

Kann uns AI helfen, besser zu testen?

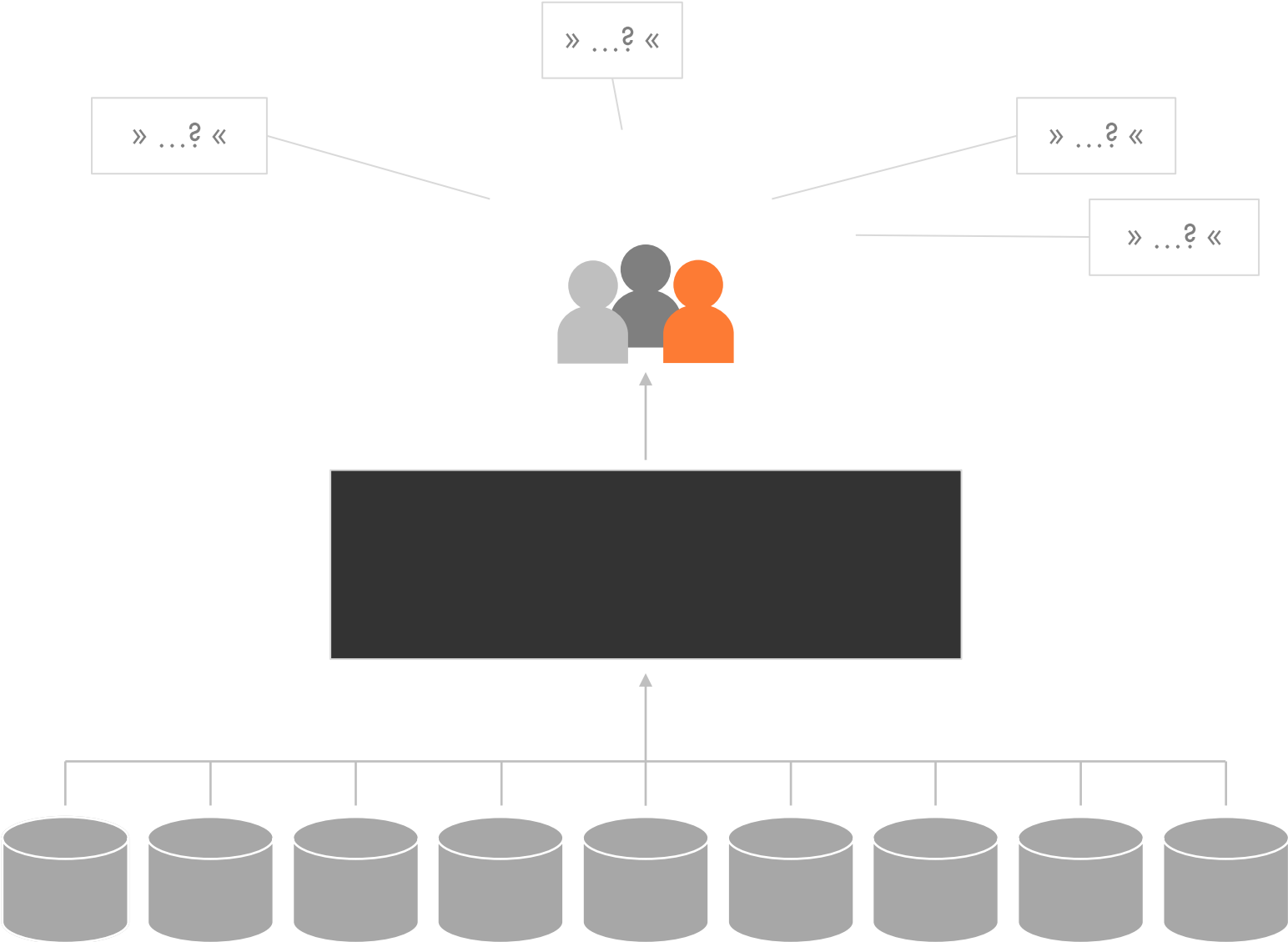
Erfahrungen mit Machine Learning in der QS in Forschung und Praxis



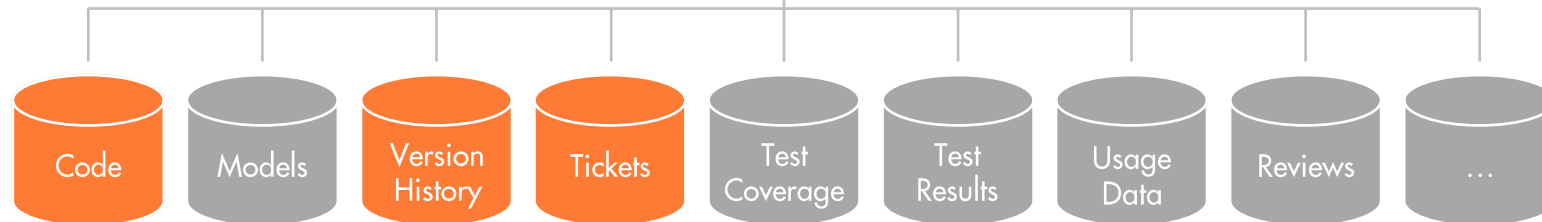
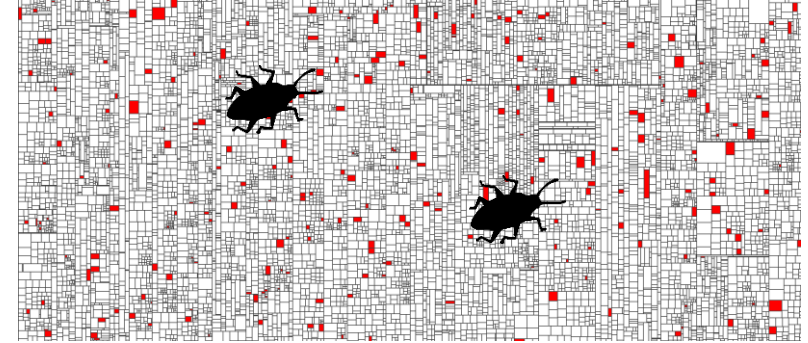
TUM



