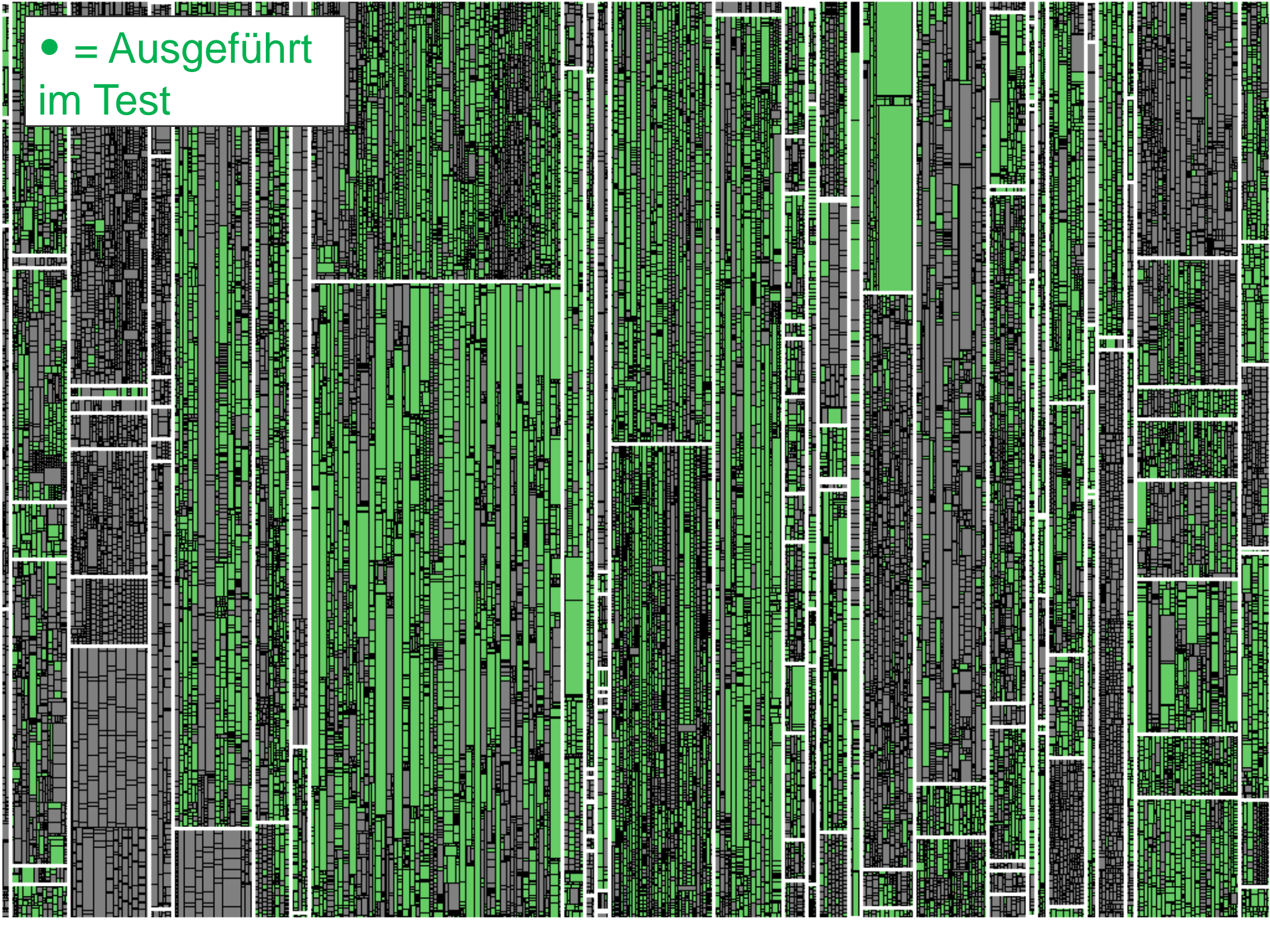


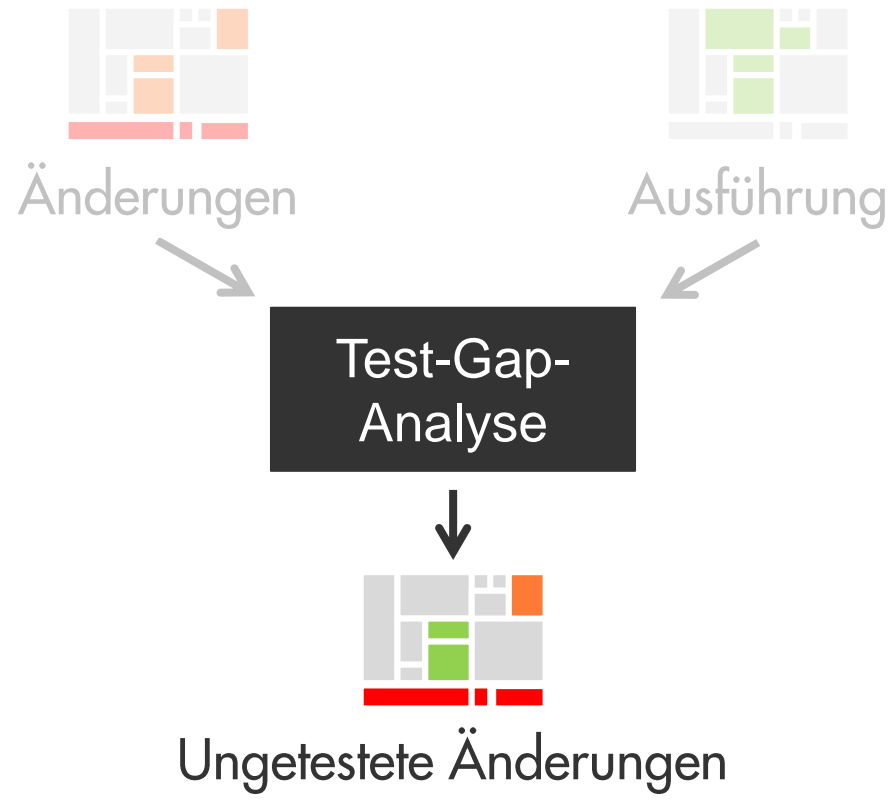
- = Modifiziert
- = Neu



