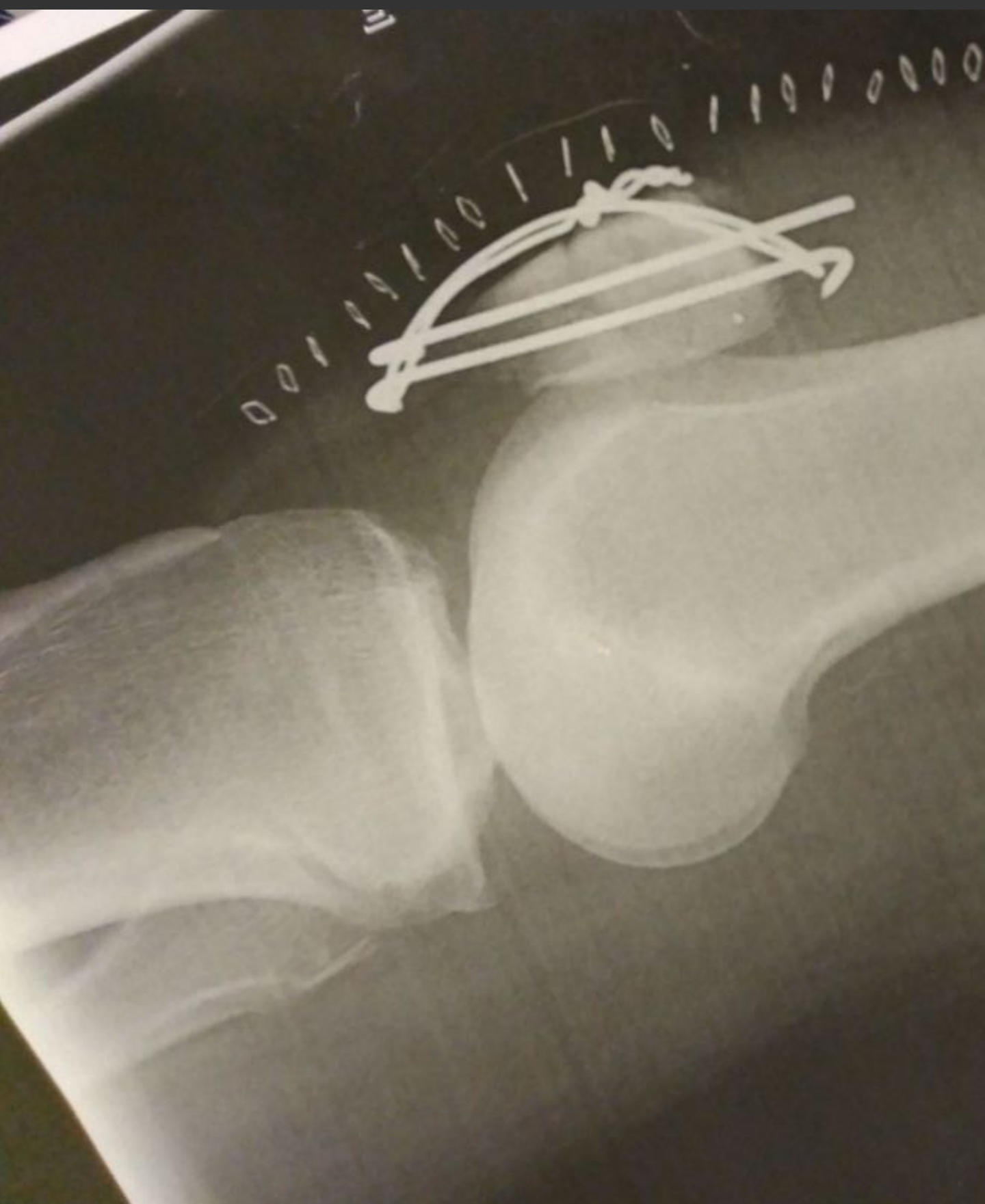
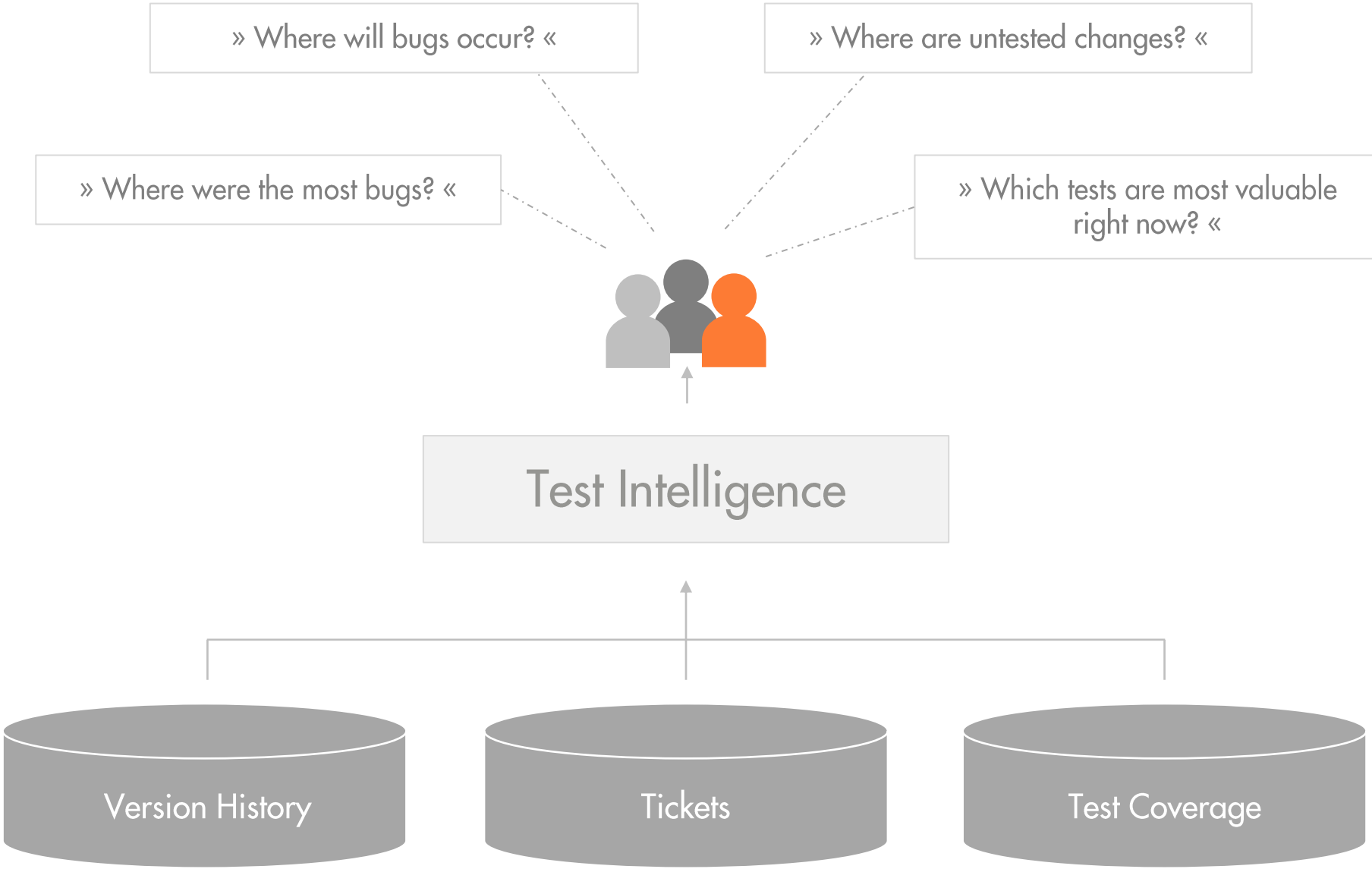
AI on Trial: How Well Can AI Help Us Find More Bugs Faster?

Elmar Jürgens
CQSE, Germany

DEEP DIVE

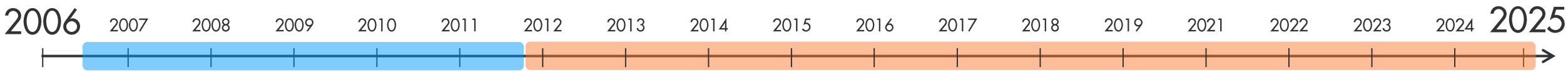
EuroSTAR 2025
EDINBURGH 3-6 JUNE

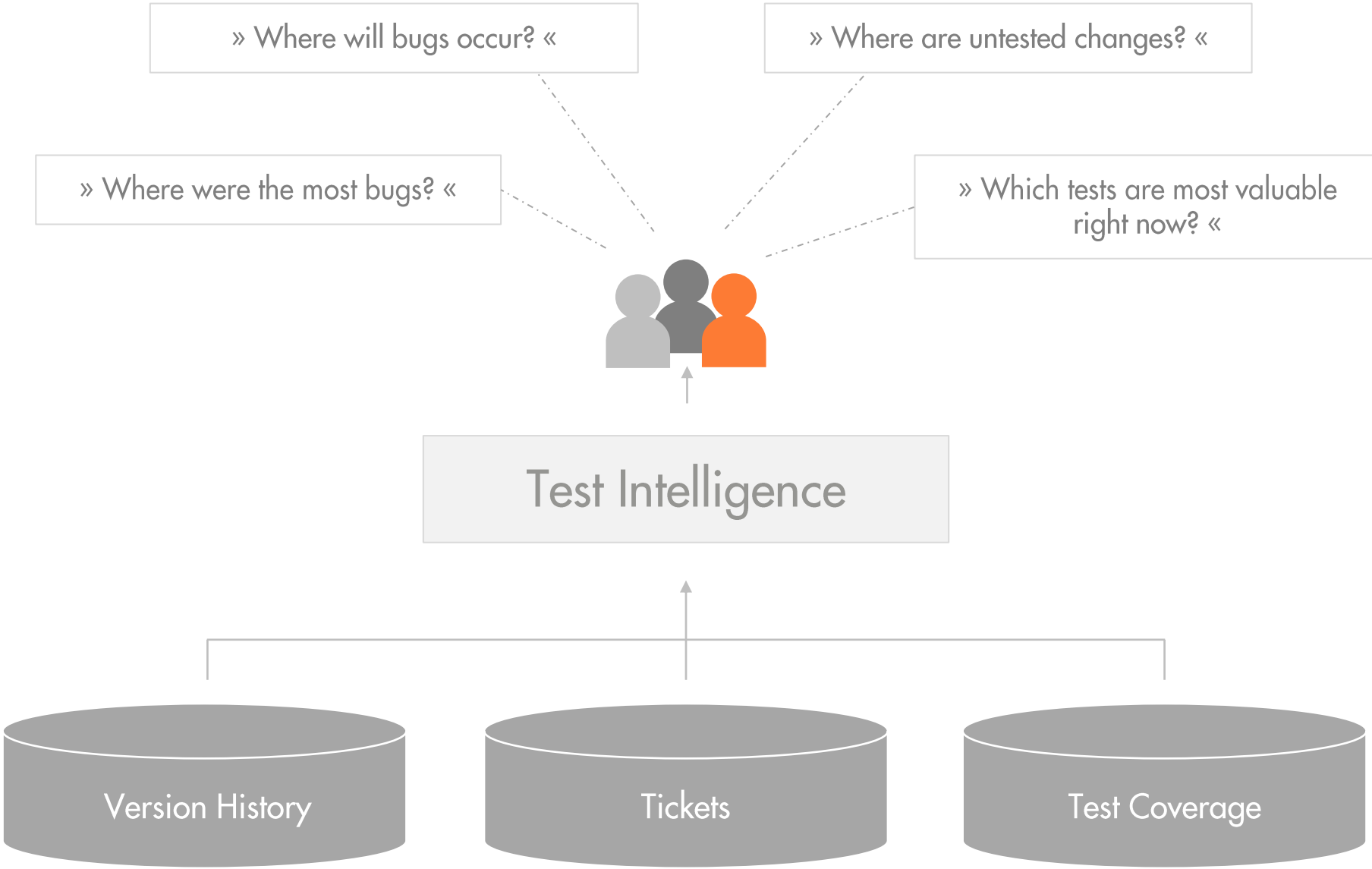


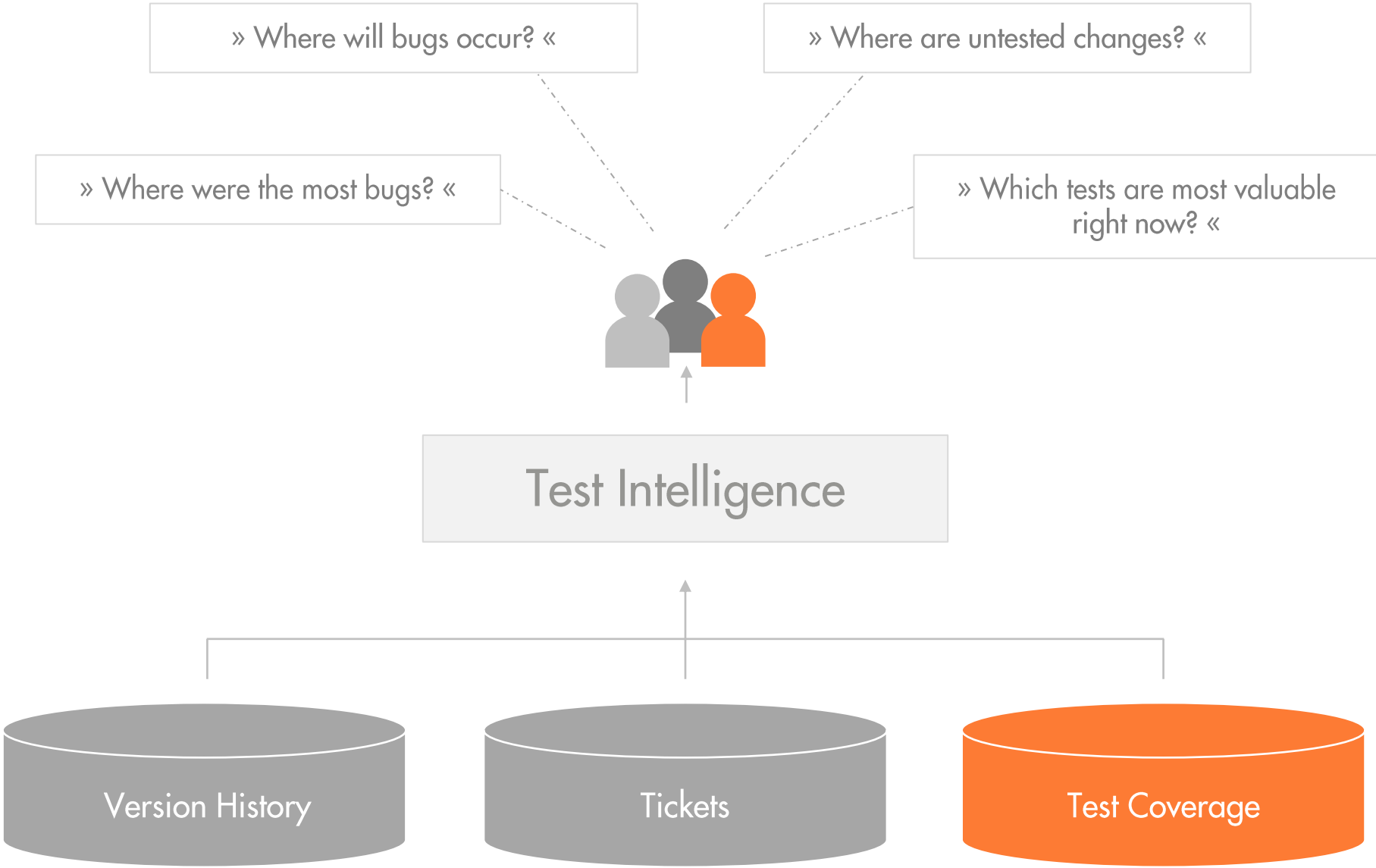

» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Test Coverage

CQSE

Teamscale




» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Test Coverage


» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Test Coverage

Picture1.png - Pixelitor 4.2.3

File Edit Layer Select Image Color Filter View Help

Sample Only the Active Layer/Mask

Untitled1 × Picture1.png ×

Layers

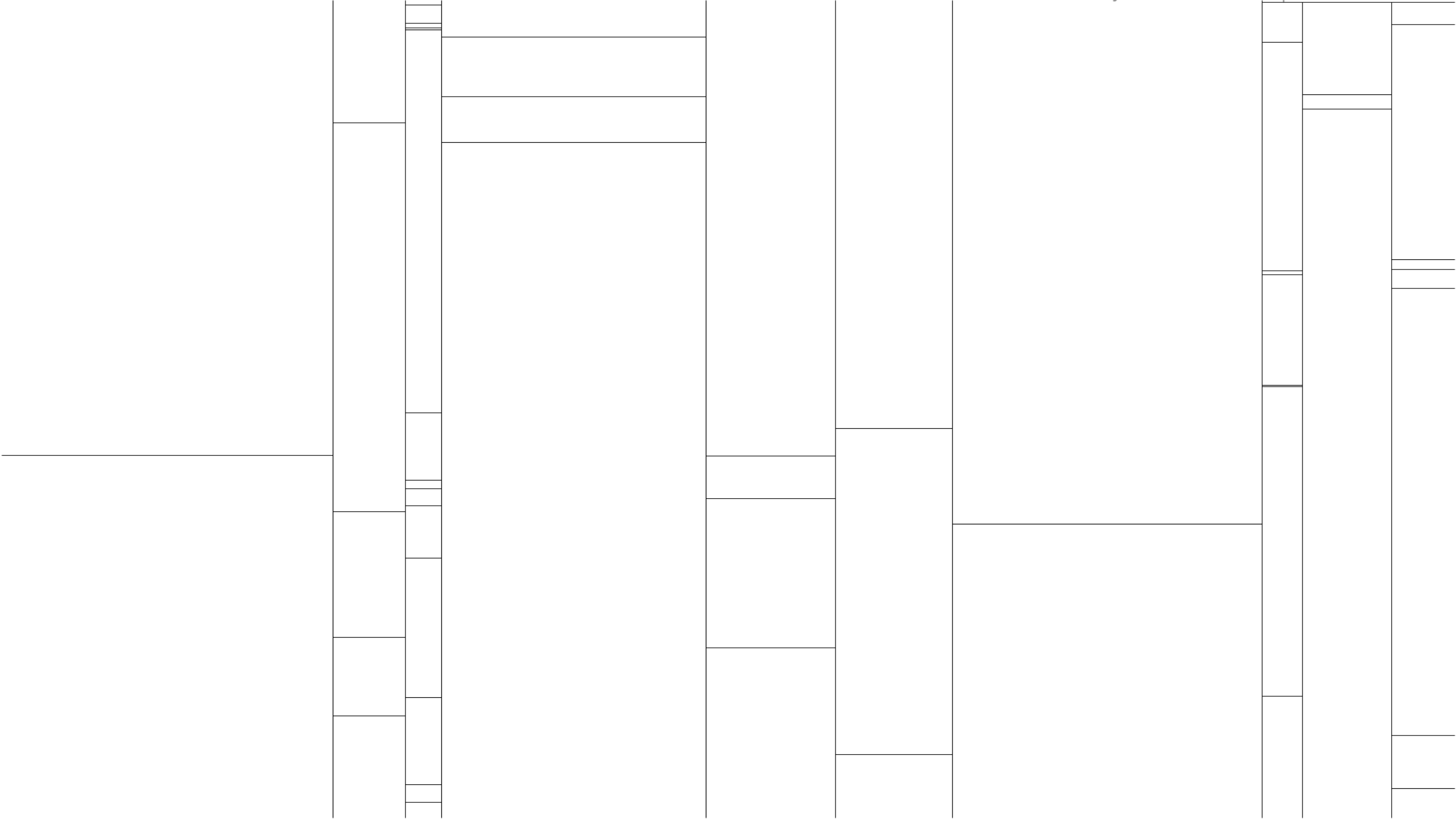
Opacity: 100 % Normal

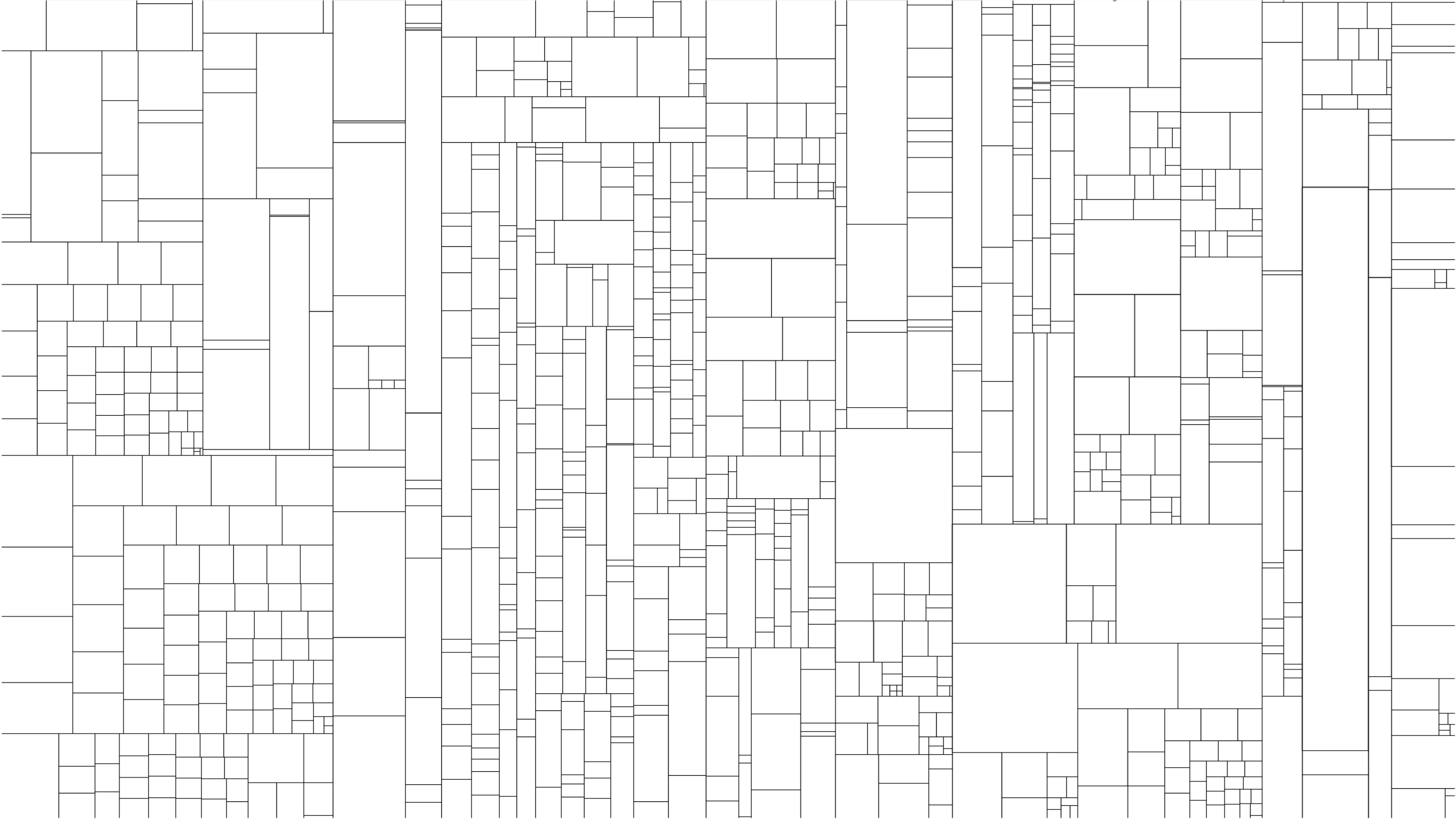
layer 1

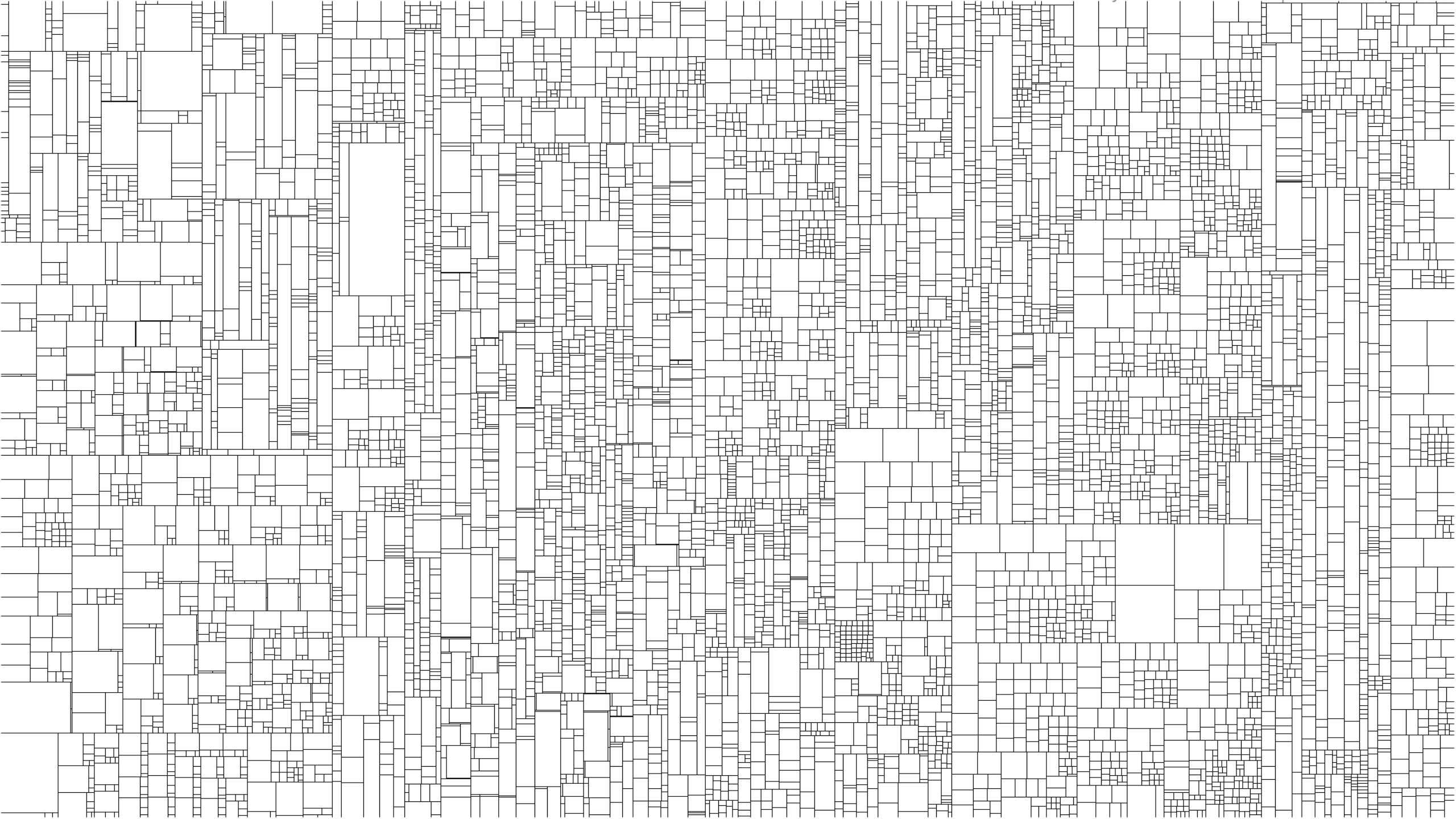
Picture1.png was opened.

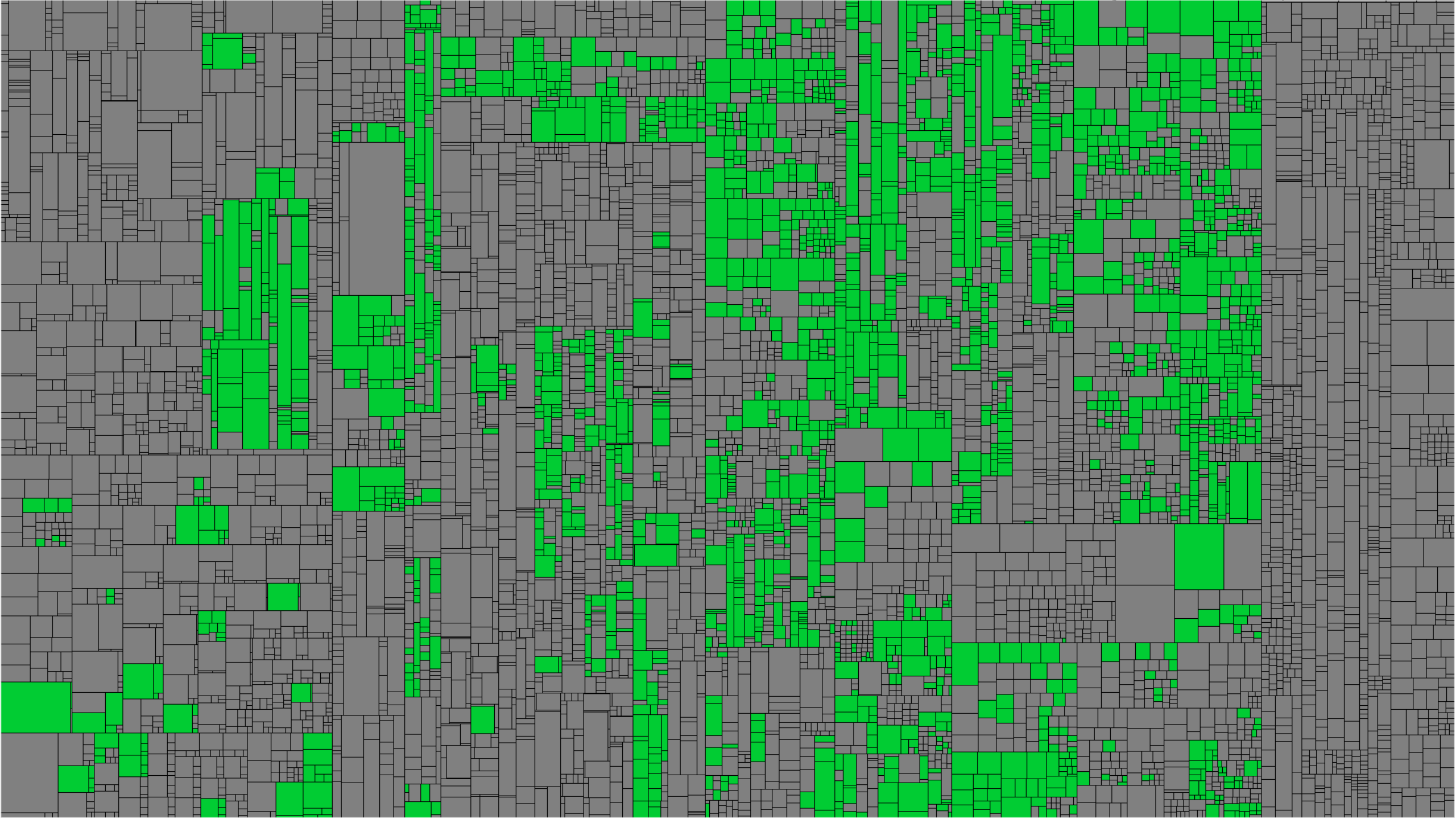
Zoom: 100% Fit 100%

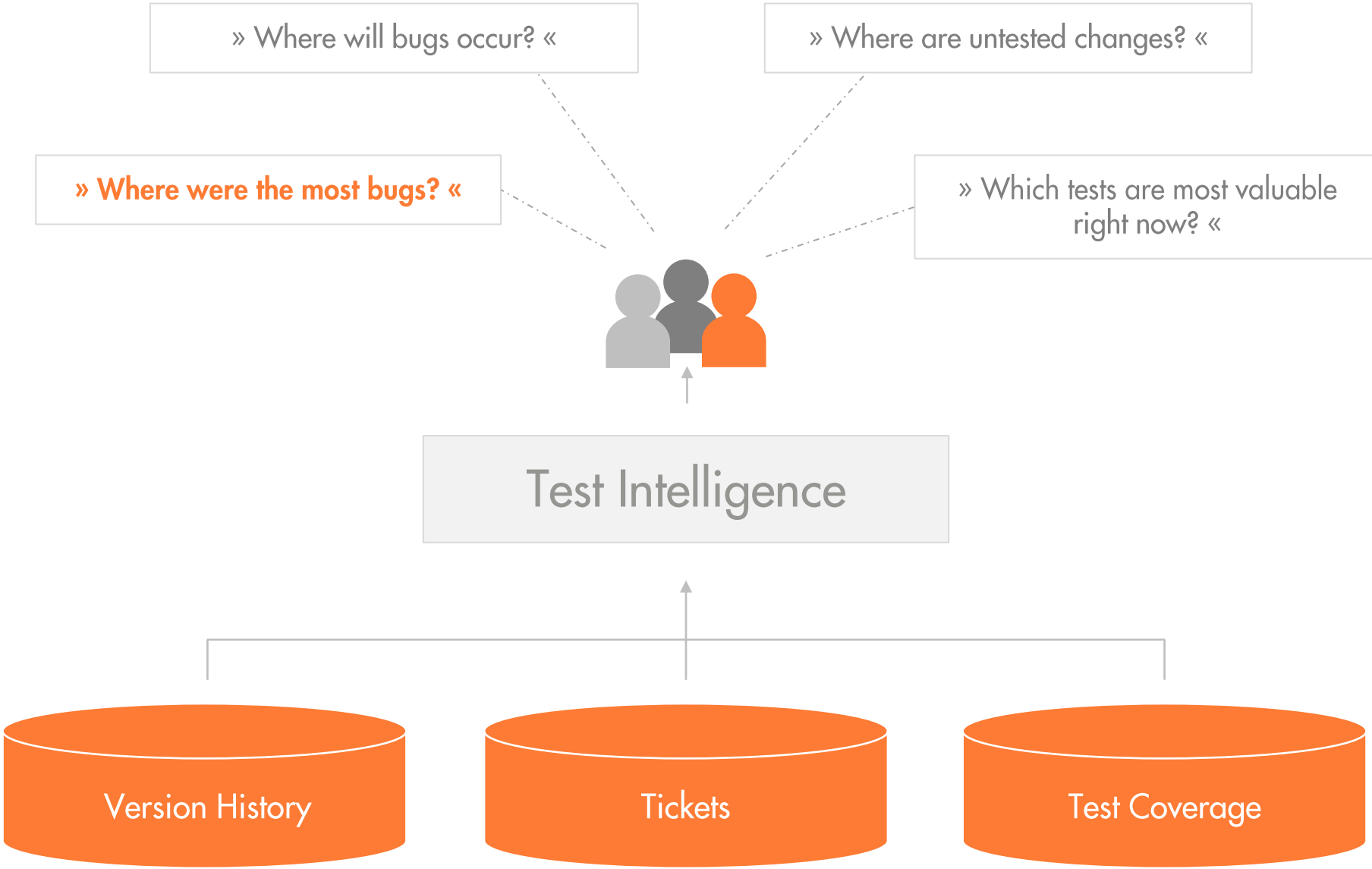
```
114
115     private static void createAndShowGUI(String[] args) {
116         assert calledOnEDT() : threadInfo();
117
118         Messages.setMsgHandler(new GUIMessageHandler());
119
120 //        GlobalKeyboardWatch.showEventsSlowerThan(100, TimeUnit.MILLISECONDS);
121
122         Theme theme = 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
```