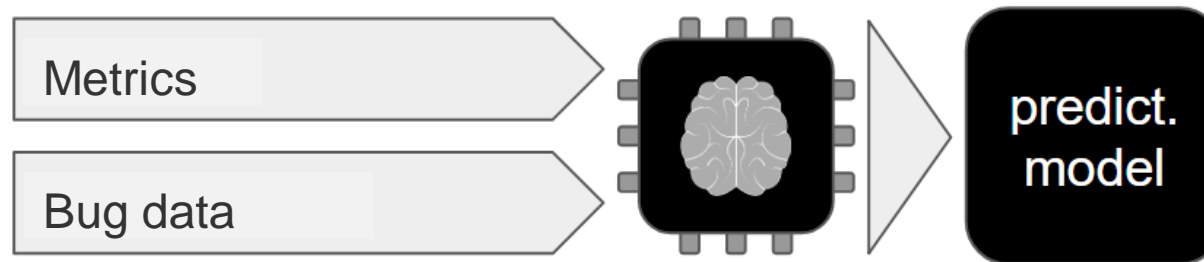
CQSE

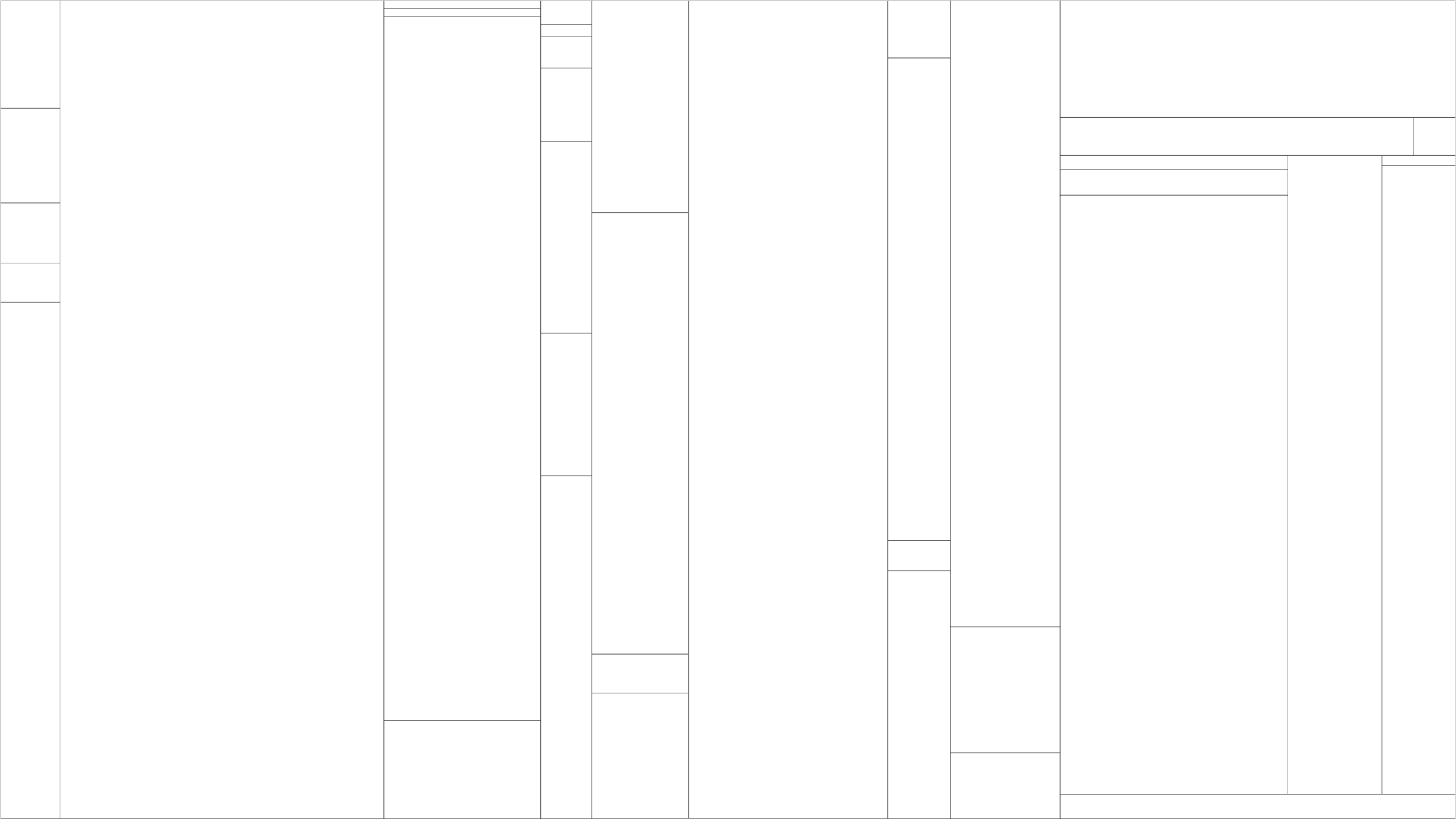


» Wo sind Fehler im Code? «

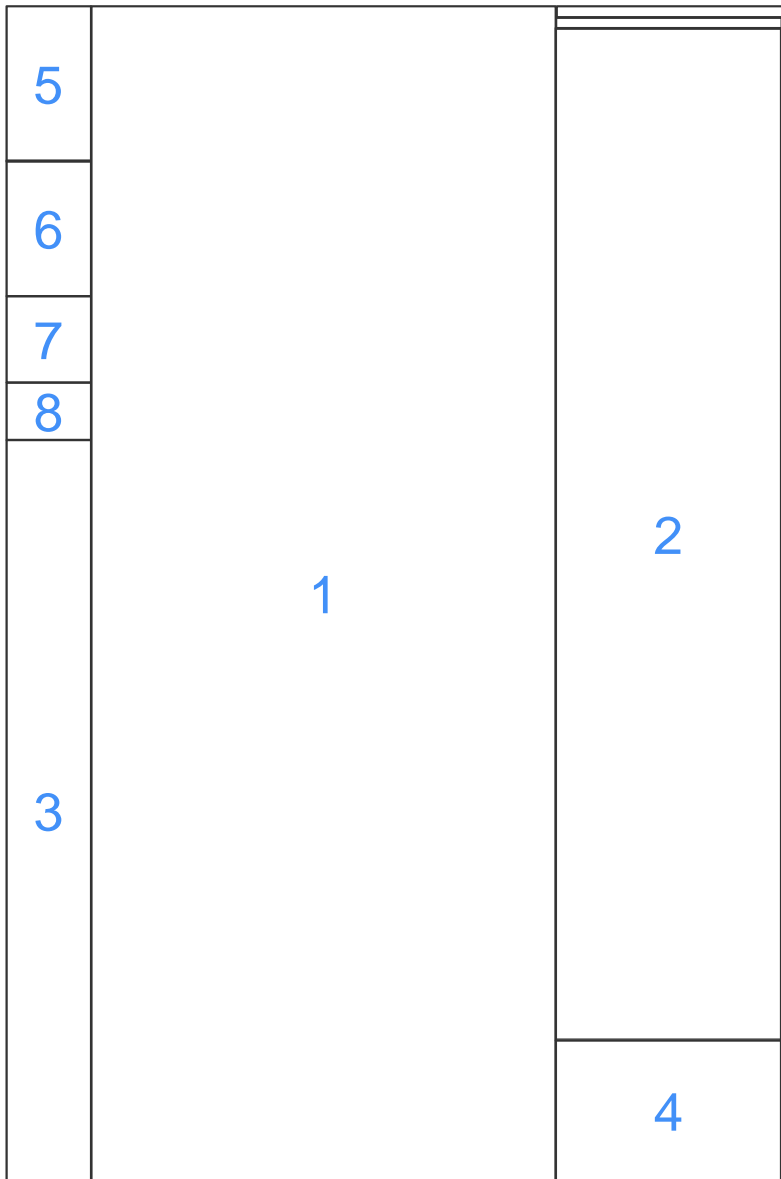


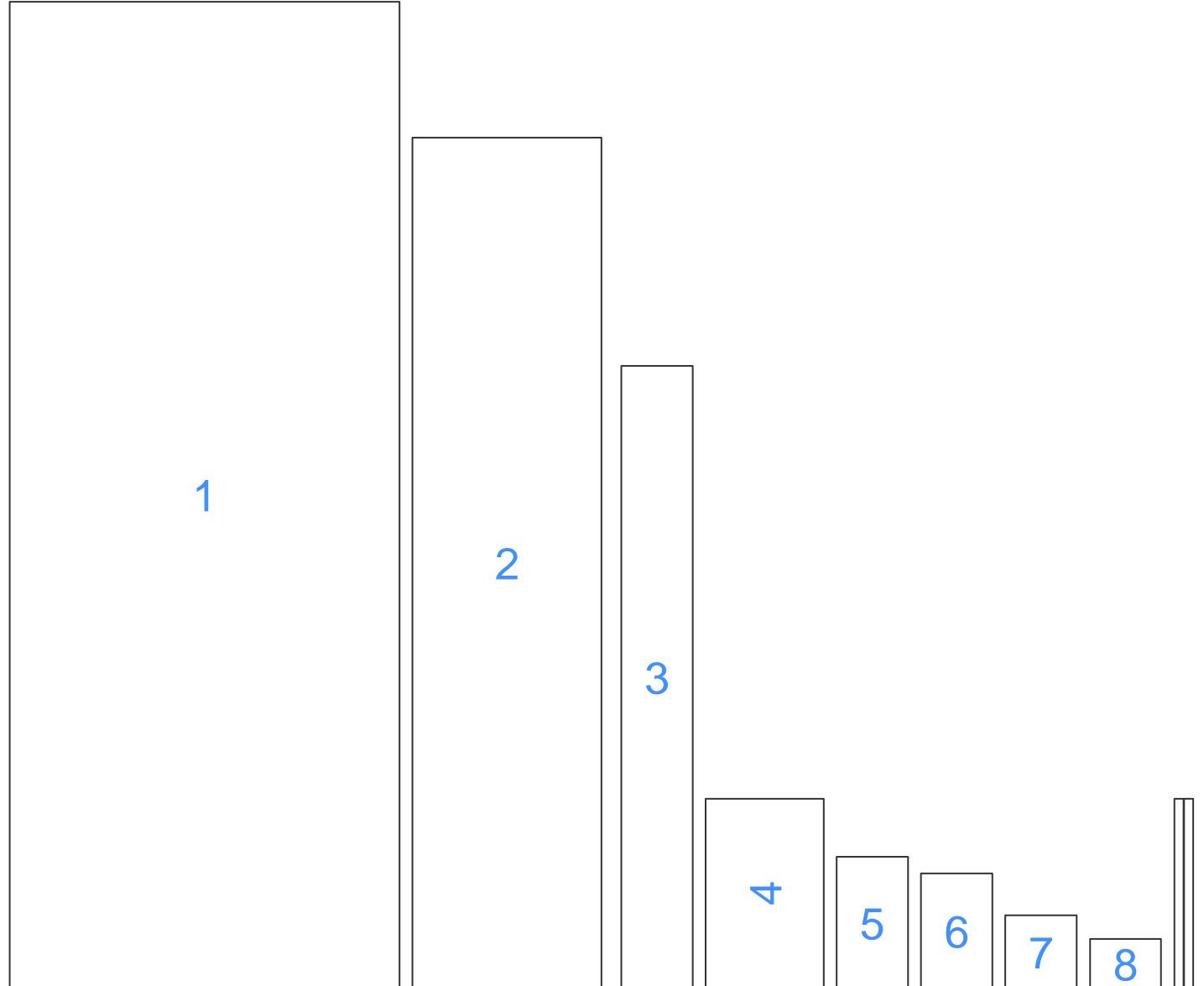
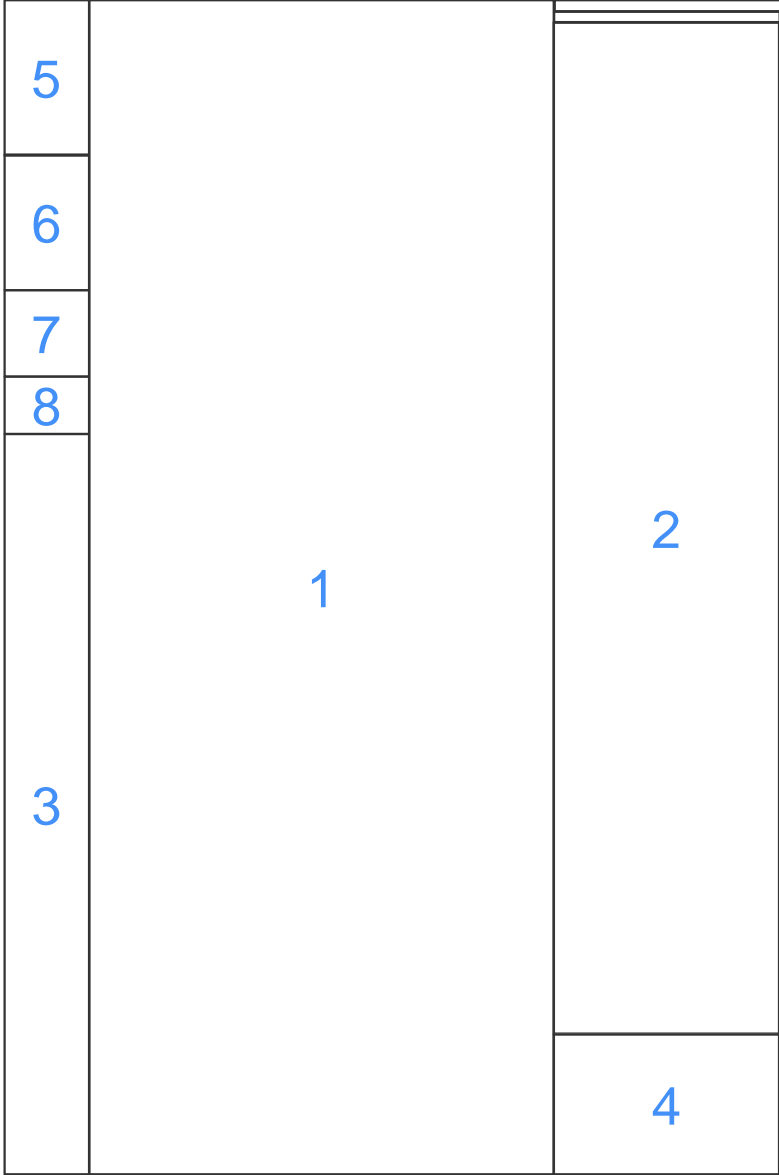
Defect Prediction