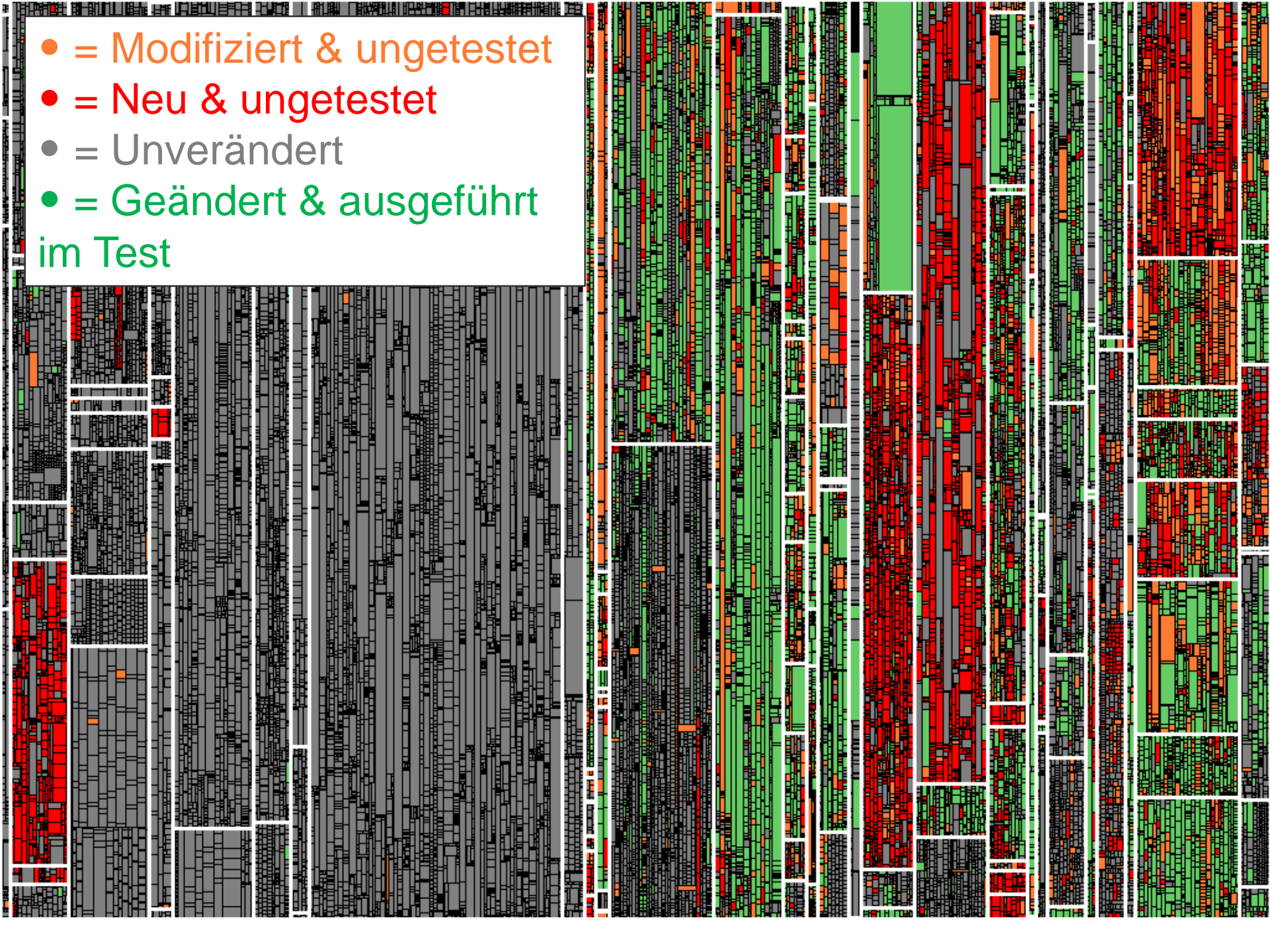


● = Ausgeführt
im Test

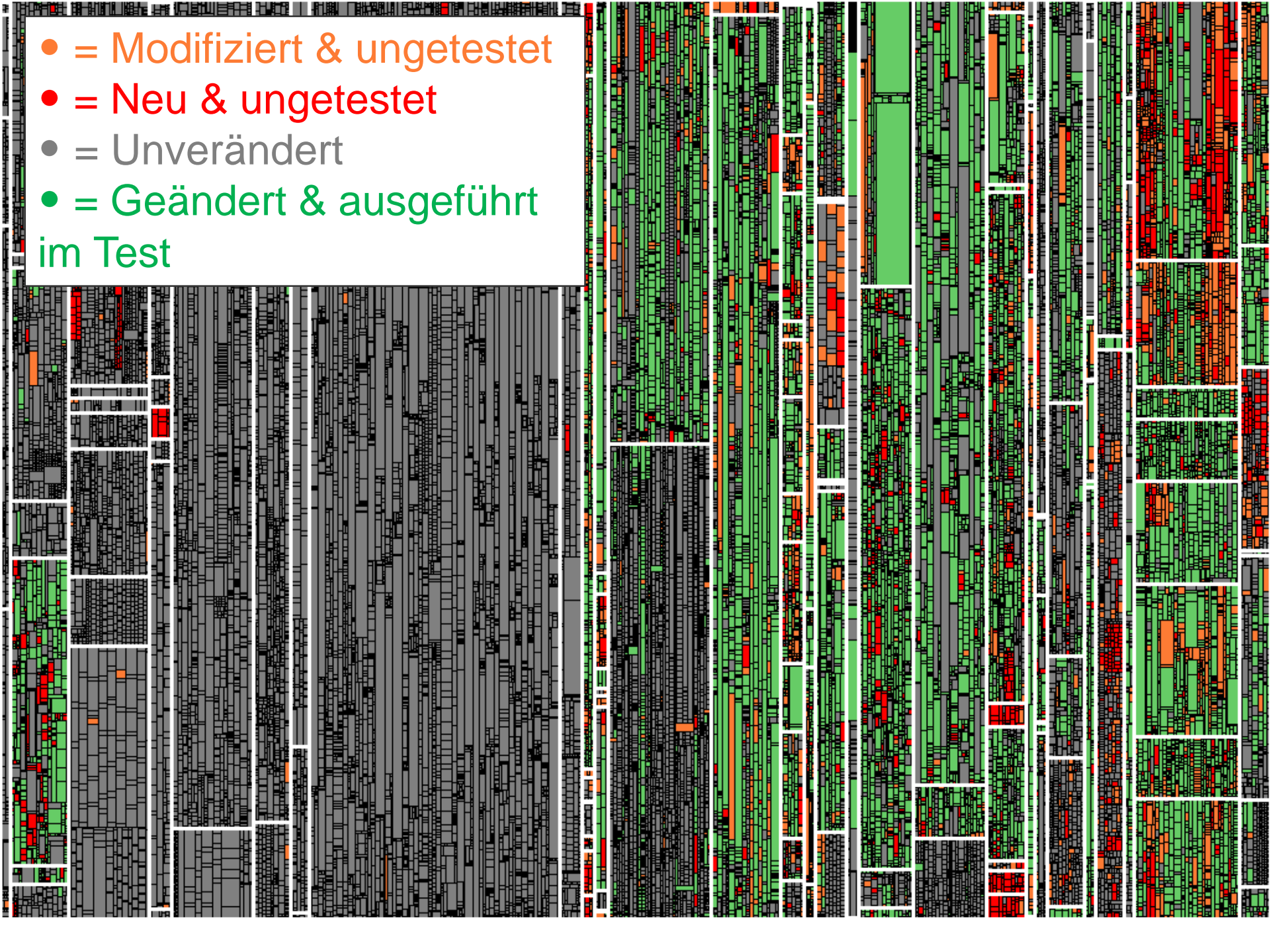




- = Modifiziert & ungetestet
- = Neu & ungetestet
- = Unverändert
- = Geändert & ausgeführt im Test