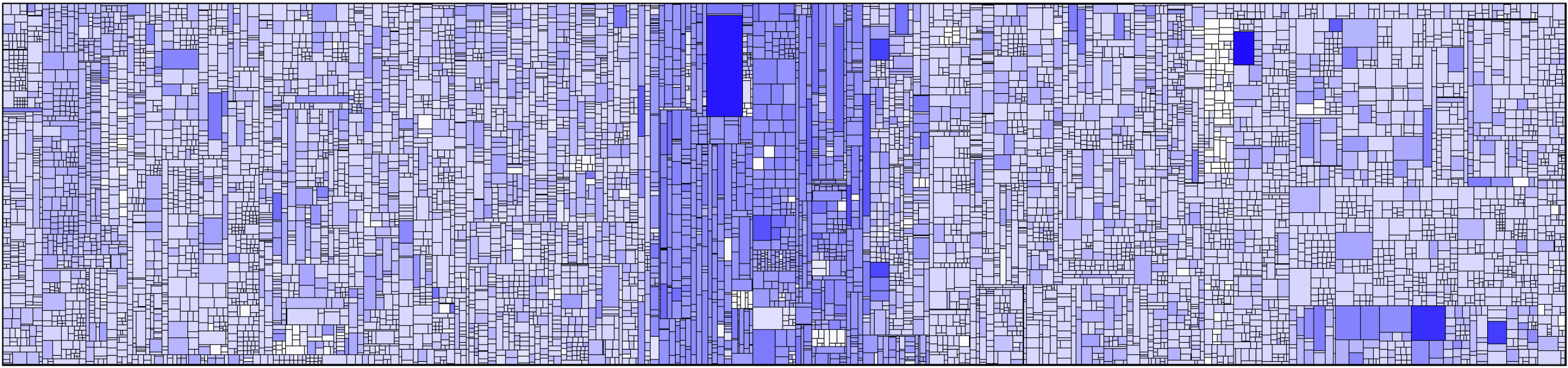


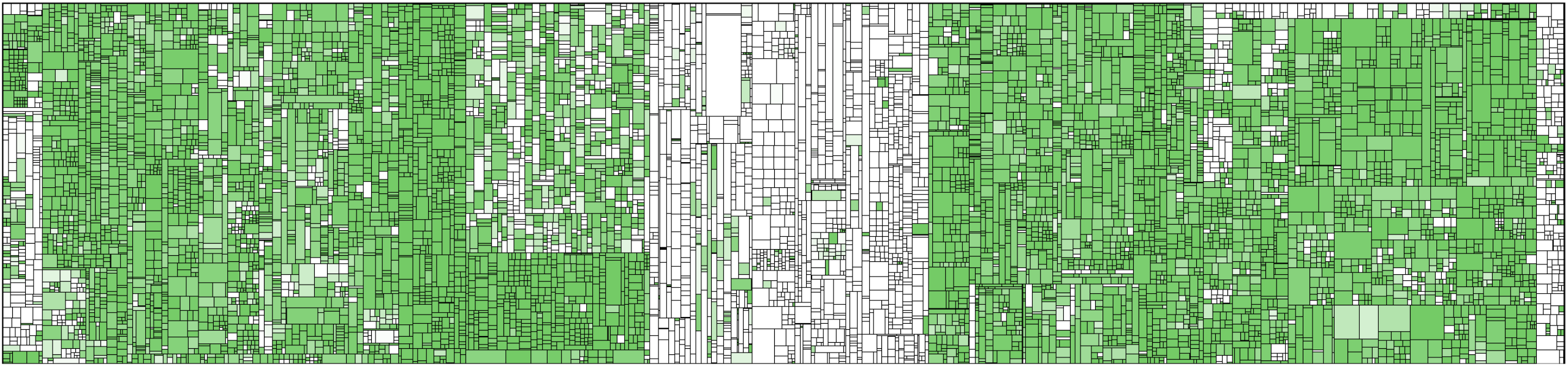


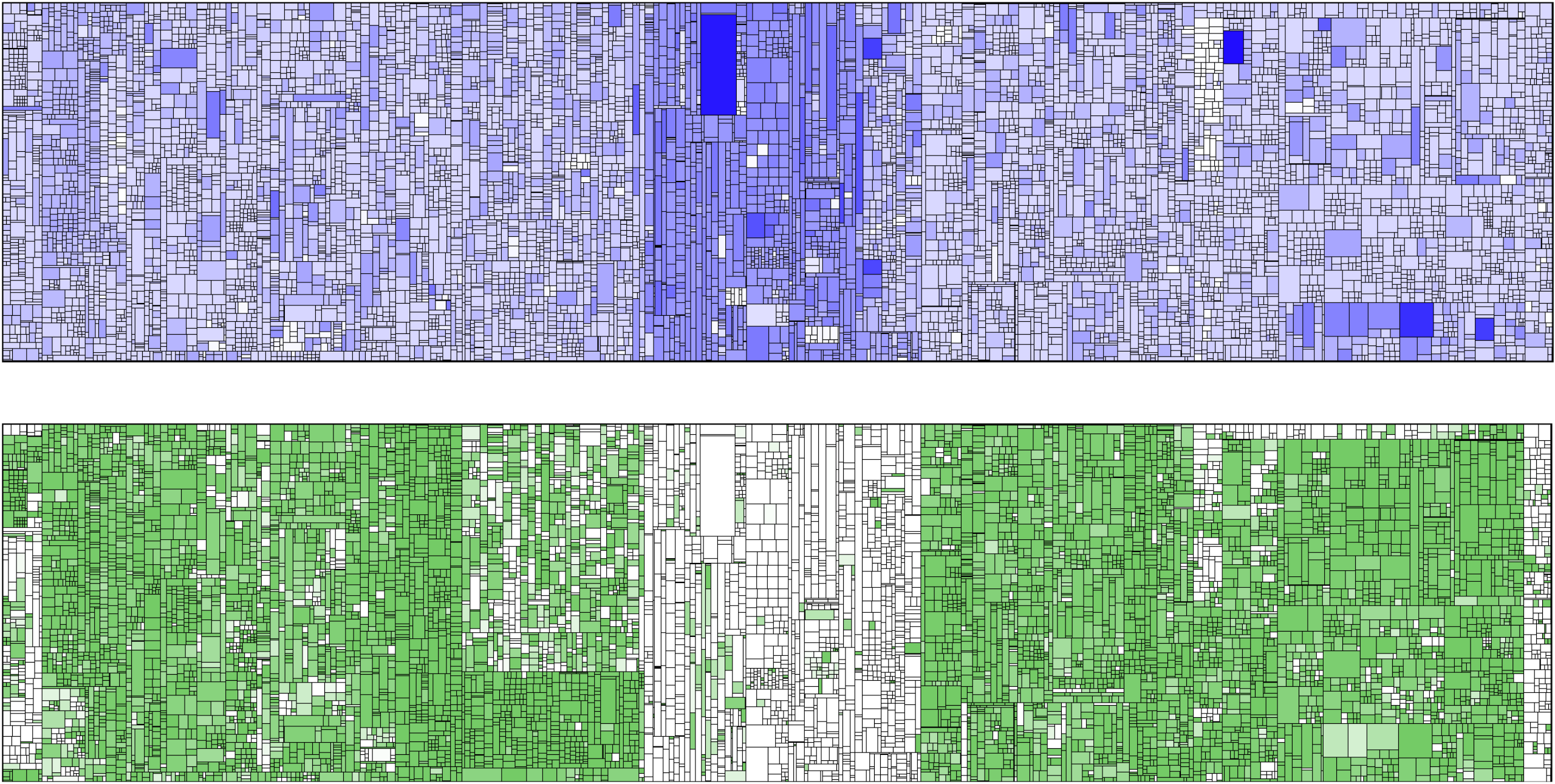


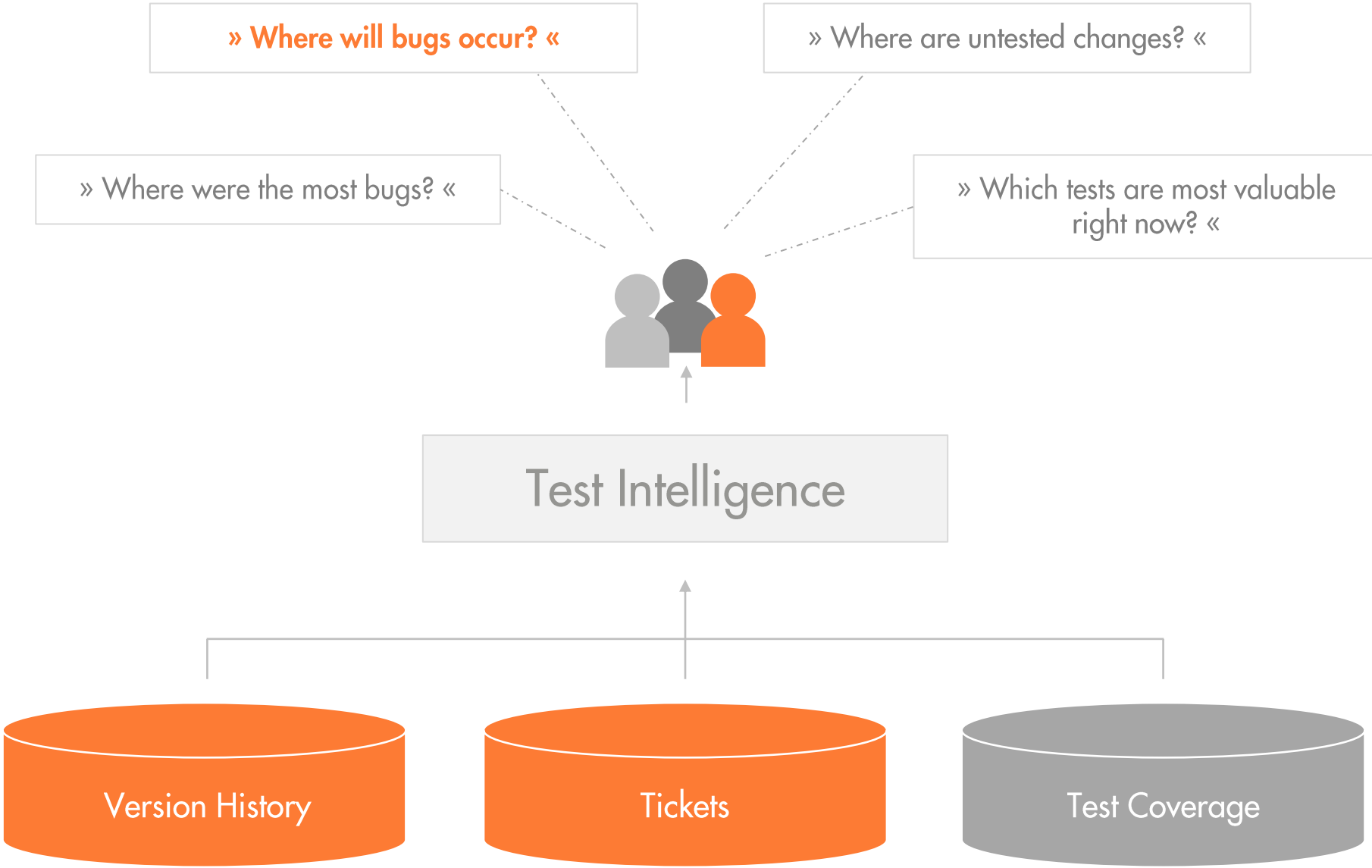


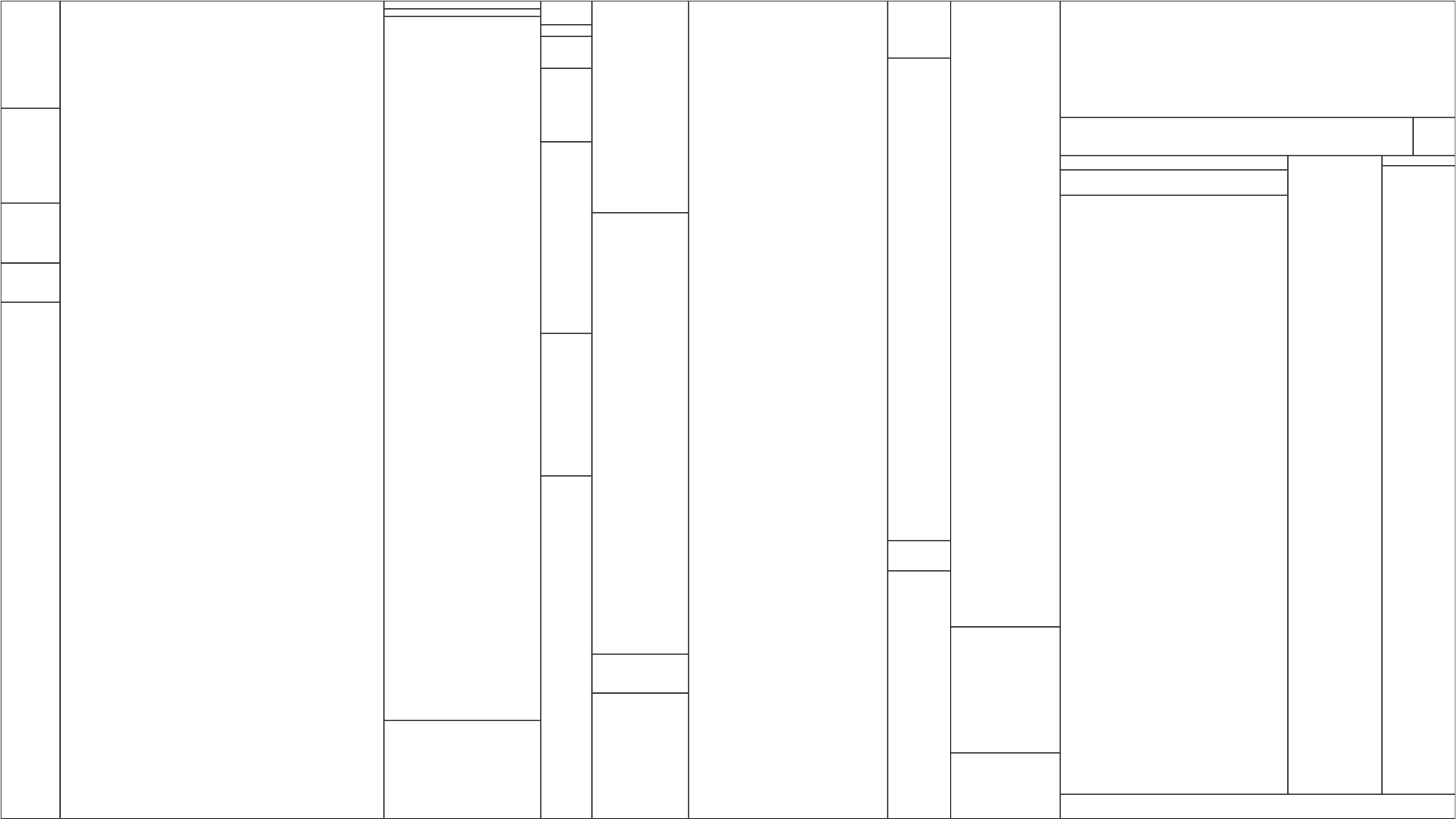

» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Test Coverage

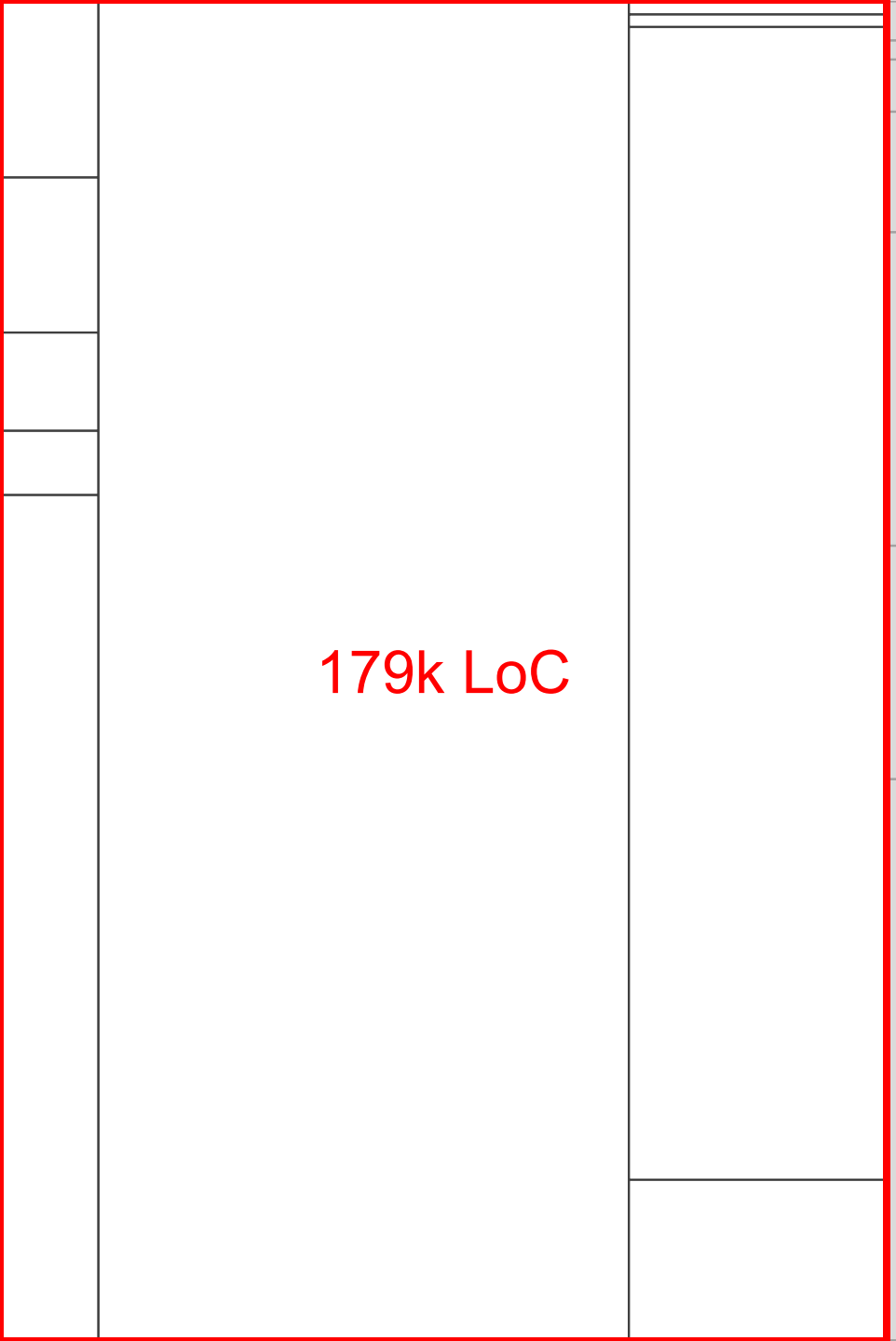


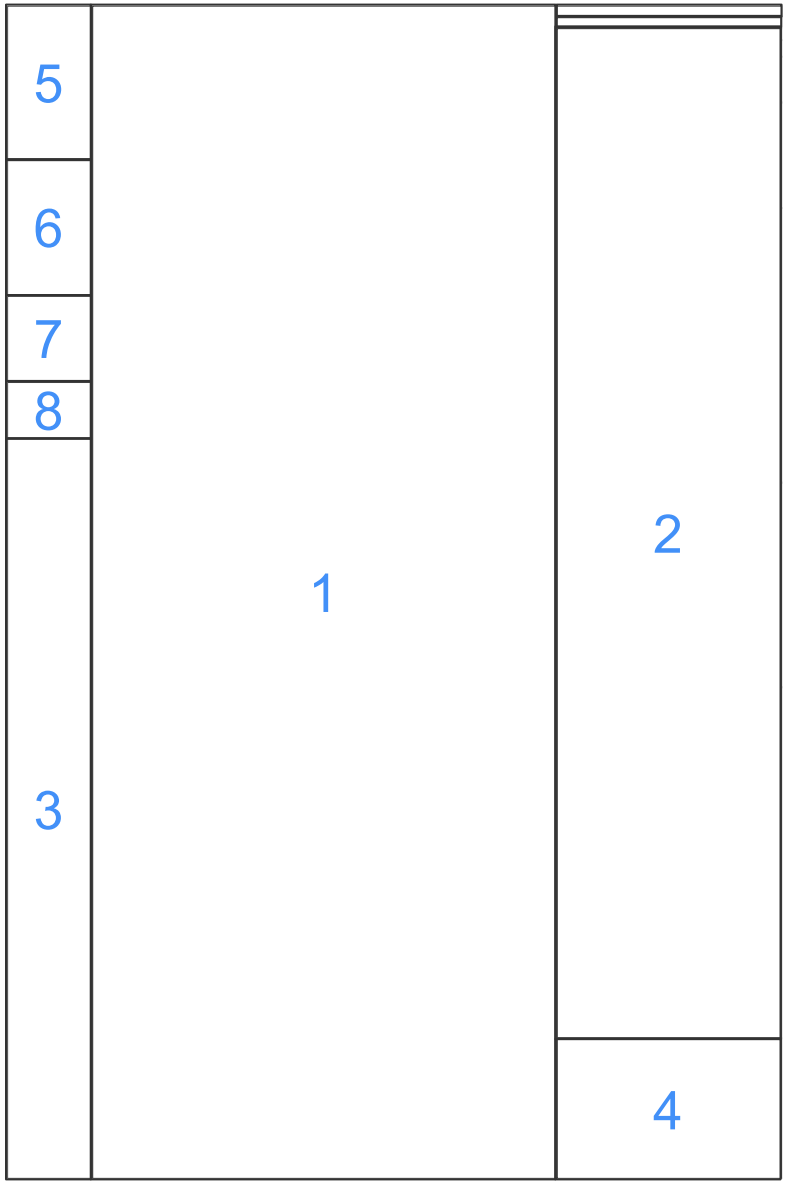


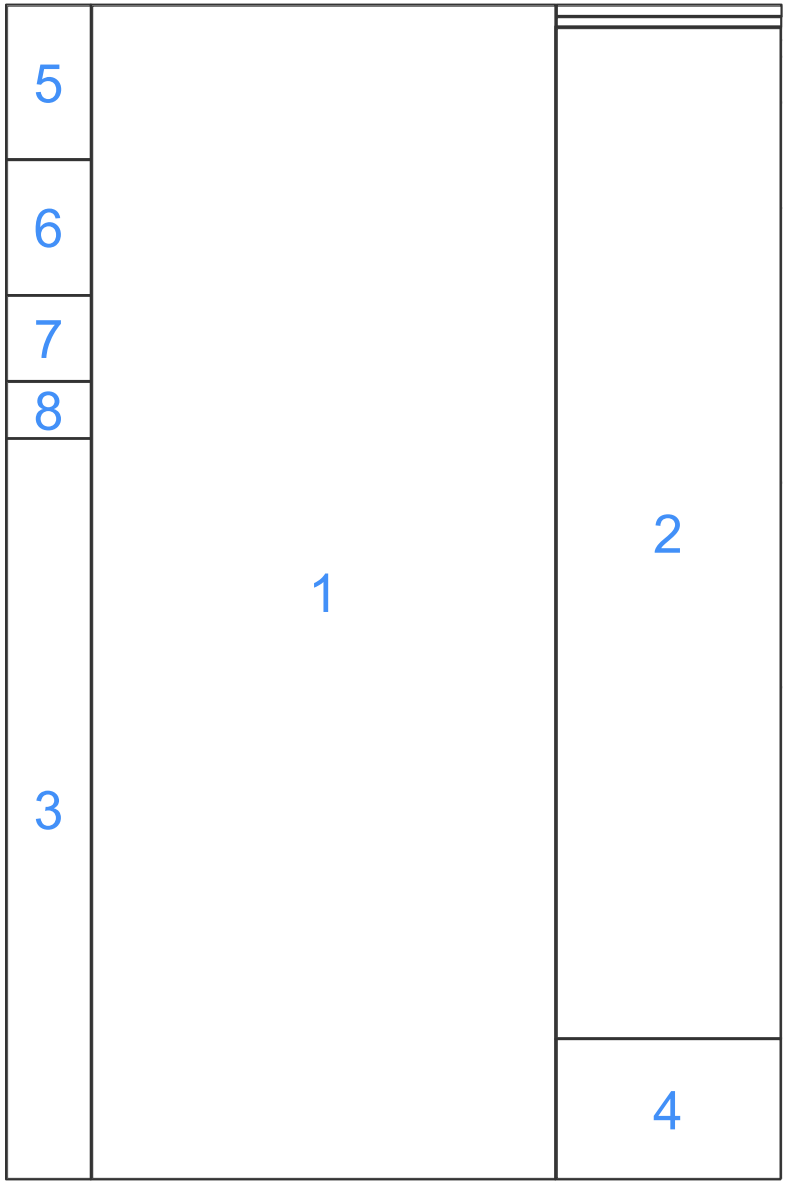


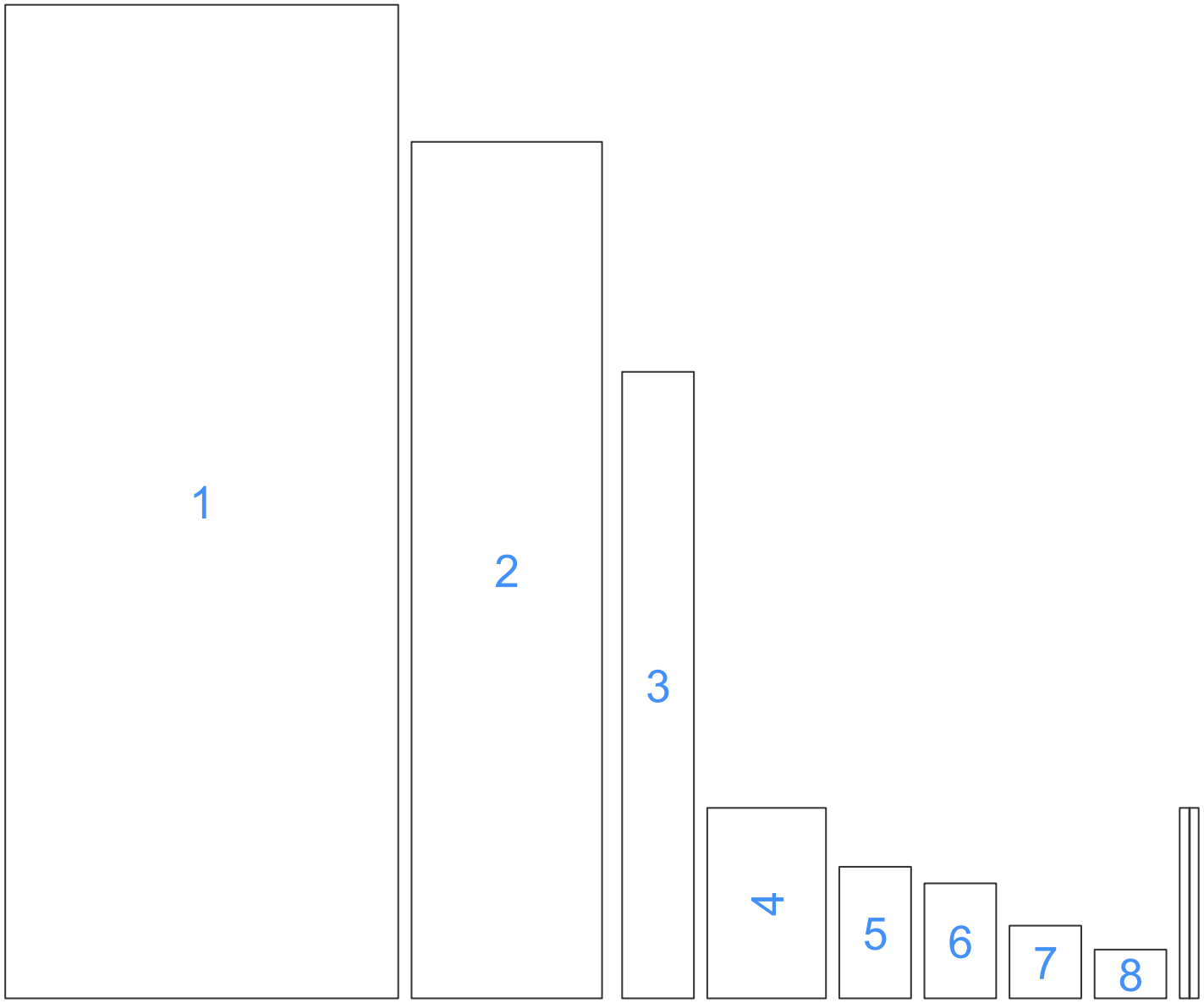

» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Test Coverage

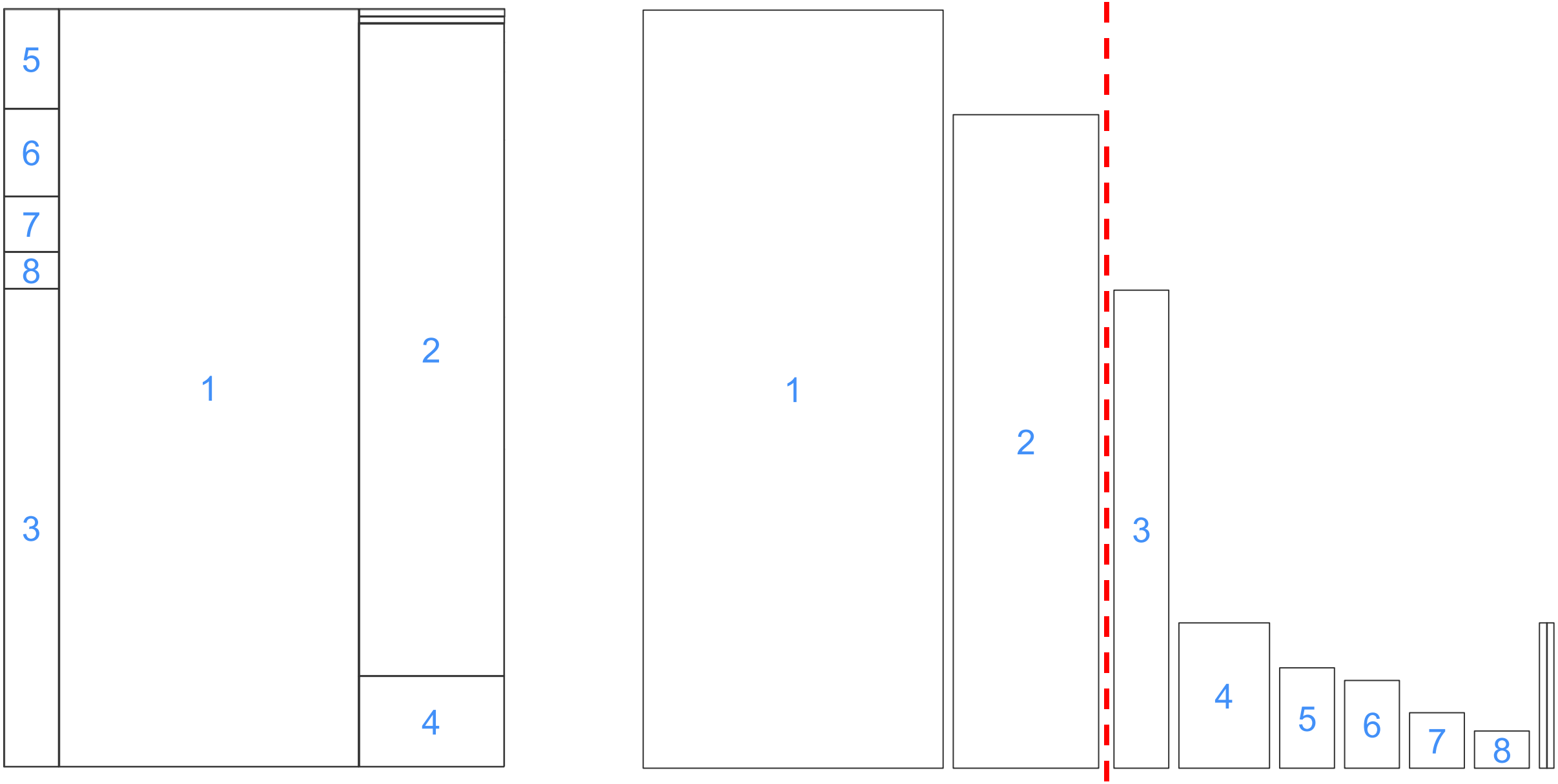


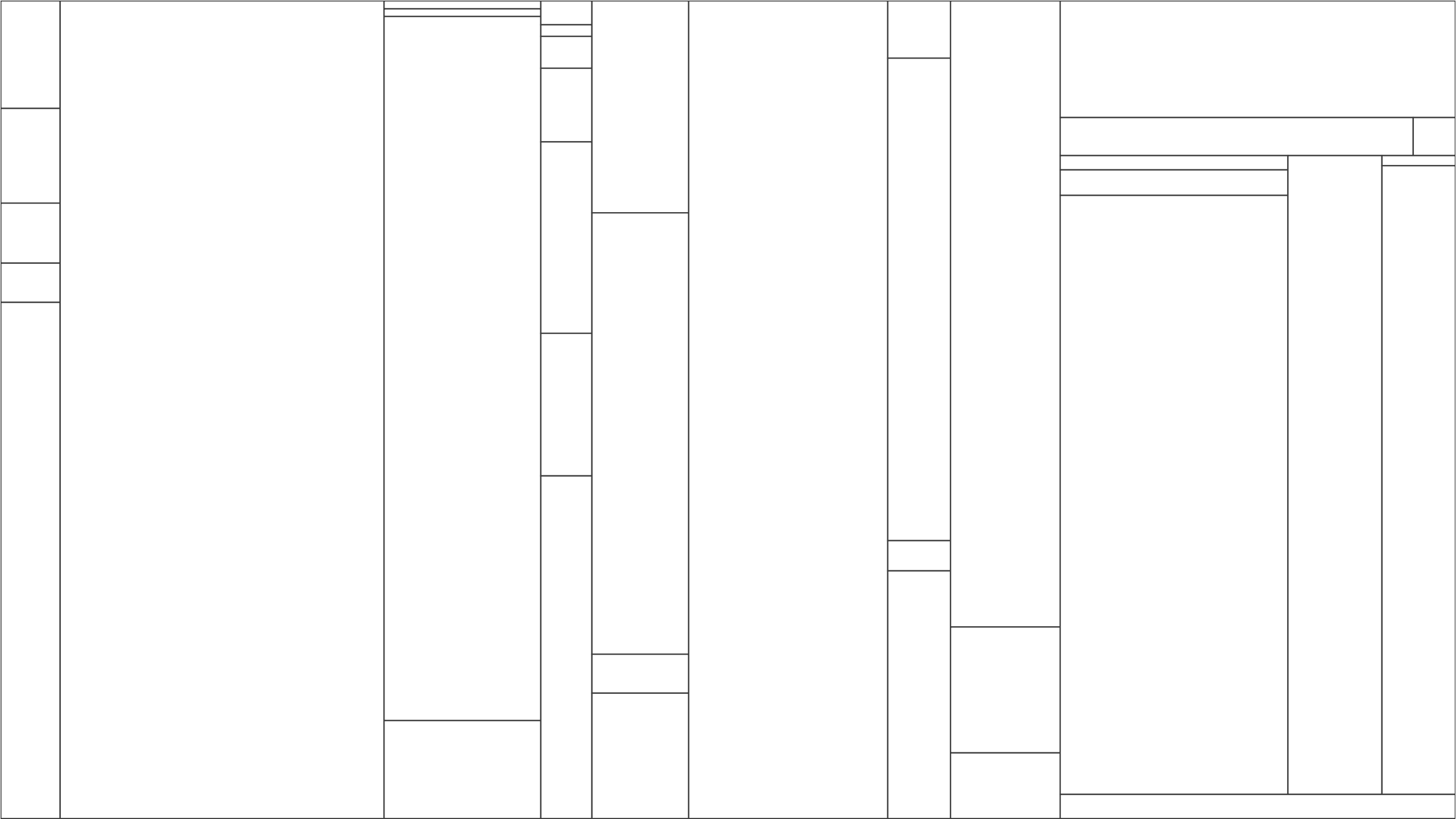

179k LoC

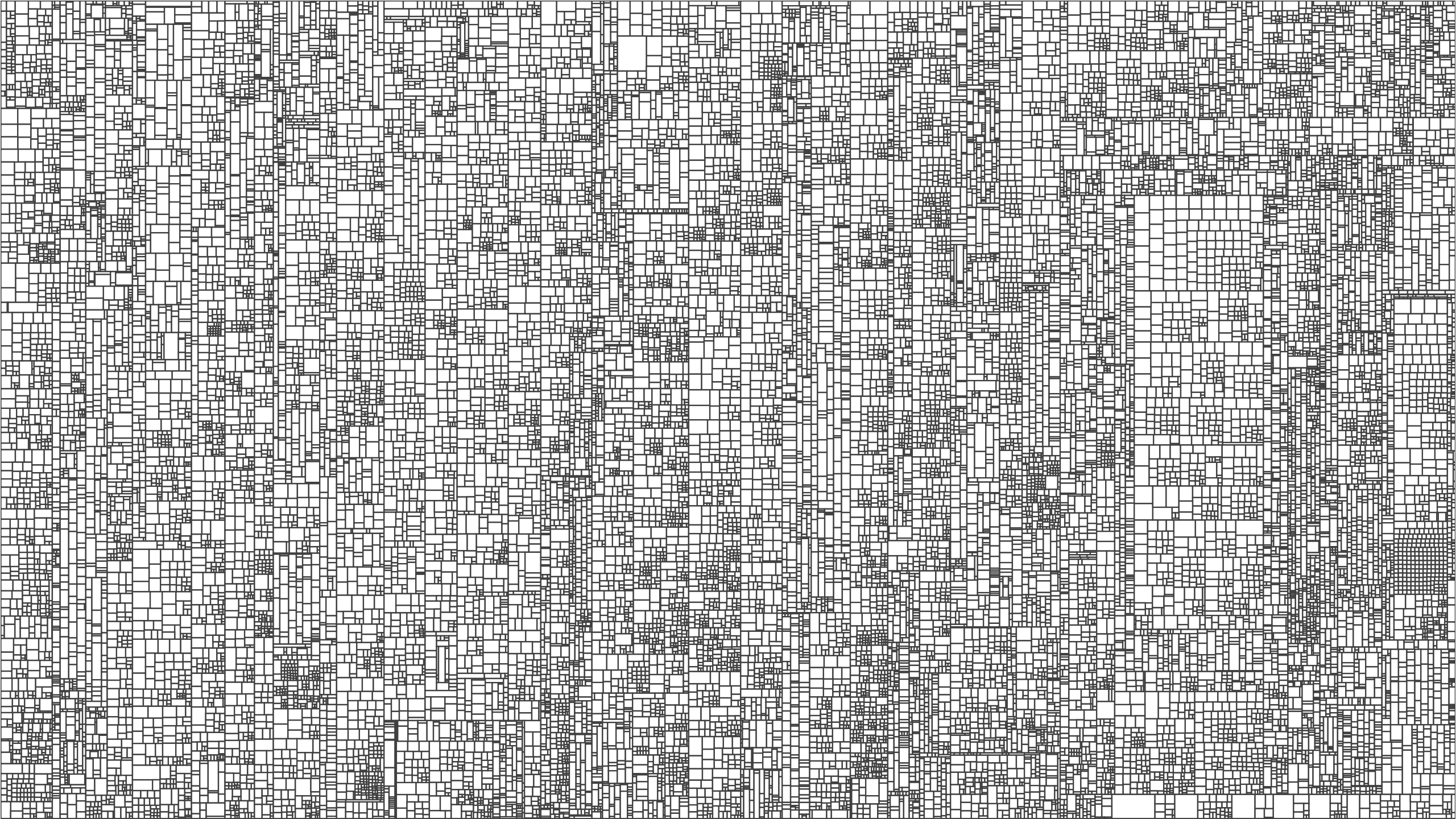

5
6
7
8
3
1
2
4

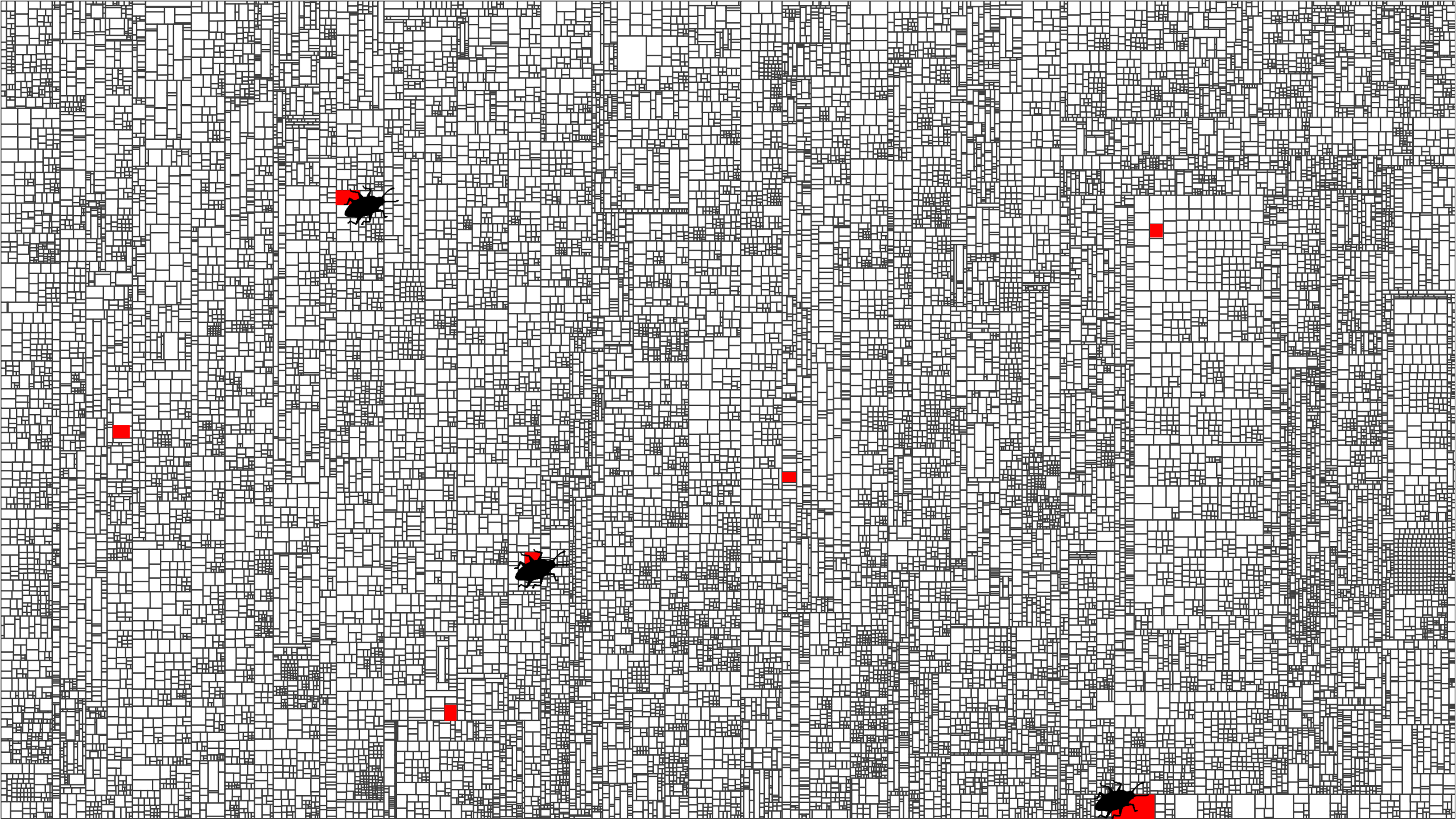

5
6
7
8
3
1
2
4