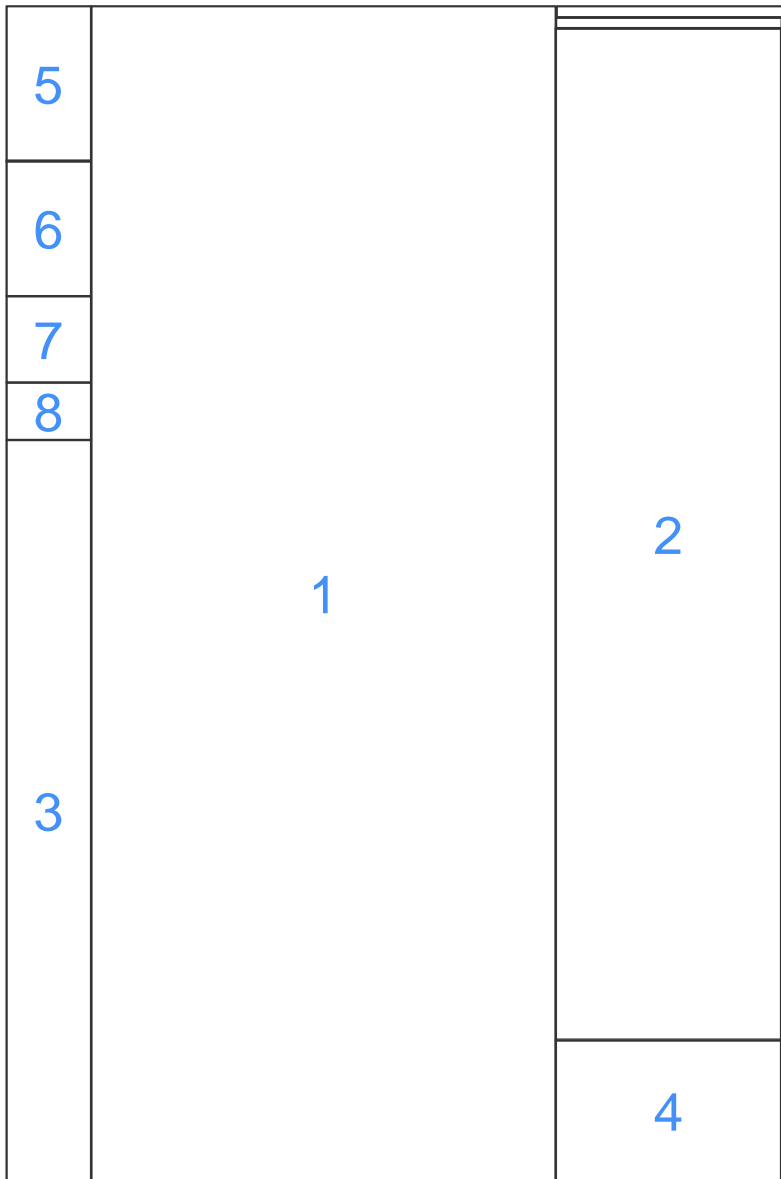




179k LoC

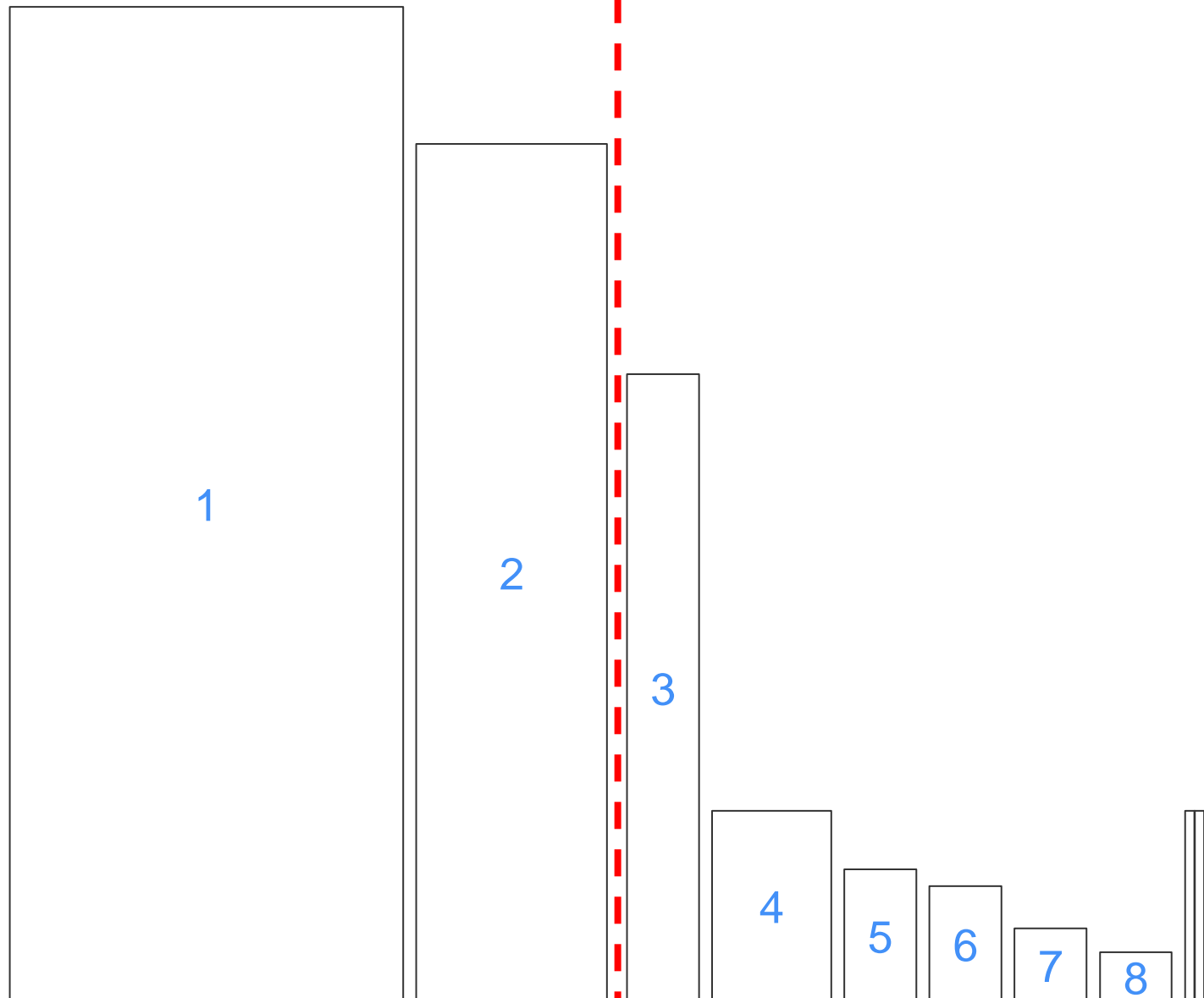


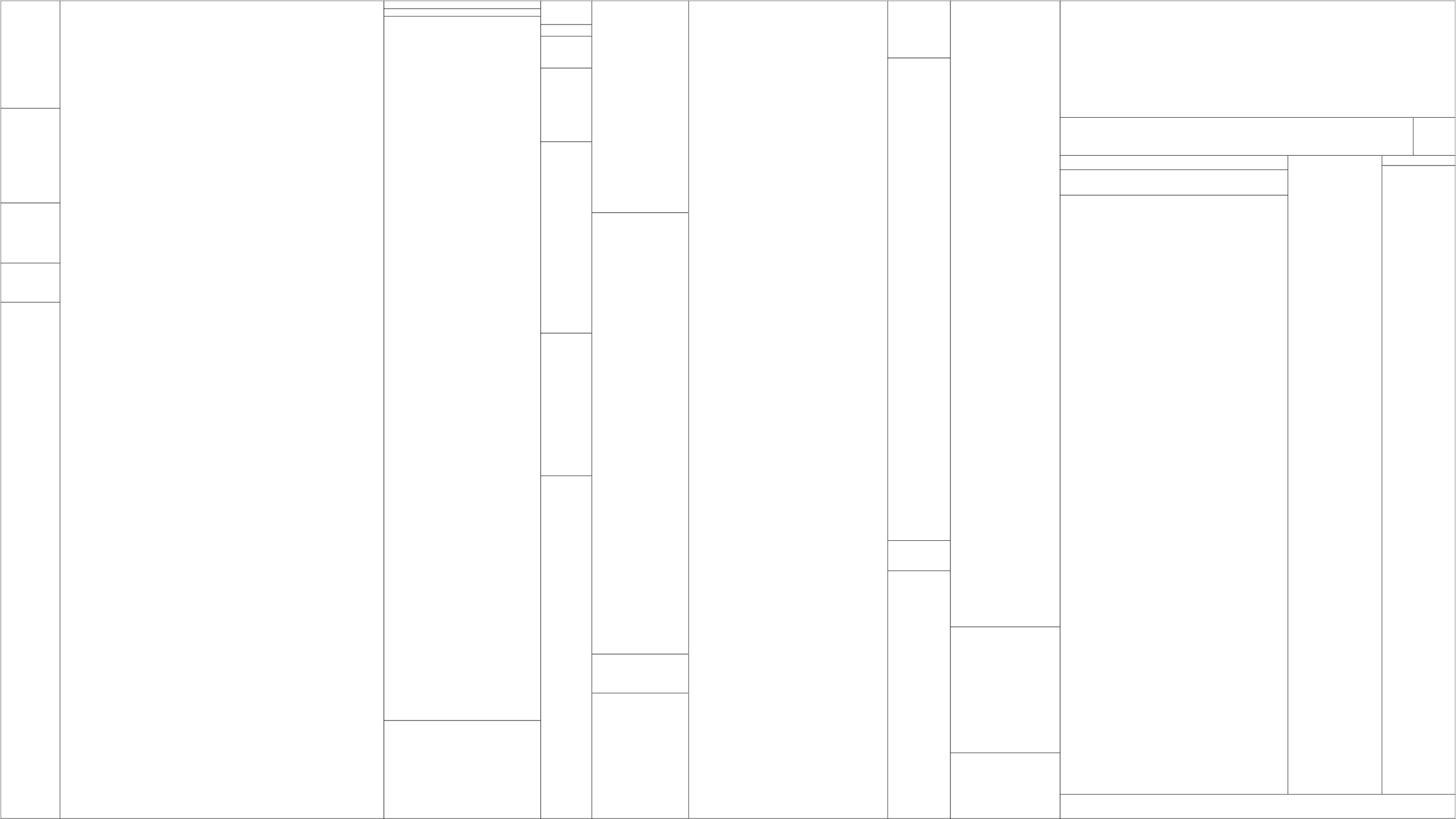


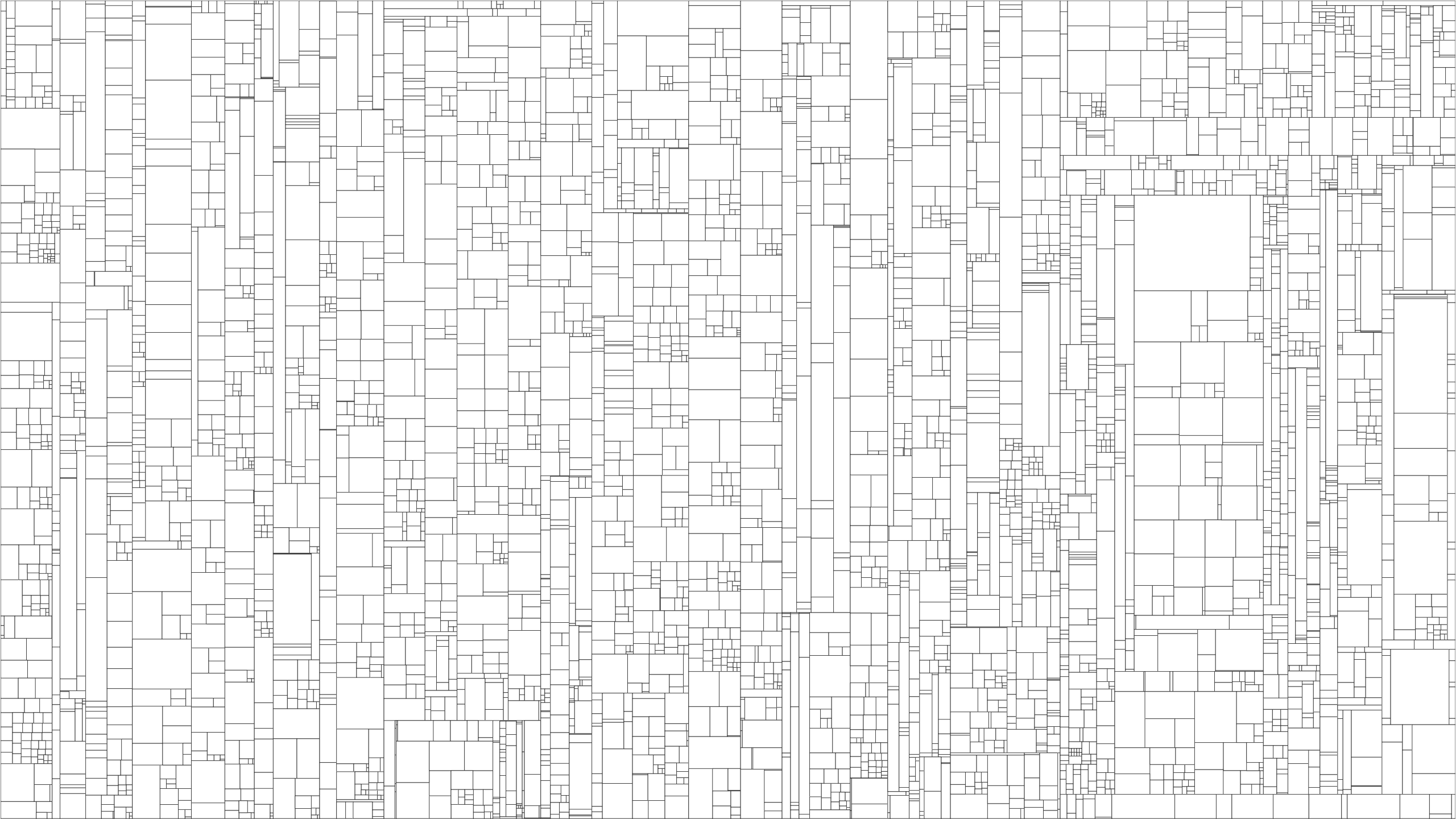


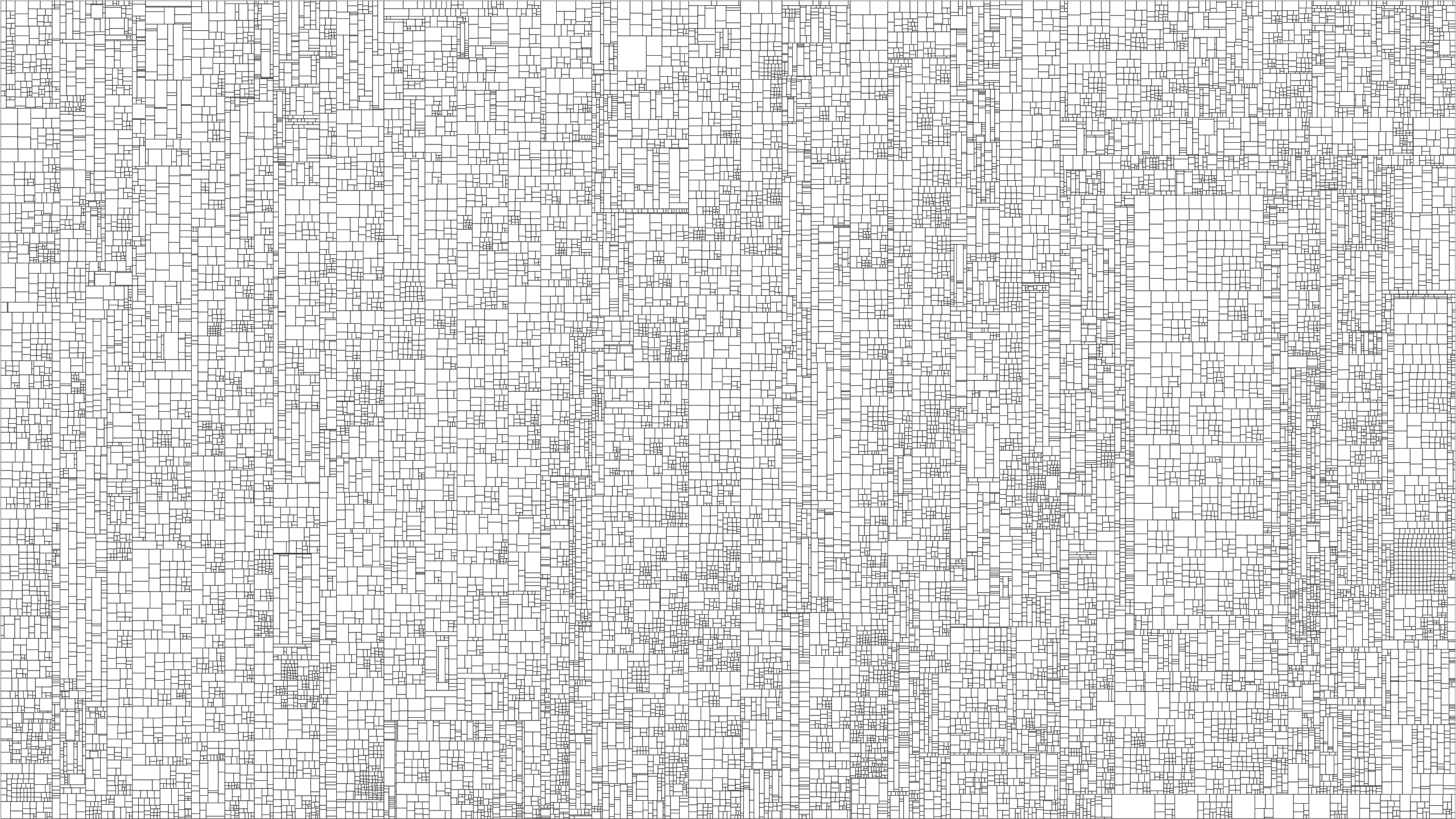
153k LoC
85%

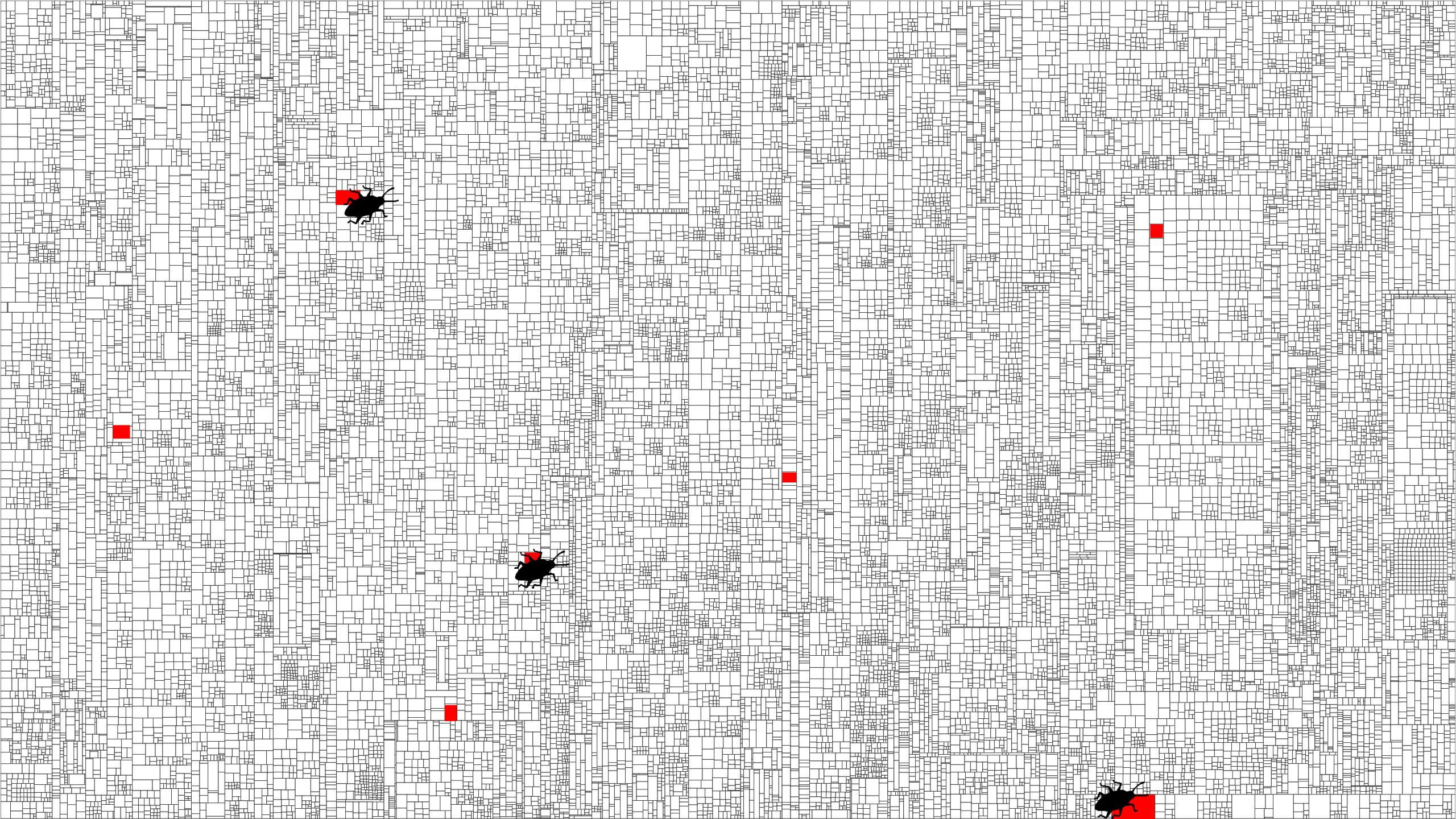
26,5k LoC
15%

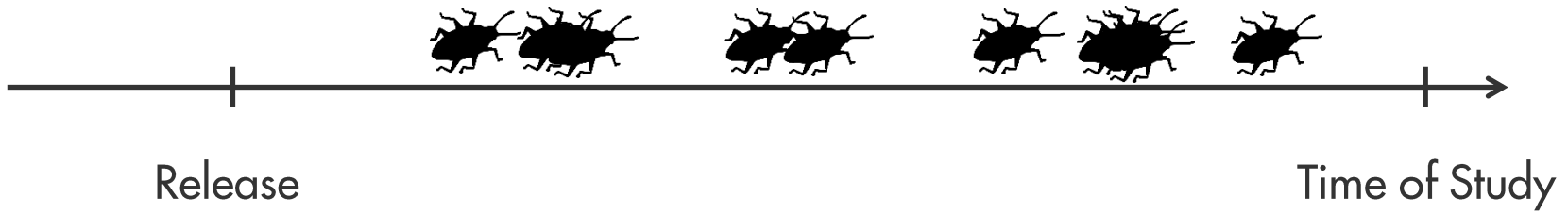
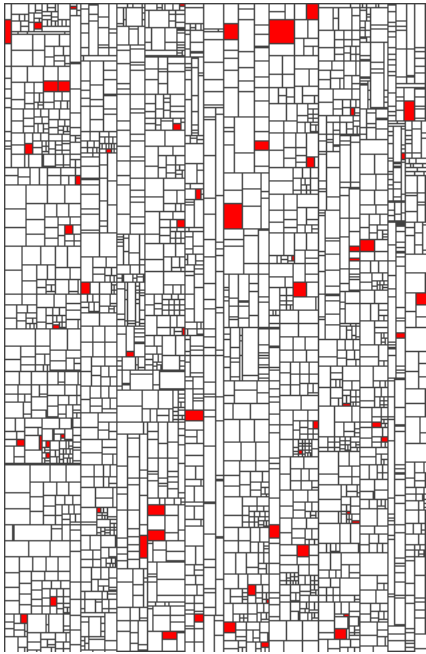