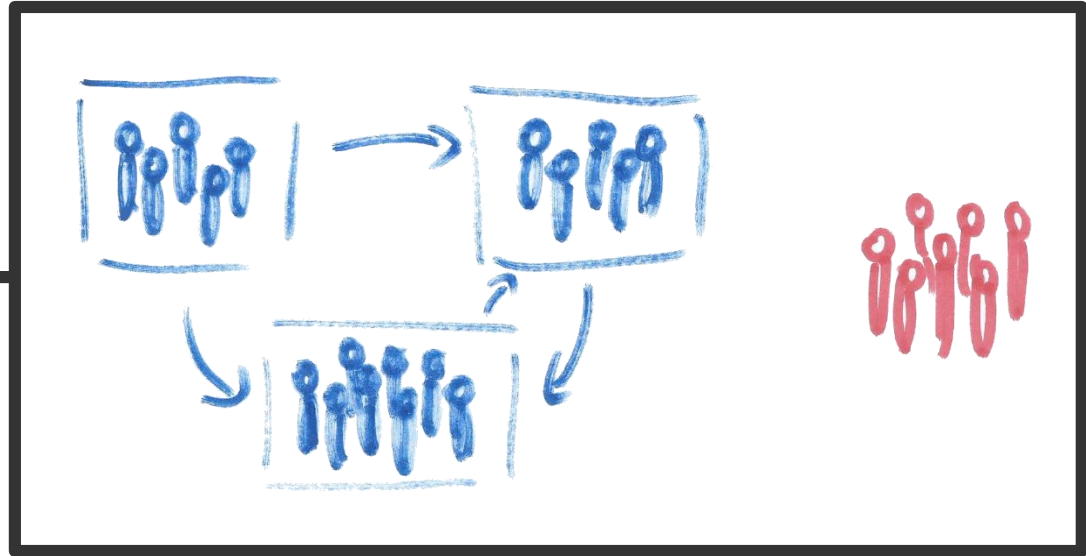
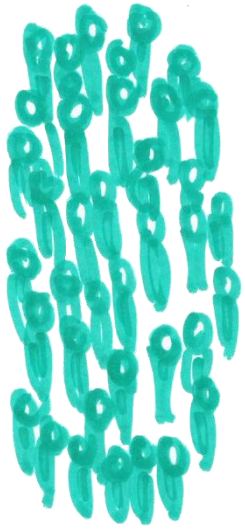


- = Modifiziert & ungetestet
- = Neu & ungetestet
- = Unverändert
- = Geändert & ausgeführt im Test