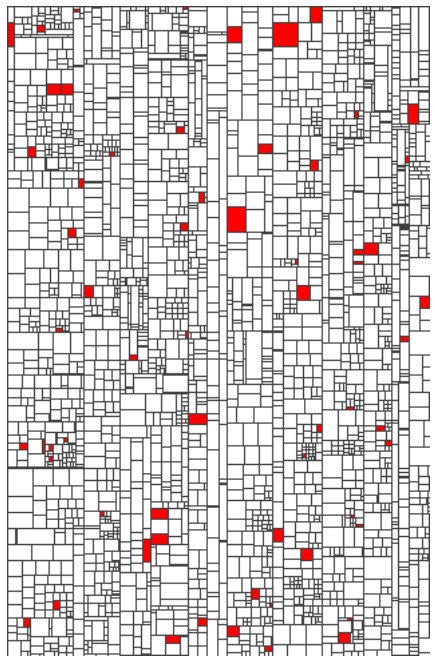

1
2
3
4
5
6
7
8

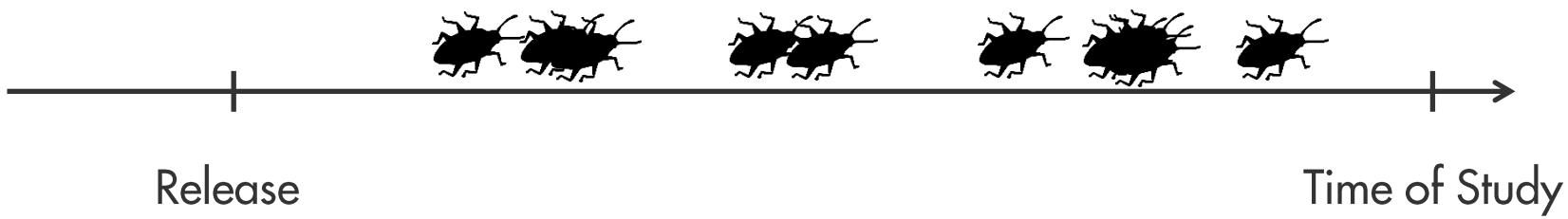

153k LoC
85%
26,5k LoC
15%
5
6
7
8
1
2
3
4
1
2
3
4
5
6
7
8

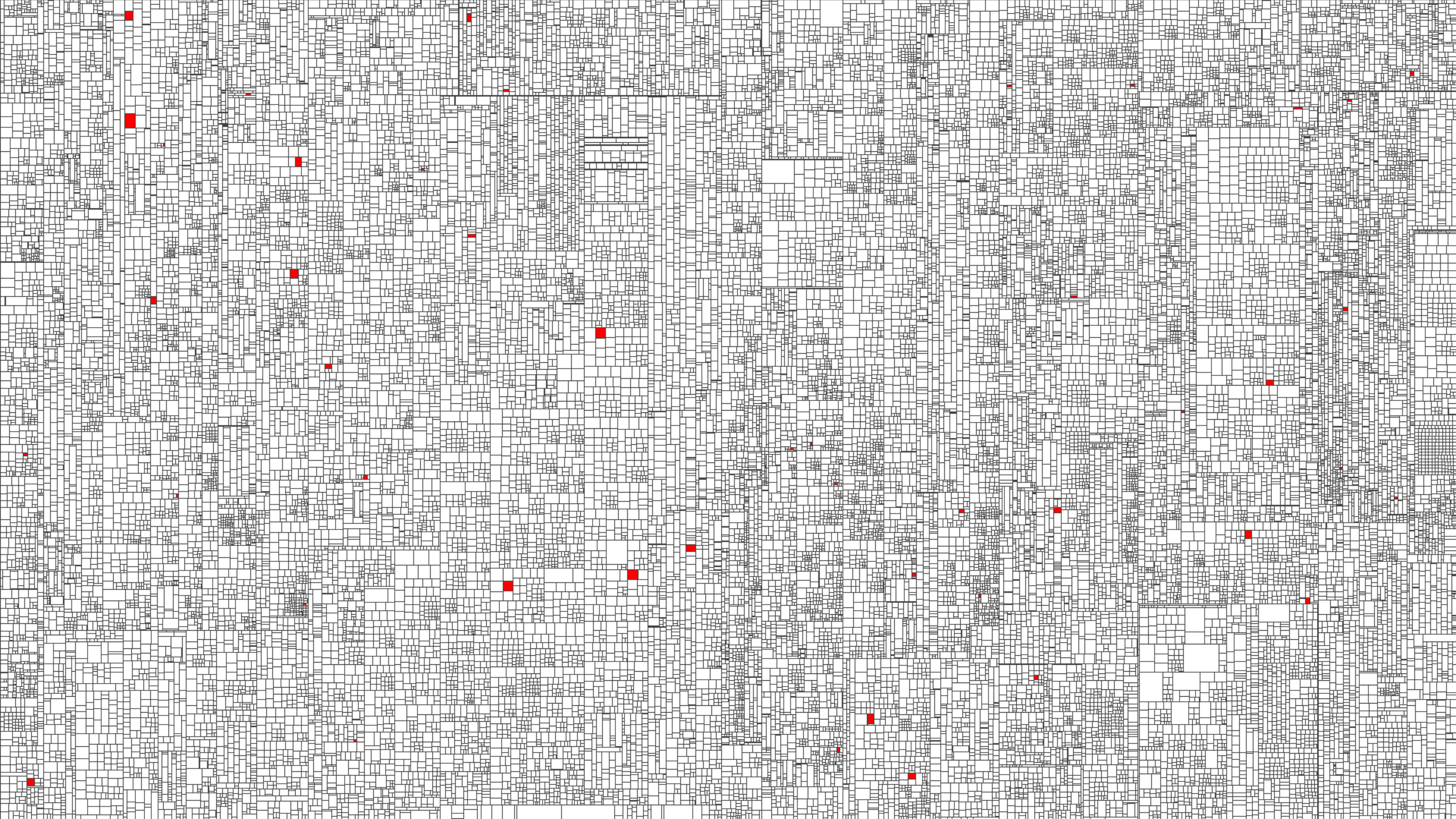









Release
Time of Study



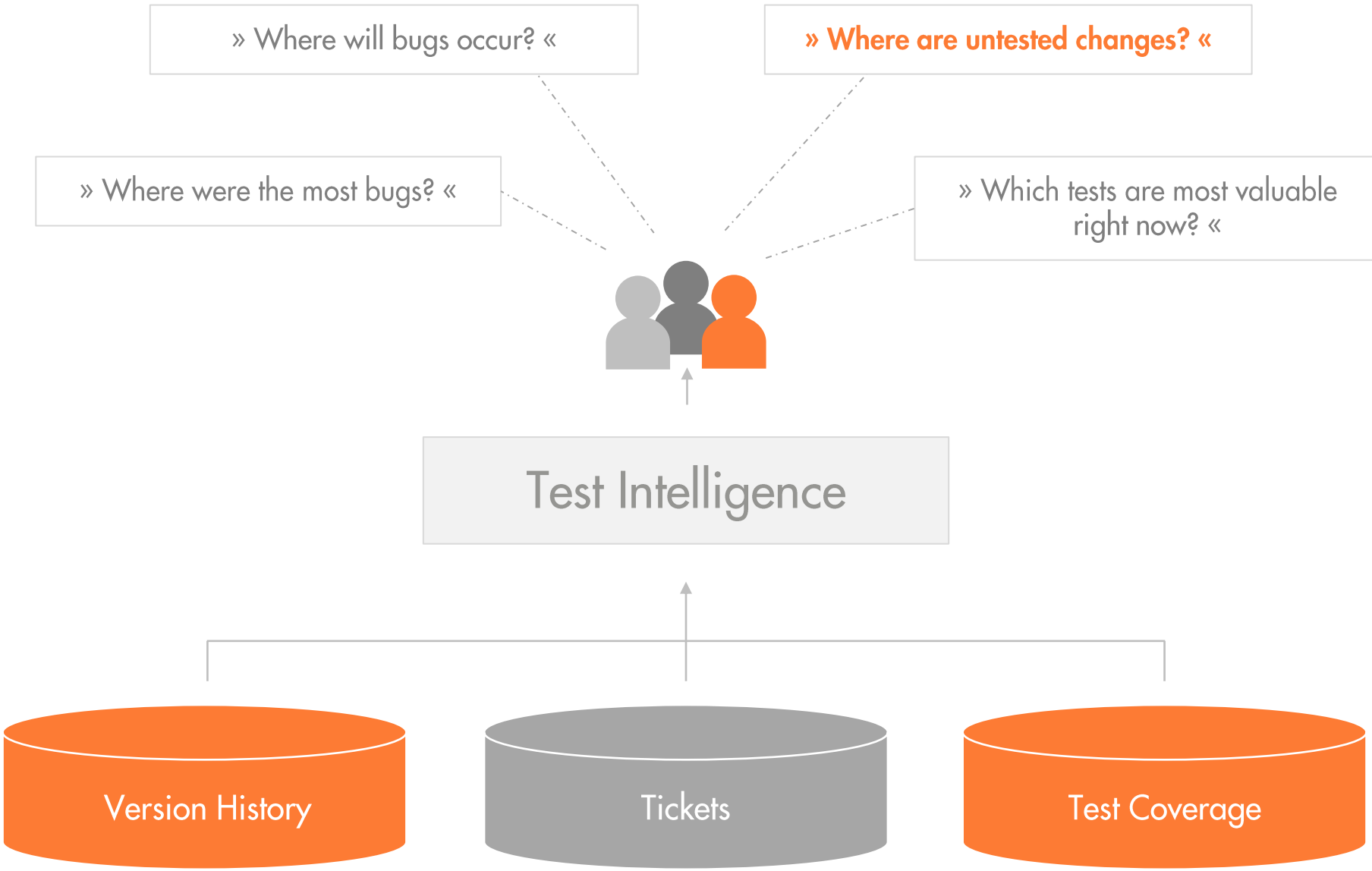
Evaluation

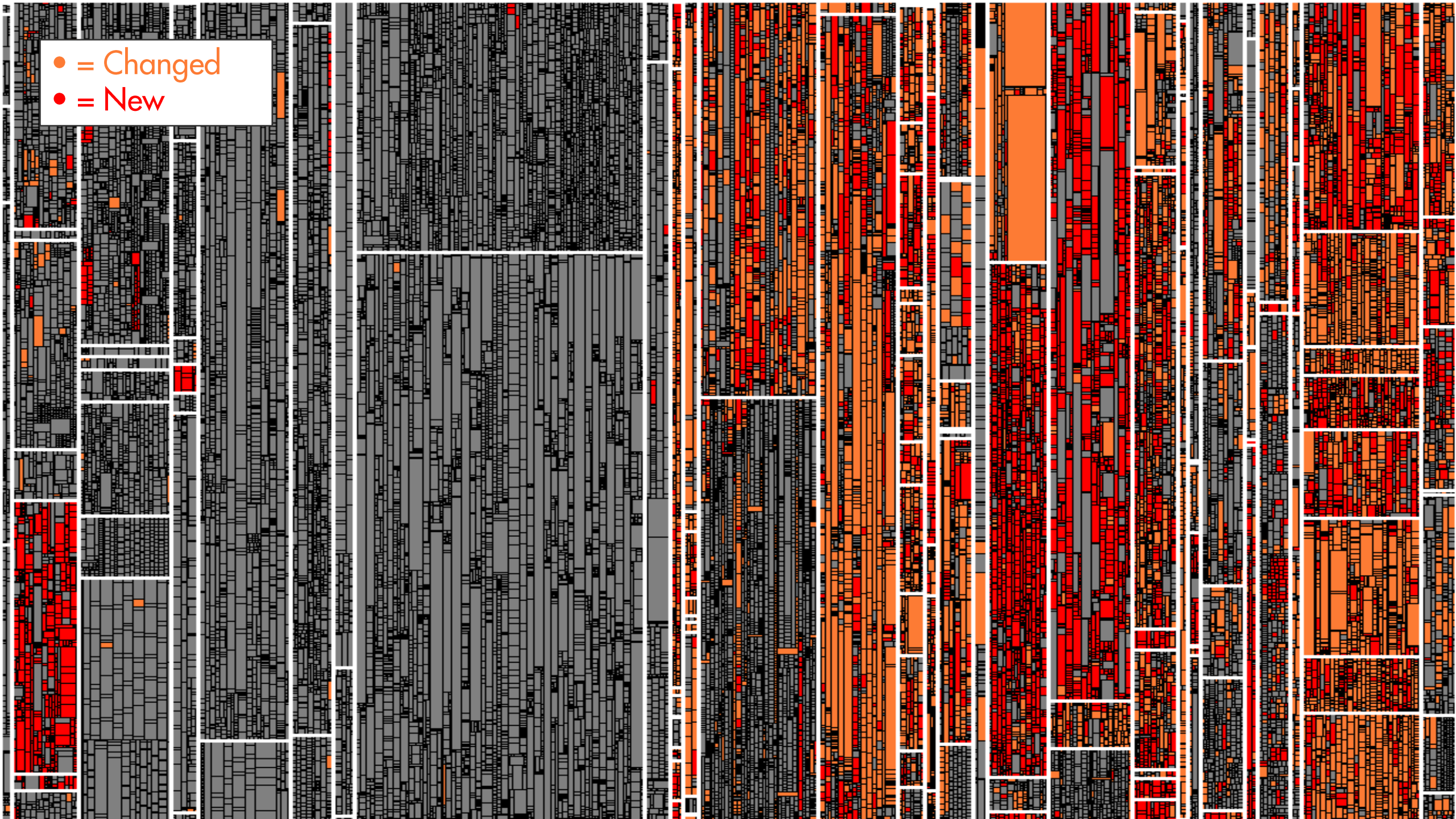
Release	# „defect prone“ Methods	# Bugs (Top 50)
1.4:	1127	0
2.0:	1176	0

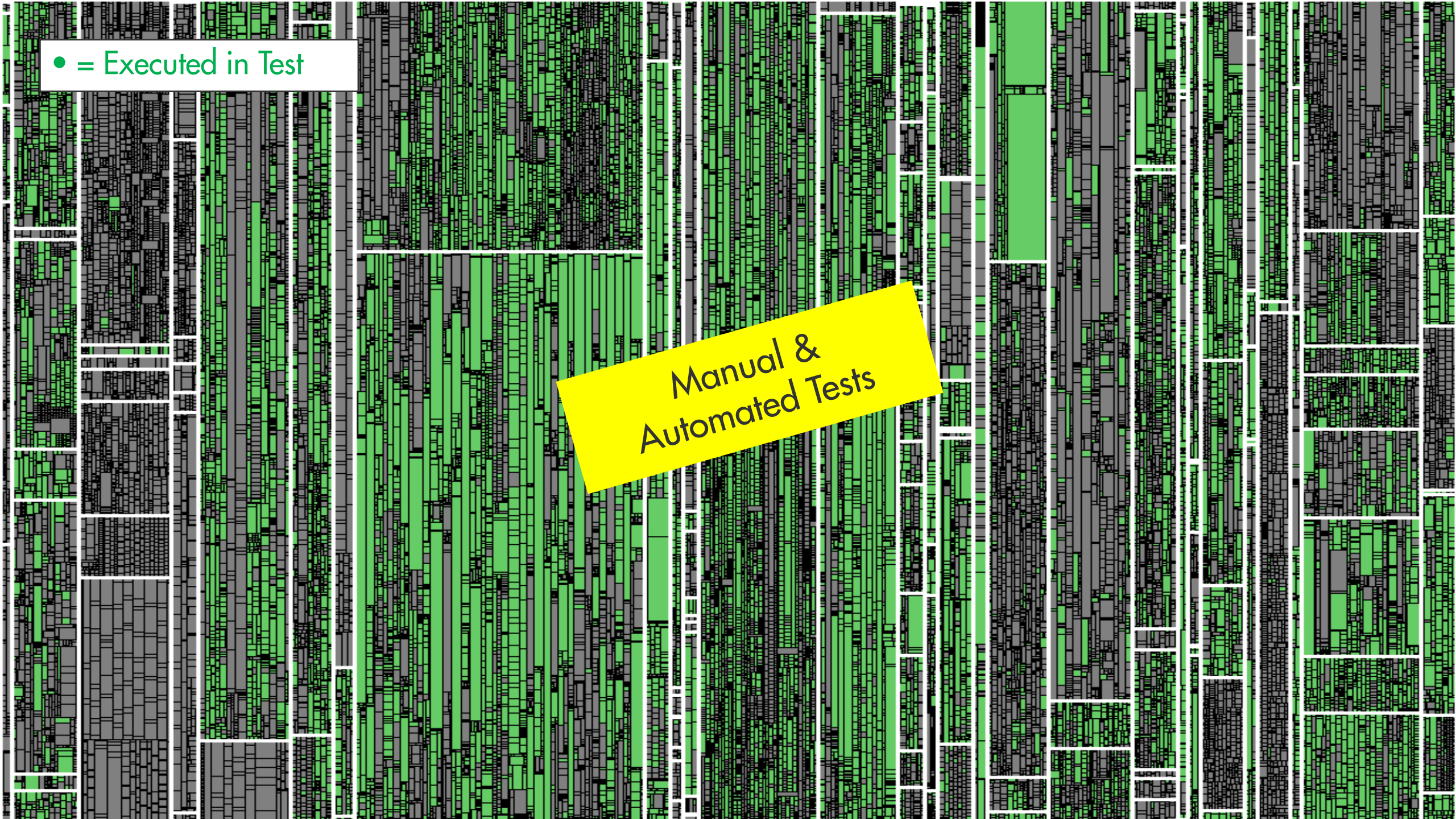
Pascarella, Palomba, Bacchelli, *Re-evaluating Method-Level Bug Prediction*, 2018:
Prediction not better than random classification.

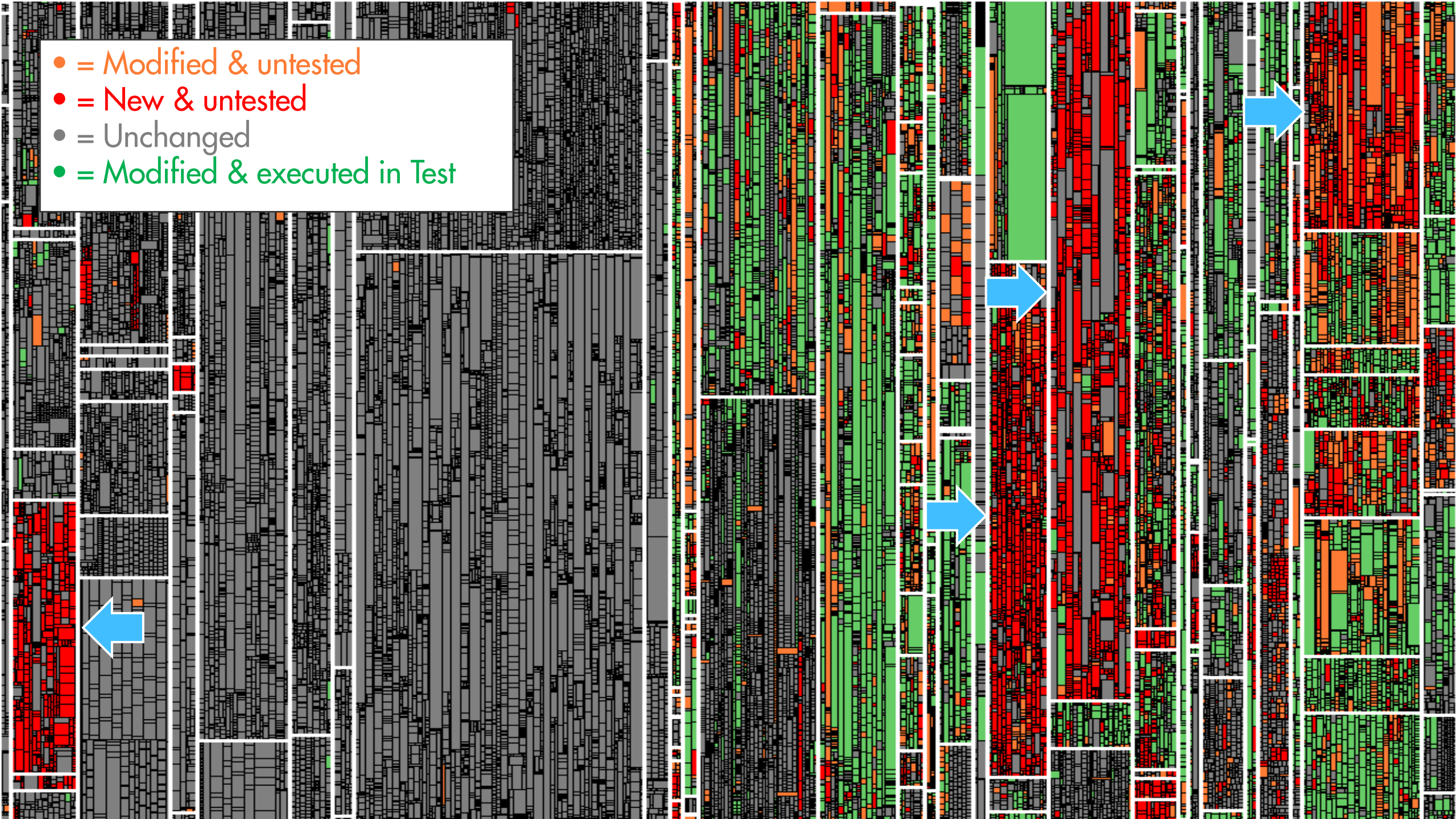
Chowdhury, Uddin, Hemmati, Holmes, *Method-Level- Bug Prediction: Problems and Promise*, 2024:

Method-Level Bug Prediction performance „extremely poor“.


» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Test Coverage


• = Changed
• = New


• = Executed in Test
Manual &
Automated Tests


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

Issue #	Subject			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

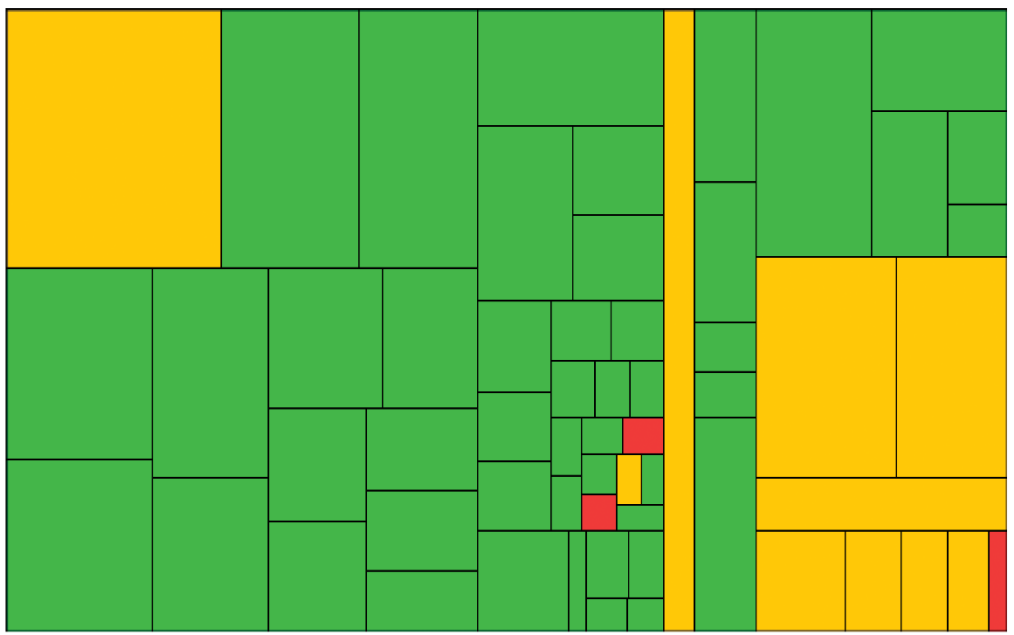
Assignee:

Project	Type	Priority	Resolution	Fix Version
TS	Maintenance	Normal	Green	Teamscale 4.5

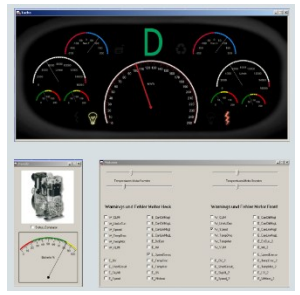
Component	Labels	Affected Version	Customer	Customer Issue
Backend	Performance			

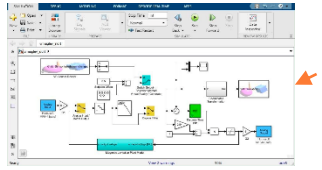
Epic Name	Freshdesk URL	Merge Request
		https://git.cqse.eu/cqse/teamscale/3621

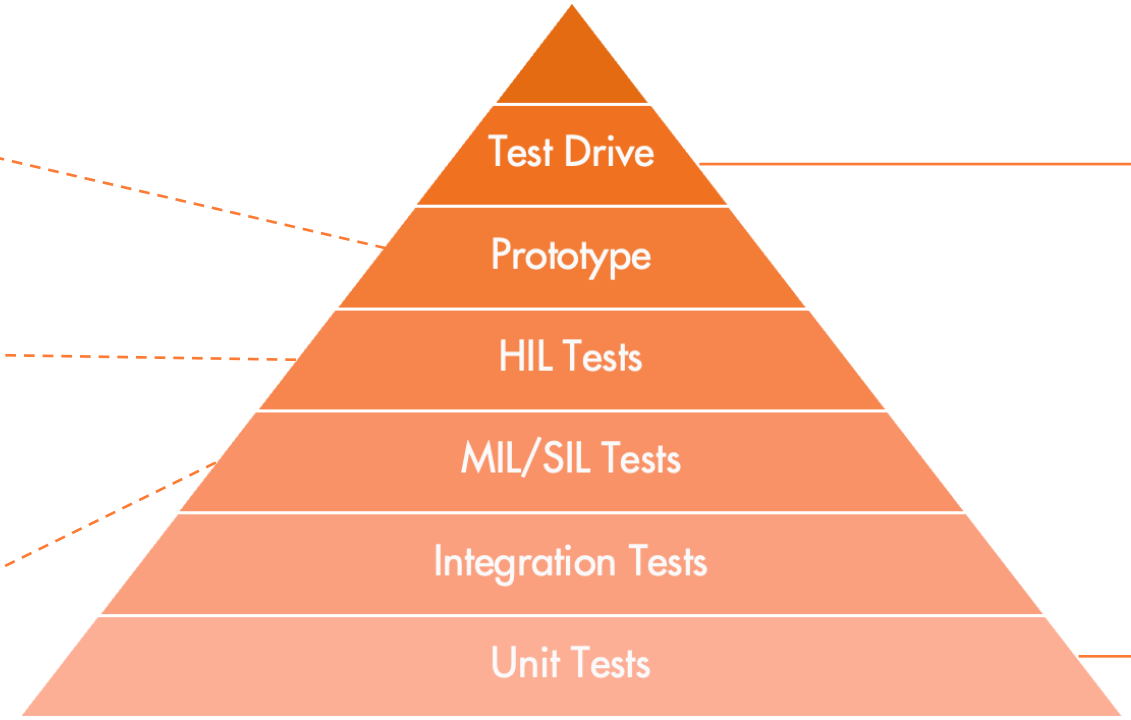
Aug 15 2018 12:37–Now | Test Gap: 22%



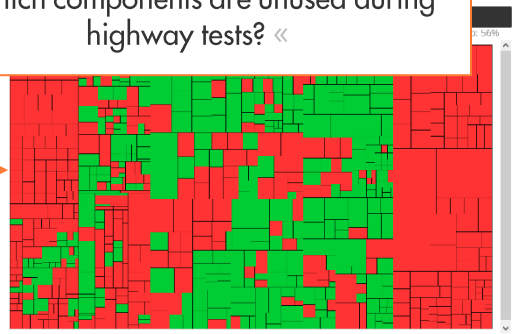
Test Gaps in different Test Environments

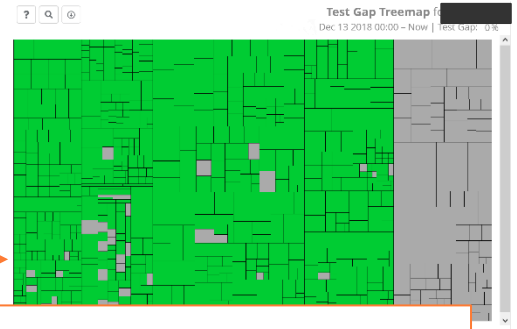






» Which components are unused during highway tests? «

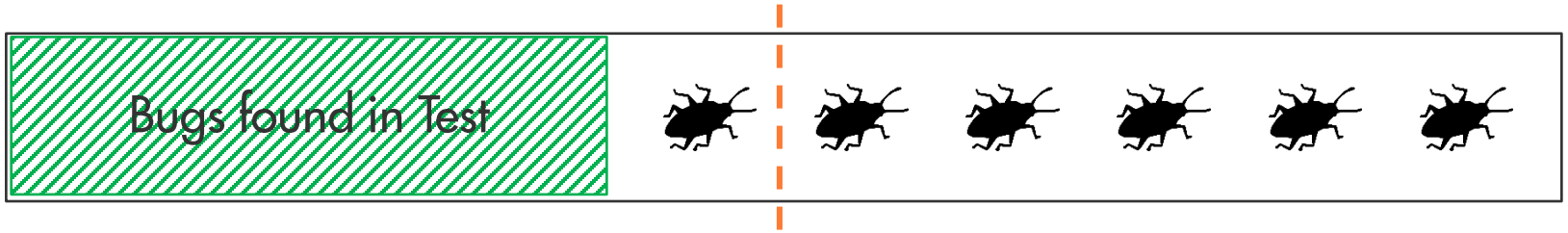




» Are we compliant with standards? «

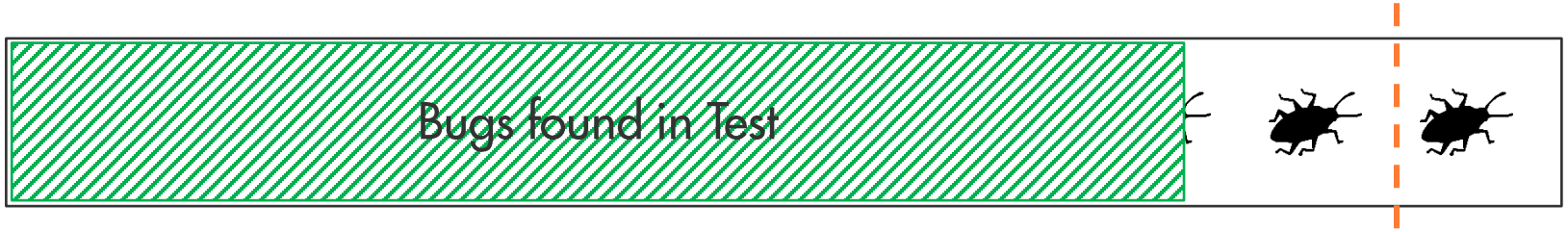
No Test-Gap-Analyse

%Remaining Bugs = 60%



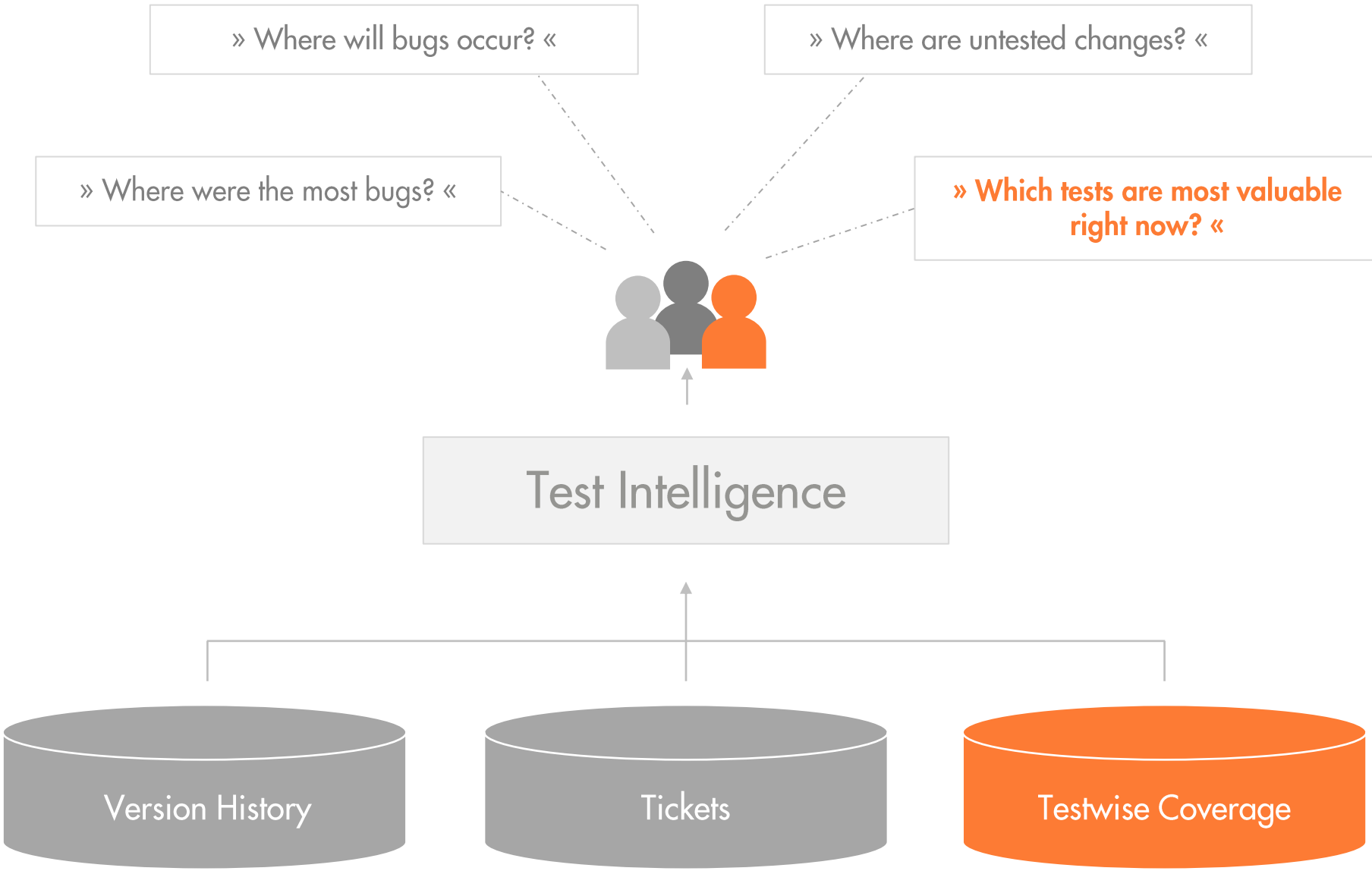
With Test-Gap-Analyse

%Remaining Bugs = 28%

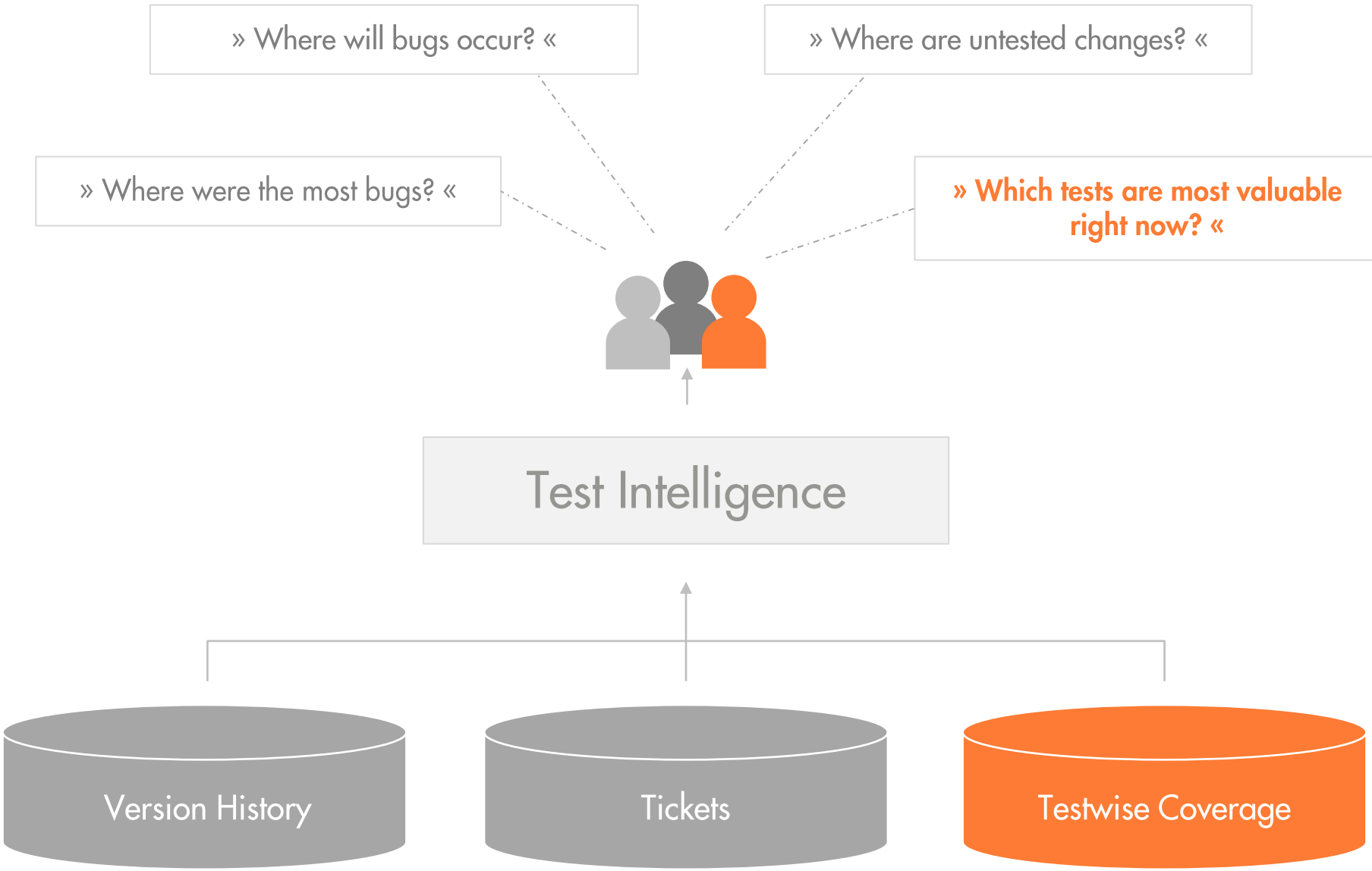


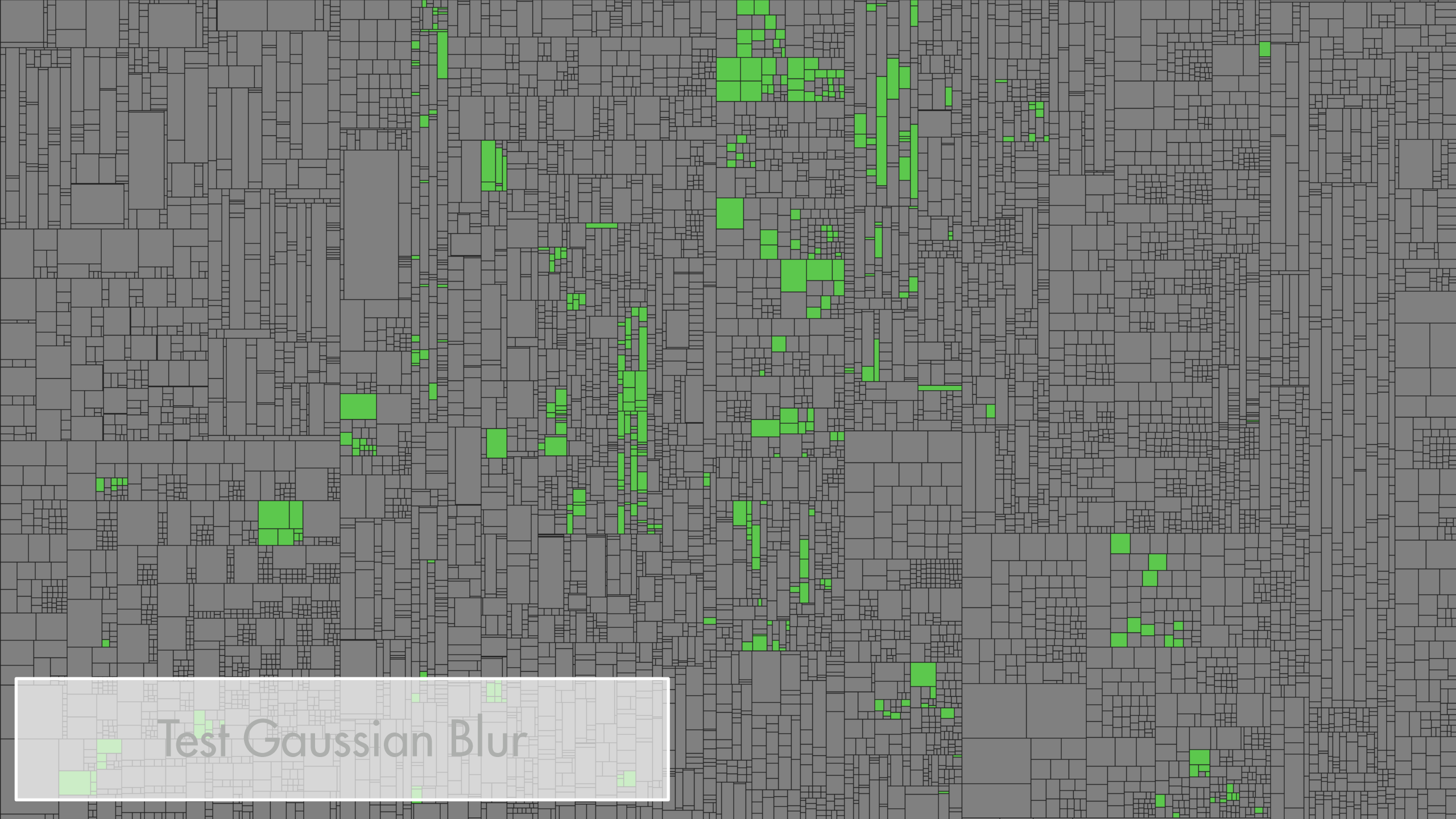
Reduced Production Bugs = **50%**

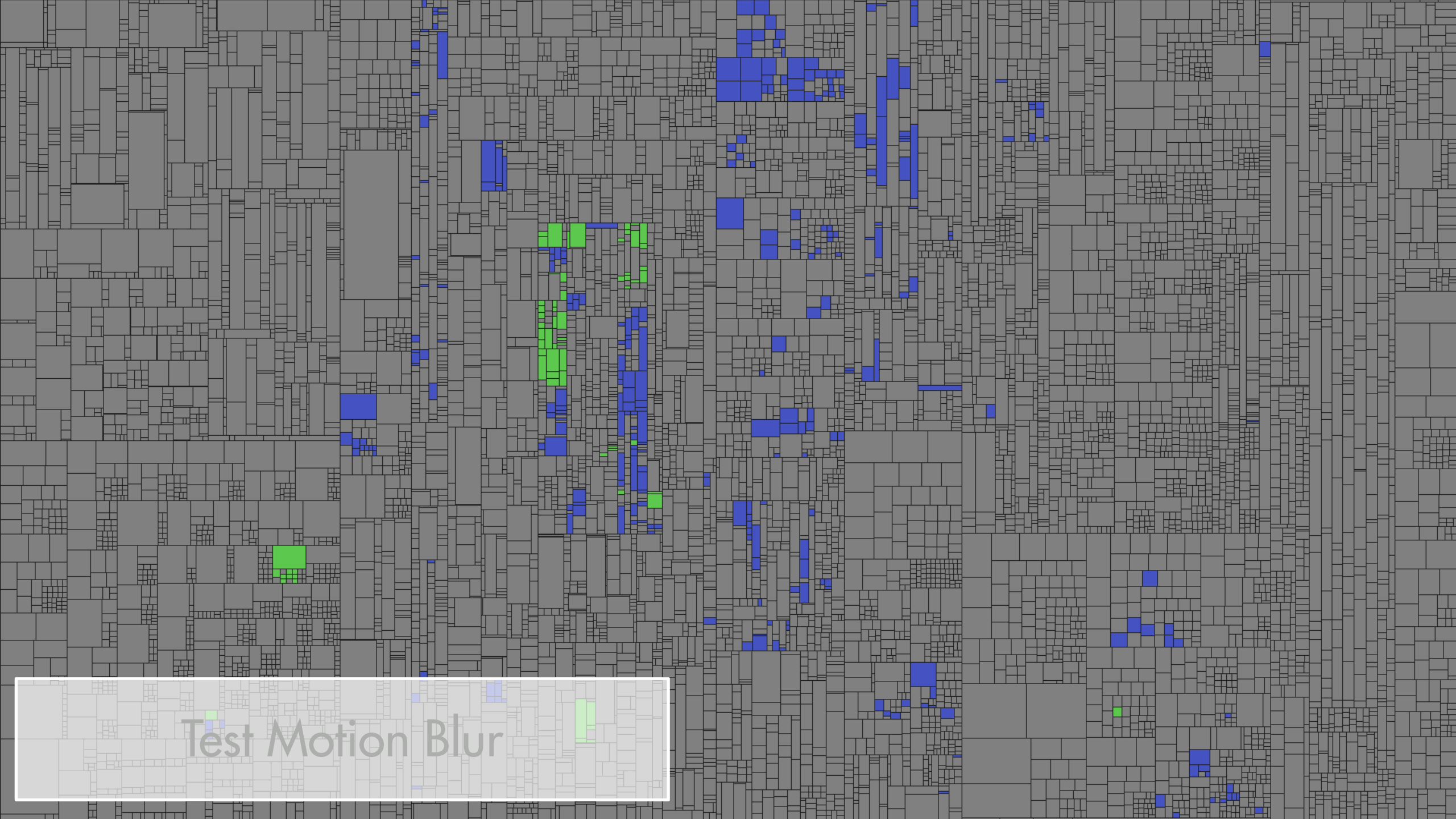
Test-Gap-Analysis reduced production bugs in the applications of Munich Re by ½