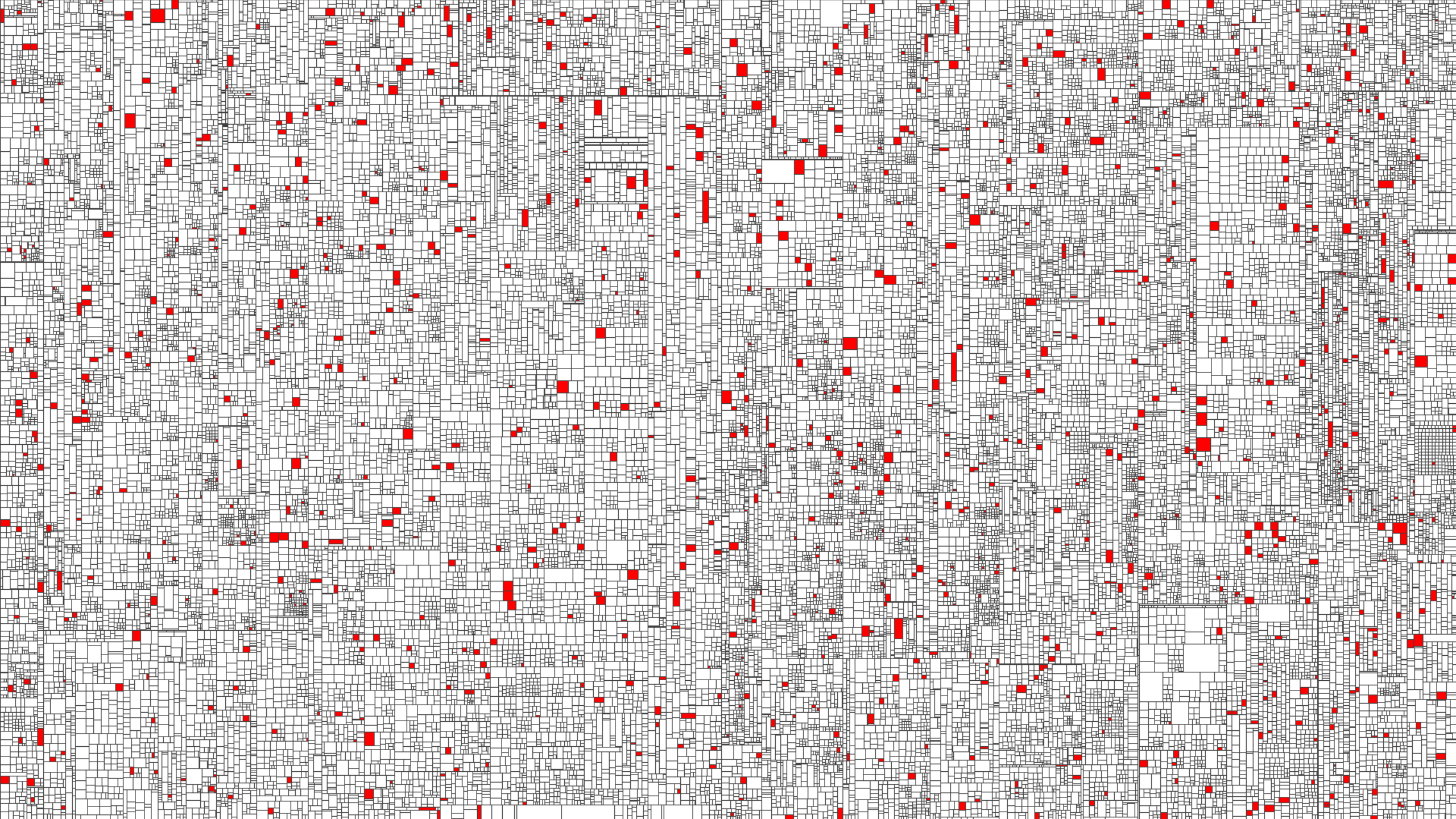














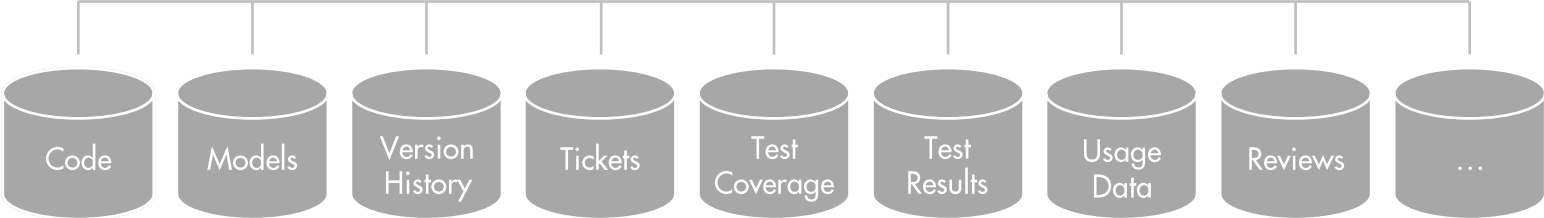
Evaluation

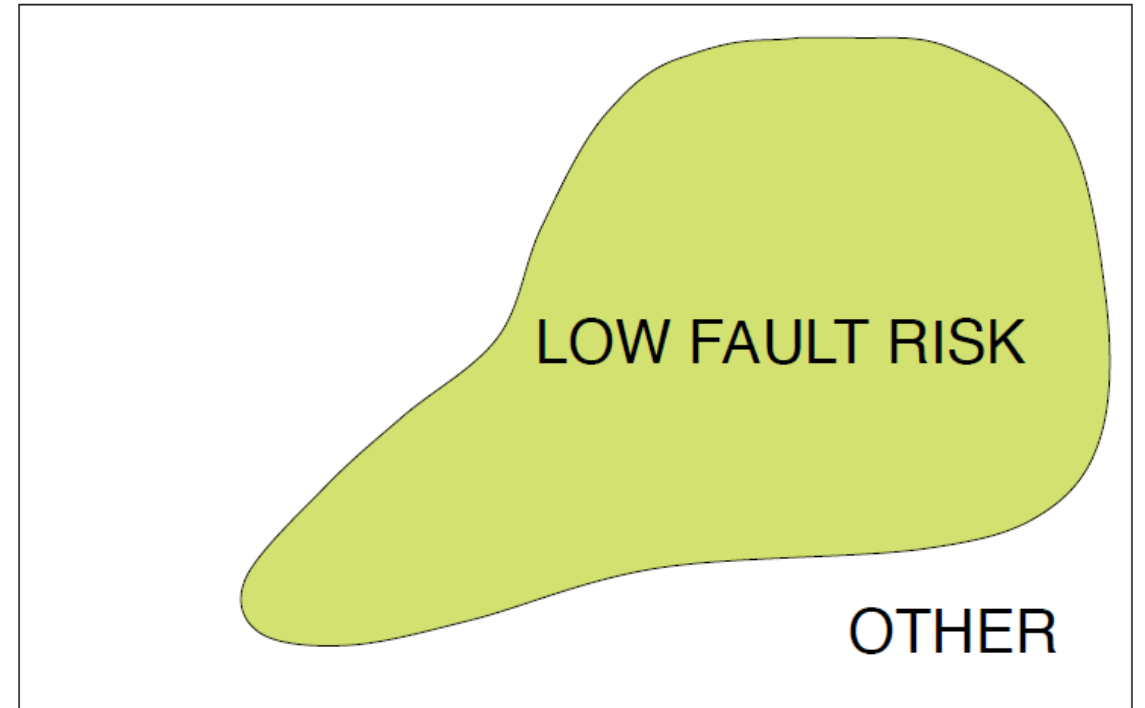
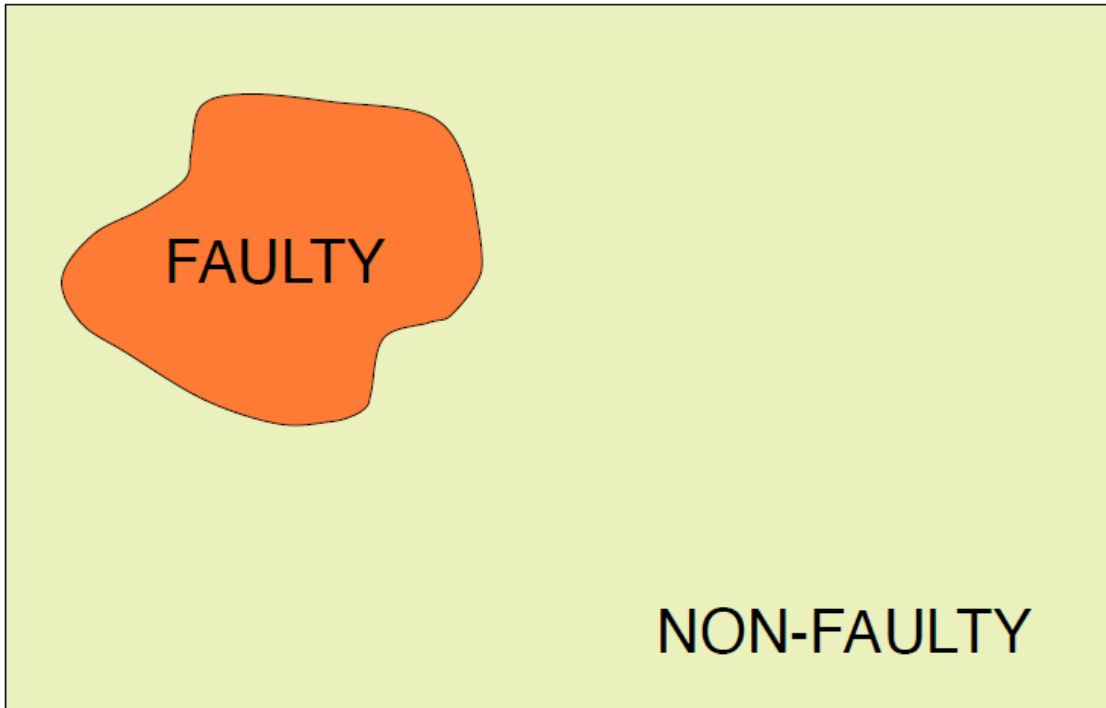
Release	# „defect prone“ Methods	# Bugs (Top 50)
1.4:	1127	
2.0:	1176	-

Pascarella, Palomba, Bacchelli, *Re-evaluating Method-Level Bug Prediction*, 2018:
Prediction nicht besser als zufällige Klassifikation.

» Wo sind keine Fehler im Code? «

» Wo sind Fehler im Code? «

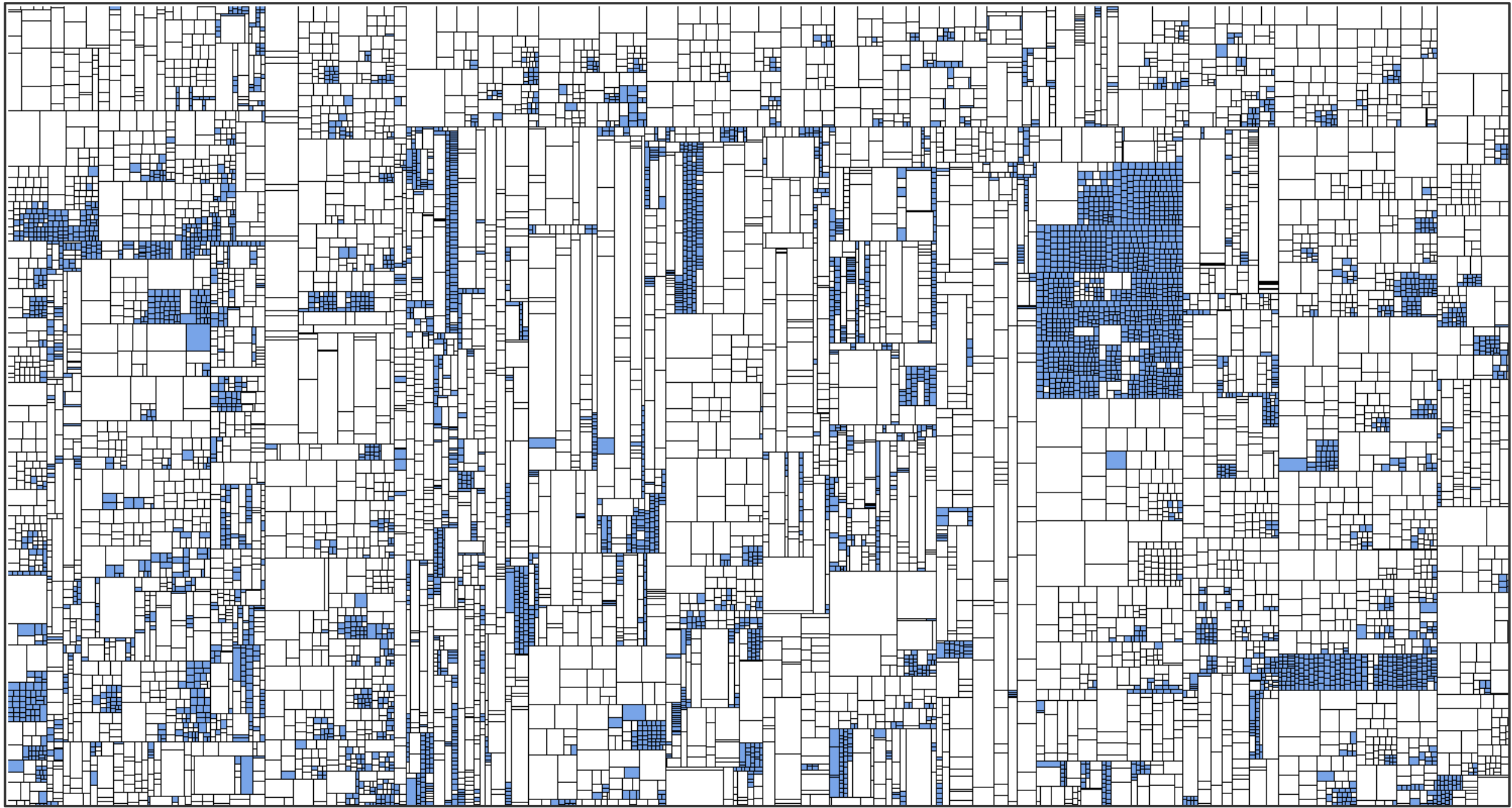




	Metric Name	Type
M1	Source Lines of Code (SLOC)	length
M2	Cyclomatic Complexity (CC)	complexity
M3	Max. Nesting Depth	max. value
M4	Max. Method Chaining	max. value
M5	Unique Variable Identifiers	unique count
M6	Anonymous Class Declarations	count
M7	Arithmetic In- or Decrementations	count
M8	Arithmetic Infix Operations	count
M9	Array Accesses	count
M10	Array Creations	count
M11	Assignments	count
M12	Boolean Operators	count
M13	Cast Expressions	count
M14	Catch Clauses	count
M15	Comparison Operators	count
M16	If Conditions	count
M17	Inner Method Declarations	count
M18	Instance-of Checks	count
M19	Instantiations	count
M20	Loops	count
M21	Method Invocations	count
M22	Null Checks	count
M23	Null Literals	count
M24	Return Statements	count
M25	String Literals	count

M26	Super-Method Invocations	count
M27	Switch-Case Blocks	count
M28	Synchronized Blocks	count
M29	Ternary Operations	count
M30	Throw Statements	count
M31	Try Blocks	count
M32	Any Conditions	count
M33	Any Arithmetic Operations	count
M34	Is Constructor	category
M35	Is Setter	category
M36	Is Getter	category
M37	Is Empty Method	category
M38	Is Delegation Method	category
M39	Is ToString Method	category

#	Rule	Support	Confidence ↓
1	{ <i>NoMethodInvocations</i> , <i>SlocLessThan4</i> , <i>NoArithmeticOperations</i> , <i>NoNullLiterals</i> } ⇒ { <i>NotFaulty</i> }	10.43%	96.76%
2	{ <i>NoMethodInvocations</i> , <i>SlocLessThan4</i> , <i>NoArithmeticOperations</i> } ⇒ { <i>NotFaulty</i> }	11.03%	96.09%
3	{ <i>NoMethodInvocations</i> , <i>SlocLessThan4</i> , <i>NoCastExpressions</i> , <i>NoNullLiterals</i> } ⇒ { <i>NotFaulty</i> }	10.43%	95.43%
4	{ <i>NoMethodInvocations</i> , <i>SlocLessThan4</i> , <i>NoCastExpressions</i> , <i>NoInstantiations</i> } ⇒ { <i>NotFaulty</i> }	10.13%	95.31%
5	{ <i>NoMethodInvocations</i> , <i>SlocLessThan4</i> , <i>NoCastExpressions</i> } ⇒ { <i>NotFaulty</i> }	11.03%	94.85%