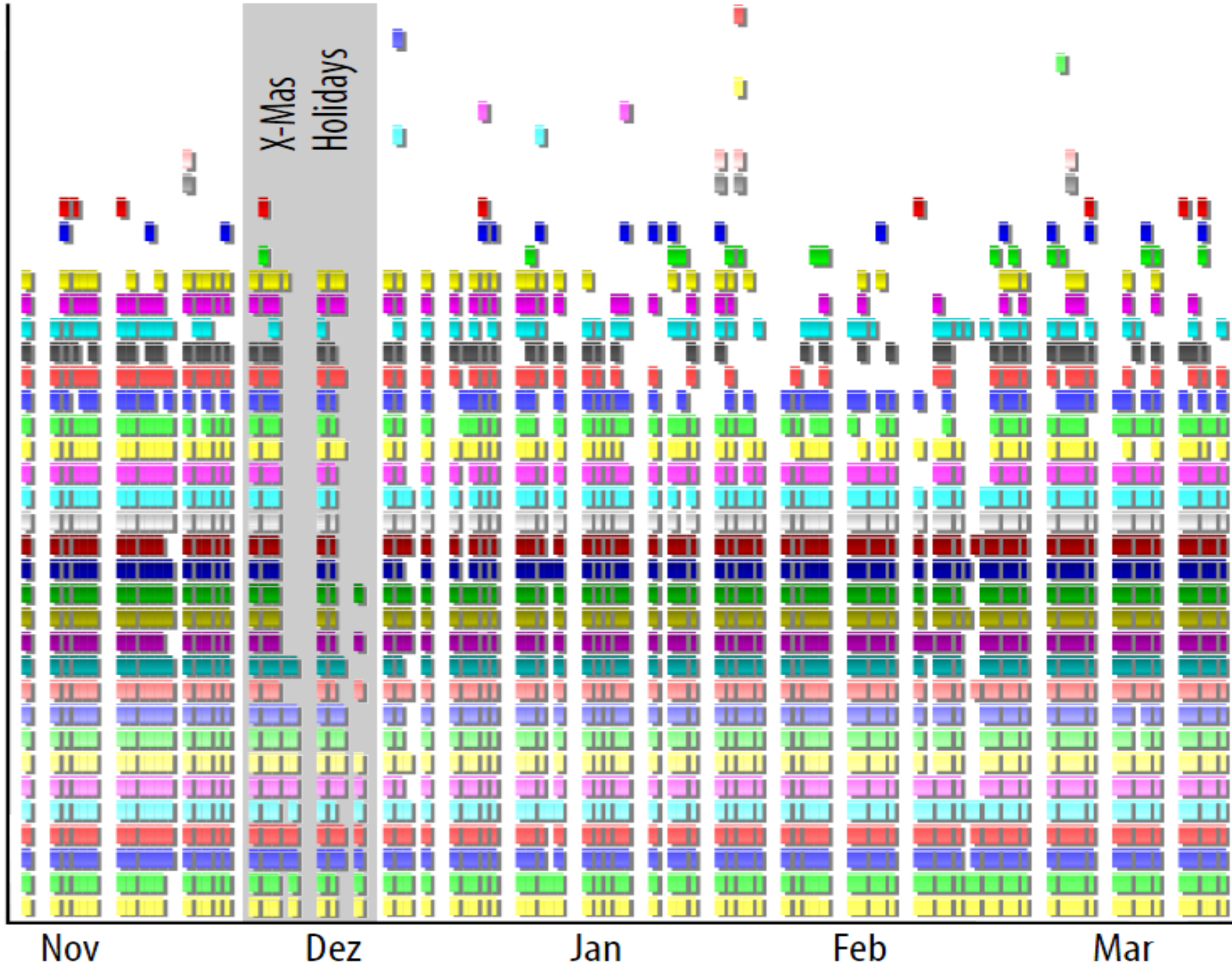




100% Change Coverage → 0 Fehler

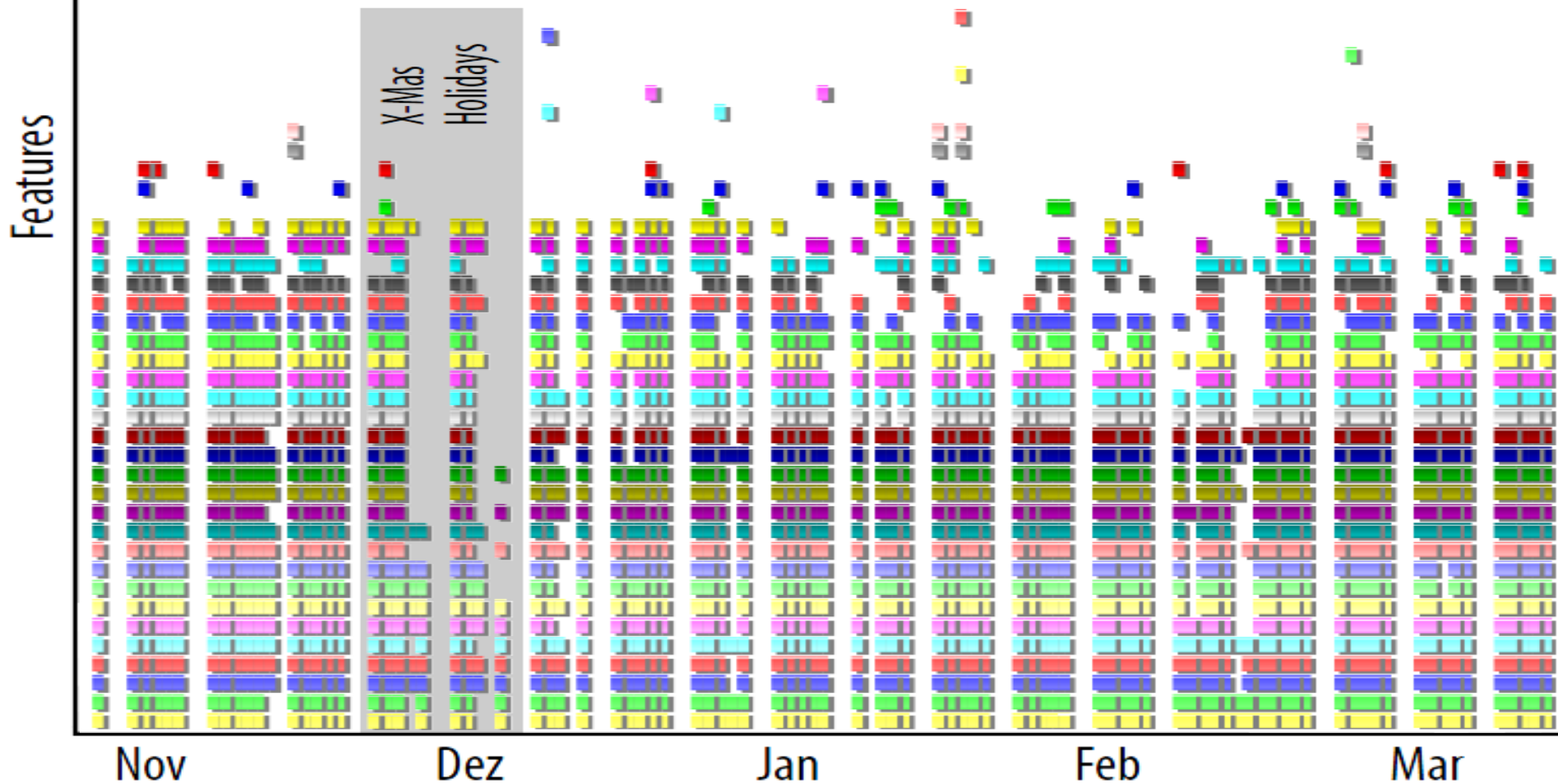


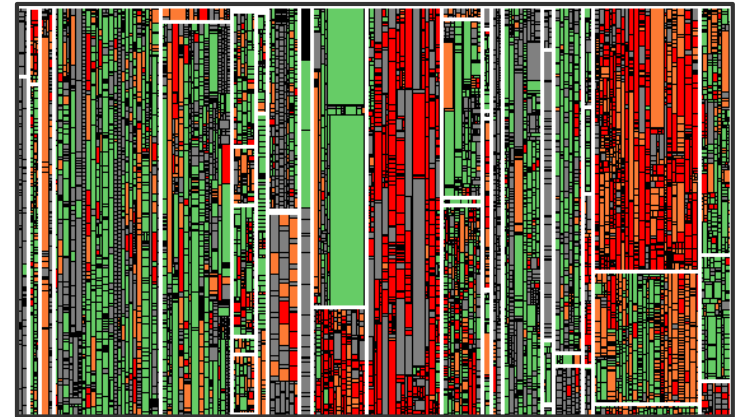
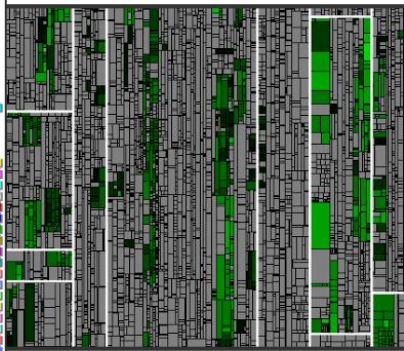