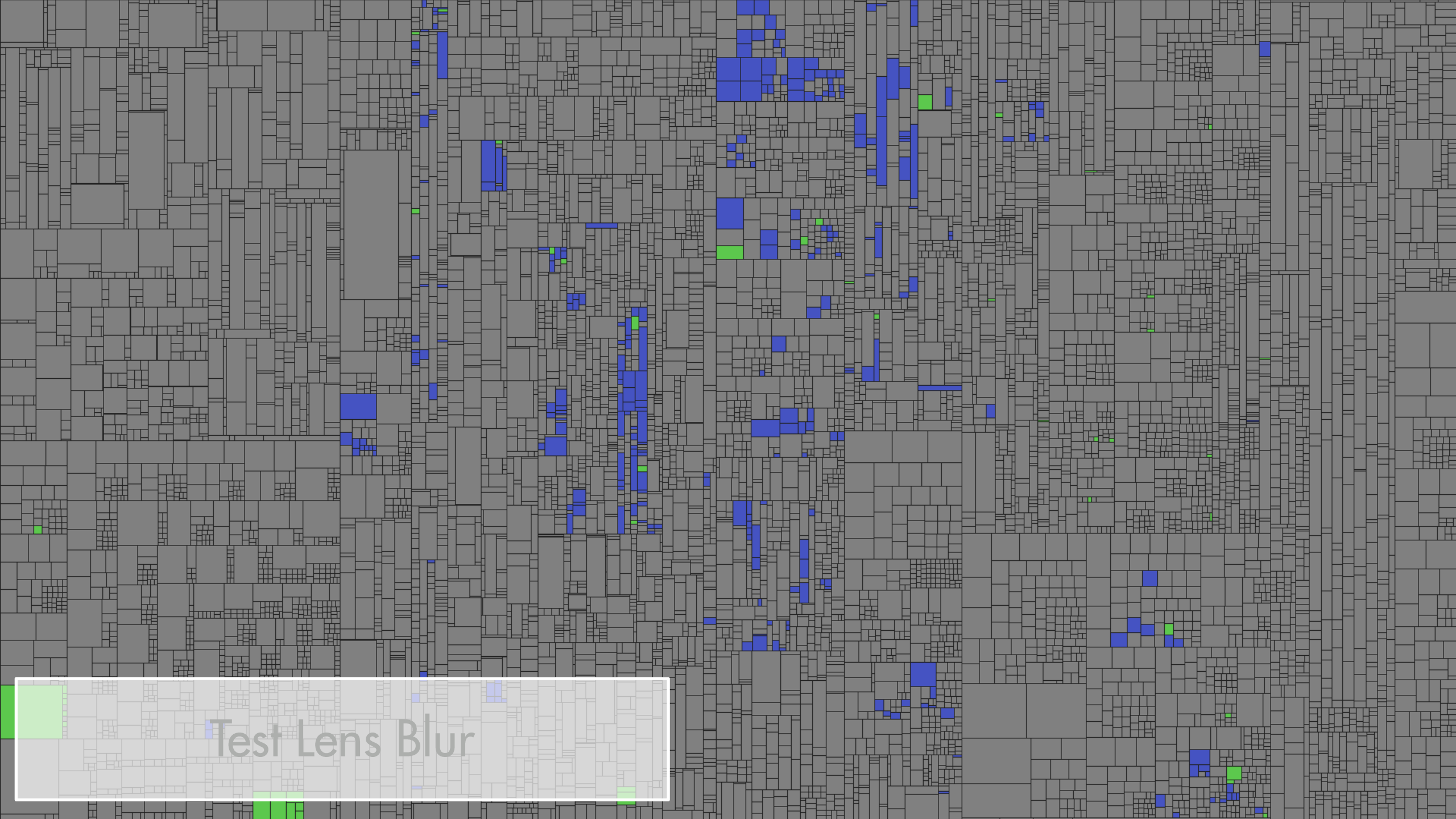

» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Testwise Coverage

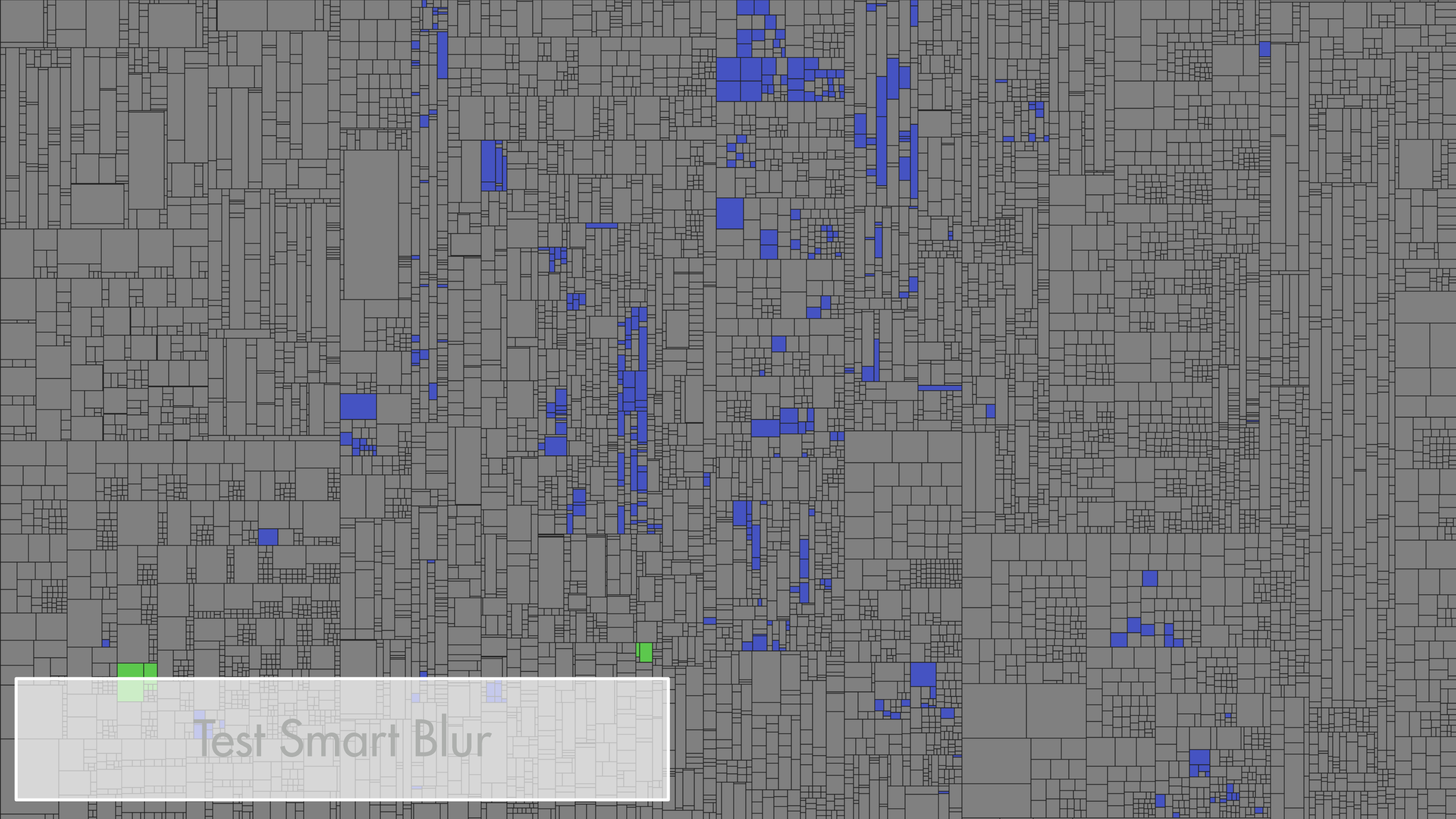
15 min Break

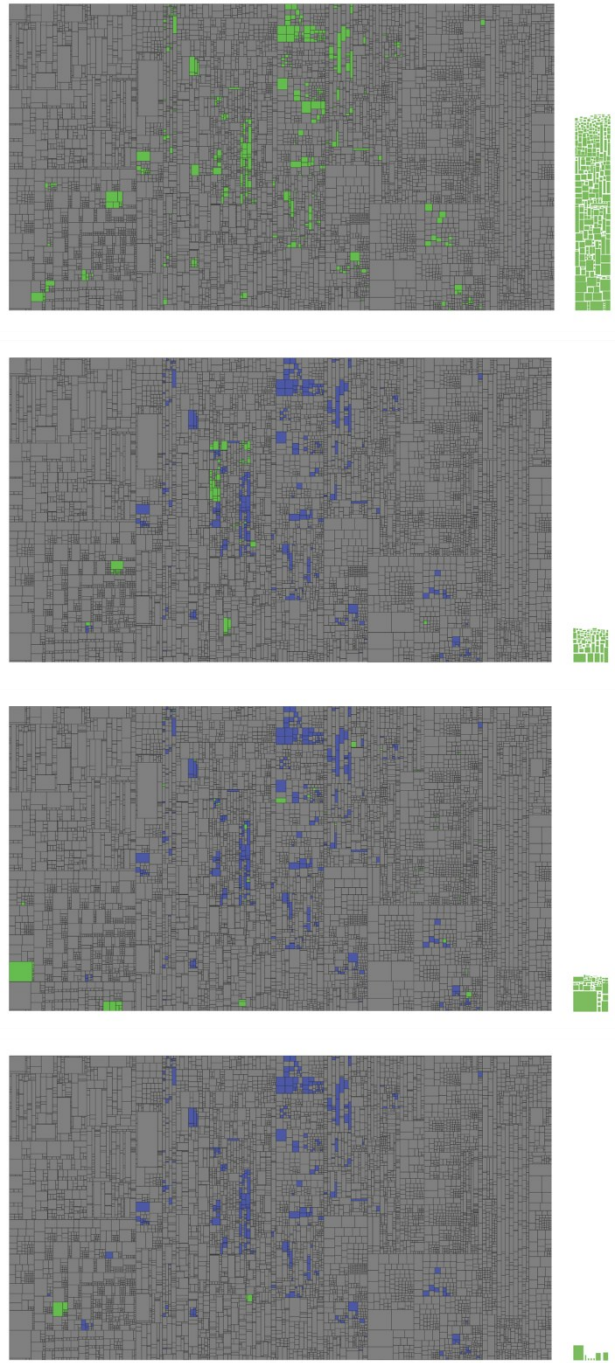

» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Testwise Coverage


Test Gaussian Blur

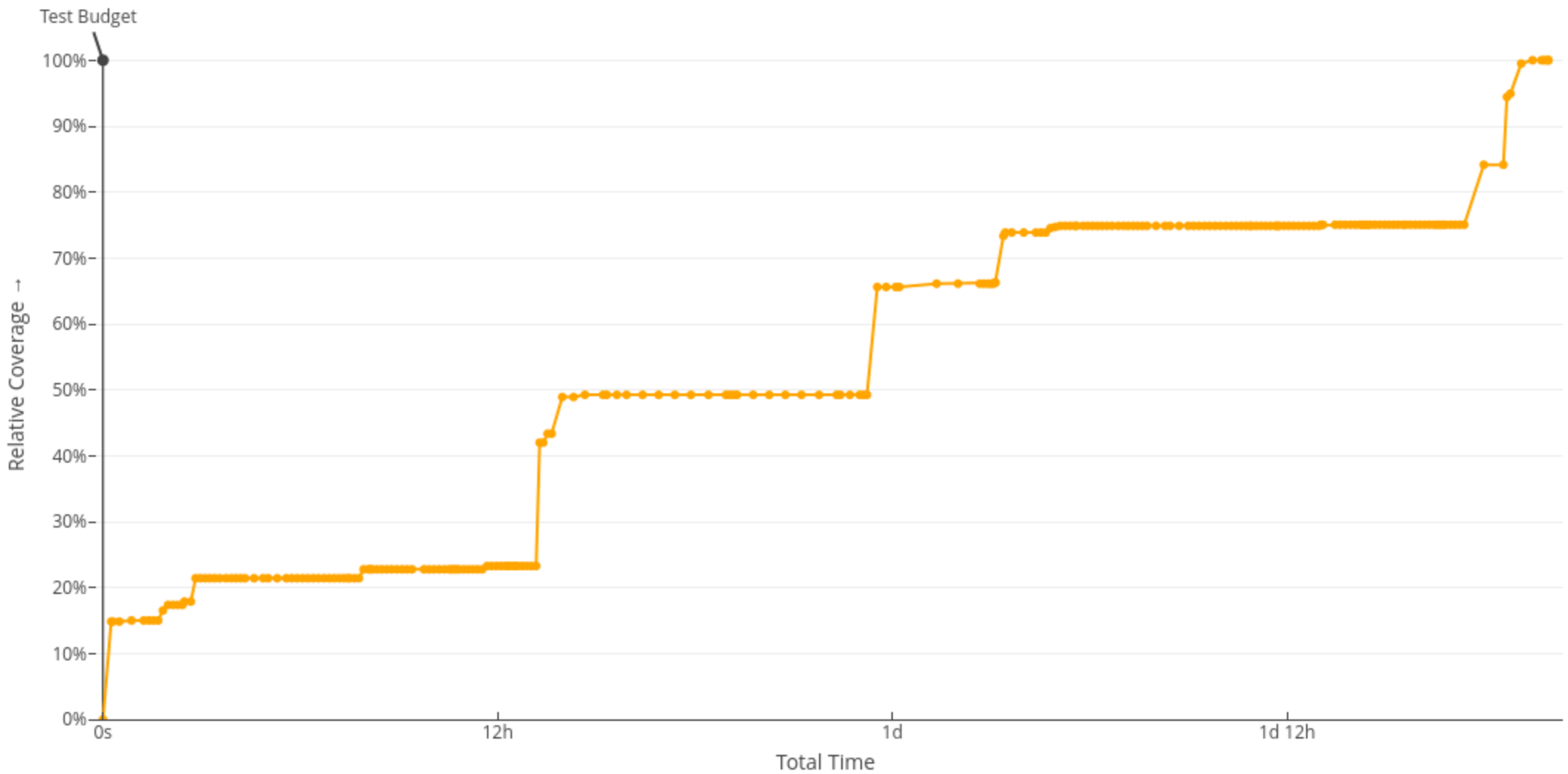

Test Motion Blur


Test Lens Blur


Test Smart Blur



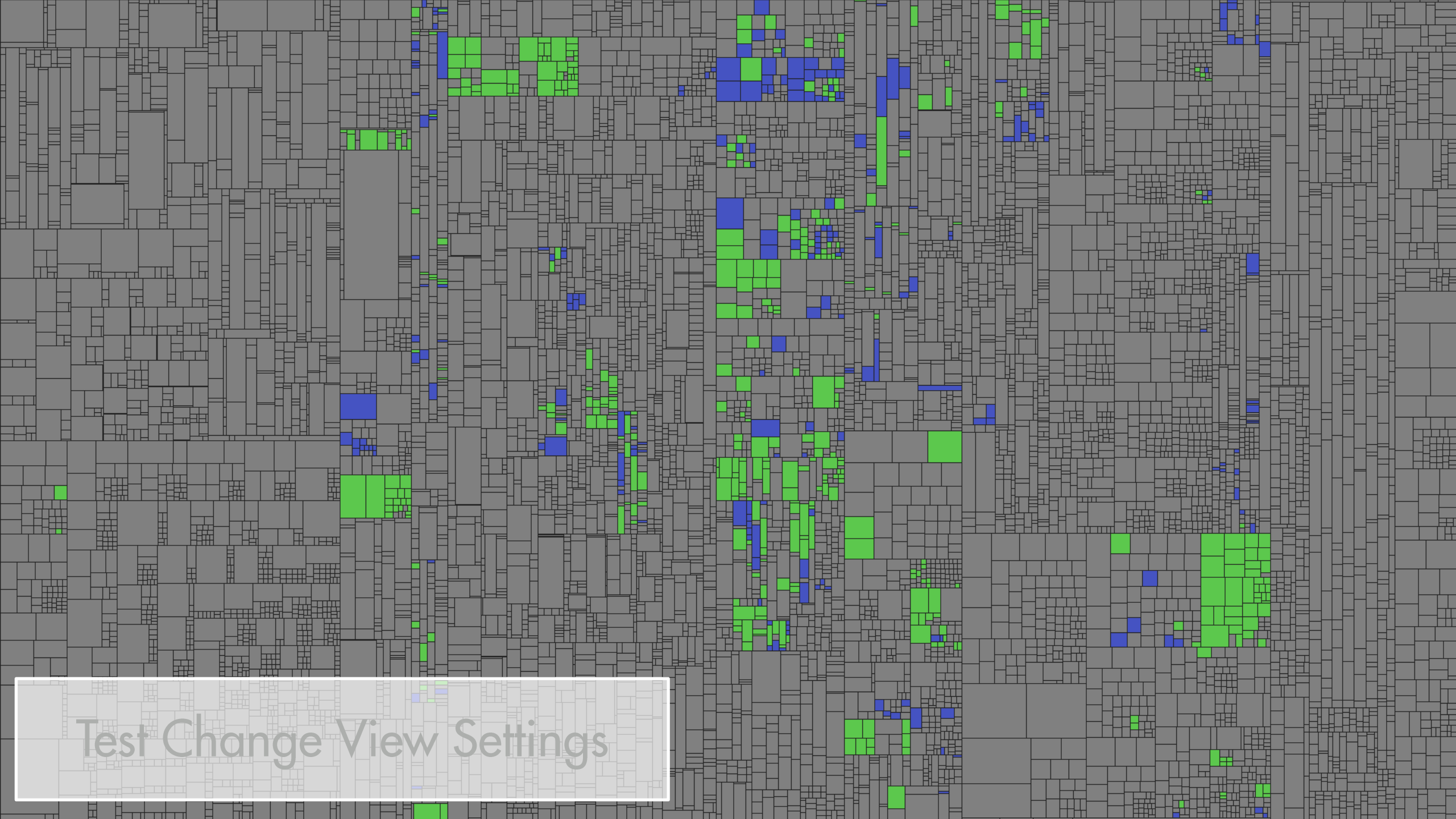
Coverage over Time

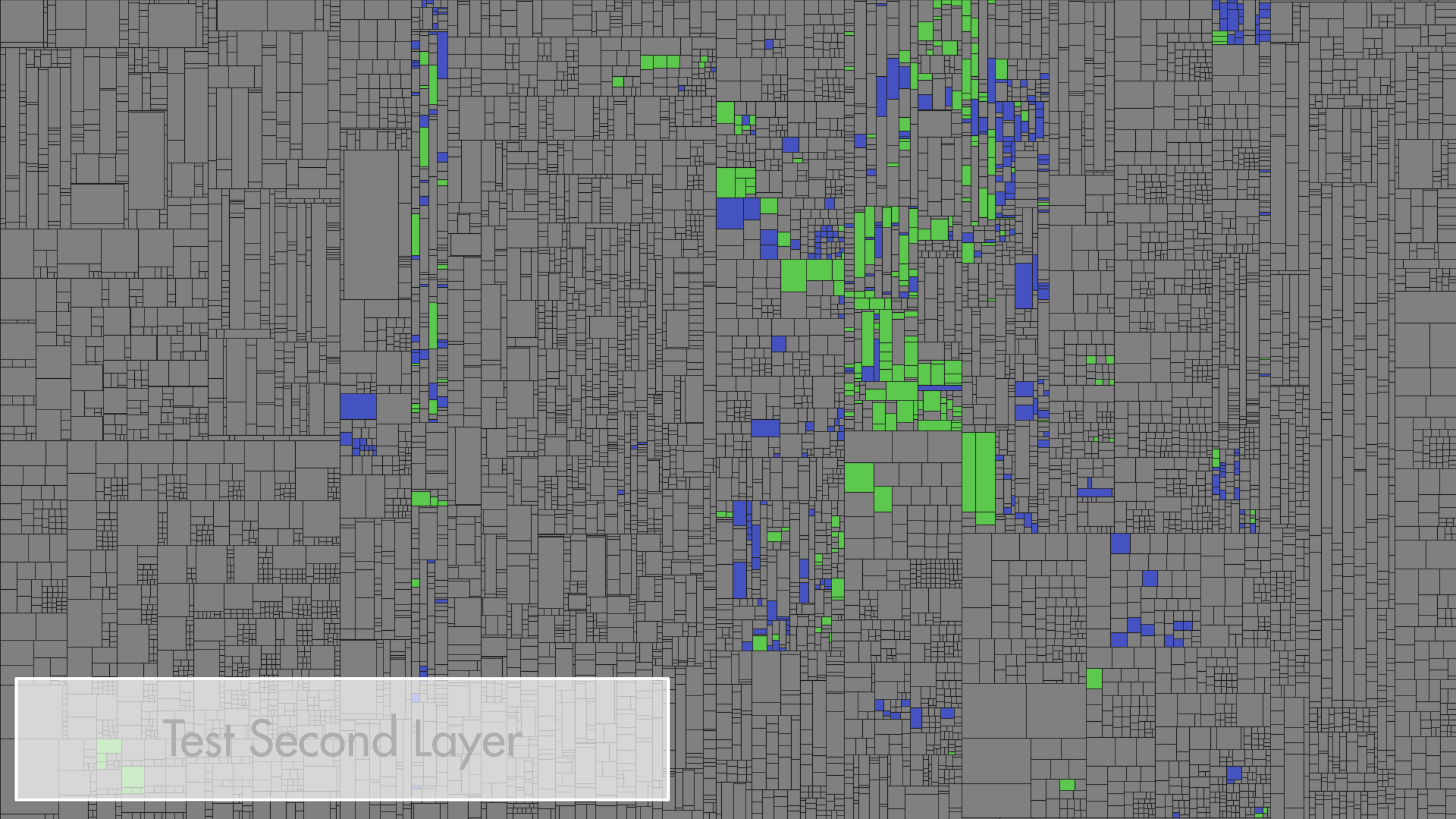


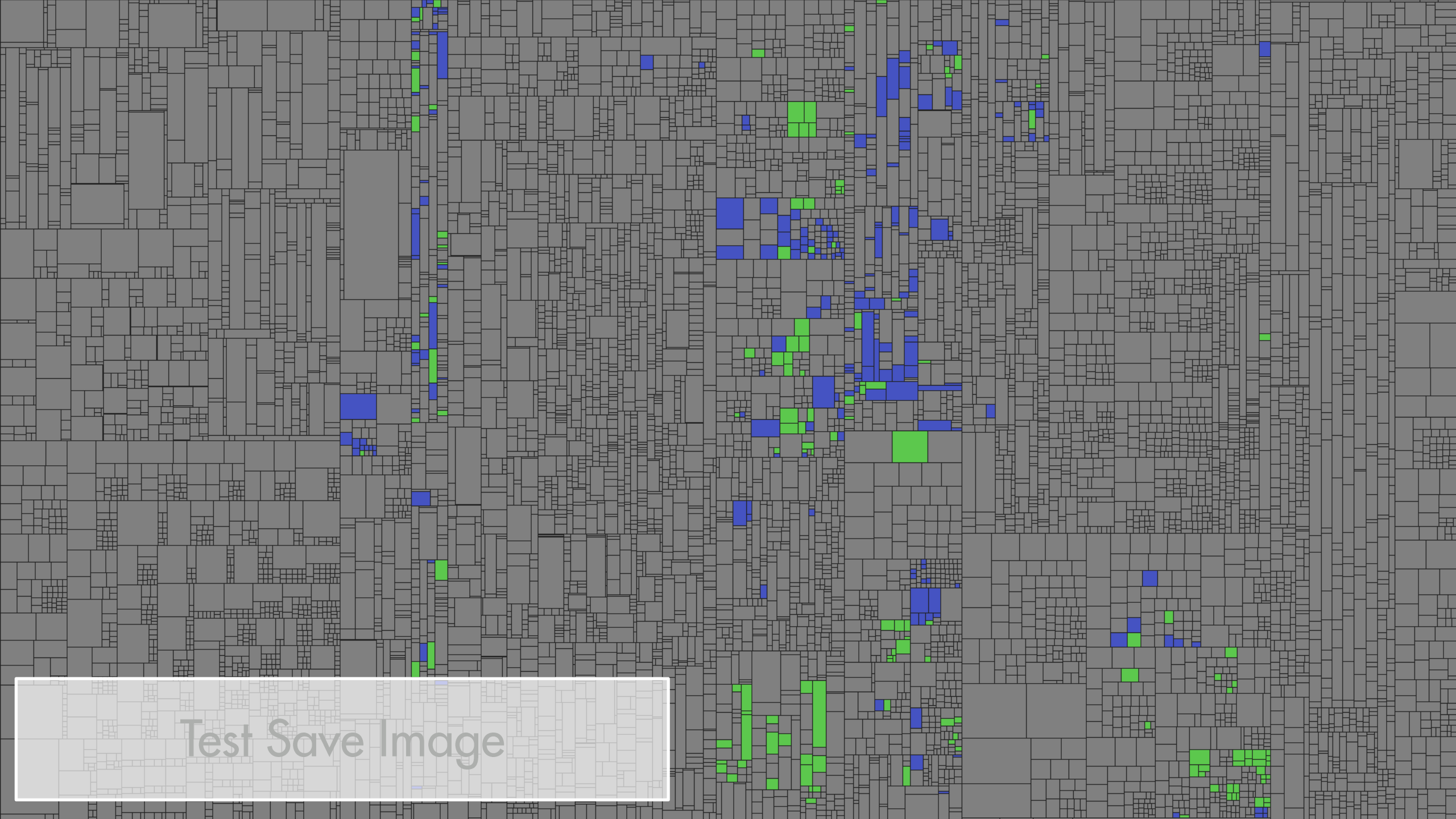
Results for Test Query & Budget Restriction

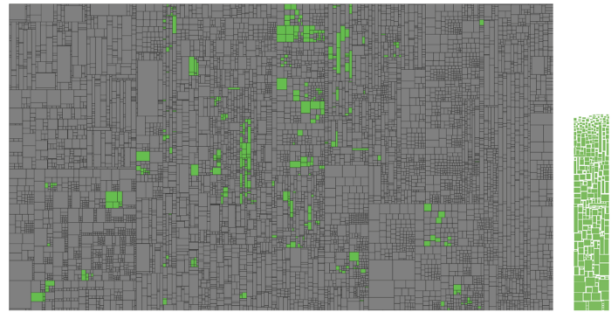
Relative Coverage: 0%, Selected Tests: 0 out of 236 (0%)

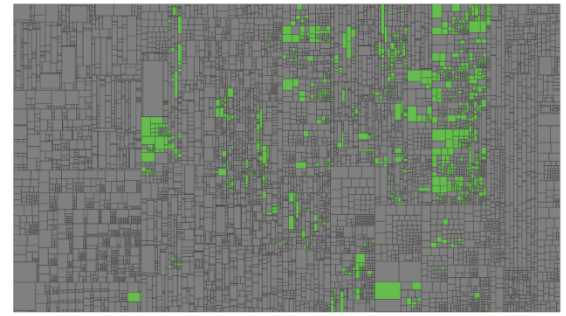

Test Create and Modify
Selection

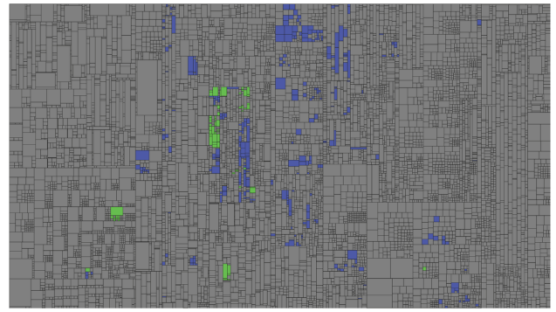

Test Change View Settings

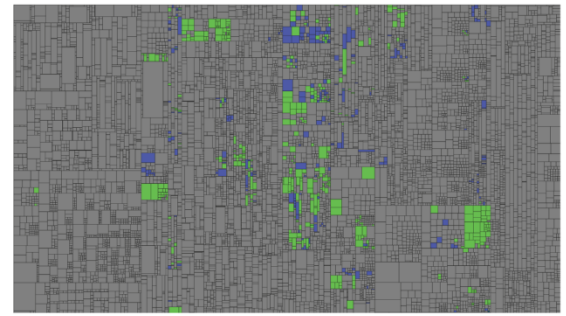

Test Second Layer

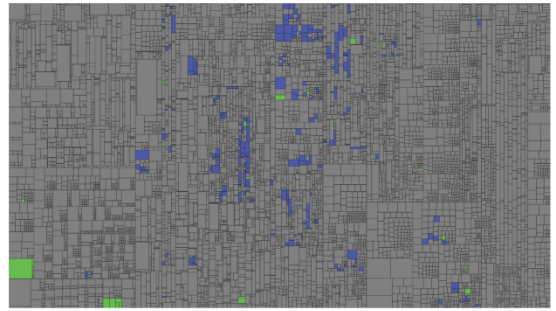

Test Save Image

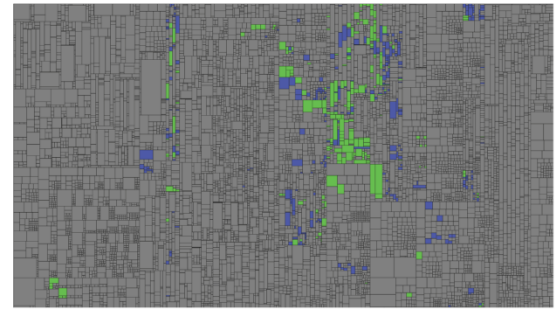


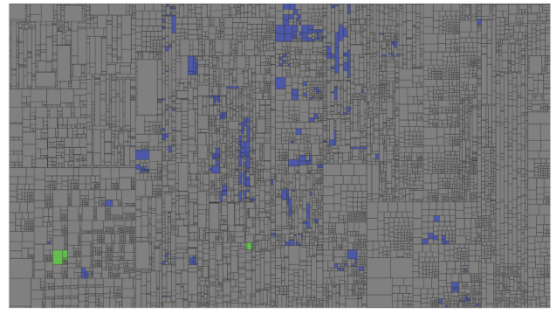


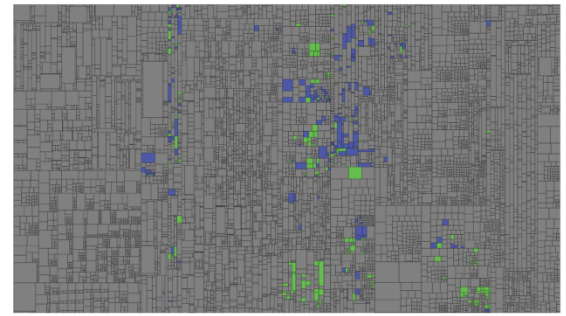




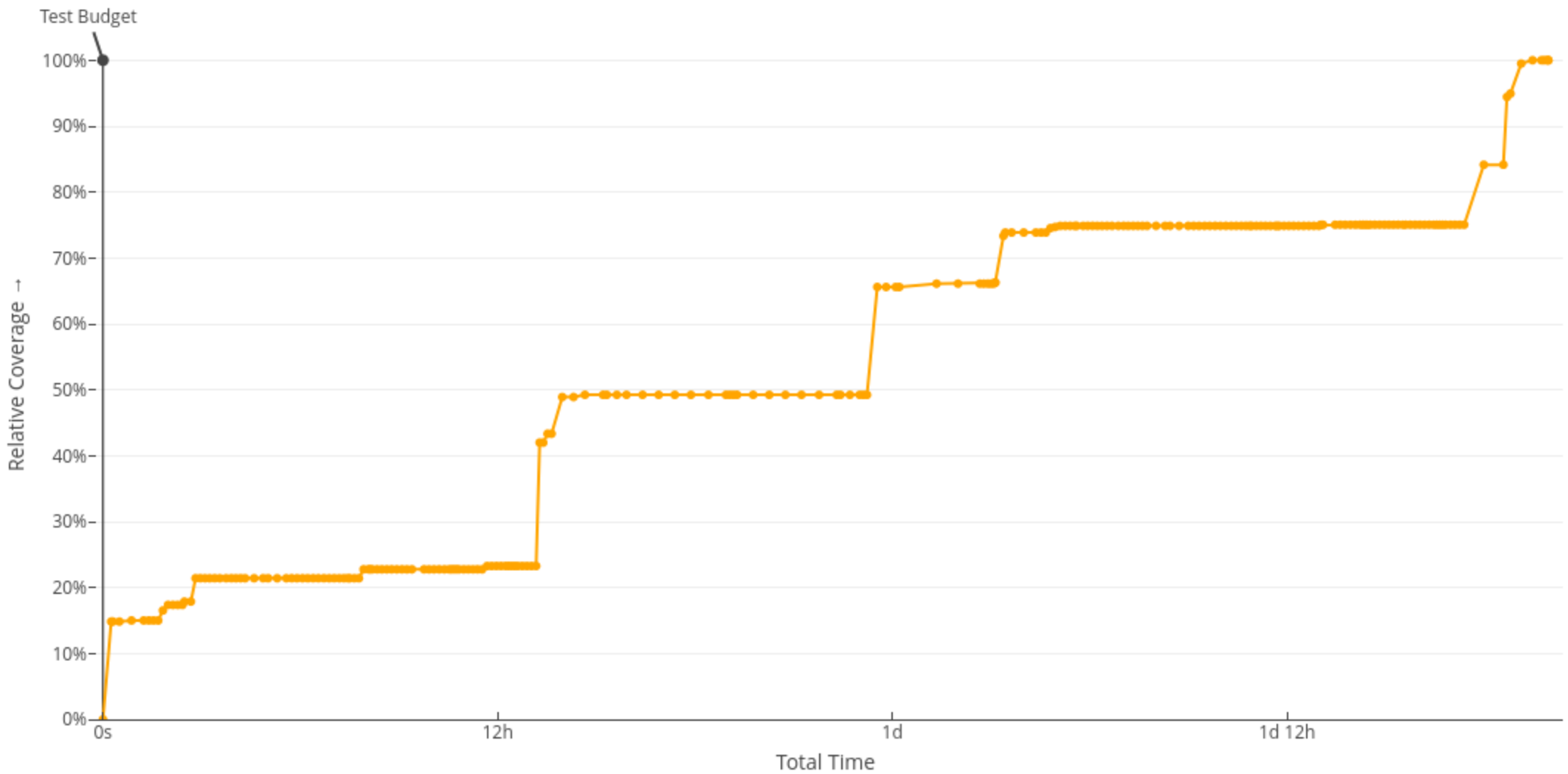








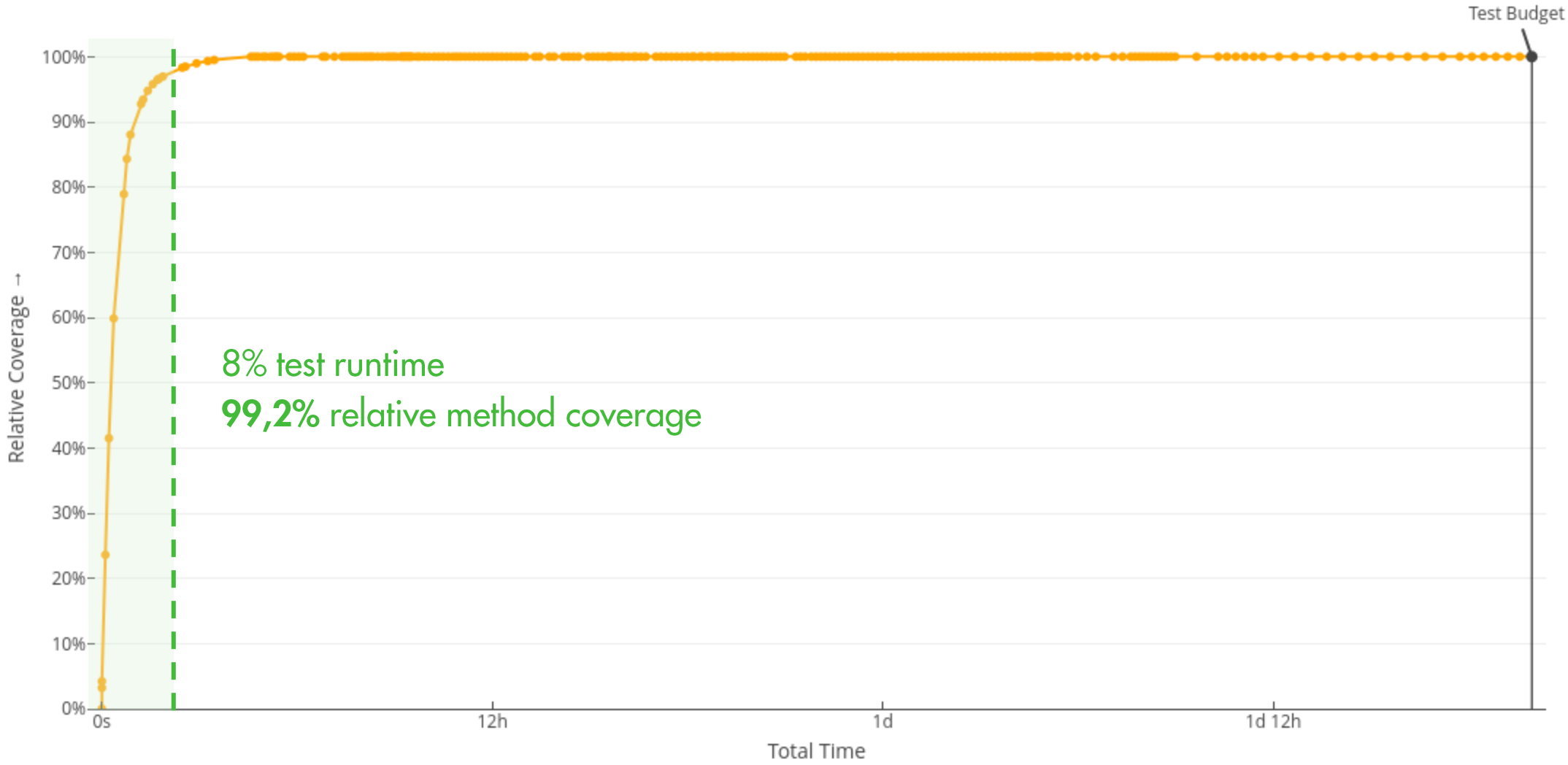
Coverage over Time



Results for Test Query & Budget Restriction

Relative Coverage: 0%, Selected Tests: 0 out of 236 (0%)