LOW FAULT RISK

Zentrale Ergebnisse

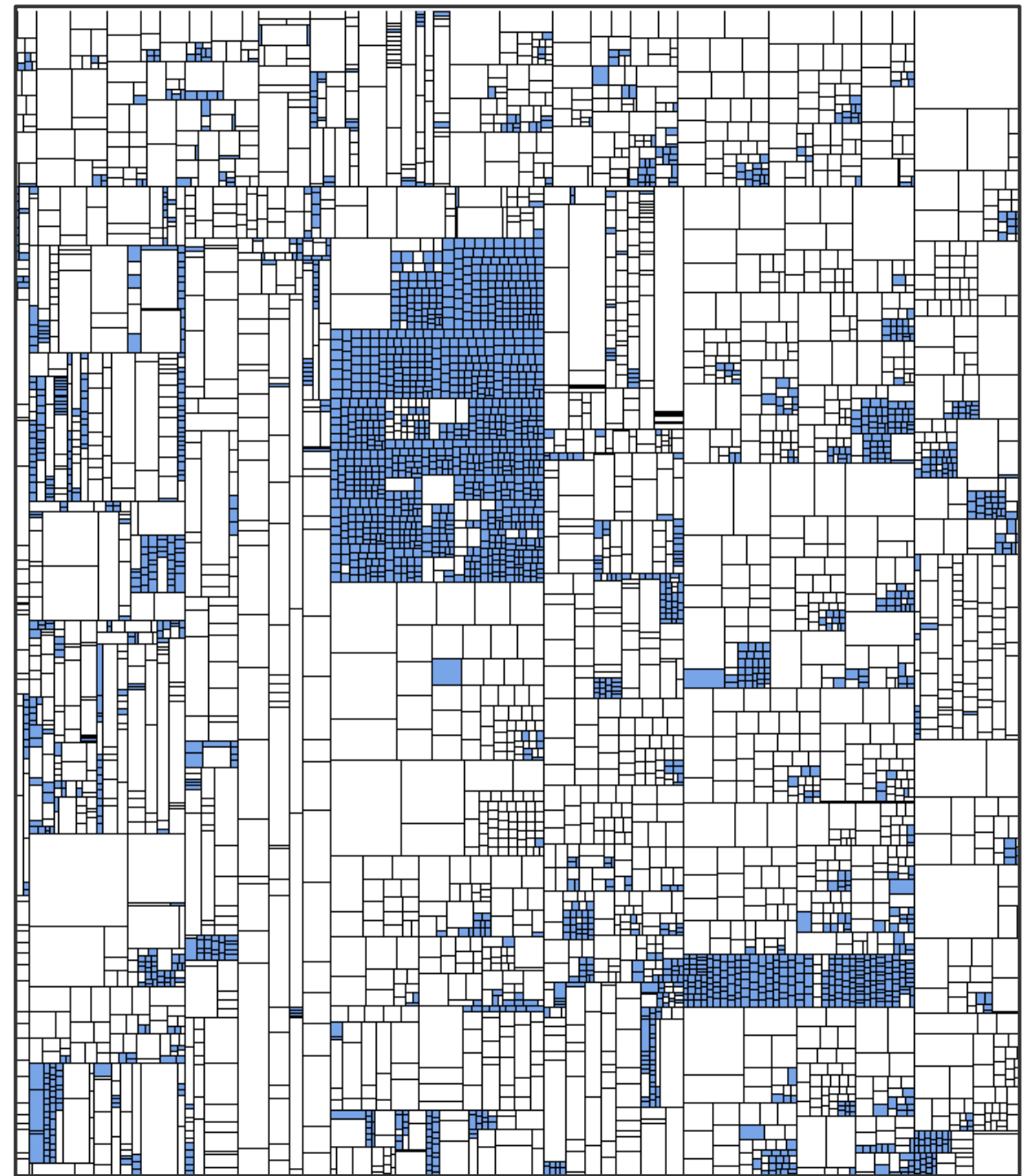
Low Fault Risk

- 16%-80% aller Methoden (Median: 31%)
- 5%-78% aller SLOC (Median: 15%)

Enthalten nur 6% der Fehler

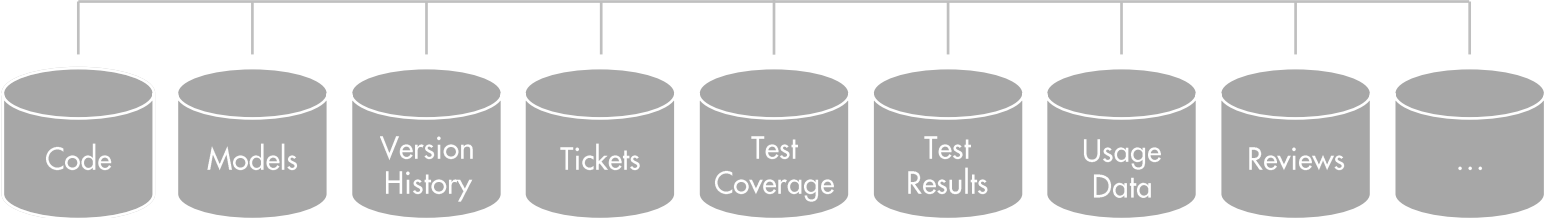
Reduktion der Fehlerdichte

- 4,3 bis 10,9 fach pro Methode
- 1,5 bis 4,4-fach pro SLOC



» Wo sind keine Fehler im Code? «

» Wo sind Fehler im Code? «



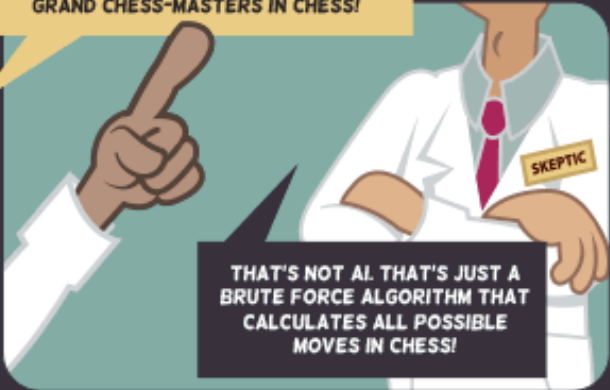
Artificial Intelligence

"Development of computer systems able to perform tasks that normally require human intelligence"

The Curse of AI

"As soon as a problem in AI is solved, it is no longer considered AI because we know how it works. AI invents itself out of existence."

BEHOLD! A COMPUTER THAT CAN BEAT GRAND CHESS-MASTERS IN CHESS!



THAT'S NOT AI. THAT'S JUST A BRUTE FORCE ALGORITHM THAT CALCULATES ALL POSSIBLE MOVES IN CHESS!

A PROGRAM THAT CAN CONVERSE AND BE YOUR THERAPIST!



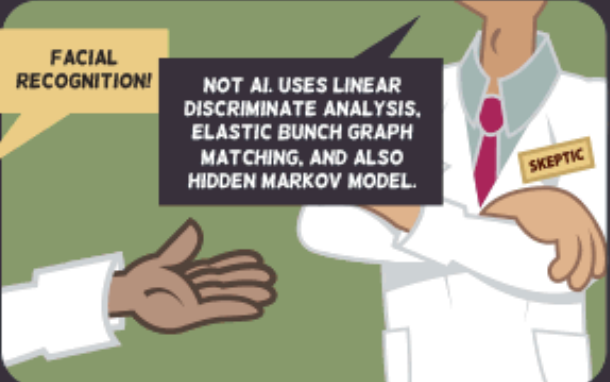
THAT'S JUST A DUMB CHATBOT THAT JUST SCANS FOR KEYWORDS AND PARROTS BACK WHAT YOU SAY.

SPEECH RECOGNITION!



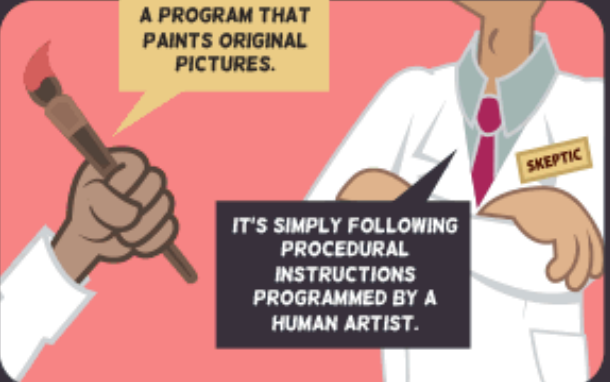
THAT'S JUST STATISTICAL ACOUSTIC MODELING, USING HIDDEN MARKOV MODELS.

FACIAL RECOGNITION!



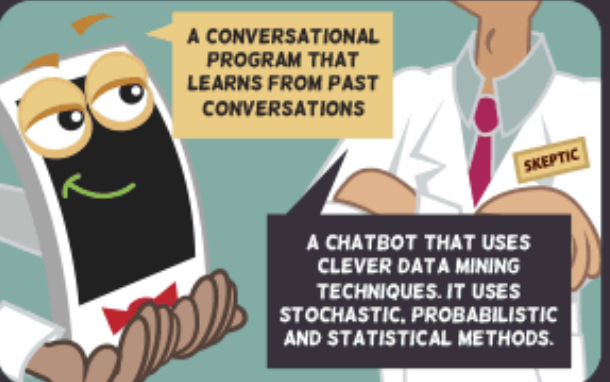
NOT AI. USES LINEAR DISCRIMINATE ANALYSIS, ELASTIC BUNCH GRAPH MATCHING, AND ALSO HIDDEN MARKOV MODEL.

A PROGRAM THAT PAINTS ORIGINAL PICTURES.



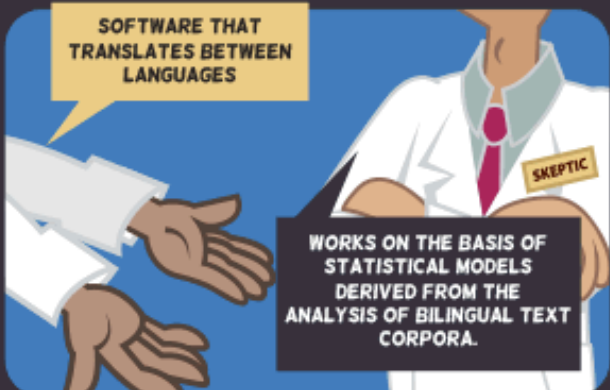
IT'S SIMPLY FOLLOWING PROCEDURAL INSTRUCTIONS PROGRAMMED BY A HUMAN ARTIST.

A CONVERSATIONAL PROGRAM THAT LEARNS FROM PAST CONVERSATIONS



A CHATBOT THAT USES CLEVER DATA MINING TECHNIQUES. IT USES STOCHASTIC, PROBABILISTIC AND STATISTICAL METHODS.

SOFTWARE THAT TRANSLATES BETWEEN LANGUAGES



WORKS ON THE BASIS OF STATISTICAL MODELS DERIVED FROM THE ANALYSIS OF BILINGUAL TEXT CORPORA.

...SELF DRIVING CARS...

...OPTICAL CHARACTER RECOGNITION...

...COMPUTER PLAYERS IN GAMES...

...HANDWRITING RECOGNITION...

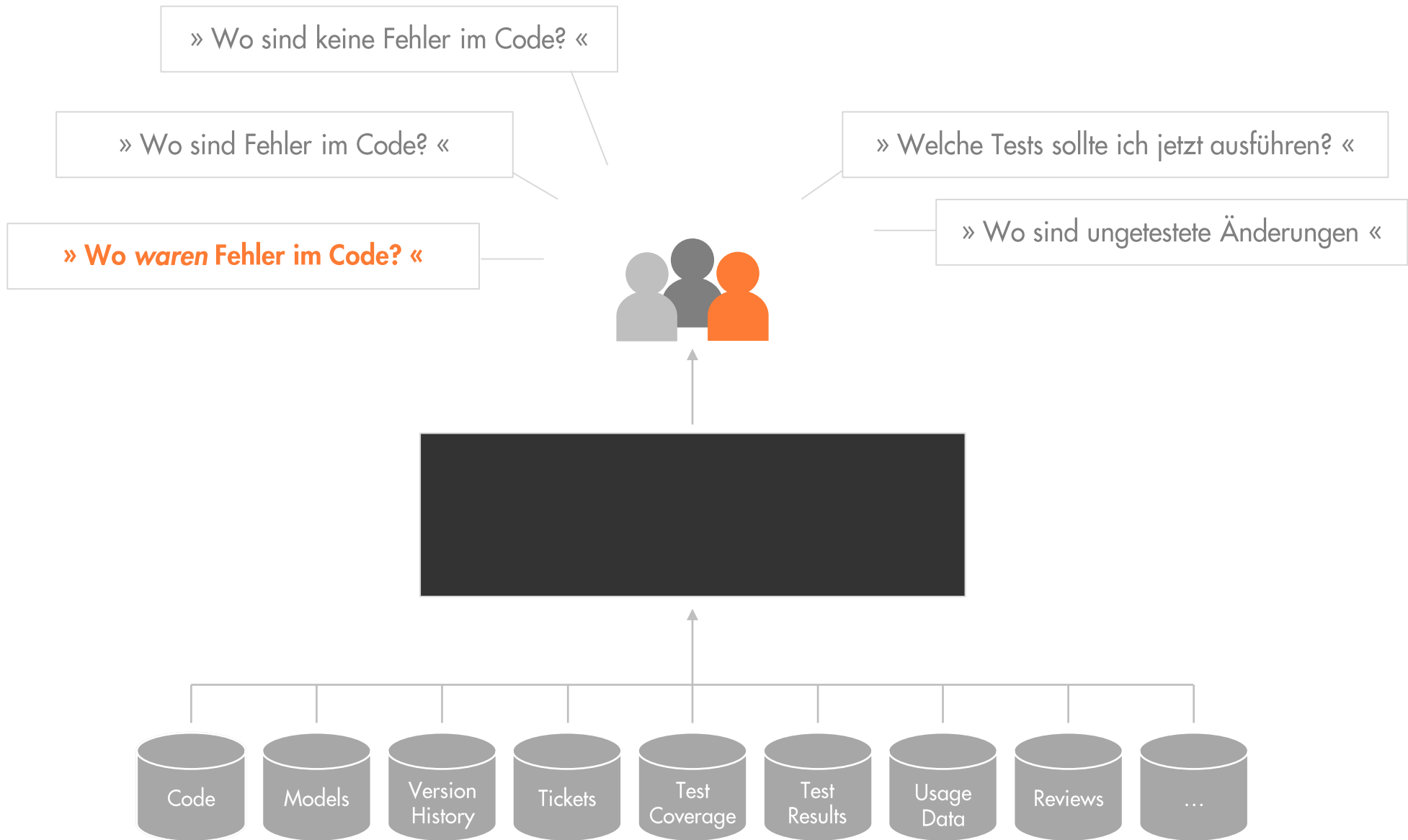
...EXPERT SYSTEMS...

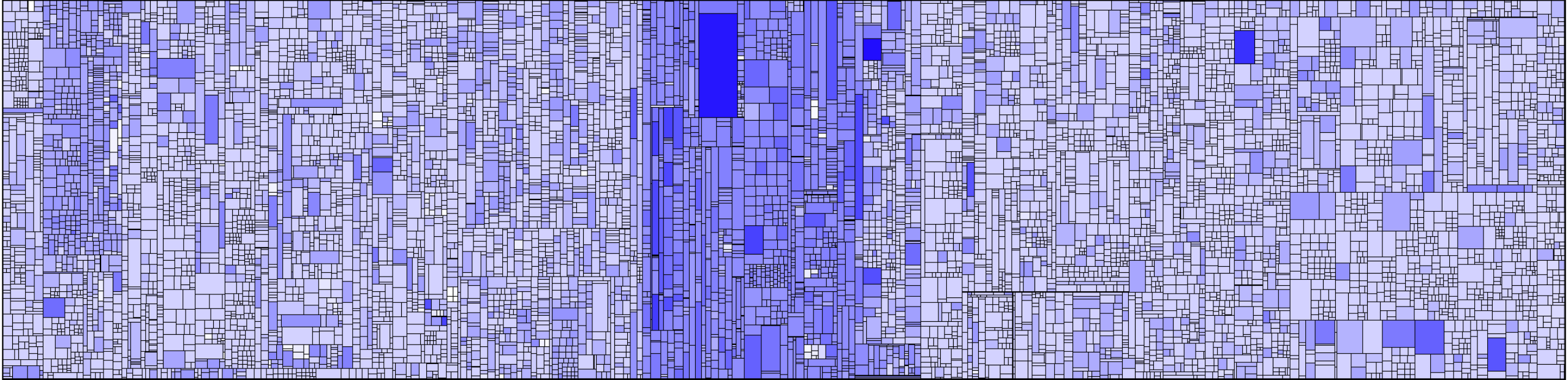
...NEURAL NETWORKS...