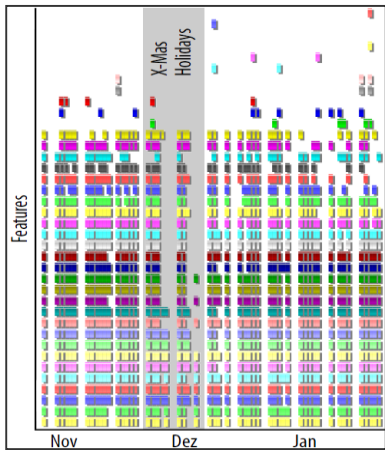
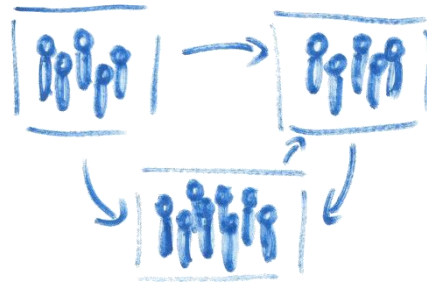
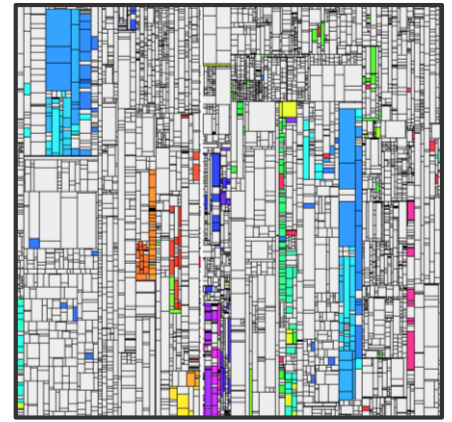
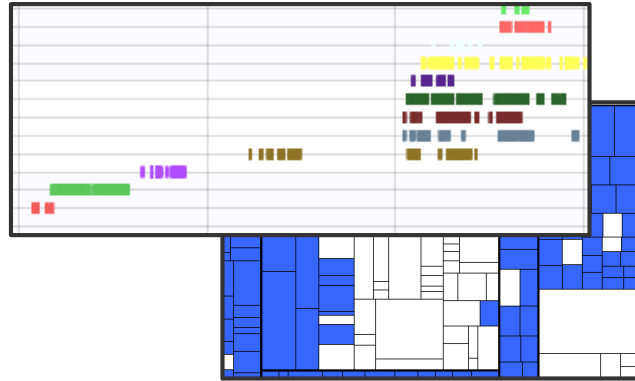
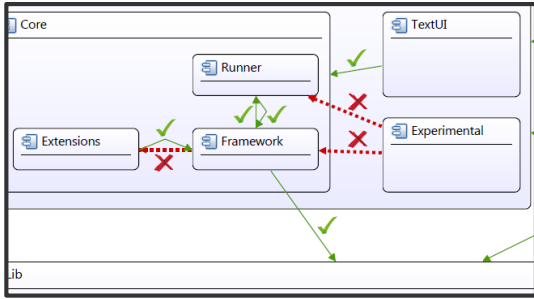
Features