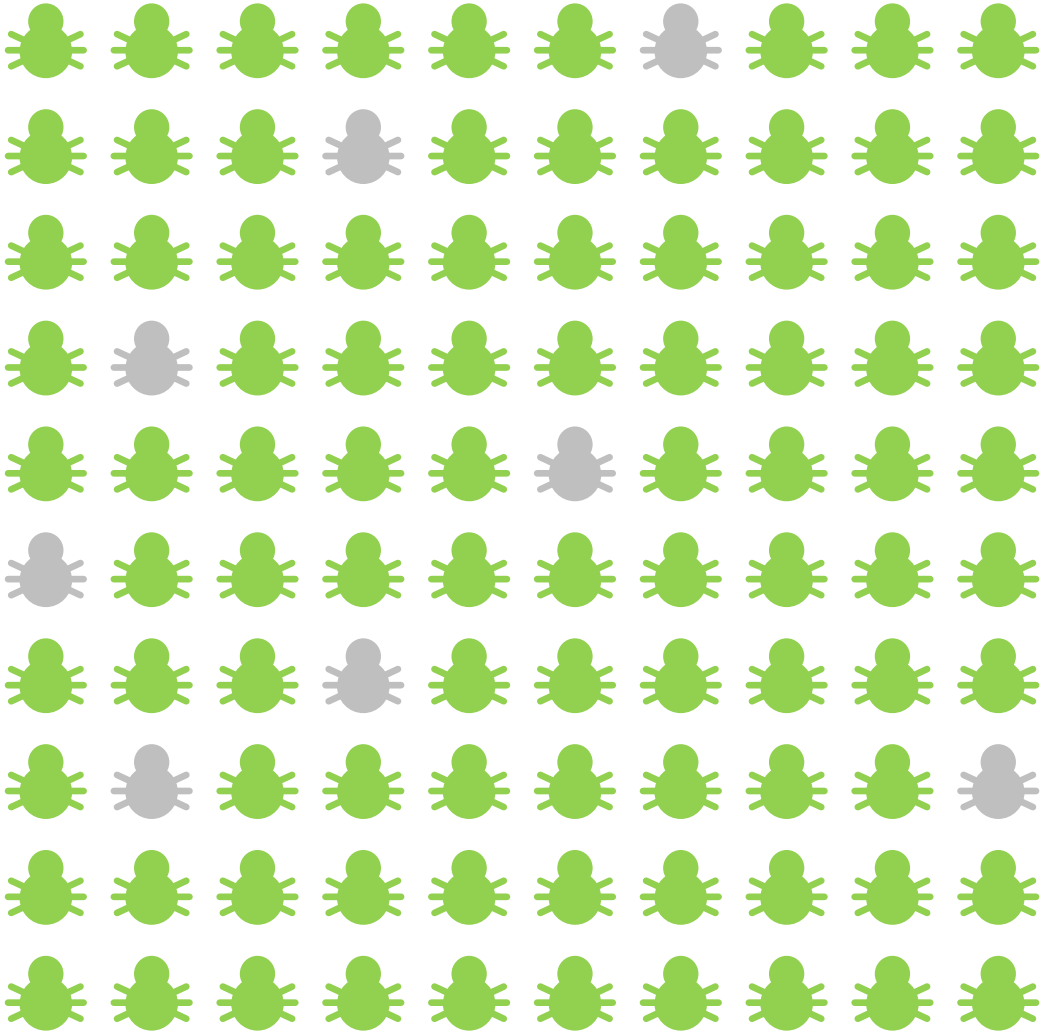
Coverage over Time



Results for Test Query & Budget Restriction

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

90 %



Industry Example

BVK Bayerische Versorgungskammer

Before: nightly „Re-run all“

Feedback late next day

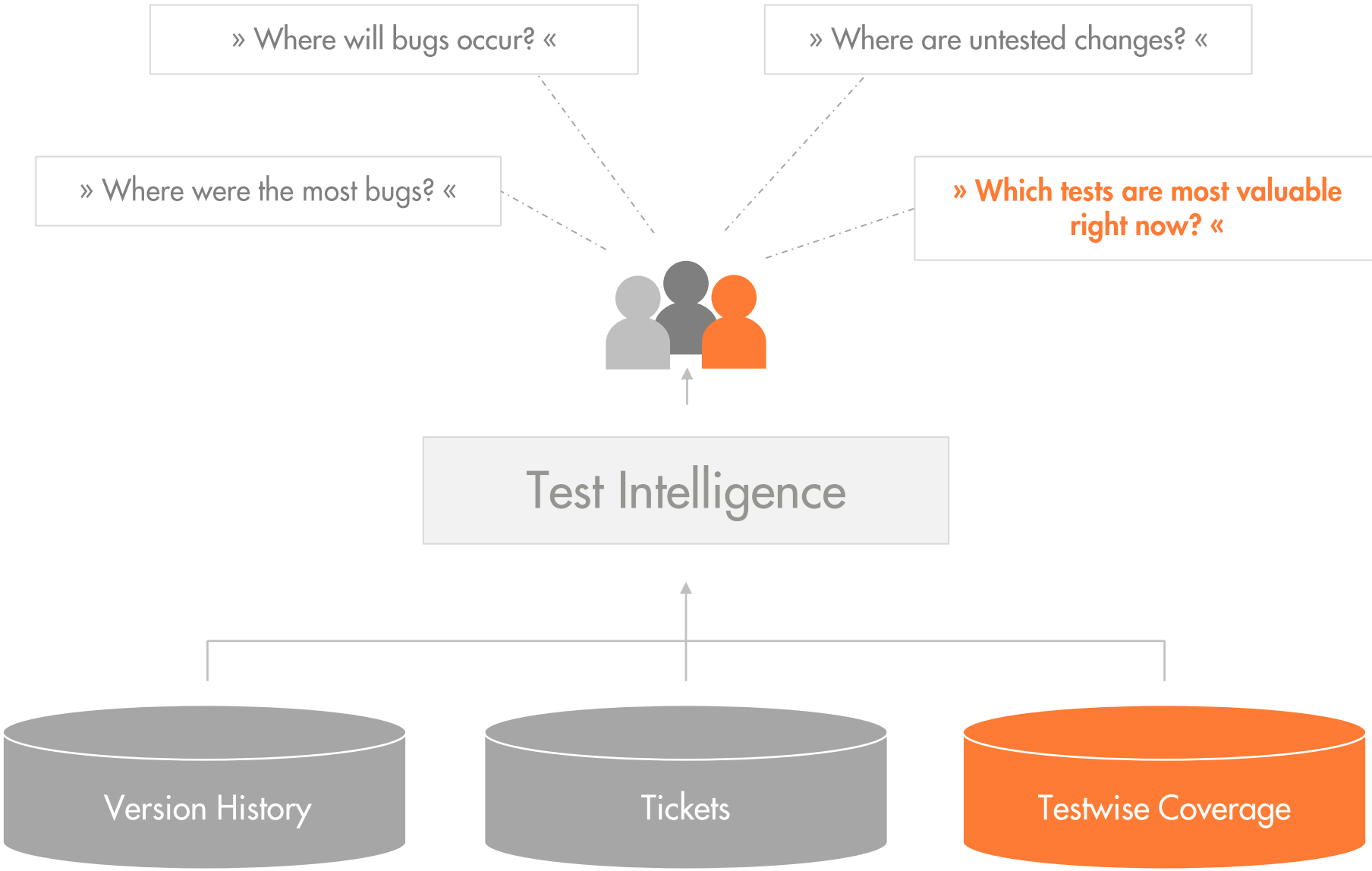
Build Test

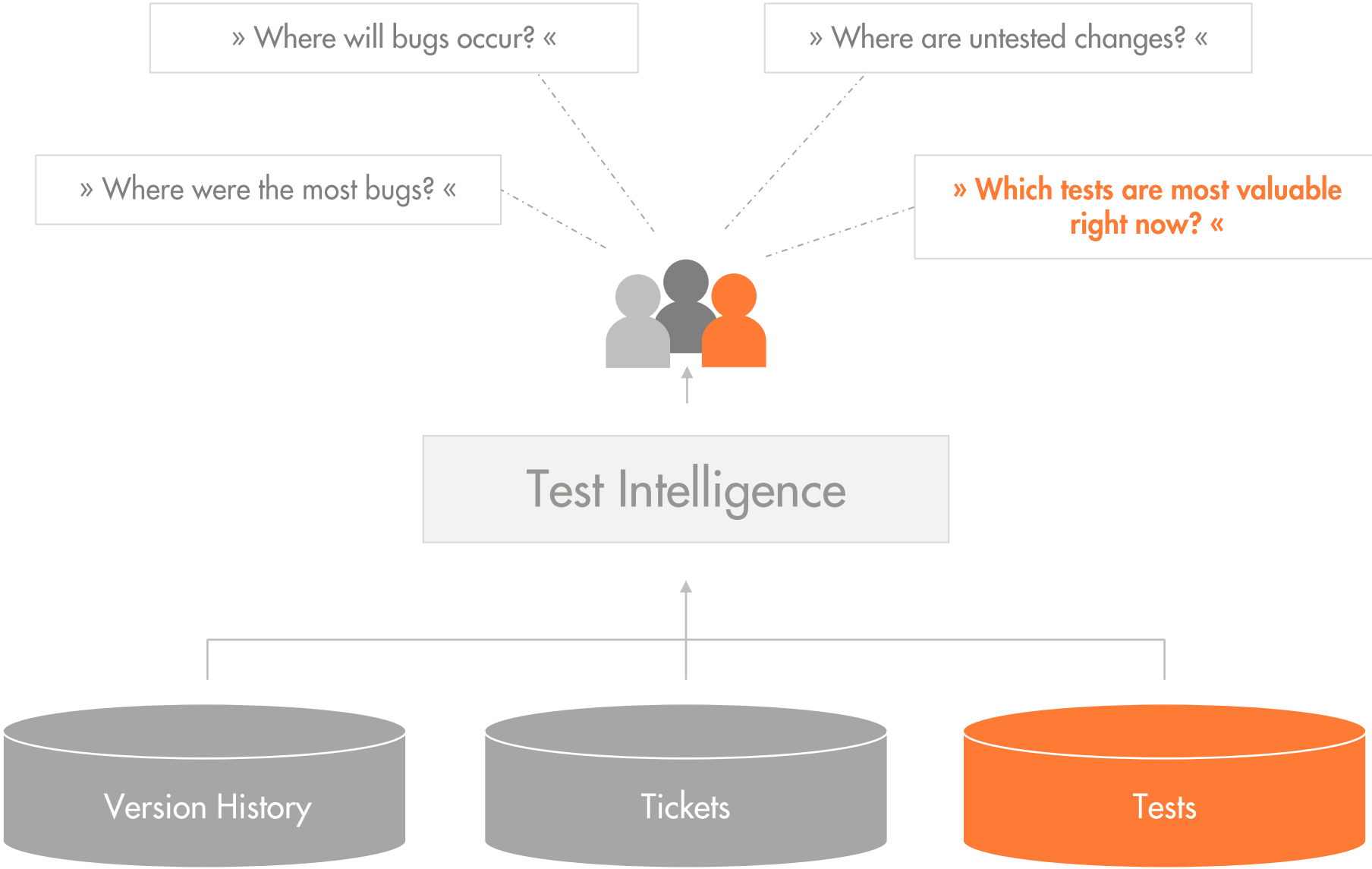
With only 5% test time budget, we find **70% of failing commits**

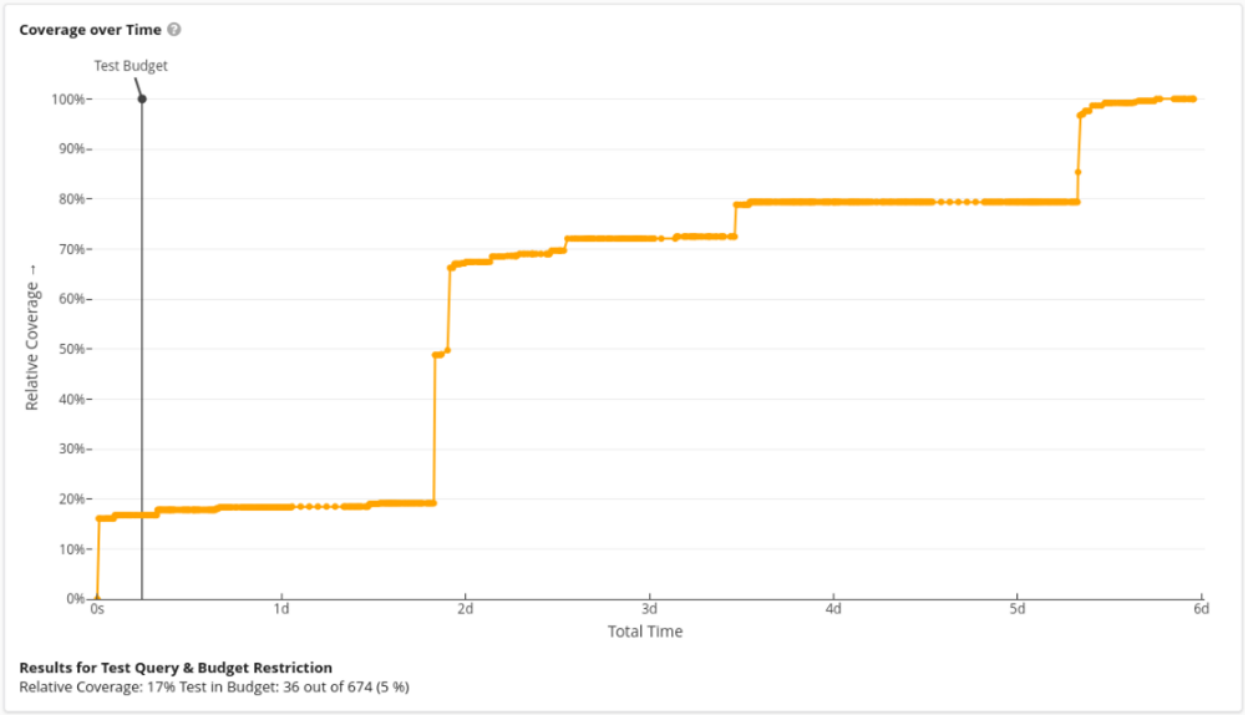
During the day 1h test budget or „pareto test list“

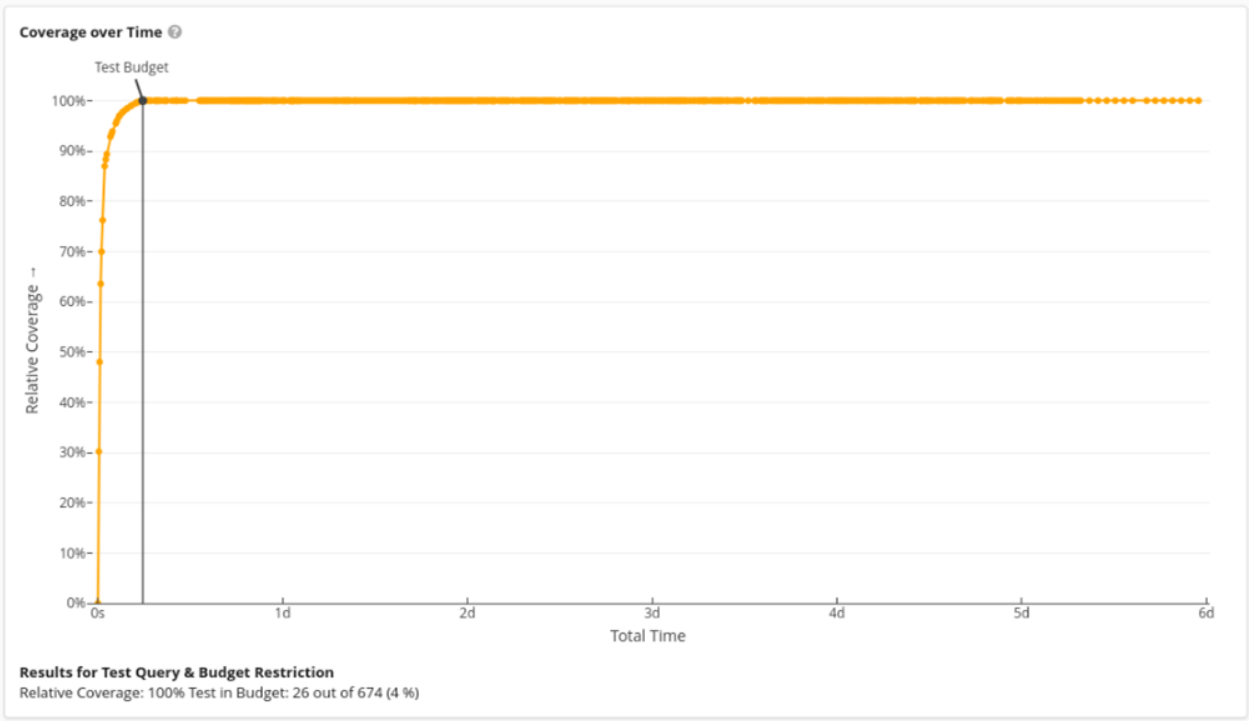
Build Test 1h

Nightly: „Re-run all“

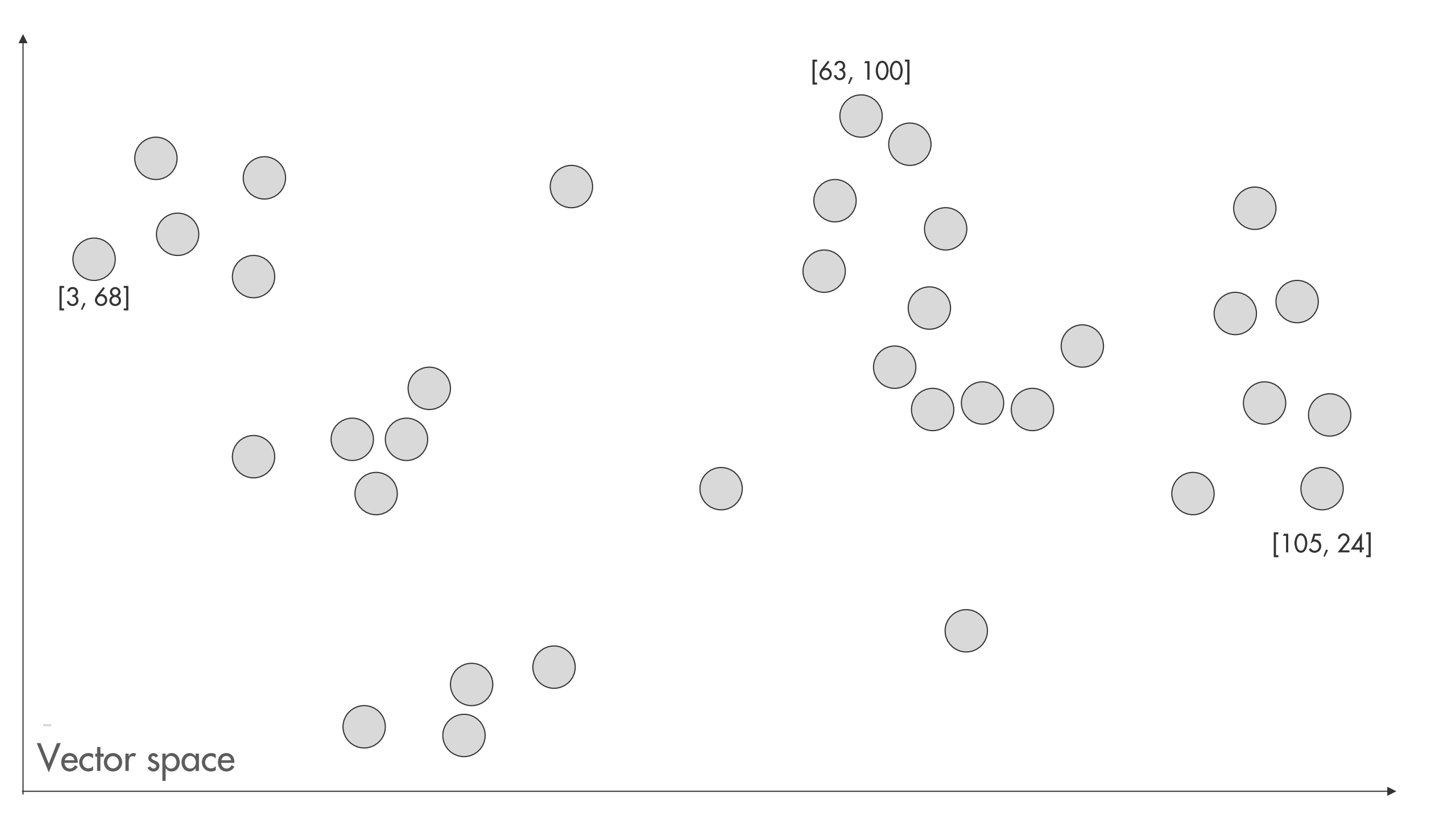

» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Testwise Coverage

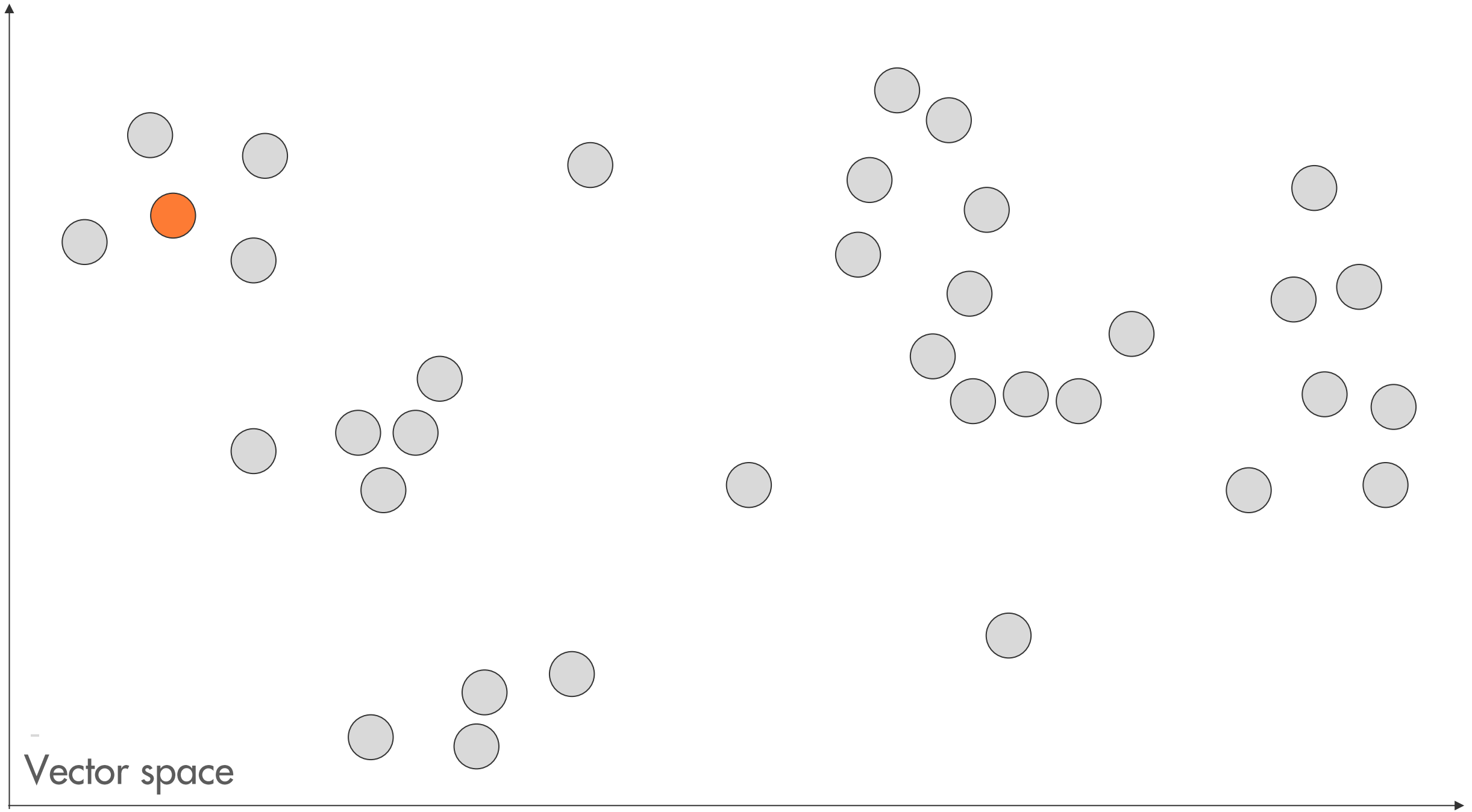

» Where will bugs occur? «
» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Tests

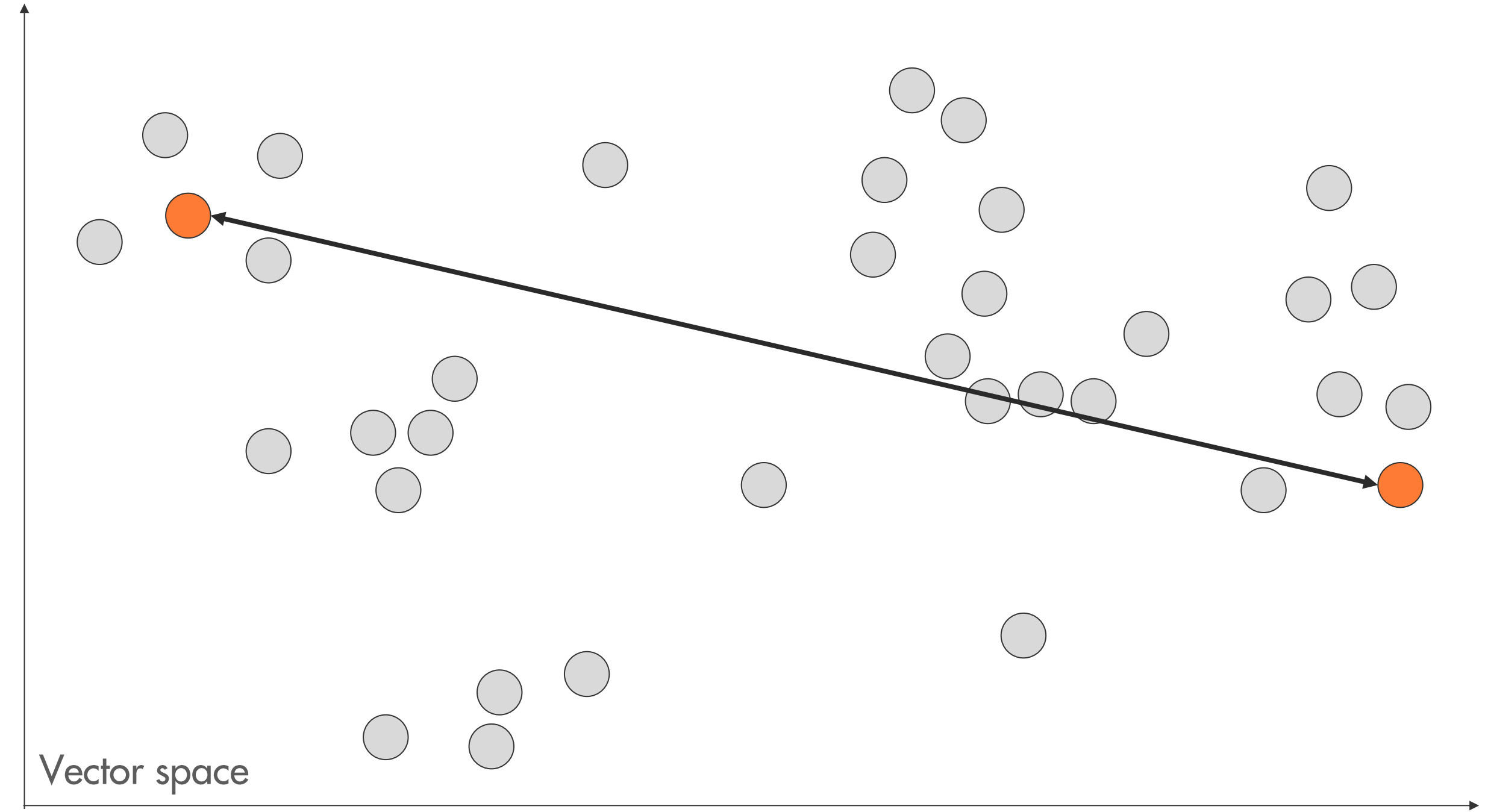


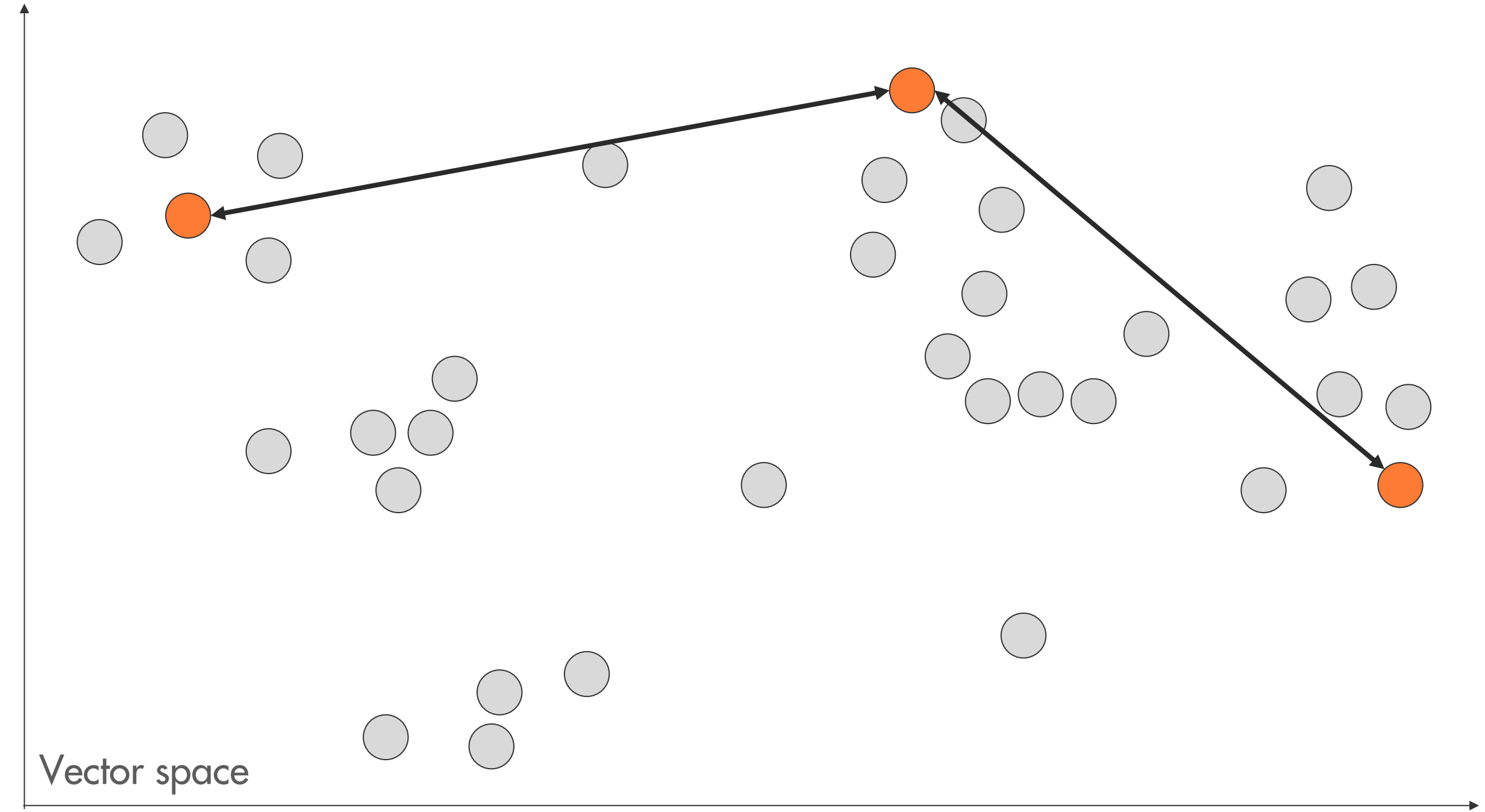


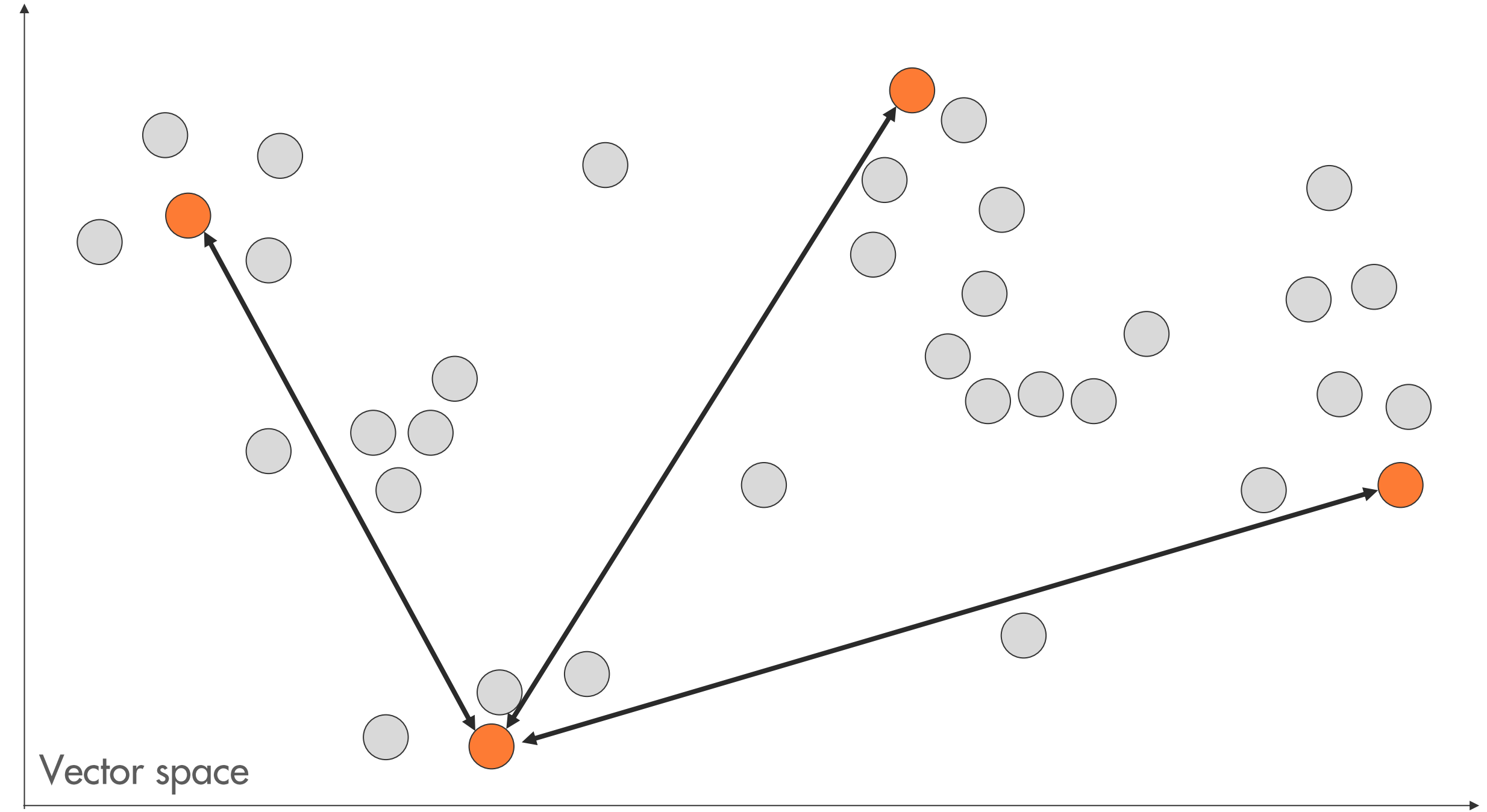
Idea: **Tests with dissimilar content**


[63, 100]
[3, 68]
[105, 24]
Vector space

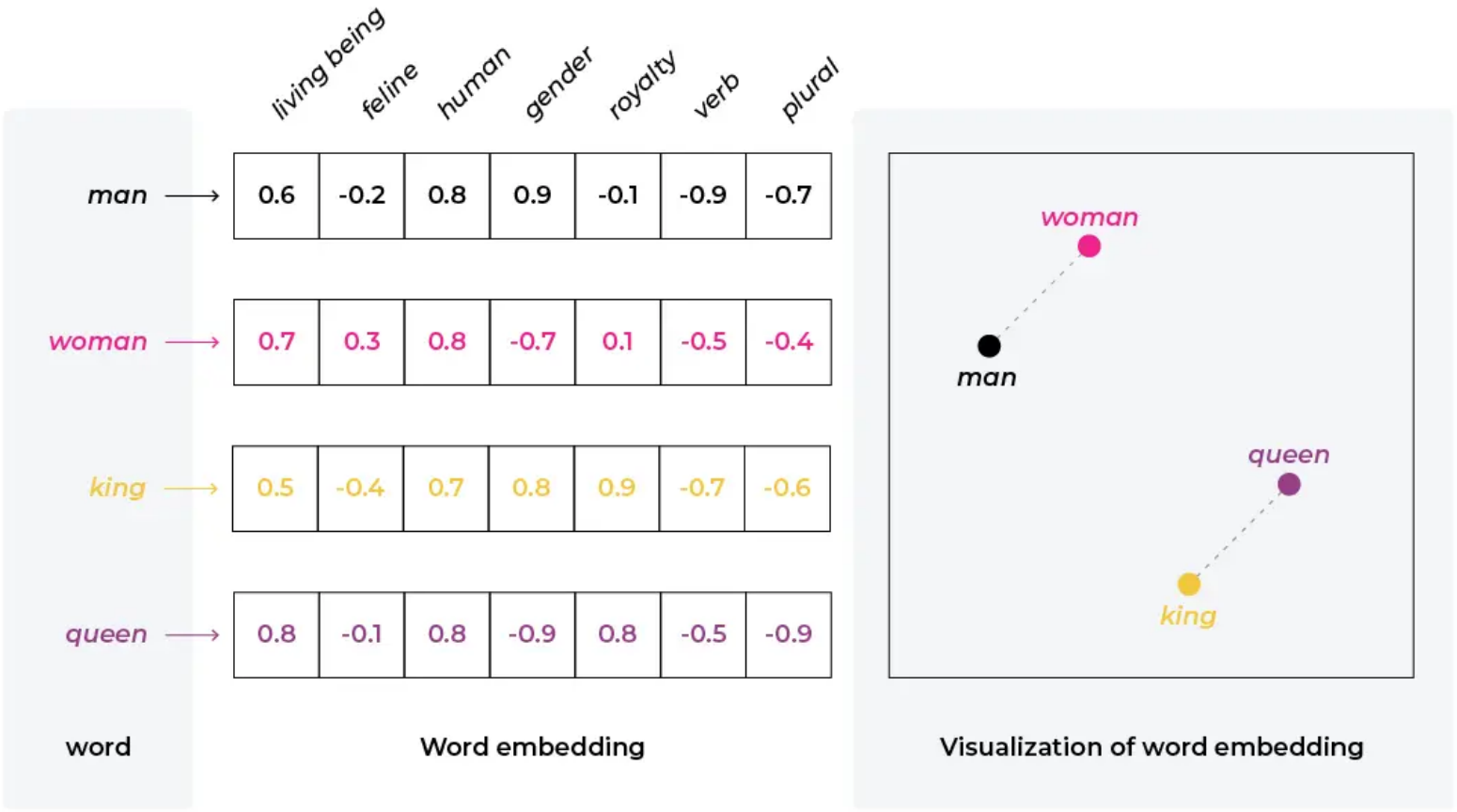

Vector space


Vector space


Vector space


Vector space

Large Language Models (AI)