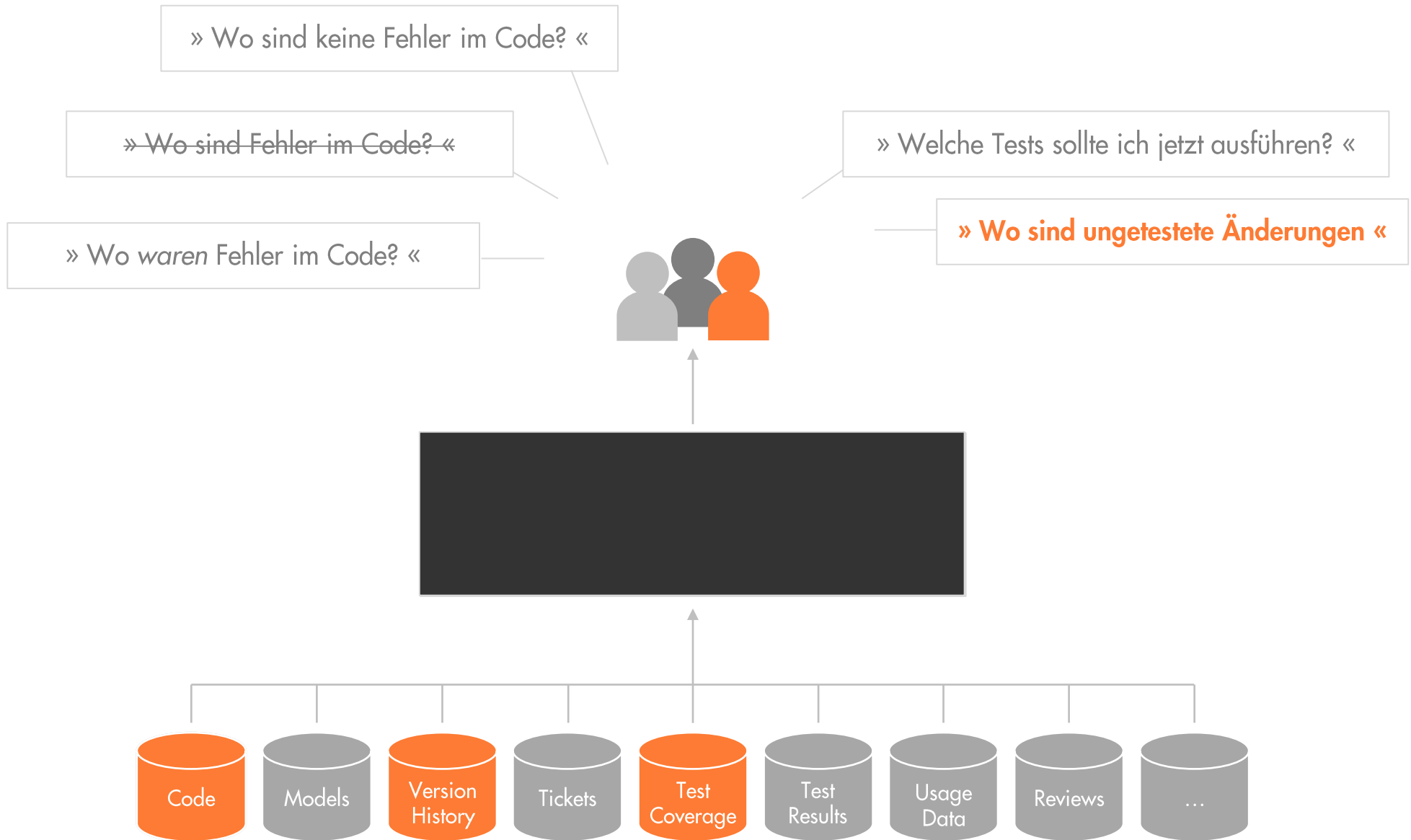
ALGORITHMS!



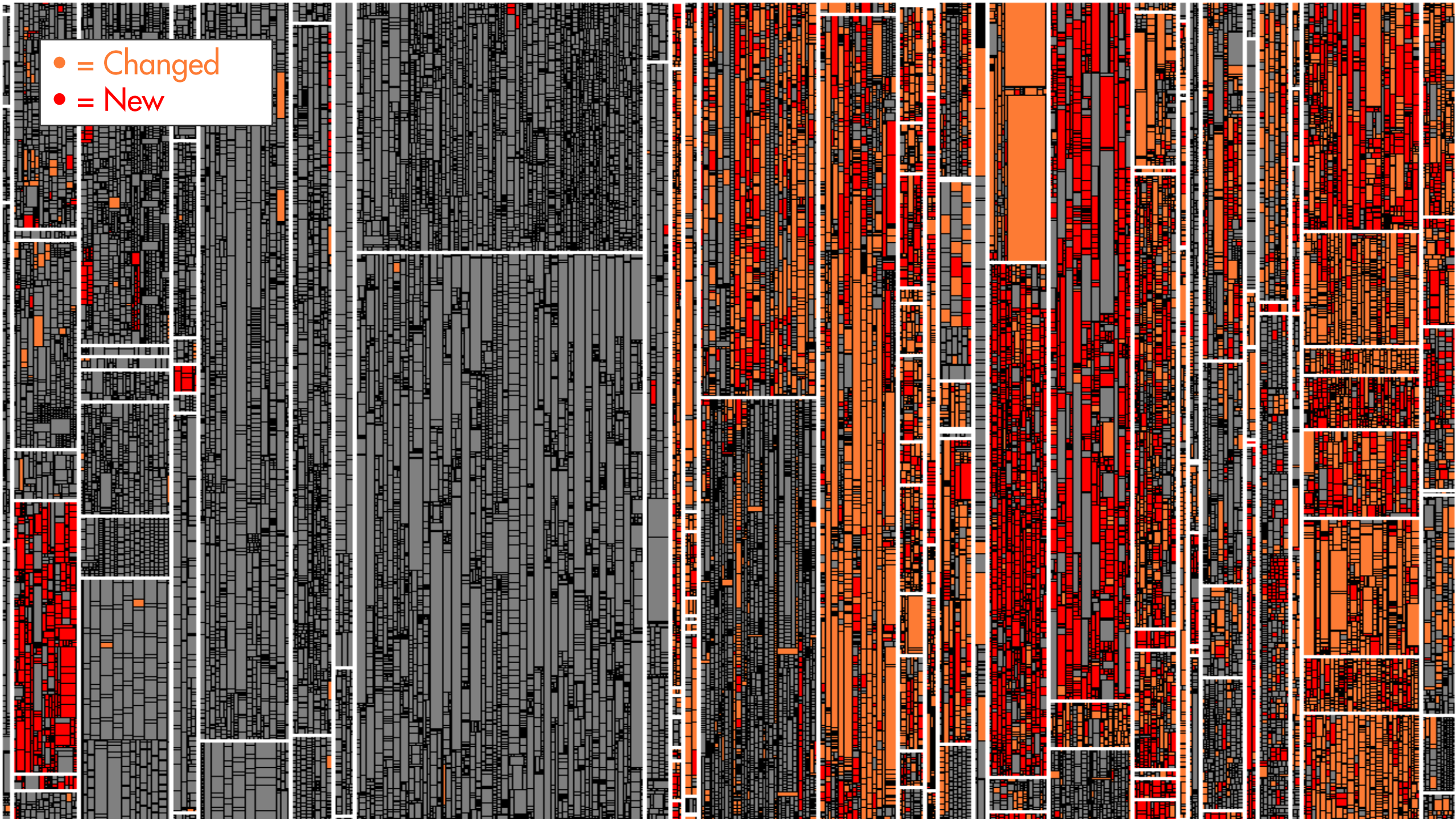
SCALDENER





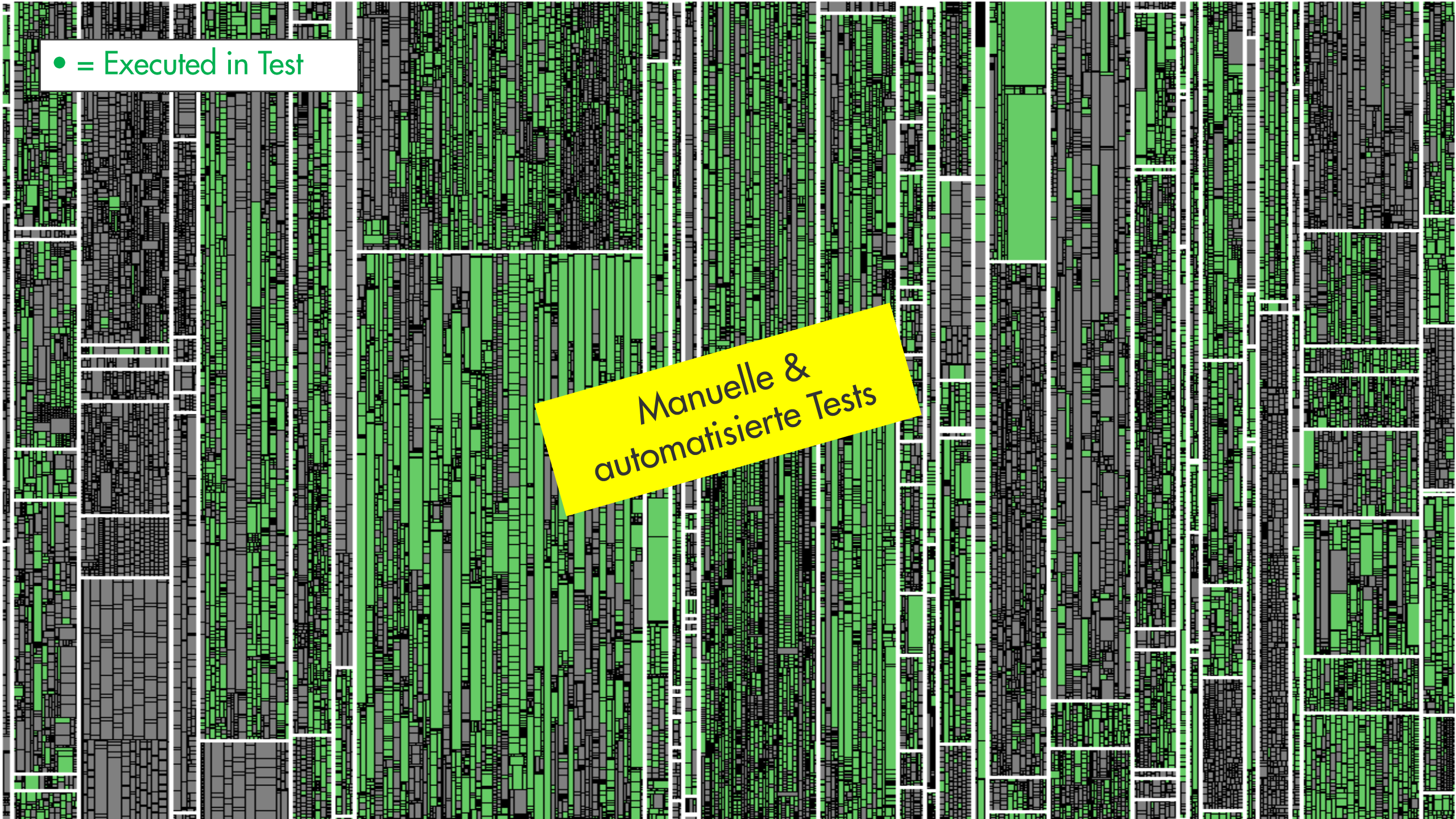


- = Changed
- = New

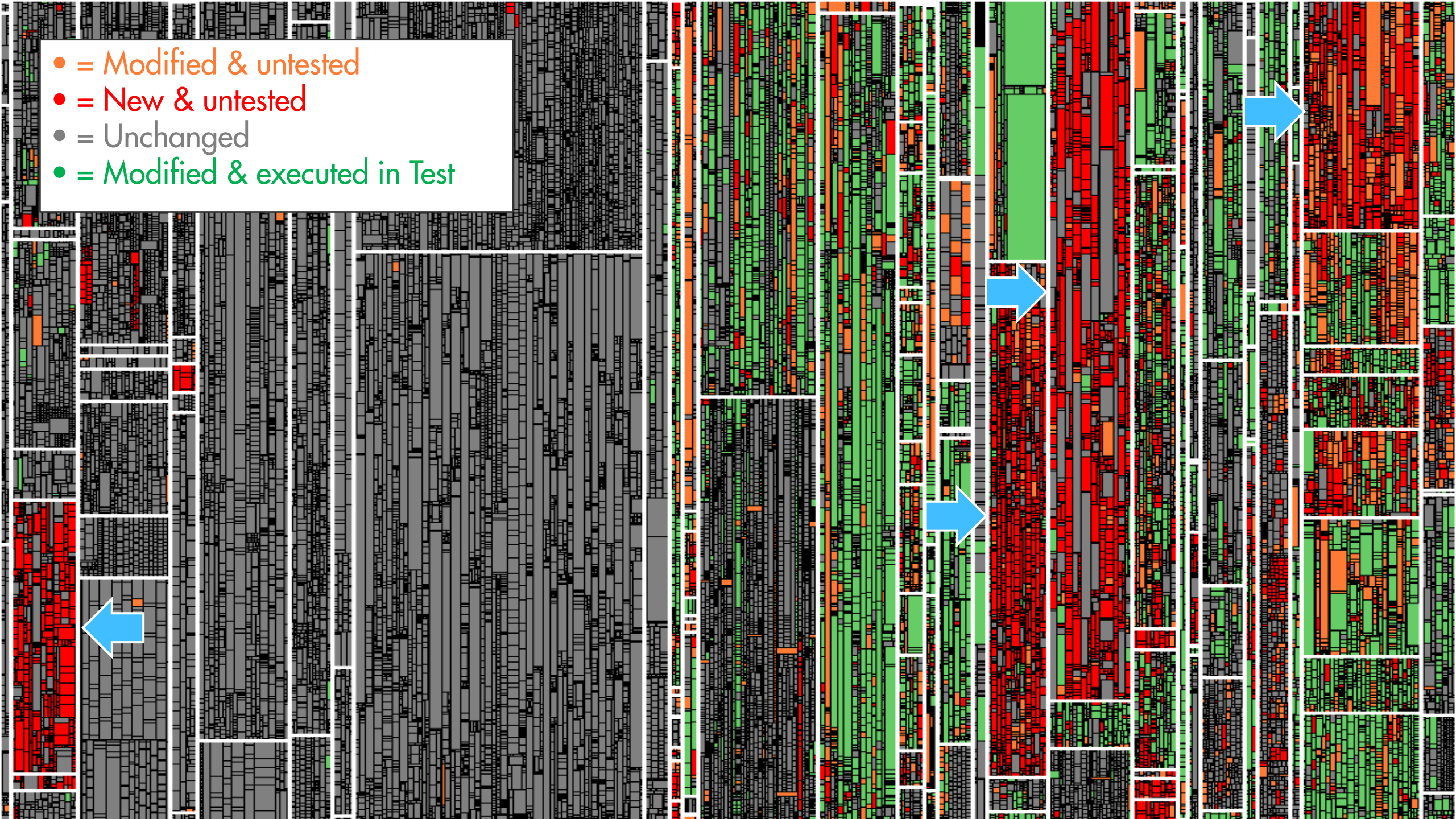



















● = Executed in Test





Manuelle &
automatisierte 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% 

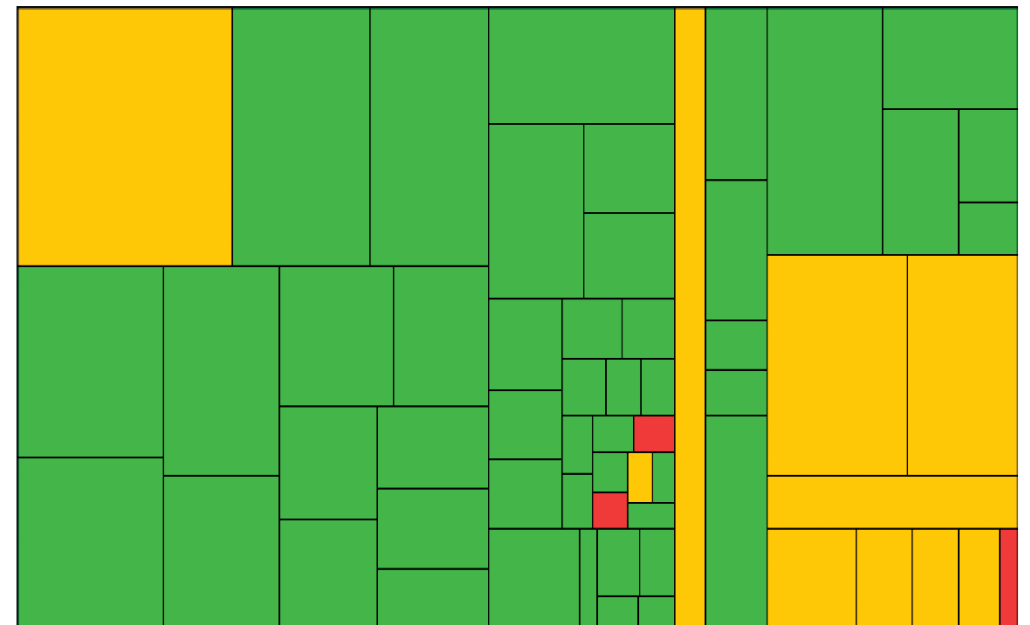
Done Issue TS-14733 - Remove Dataflow blocks