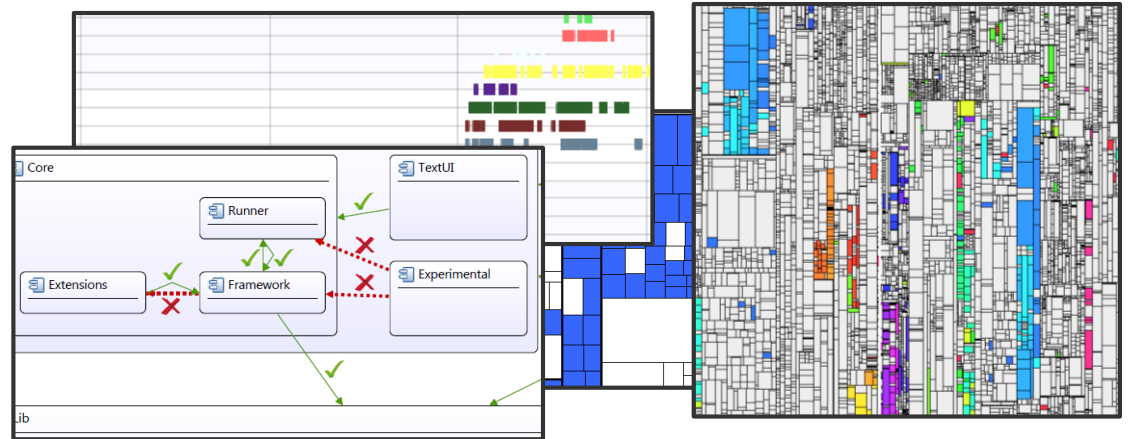
28% Features ungenutzt (15/53)

- 11 unerwartet

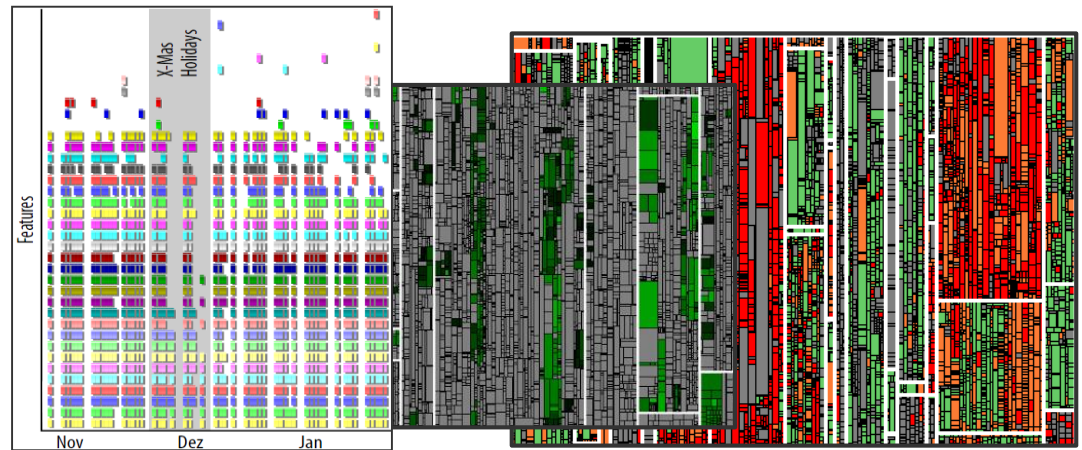




Statische Analysen



Dynamische Analysen



Fazit

Es treten oft Probleme auf, wenn Architektur und Organisation nicht gut zusammenpassen.

Es gibt nützliche dynamische und statische Analysen, die Symptome ermitteln können, damit Ursachen abgestellt werden können.

Sie sind zu einem großen Grad automatisiert, erfordern aber Erfahrung in Konfiguration und Interpretation. Meine Kollegen und ich unterstützen gerne.

<https://www.cqse.eu/en/blog>

Bridge your Test Gaps with Teamscale

Posted on 04/27/2016 by Dr. Dennis Pagano

Many companies employ sophisticated testing processes, but still bugs find their way into production. Often they hide among the subset of changes that were not tested. Actually, we found that untested changes are five times more error prone than the rest of the system.



The screenshot shows the Teamscale dashboard interface. The main area displays a treemap visualization titled "Test Gaps for pinta" with a "Test Gap: 77.78%" indicator. The treemap consists of numerous small rectangular blocks, some colored red, yellow, and green, representing different components and their associated test gaps. To the right of the treemap is a sidebar titled "Editing Test Gaps Pinta" which contains a list of visualization options: Label, Numeric Metric Value, Assessment Pie Chart, Metrics Table, Metrics Bar Chart, Metric Distribution Table, Metric Distribution Pie Chart, Metrics Hotspot Table, Metrics Change Table, Findings Summary Table, Findings Summary Bar Chart, Findings Churn, Metrics Trend Chart, Assessment Trend Chart, Treemap, Test Gap Treemap, Test Gap Trend Chart, Code City, and Metric File Distribution. At the bottom of the sidebar are "Save", "Save as", and "Cancel" buttons.