An Evaluation of Distance Based Test Suite Reduction Techniques

Alessandro Escher
Technical University of Munich
Munich, Germany
alessandro.escher@tum.de

Raphael Nömmer
Technical University of Munich
Munich, Germany
noemmer@cqse.eu

***Abstract*—Efficient test suite selection is crucial in software testing due to the high cost of running extensive tests, particularly on large industry projects. Coverage-based techniques aim to maximize system execution within time constraints but often suffer from costly and complex coverage recording processes. This study explores alternative selection methods using test metadata and source code. Hierarchical Agglomerative Clustering (HAC) and a greedy approach were evaluated alongside distance measures based on package path distance and vector representations of test code.**

Evaluation on a variety of open-source projects and a large industry project revealed that while the proposed methods maintained decent coverage, they did not significantly outperform a strictly time-based selection. We note that HAC lacks a clear time-budget stopping criterion and performs worse than the greedy approach and random selection. Furthermore, techniques that rely on execution times tend to neglect longer-running tests, which can have an impact on fault detection, particularly in industry projects.

This study emphasizes the importance of effective test selection methods that balance coverage, cost, and fault detection. We suggest that a simple yet effective baseline such as lowest execution time first is a more robust baseline than a random selection, especially for a cost based evaluation, and underline the need for more competitive baseline methods in test suite optimization research.

***Index Terms*—test selection, test suite reduction, clustering, code embeddings, topic model**

I. Introduction

Software testing is an integral part of the software development lifecycle of any application. In order to validate that the program works as intended and provides the required functionality, a suite of tests is run—each focusing on different components of the system and at differing granularities—at various points in time before the software is released. Regression testing is a popular approach for this. The test suite is run at different intervals, depending on the size of the suite and requirements of the project. Most often this is done whenever a change is made to the system as this is typically where faults are introduced [1]. For large industry systems where test suites can reach hours or days of execution time, this takes up a significant amount of resources [2]–[6], causing additional costs for the company and resulting in slower feedback for the developers. Test Case Selection (TCS) aims to alleviate these issues by selecting a subset of the test suite, picking relevant tests and omitting redundant ones. Many TCS approaches rely on the test coverage—be that at the statement, branch or method level—of the test suite in order to determine which tests to choose. Recording and storing this coverage data can become a cumbersome process, especially for large and complex software systems that use multiple programming languages and frameworks [7]. Because of this, a company will have to struggle with the high cost and maintenance effort, and may only decide to do adopt this approach in a limited manner [8]. Being able to use an alternative approach that is not based on coverage data but instead uses readily available data would allow for TCS to be performed on all projects, no matter their priority. Additionally, it would allow the developers of a project to gain immediate benefits of TCS in case the coverage recording process is not set up yet.

In this study we focus on exploring alternative approaches to coverage-based test suite selection, aiming to address the challenges associated with the expense and complexity of traditional methods. Specifically, we investigate the feasibility of using test metadata and source code for a more efficient test selection. We examine a clustering and a greedy approach in conjunction with various distance measures based on package path distance and vector representations of test code. The practical effectiveness of these techniques in maintaining coverage and detecting faults is evaluated across a variety of open source projects as well as a large industry project.

The rest of this research is structured as follows. Section II gives background information about some of the techniques and concepts used. In Section III, we explain our TCS approaches and the different combination of parameters that we apply. Afterwards in Section IV we detail our empirical evaluation of our proposed implementation and lastly, we offer our concluding thoughts in Section V.

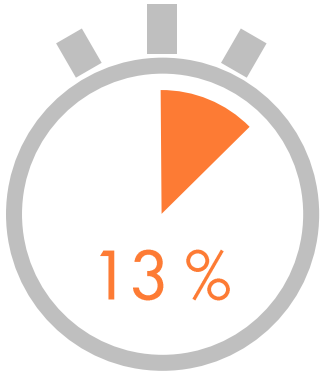
II. Related Work & Background

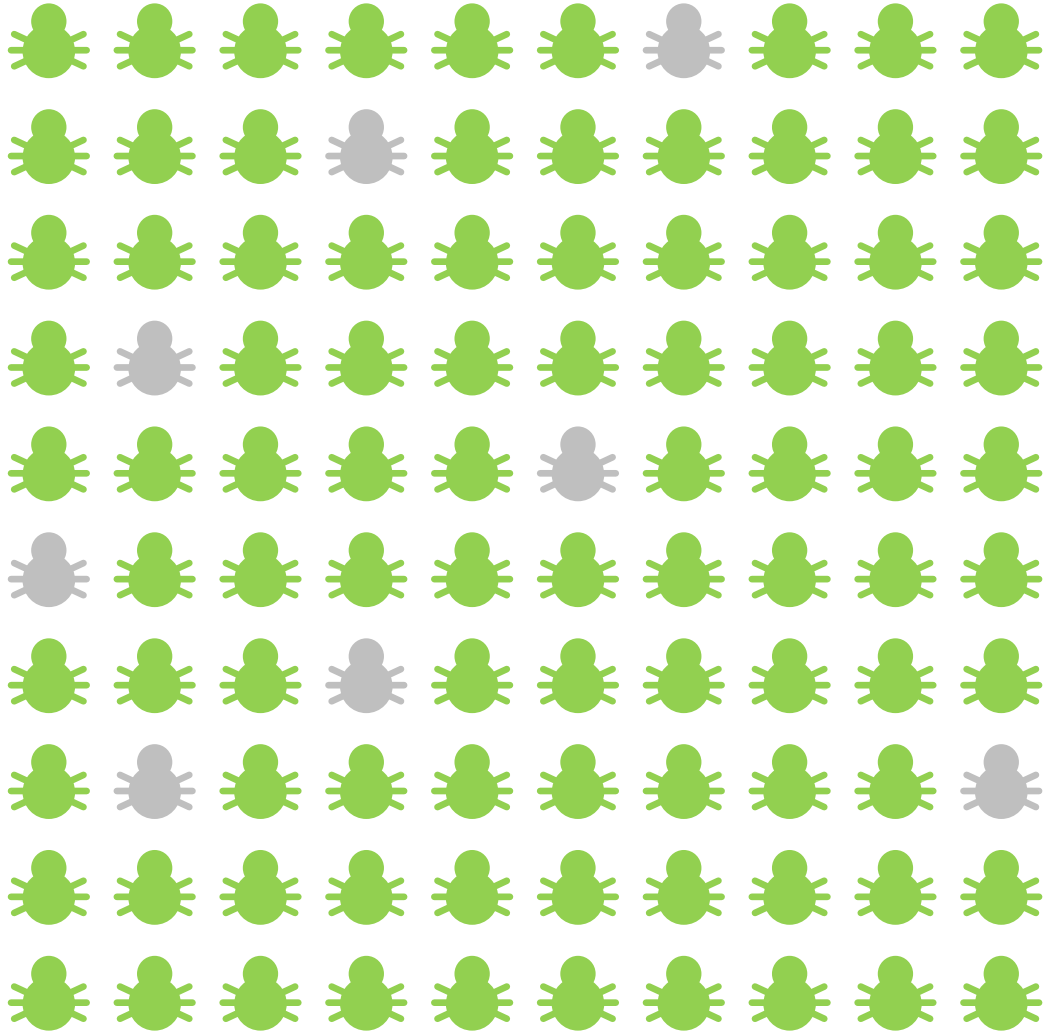
This section gives background information about the concept of test selection and some of the techniques that were used and offers insight into how they have been applied in related works.

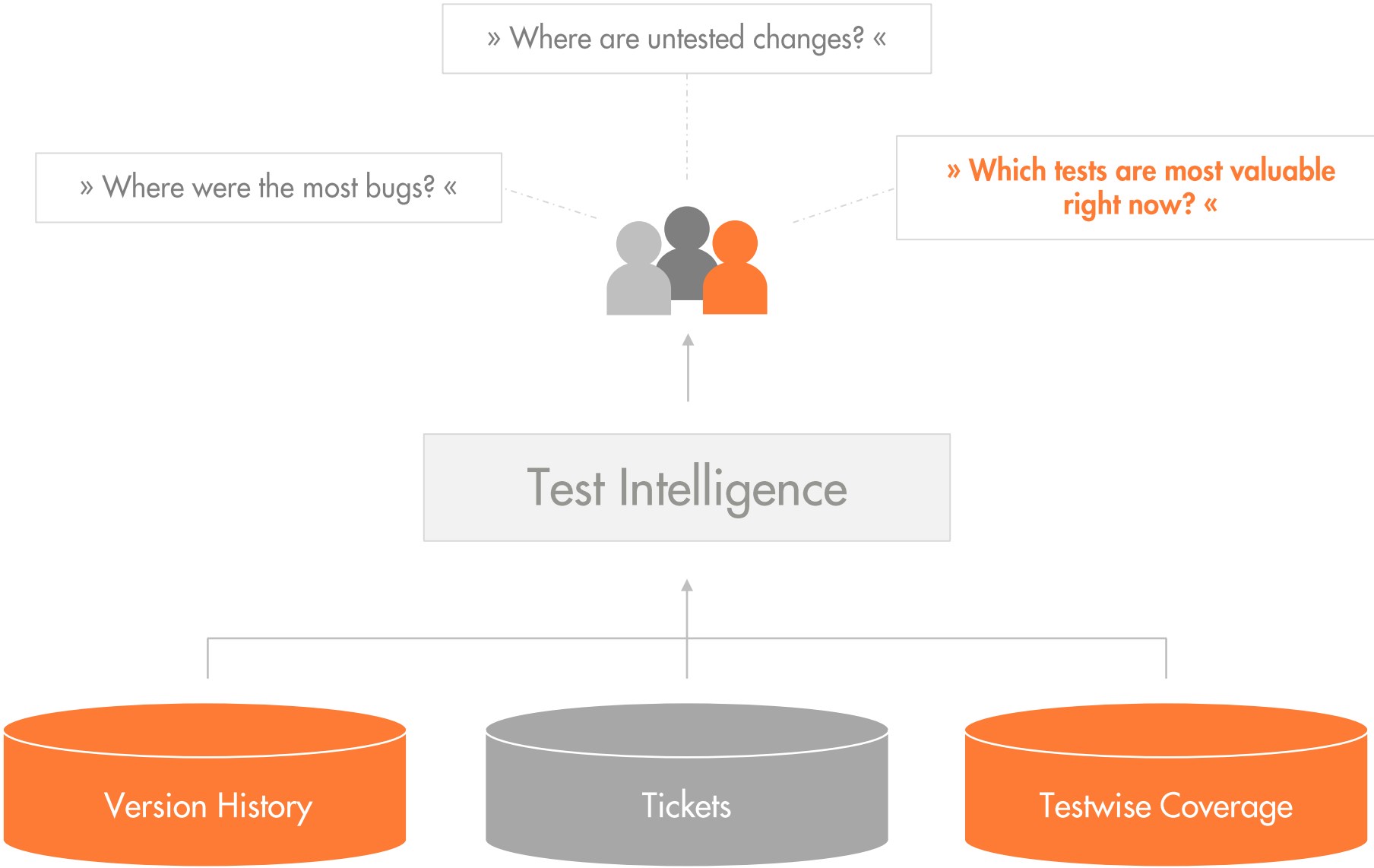
A. Test Suite Optimization

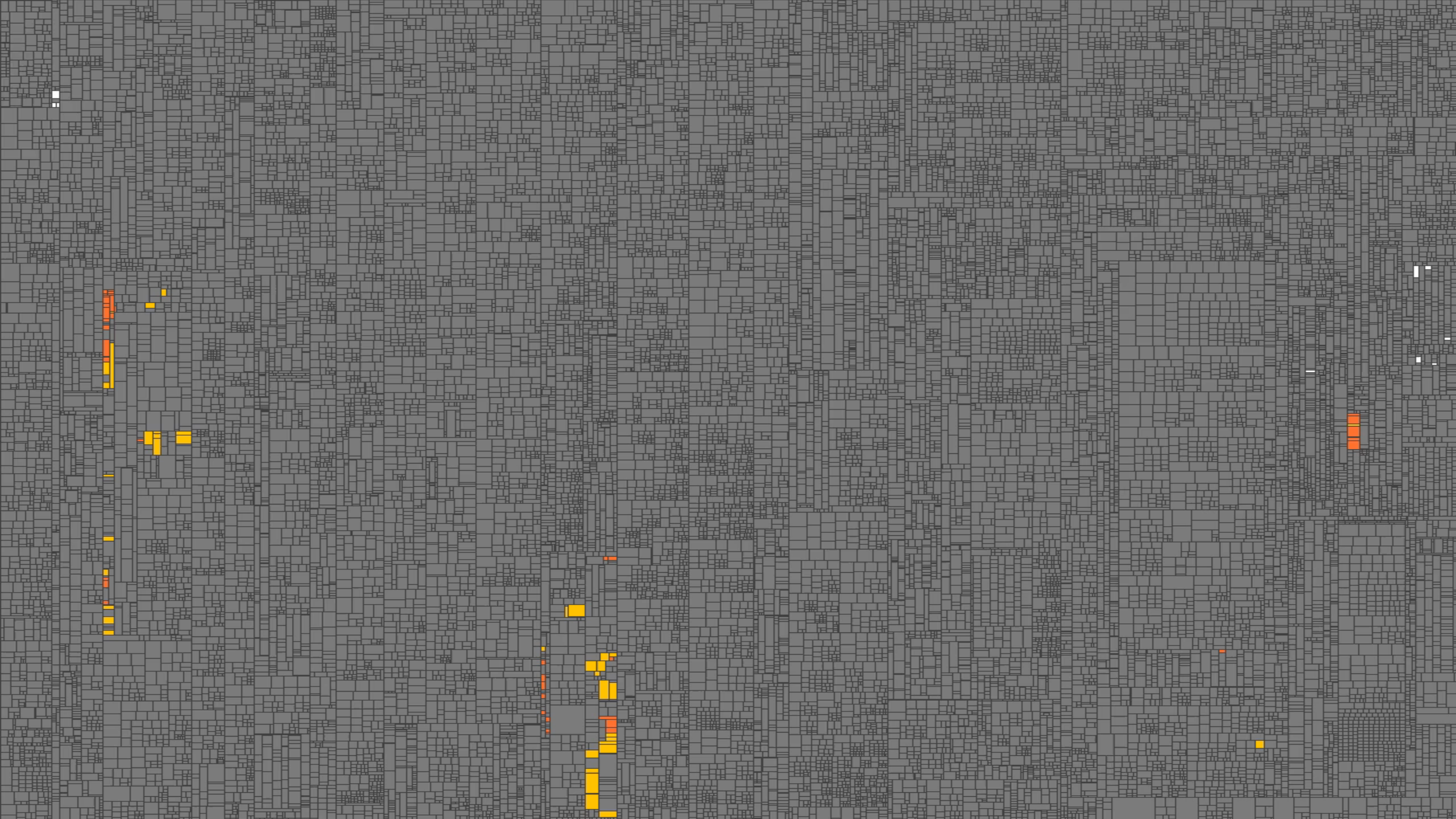
Optimizing a test suite entails maximizing its effectiveness, that is its achieved coverage and fault detection for a given cost in execution time [2]. There are different principles that

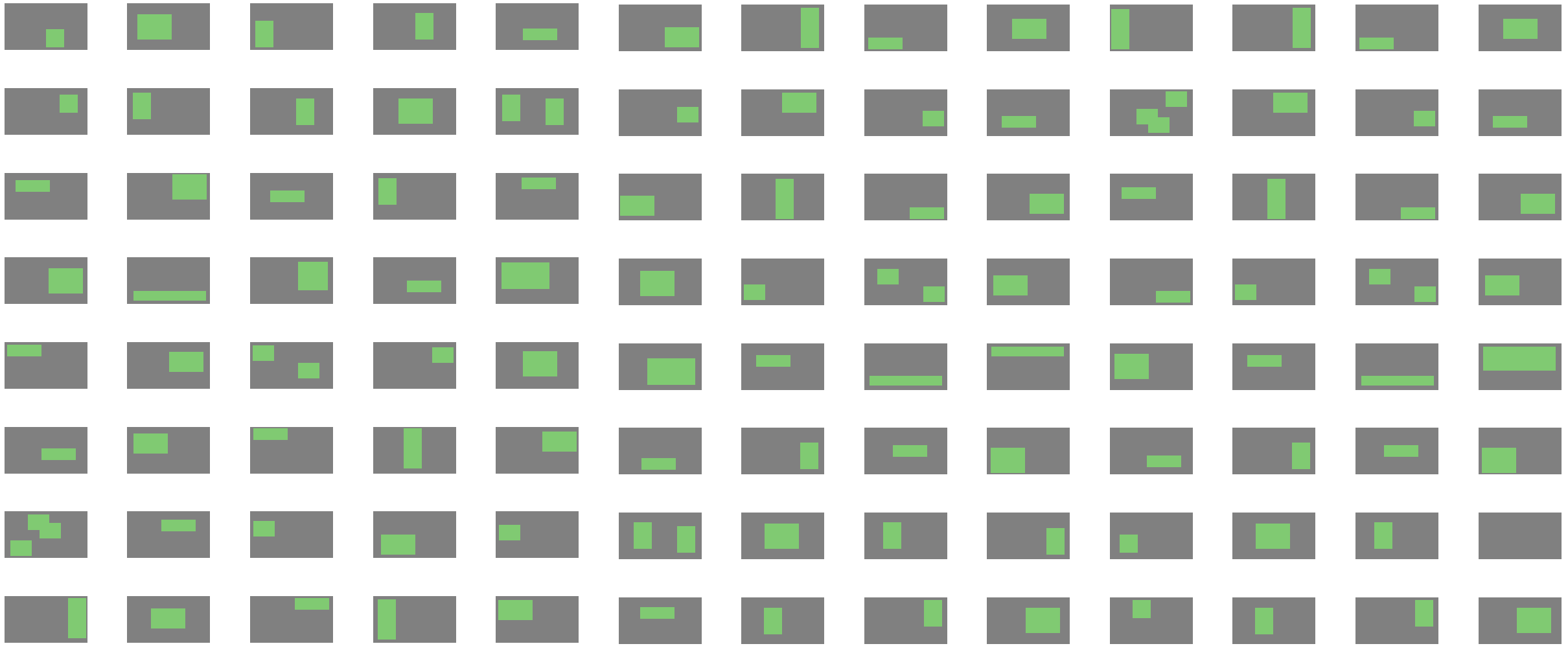
AI Test Clustering



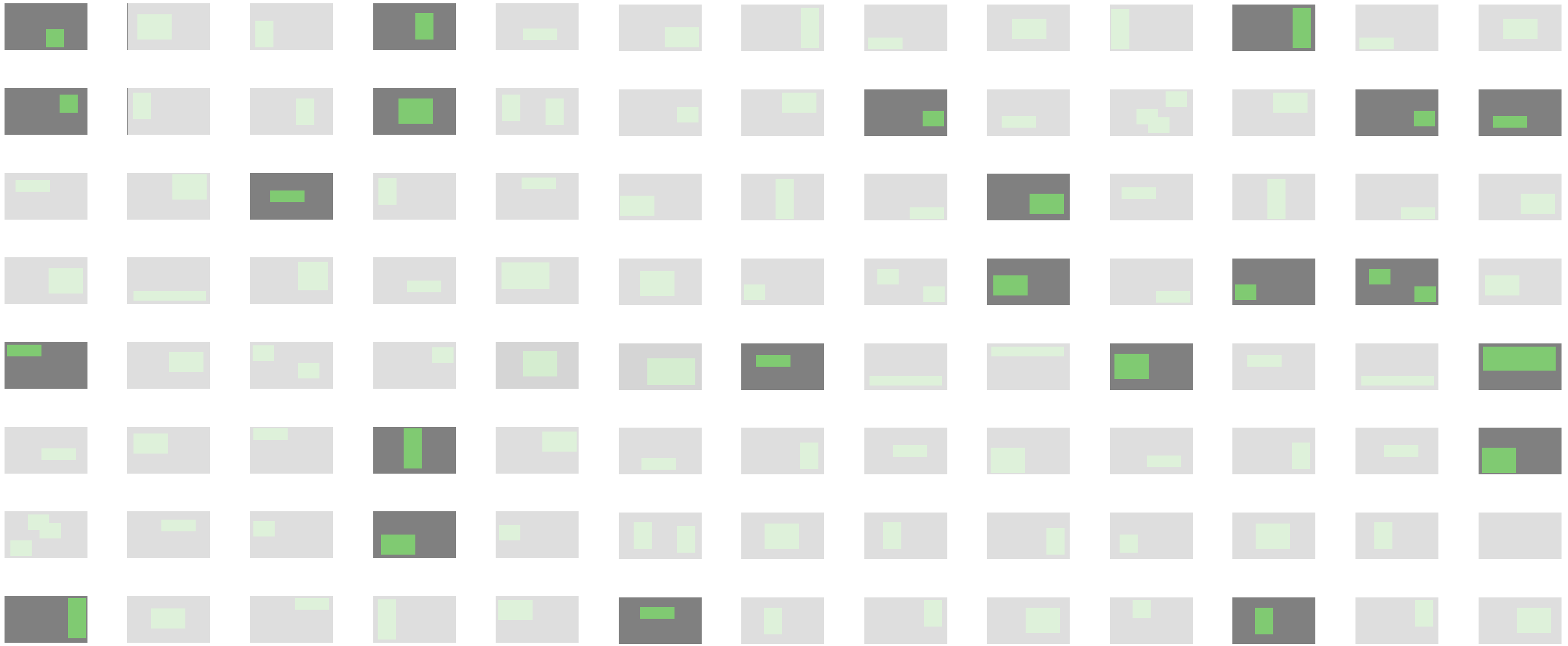



» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Testwise Coverage

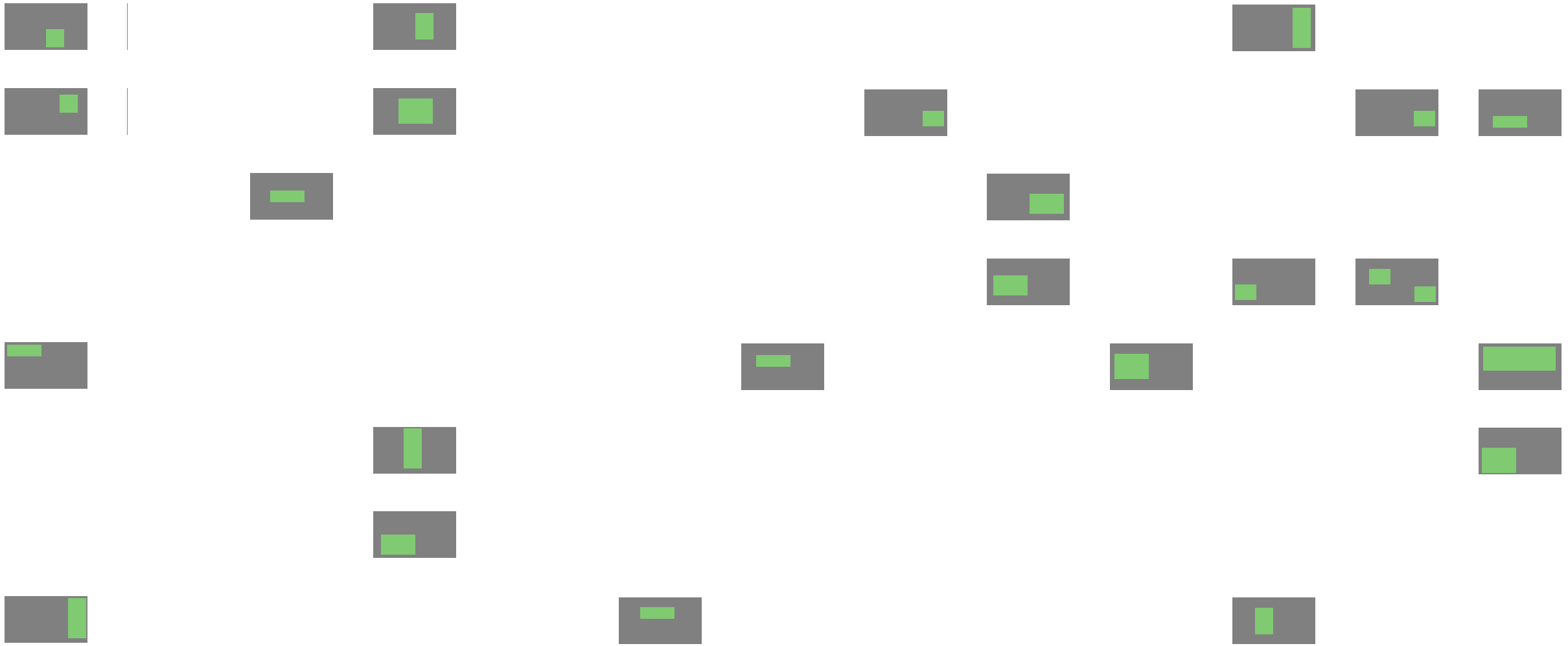




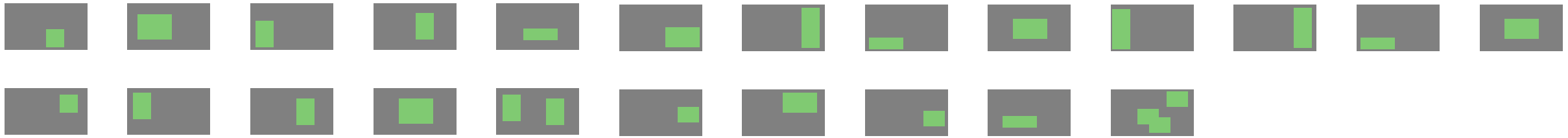
Step 1: Selection of Impacted Tests



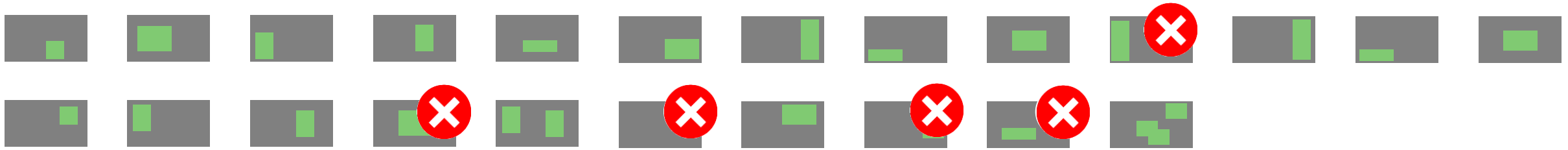
Step 1: Selection of Impacted Tests



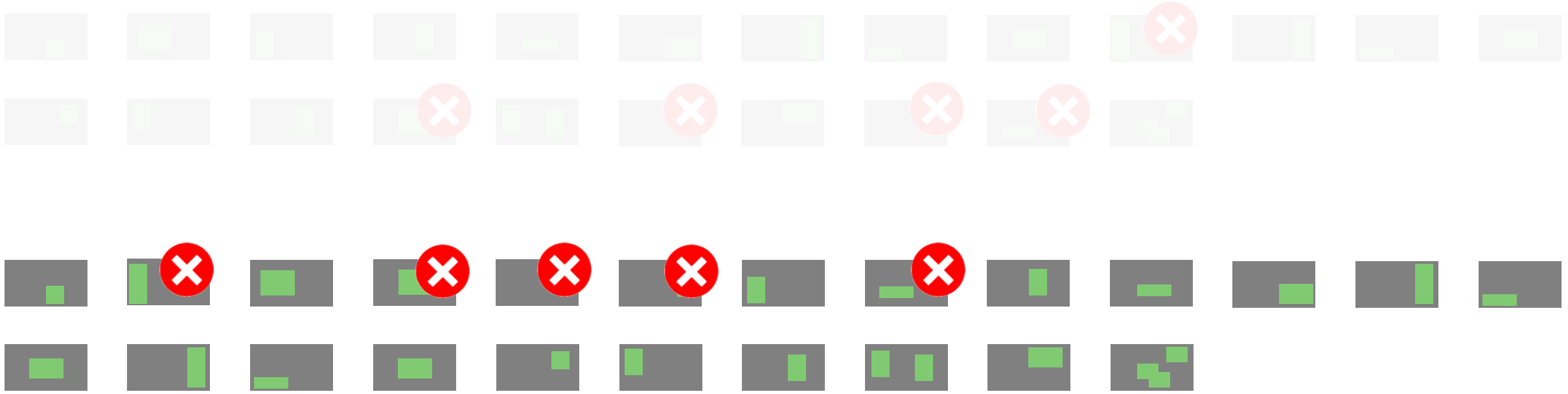
Step 1: Selection of Impacted Tests

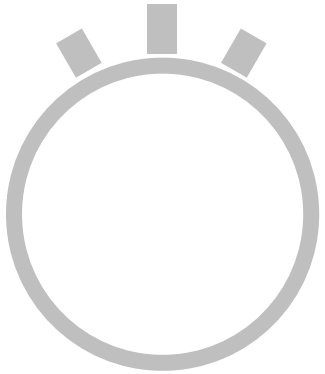


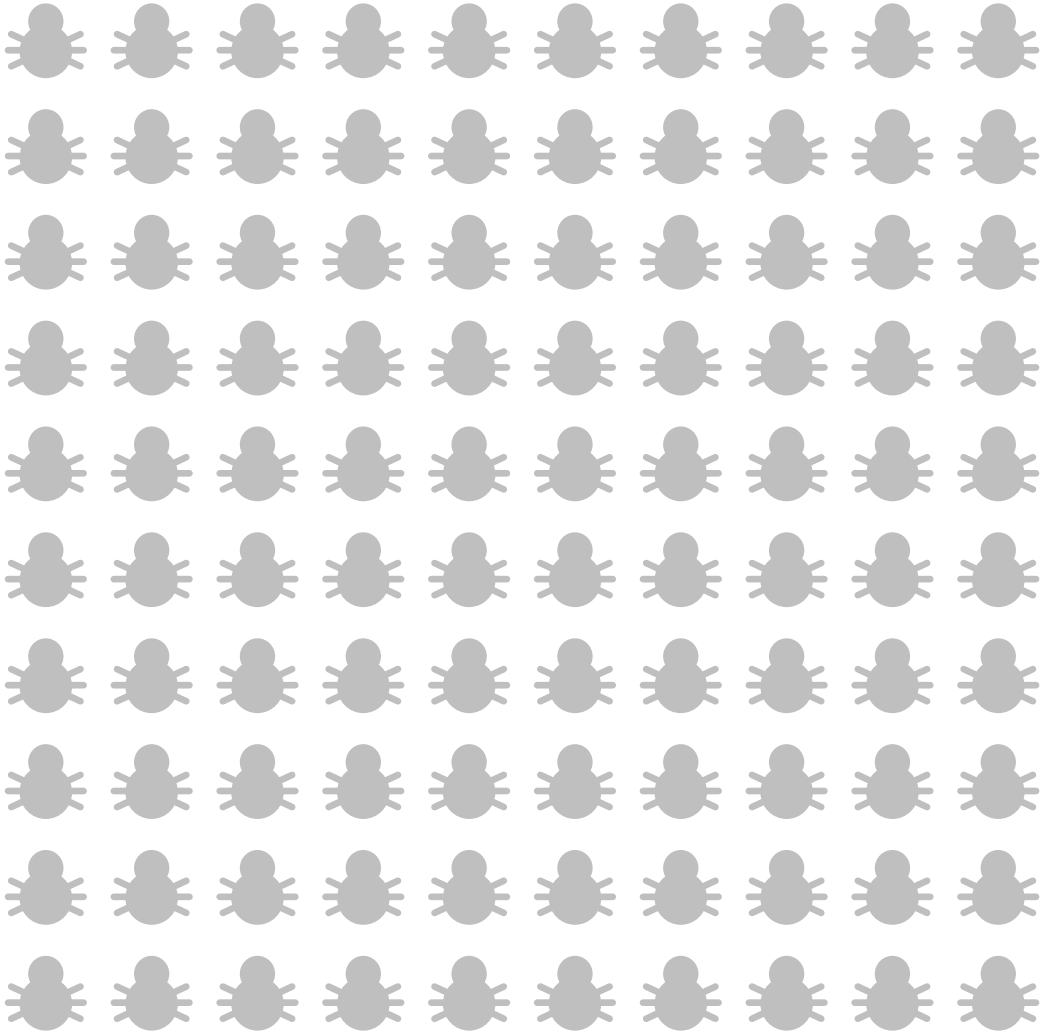
Step 2: Prioritization of Impacted Tests