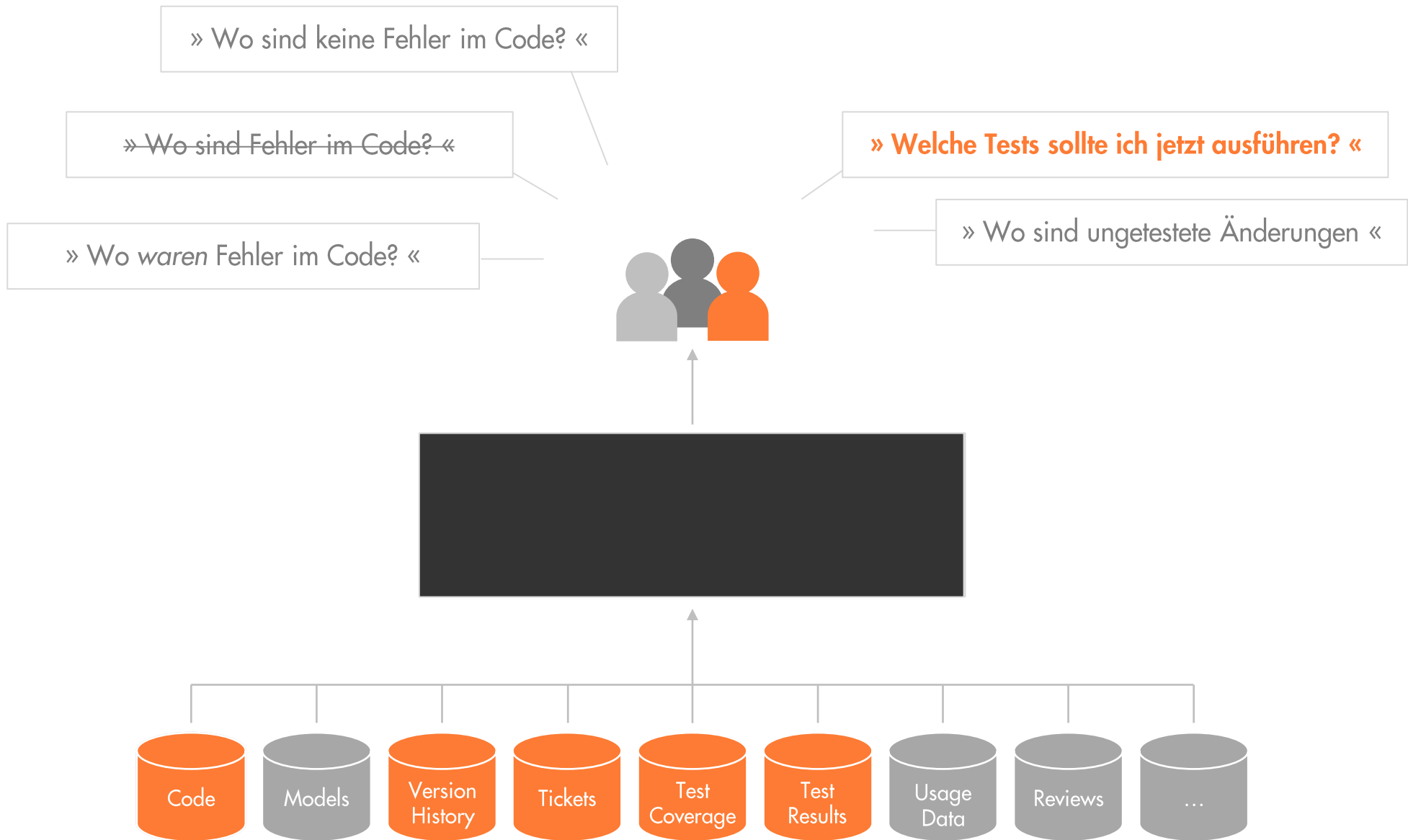
Creator:  (on Apr 06 2018 19:44) Last update: Aug 24 2018 09:32

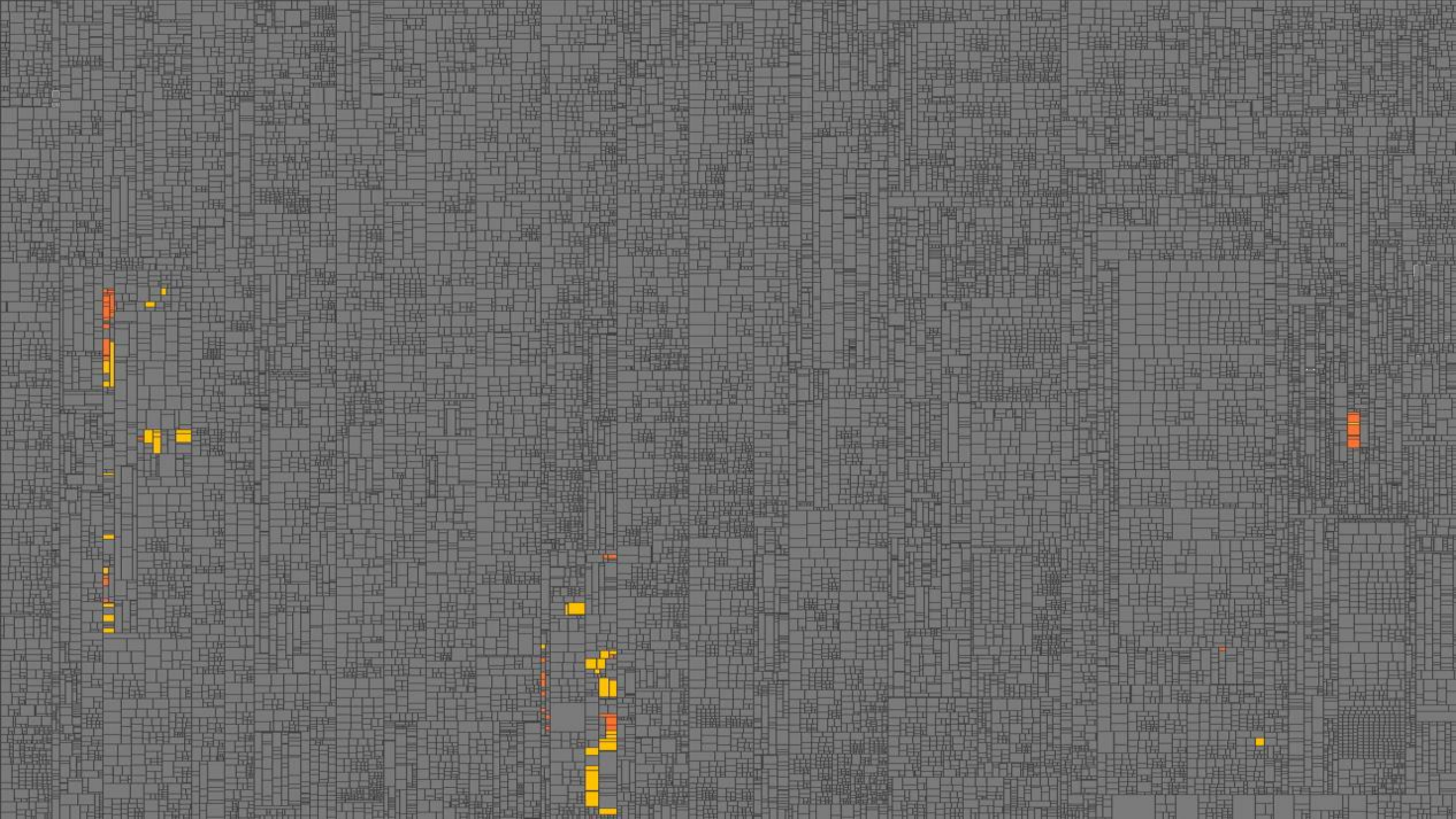
Assignee: 

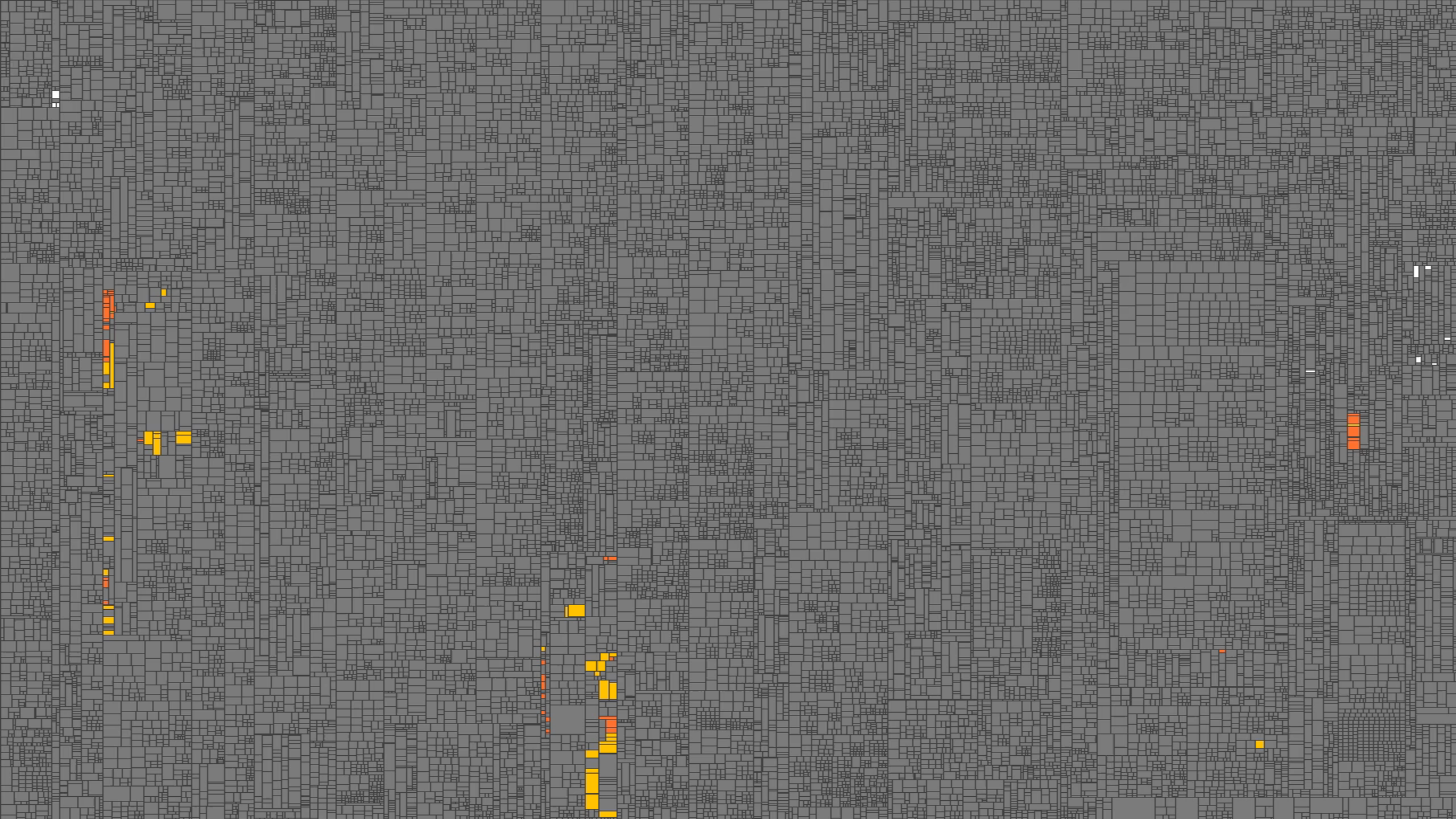
Project	Type	Priority	Resolution	Fix Version
TS	Maintenance	Normal	Green	Teamscale 4.5
Component	Labels	Affected Version	Customer	Customer Issue
Backend	Performance			
Epic Name	Freshdesk URL	Merge Request		
		https://git.cqse.eu/cqse/teamscale/3621		