To avoid code that is not consciously

A short quick reference

Present

Obviously

To use

Please contact

Test

Test Gaps tested test gaps

regularly.

average can be found here.

changes made on the way have been want to show you how to find

Testing Changes in SAP BW Applications

Posted on 04/29/2015 by Dr. Andreas Göb

As my colleague Fabian explained a few weeks ago, a combination of change detection and execution logging can substantially increase transparency regarding which recent changes of a software system have actually been covered by the testing process. I will not repeat all the details of the Test Gap Analysis approach here, but instead just summarize the core idea: Untested new or changed (informative) information. Therefore it makes sense to use those changed but untested areas.

Several times in the main from Python containing may provide Analysis in

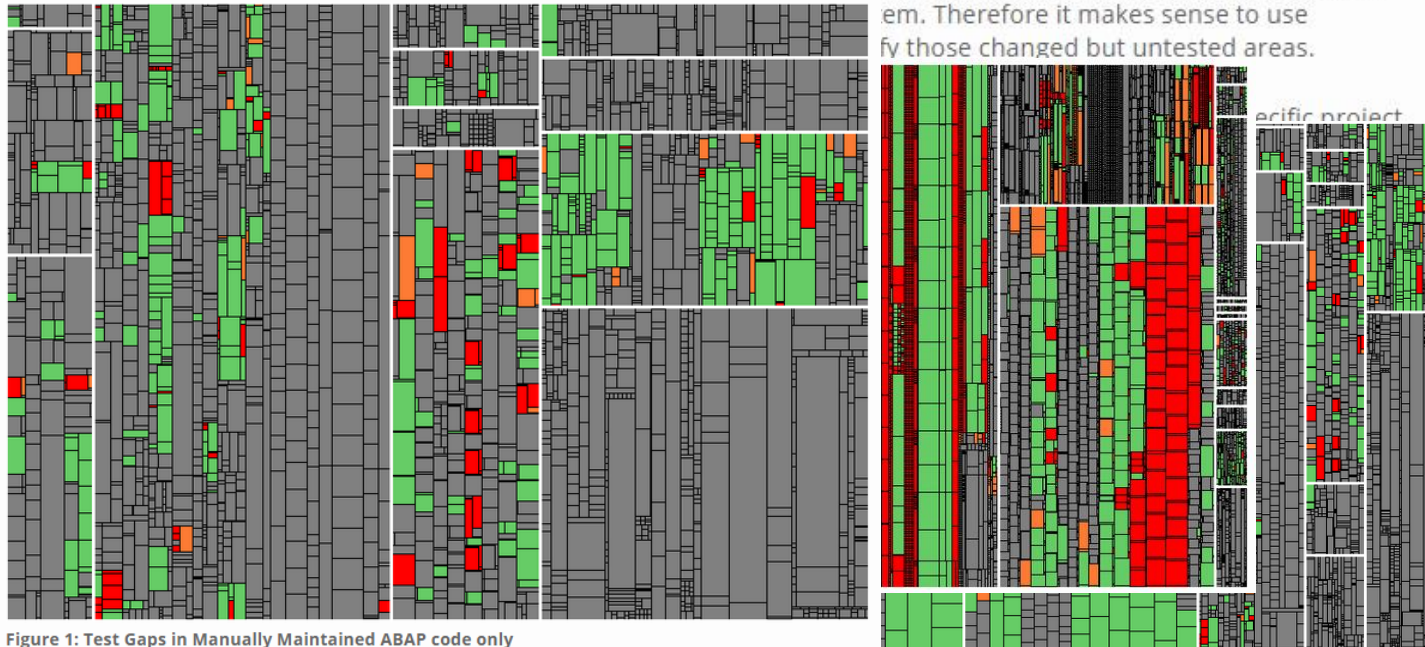


Figure 1: Test Gaps in Manually Maintained ABAP code only



Software Quality Blog

Practical Guide to Code Clones (Part 1)

Posted on 07/16/2014 by Dr. Benjamin Hummel

One well known principle in software engineering states *don't repeat yourself*, also known as the DRY principle. A very obvious violation of DRY is the application of copy/paste to create duplicates of large portions of source code within the same code base. These duplicate pieces of code, also known as *code clones*, have been subject to lots of research in the last two decades. In this two-part post I want to summarize those parts of the current knowledge that I find most relevant to the practitioner, especially the impact of clones on software development.



Practical Guide to Code Clones (Part 2)

Posted on 07/30/2014 by Dr. Benjamin Hummel

In the previous part we introduced the notion of code clones and discussed, whether and under which circumstances cloning in your code base can be a problem for development and maintenance. In this post, I will introduce ways and tools to deal with code clones in your code base. After reading this, you should be able to select and apply a detection tool to inspect the clones in your own code base.

Kontakt

Dr. Elmar Jürgens · juergens@cqse.eu · +49 179 675 3863

@ElmarJuergens

www.cqse.eu/en/blog

CQSE GmbH

Lichtenbergstraße 8

85748 Garching bei München