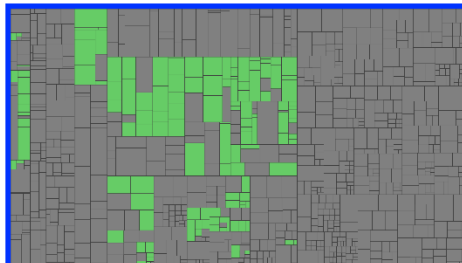
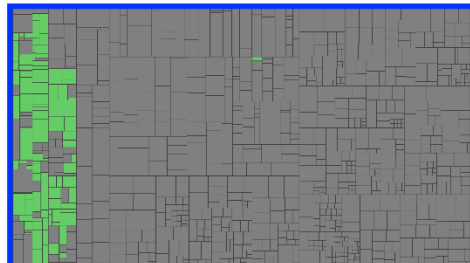
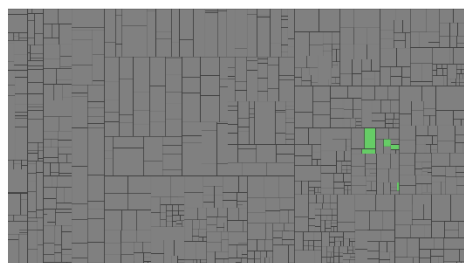
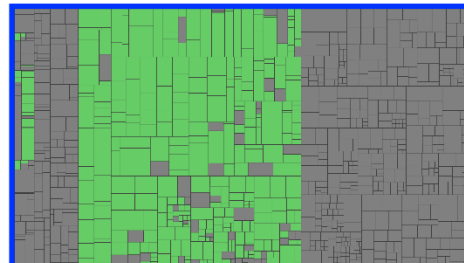
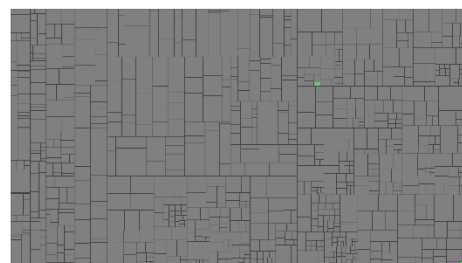
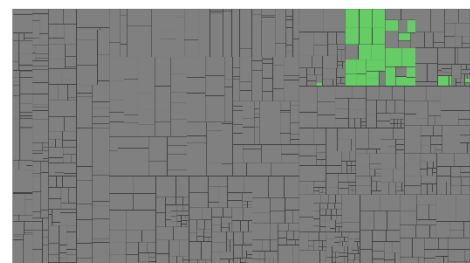
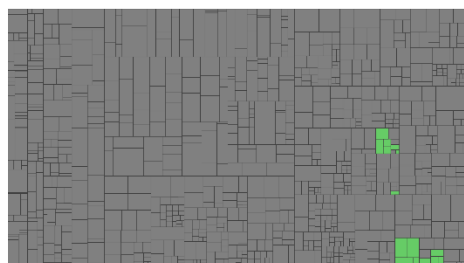
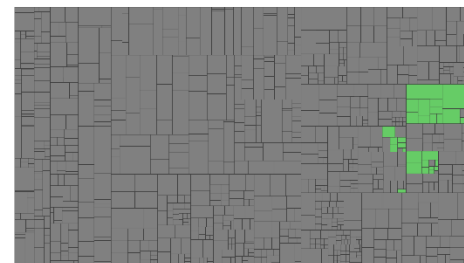
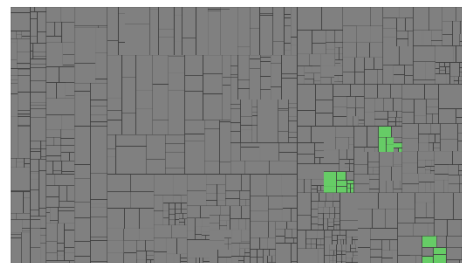
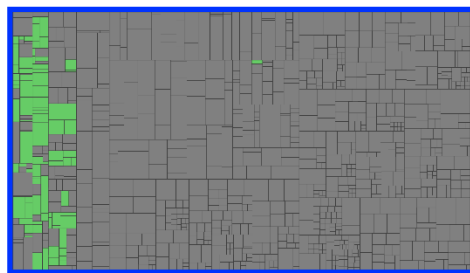
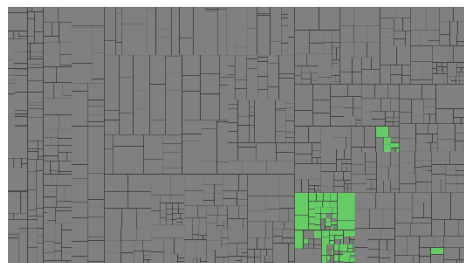
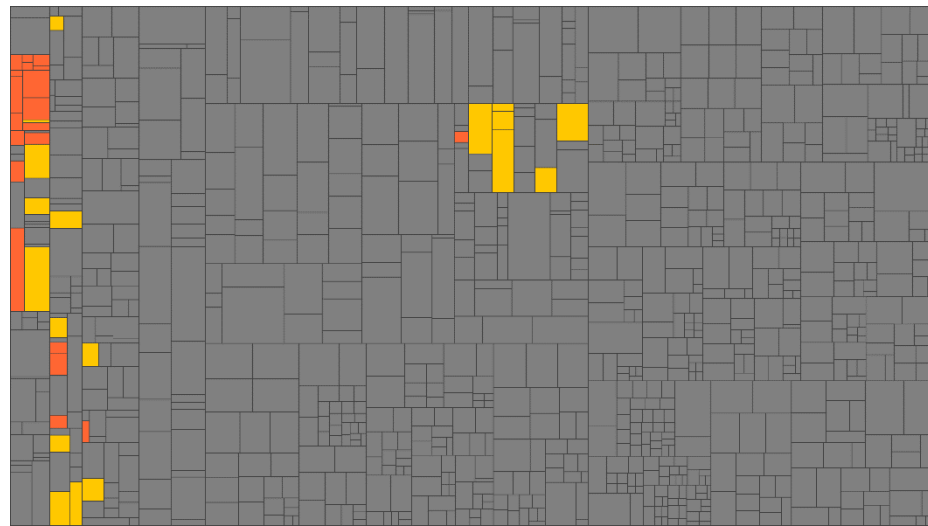
Aug 15 2018 12:37–Now | Test Gap: 22%



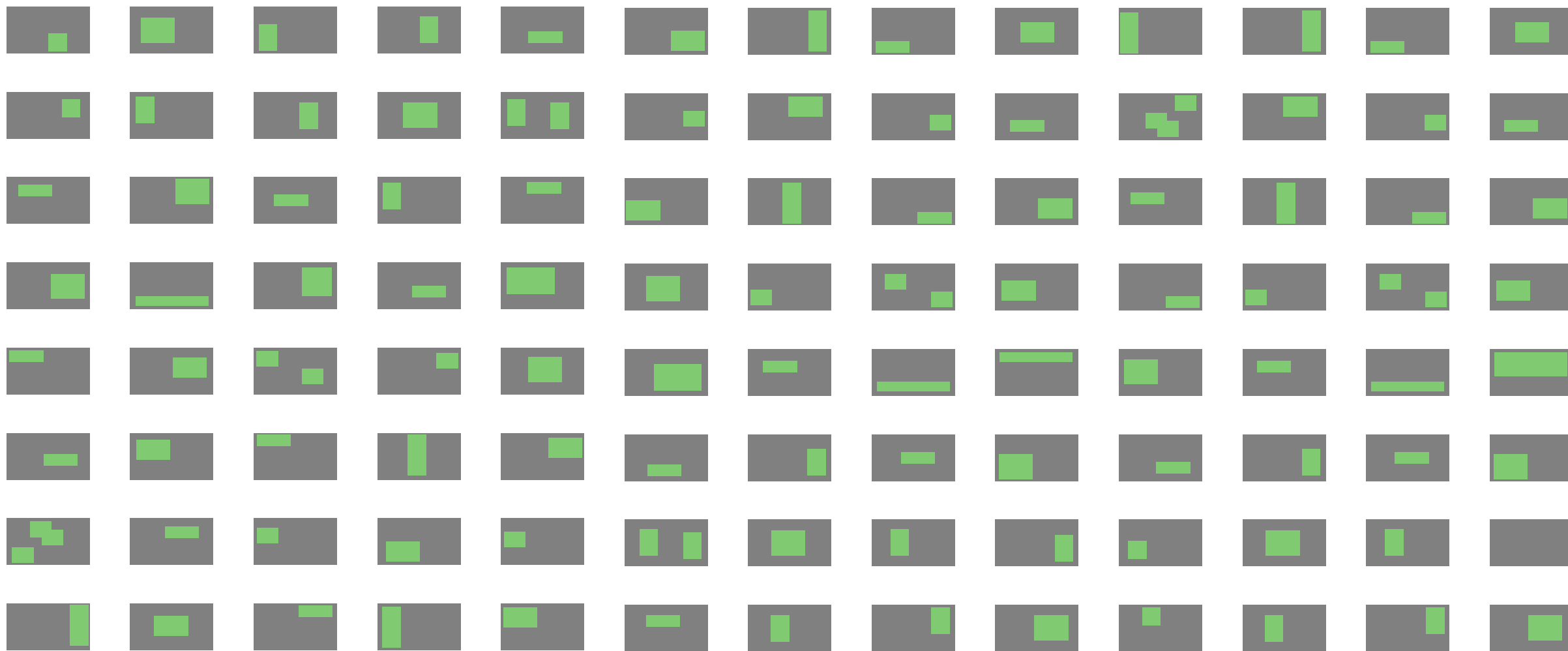


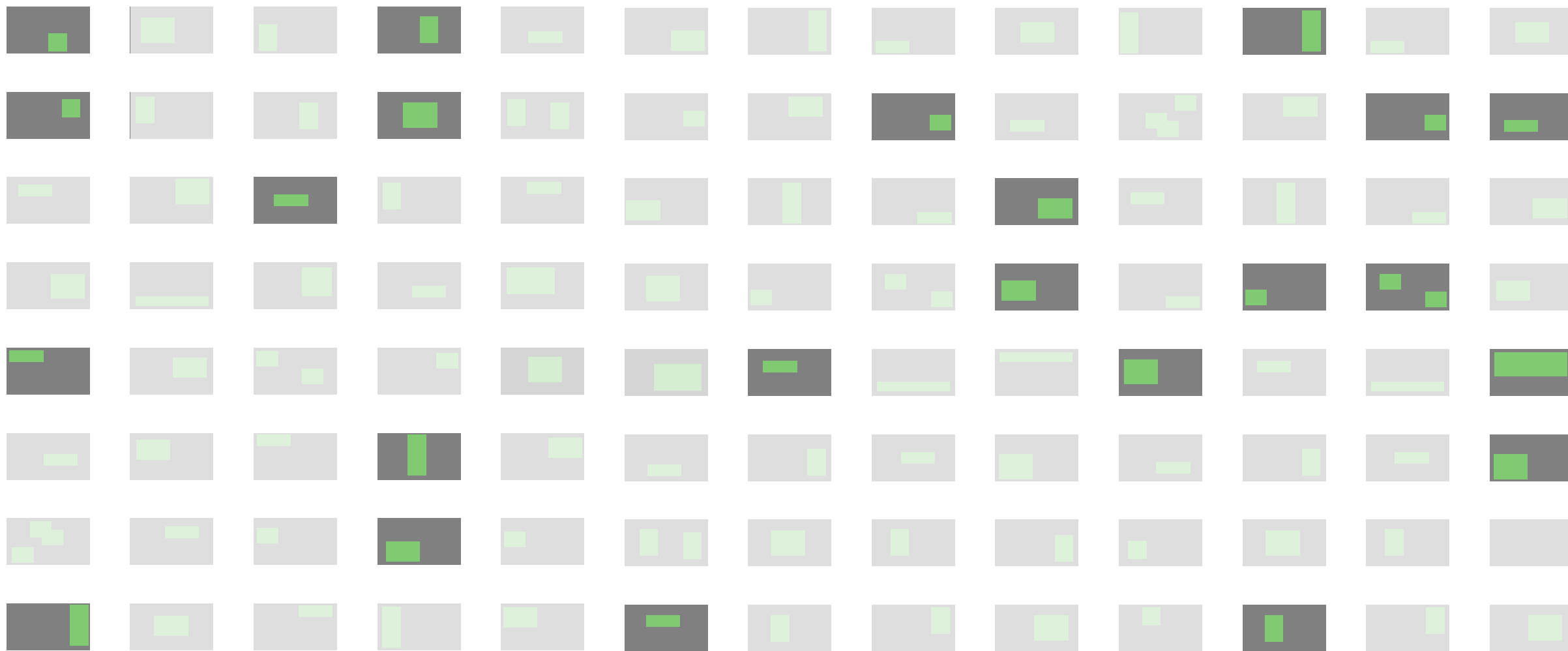


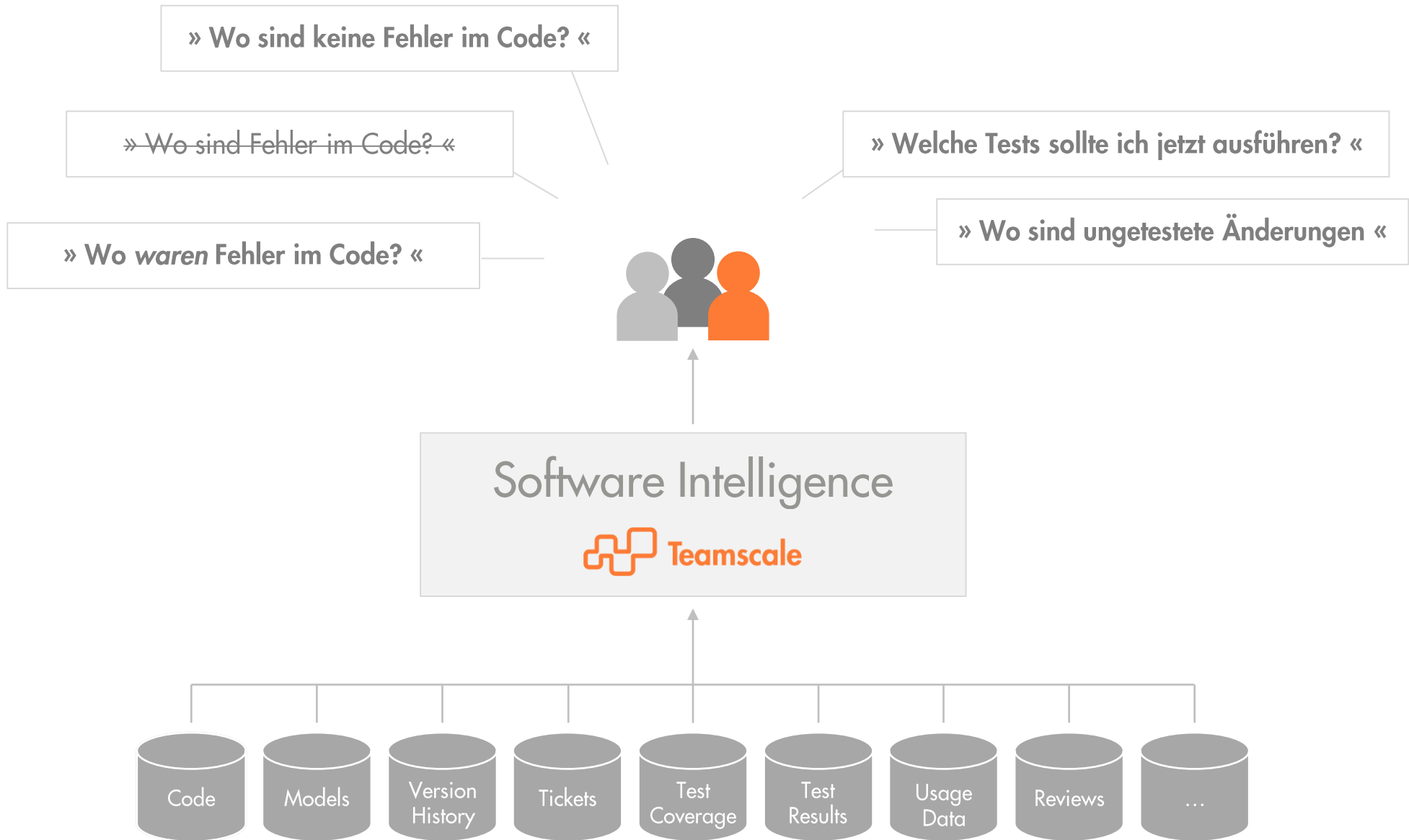




... 4000+ test cases
don't execute
changed code







- Elmar J.
- Lounge
- OOP Program
- Expo
- Event Hub
- My Program
- Attendees
- Matchmaking
- Settings
- Hilfe
- Abmelden

Software Intelligence

Welche Überraschungen lauern in Ihrer Software?

CQSE



 Teamscale

Wo sind Schwächen in der Architektur?

Wo sind Probleme in Ihrem Quelltext?

Welche Änderungen sind ungetestet?

BESCHREIBUNG

Die CQSE GmbH entwickelt die Software-Intelligence-Suite »Teamscale«. Teamscale begleitet durch die Softwareentwicklung und bietet die Möglichkeit kontinuierlich die Qualität eines Systems sichtbar zu machen. Es vereint die Anforderungen von *Projektmanagern, Entwicklern, Testern* und *Quality-Engineers* und ermöglicht ein langfristiges und erfolgreiches Qualitätscontrolling.



Elmar Juergens
Gründer CQSE GmbH und Post-Doctoral Fellow an der TU München
CQSE GmbH

SEND MESSAGE



Jakob Rott
CQSE GmbH

SEND MESSAGE



Kontakt – Ich freue mich auf Fragen 😊



Dr. Elmar Jürgens

juergens@cqse.eu

+49 179 675 3863

CQSE GmbH
Centa-Hafenbrädl-Str 59
81249 München
www.cqse.eu