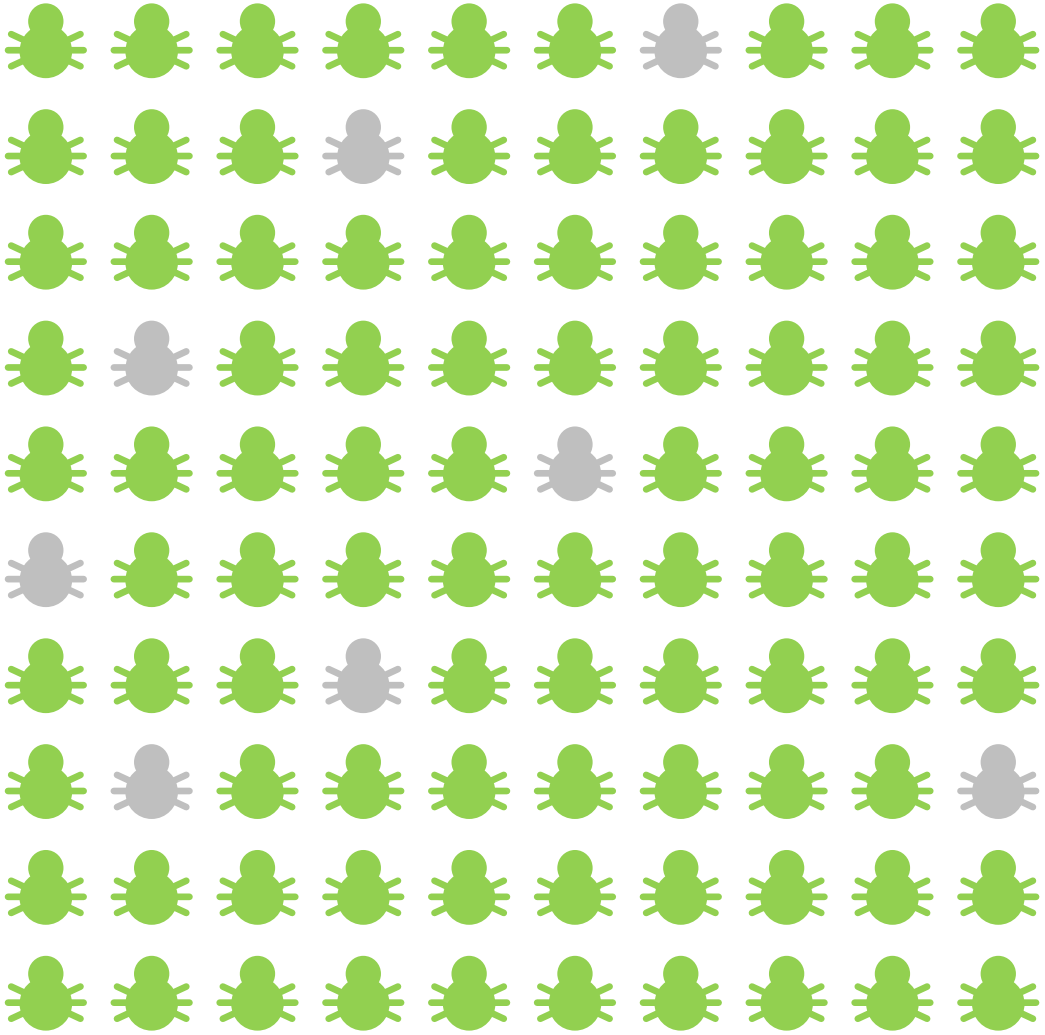
Step 2: Prioritization of Impacted Tests







90 %



Industry Example


Before

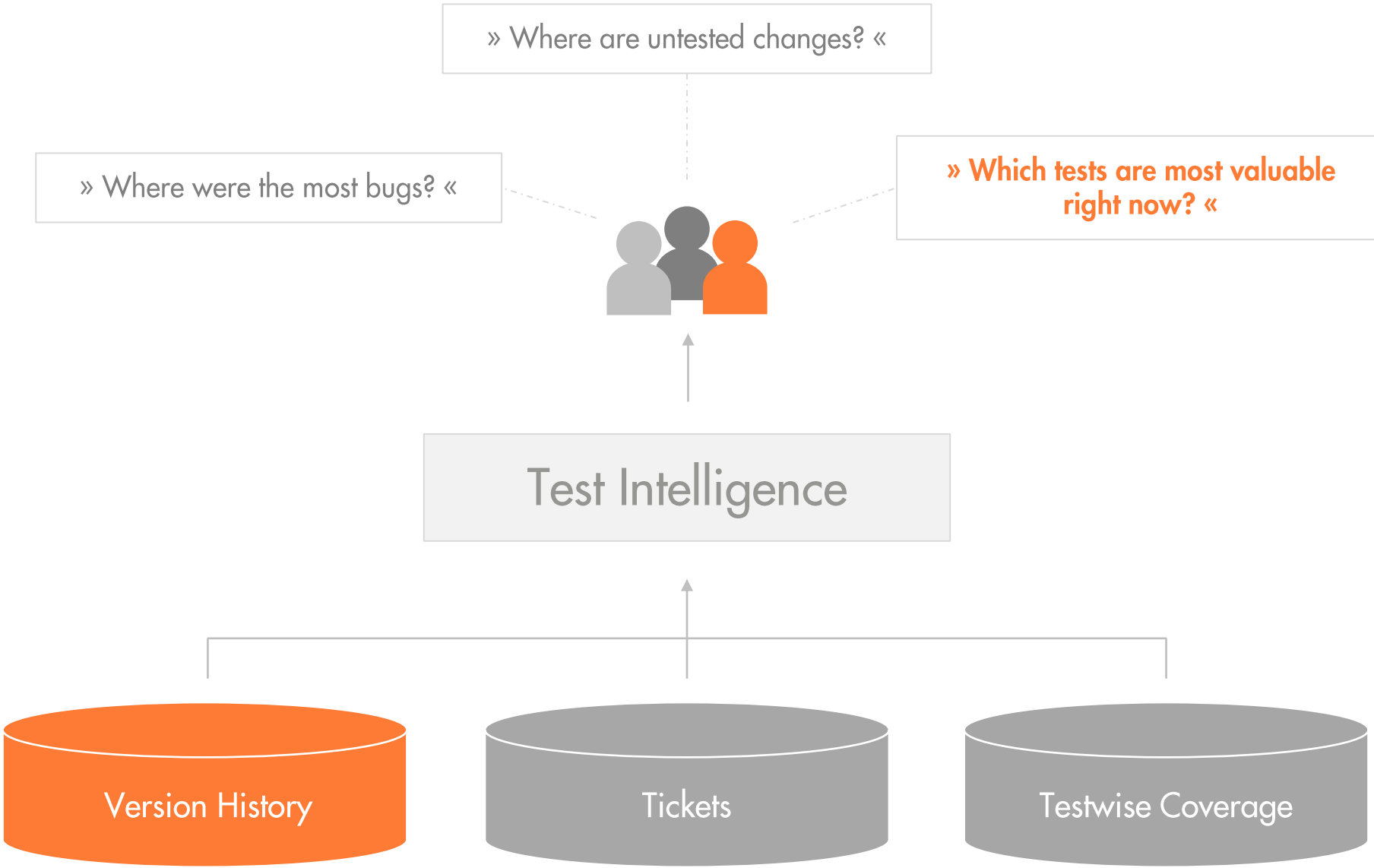
Test runtimes of 3-12h through large parameter combination space.


Goal

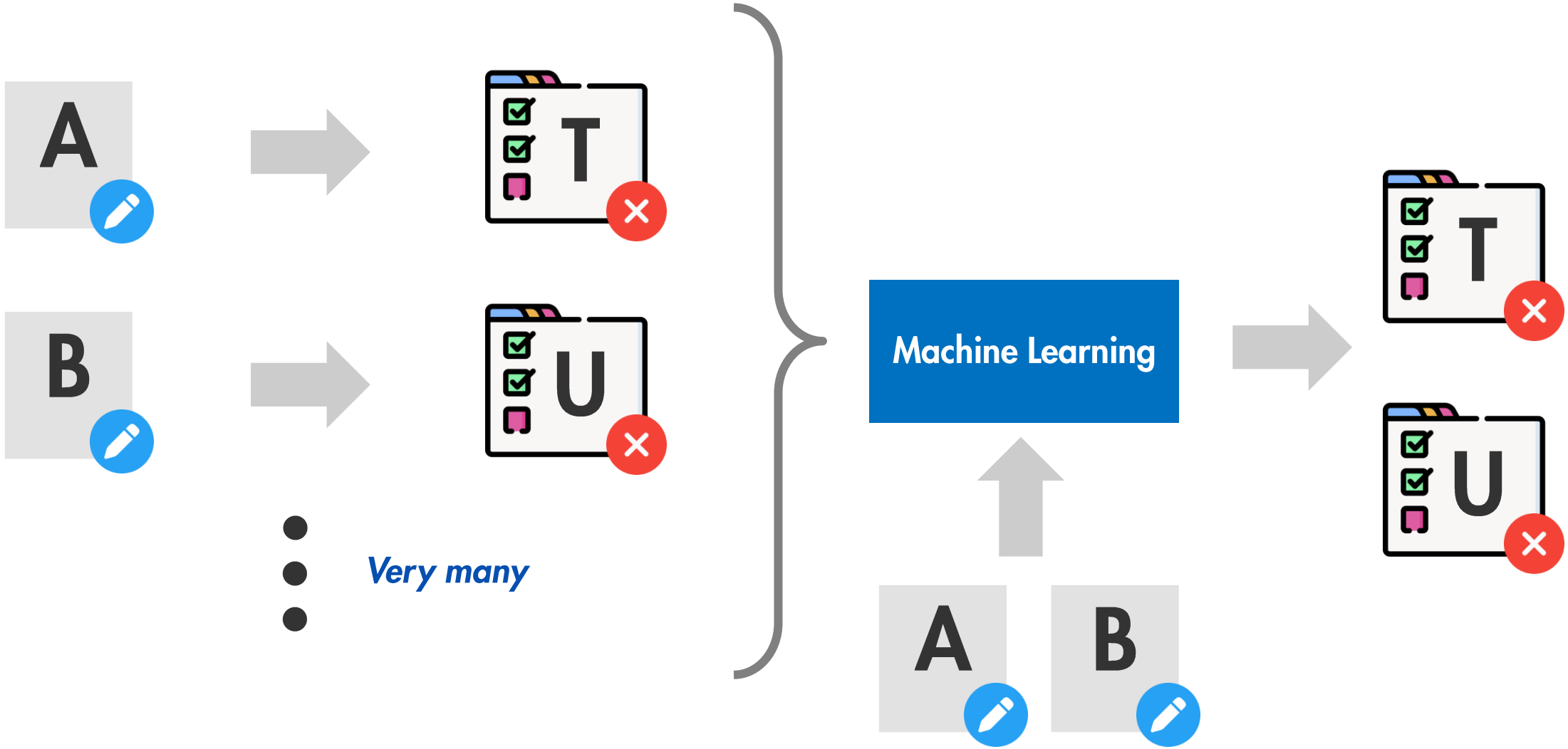
Test feedback from the CI in less than 10 minutes.

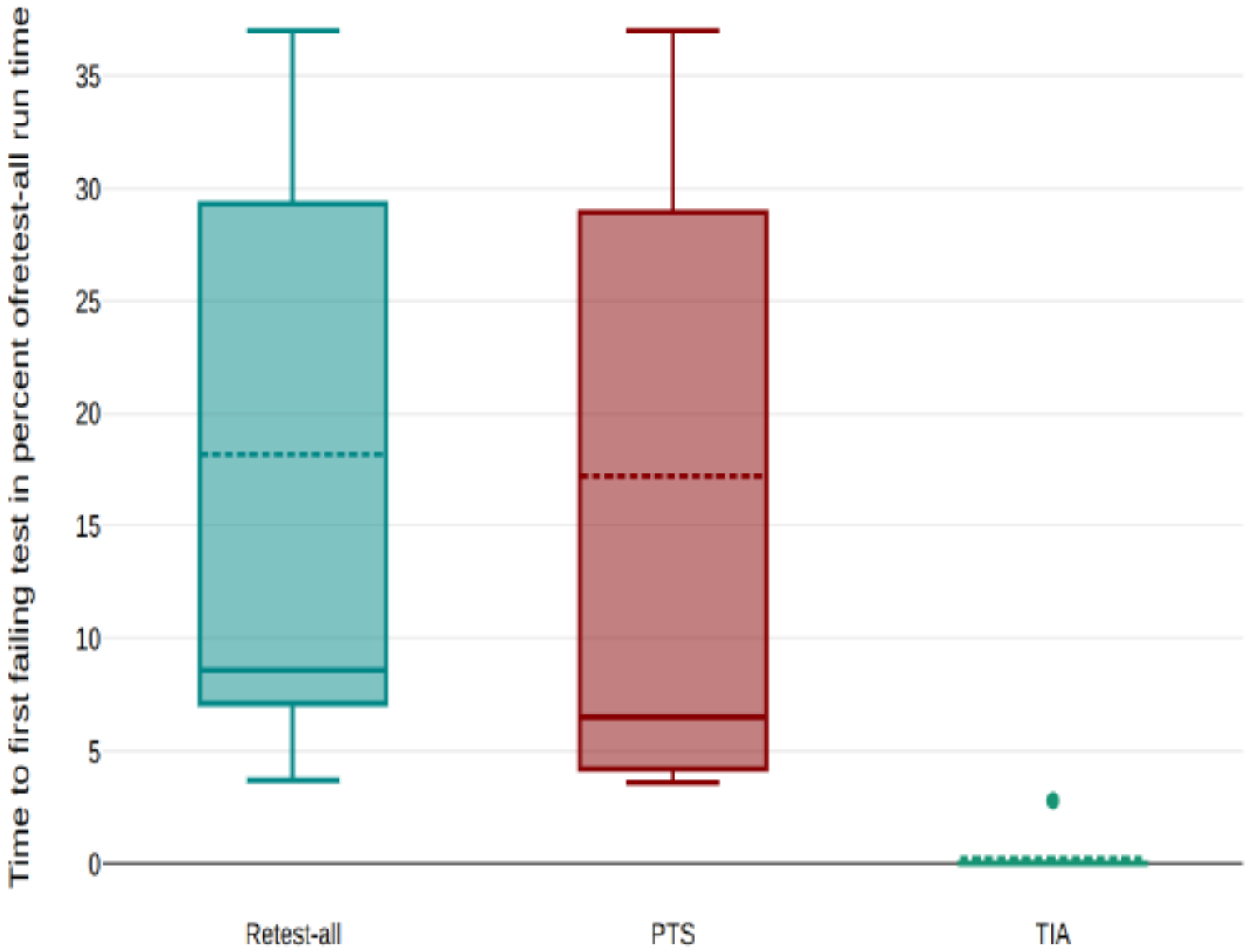

With TIA

Since the introduction of test impact analysis years ago, test feedback always arrives within 10 minutes, although the number of tests increased 57% since.

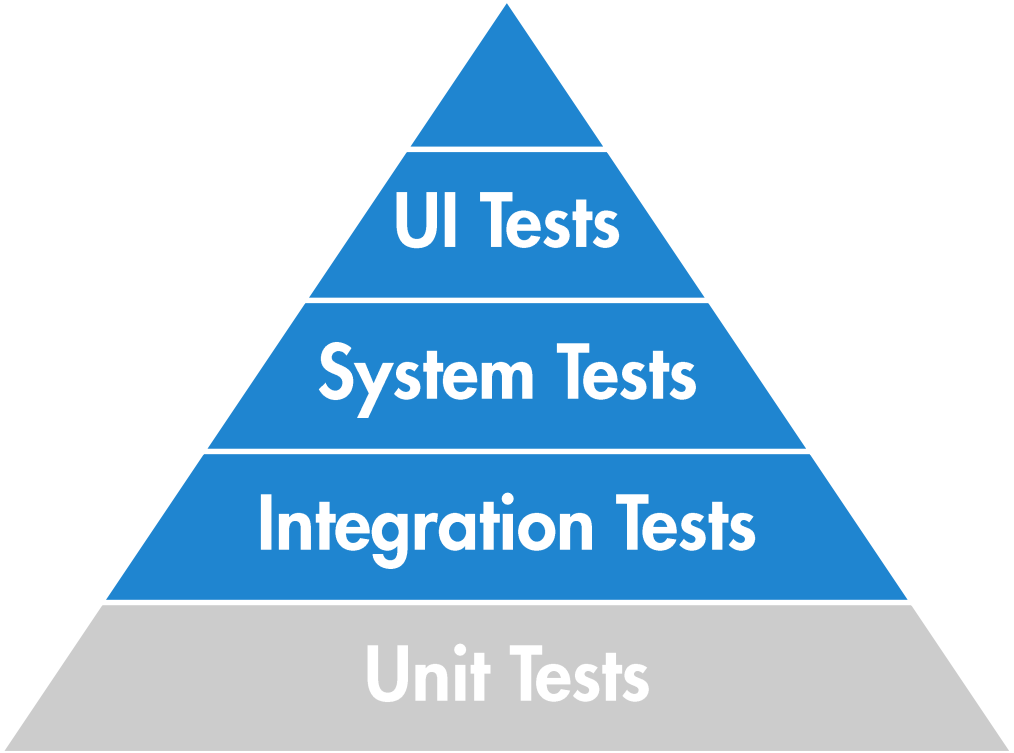

» Where are untested changes? «
» Where were the most bugs? «
» Which tests are most valuable right now? «
Test Intelligence
Version History
Tickets
Testwise Coverage

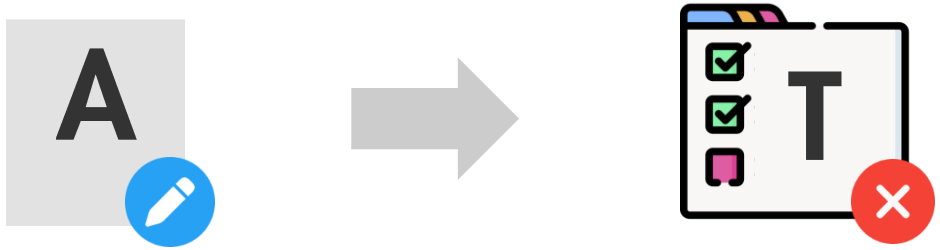
Predictive Test Selection

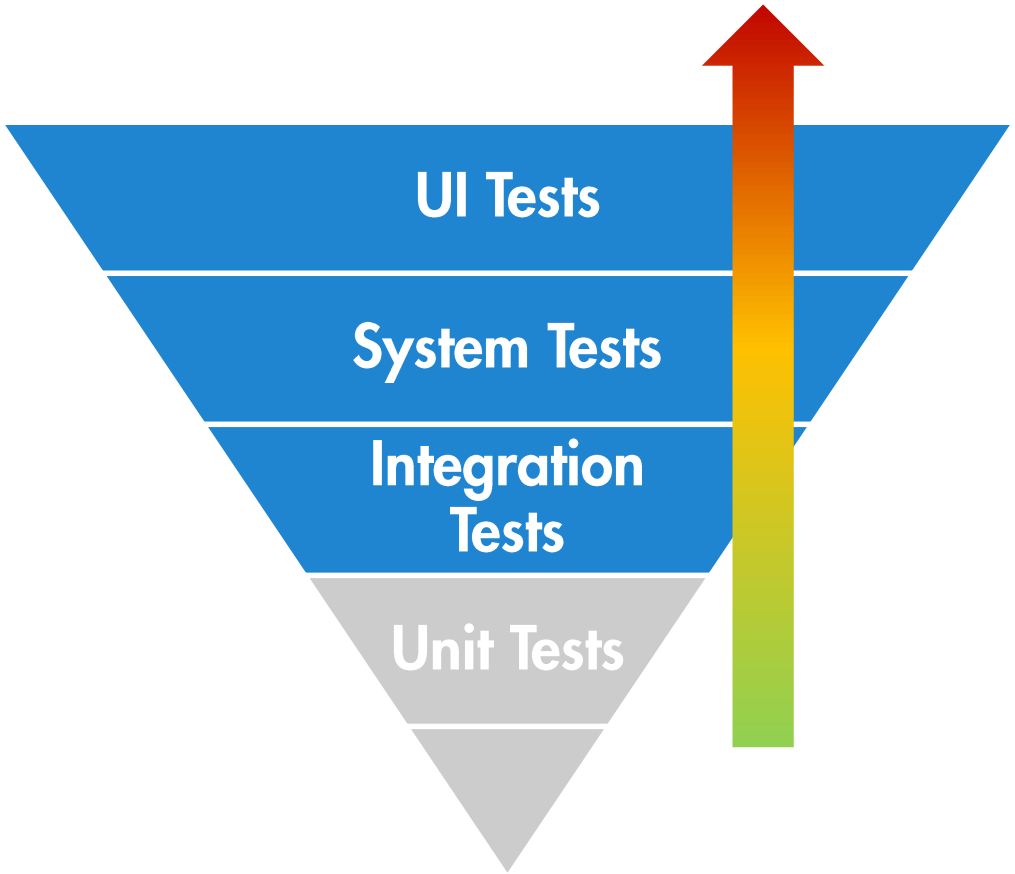


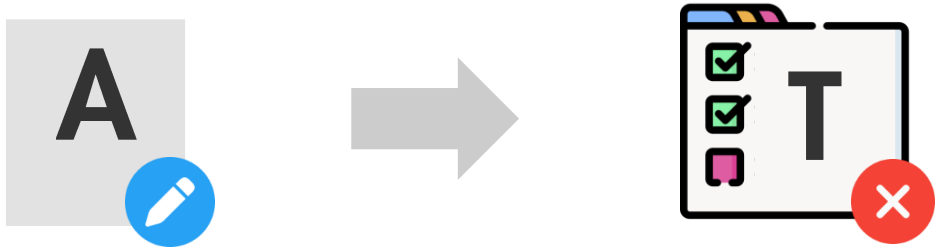


Ben Slimane, 2023, TU München: *Comparative Analysis of Different Approaches for Test Impact Analyses for Real World Test Suites*

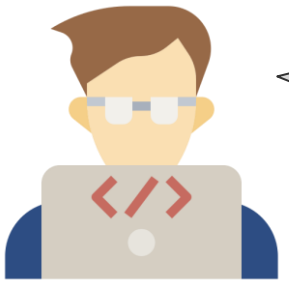

UI Tests
System Tests
Integration Tests
Unit Tests


A
T


UI Tests
System Tests
Integration Tests
Unit Tests

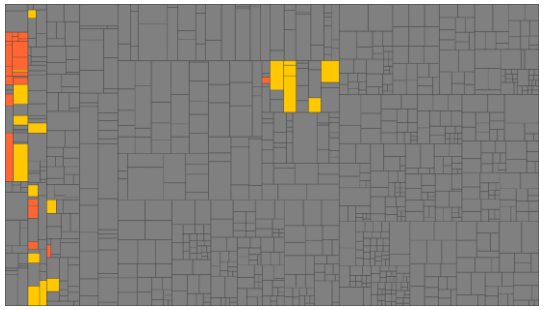

A
T

Continuous Integration



We changed **Login, Accounting und Search**.

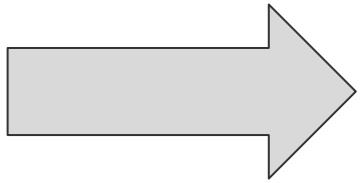
(User Story, Pull Request, Release, …)



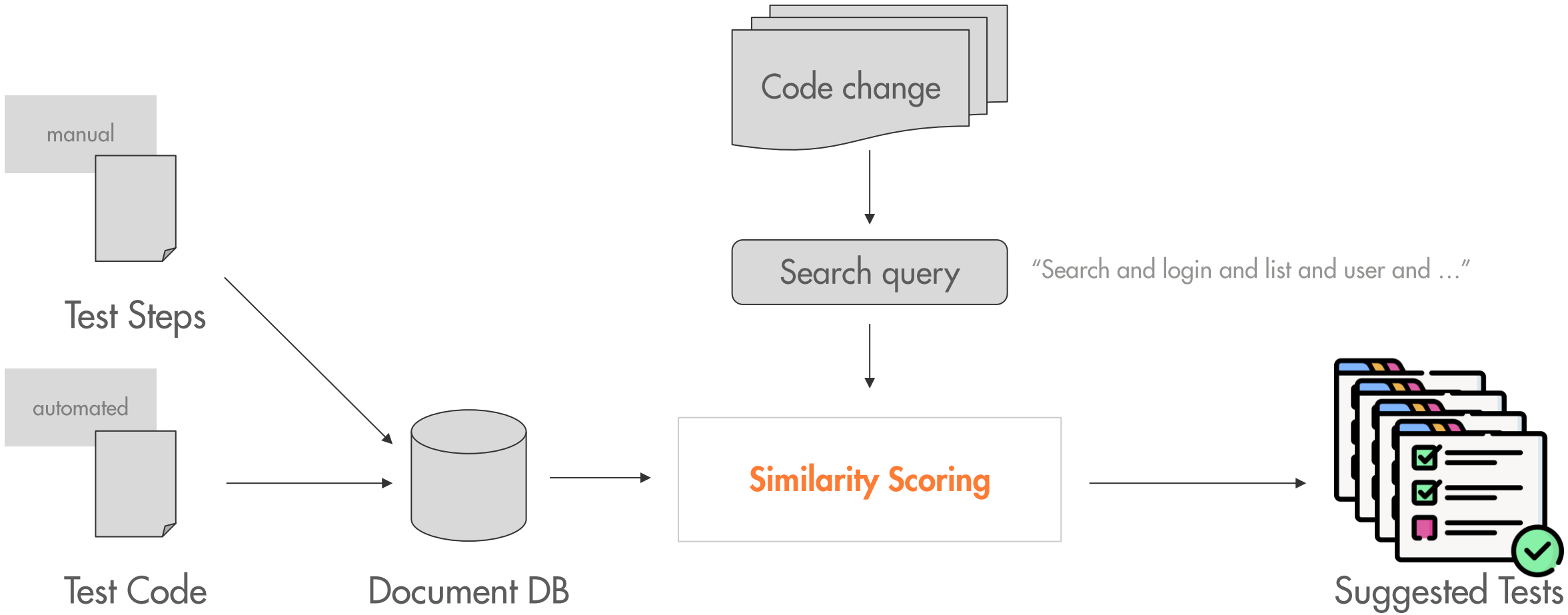
Google test cases for login

Google test cases for accounting

test cases for search



Tests the affected functionality


Code change
manual
Search query
"Search and login and list and user and …"
Test Steps
automated
Similarity Scoring
Test Code
Document DB
Suggested Tests

Changed Code

```
43
42      LOG.debug("Debit Transaction from Account: Account Updated.");
41
40    }
39
38    /*
37     * Transfer amount between two accounts
36     *
35     * Accounts should be full objects. With that said, the objects are fetched to make sure.
34     *
33     * AccountTransaction can be a partial object but must contain the transaction amount.
32     */
31    public void transfer(Account fromAccount, Account toAccount, AccountTransaction accountTransaction) {
30
29      LOG.debug("Transfer Between Accounts:");
28
27      // From Transaction
26      fromAccount = this.getAccountById(fromAccount.getId());
25      AccountTransaction fromAt = new AccountTransaction();
24      fromAt.setAmount(accountTransaction.getAmount());
23      fromAt.setTransactionDate(accountTransaction.getTransactionDate());
22      fromAt.setDescription("Transfer to Account (" + toAccount.getAccountNumber() + ")");
21      fromAt.setTransactionType(transactionTypeRepository.findByCode(Constants.ACCT_TRAN_TYPE_XFER_CODE));
20      debitTransaction(fromAccount, fromAt);
19
18      // To Transaction
17      toAccount = this.getAccountById(toAccount.getId());
16      AccountTransaction toAt = new AccountTransaction();
15      toAt.setAmount(accountTransaction.getAmount());
14      toAt.setTransactionDate(accountTransaction.getTransactionDate());
13      toAt.setDescription("Transfer from Account (" + fromAccount.getAccountNumber() + ")");
12      toAt.setTransactionType(transactionTypeRepository.findByCode(Constants.ACCT_TRAN_TYPE_XFER_CODE));
11      creditTransaction(toAccount, toAt);
10
9       LOG.debug("Transfer Between Accounts: Accounts Updated.");
8     }
7
6     /*
5      * Get Account object by Id
4      */
3     public Account getAccountById(Long id) {
2       Optional<Account> act = accountRepository.findById(id);
```

Digibank

Changed Code

```
43
42      LOG.debug("Debit Transaction from Account: Account Updated.");
41
40    }
39
38    /*
37     * Transfer amount between two accounts
36     *
35     * Accounts should be full objects. With that said, the objects are f
34     *
33     * AccountTransaction can be a partial object but must contain the tr
32     */
31    public void transfer(Account fromAccount, Account toAccount, AccountT
30
29      LOG.debug("Transfer Between Accounts:");
28
27      // From Transaction
26      fromAccount = this.getAccountById(fromAccount.getId());
25      AccountTransaction fromAt = new AccountTransaction();
24      fromAt.setAmount(accountTransaction.getAmount());
23      fromAt.setTransactionDate(accountTransaction.getTransactionDate());
22      fromAt.setDescription("Transfer to Account (" + toAccount.getAccoun
21      fromAt.setTransactionType(transactionTypeRepository.findByCode(Cons
20      debitTransaction(fromAccount, fromAt);
19
18      // To Transaction
17      toAccount = this.getAccountById(toAccount.getId());
16      AccountTransaction toAt = new AccountTransaction();
15      toAt.setAmount(accountTransaction.getAmount());
14      toAt.setTransactionDate(accountTransaction.getTransactionDate());
13      toAt.setDescription("Transfer from Account (" + fromAccount.getAcco
12      toAt.setTransactionType(transactionTypeRepository.findByCode(Consta
11      creditTransaction(toAccount, toAt);
10
9       LOG.debug("Transfer Between Accounts: Accounts Updated.");
8     }
7
6     /*
5      * Get Account object by Id
4      */
3     public Account getAccountById(Long id) {
2       Optional<Account> act = accountRepository.findById(id);
```

Code Test

```
20    public class AccountServiceTest extends IntegrationTest {
19
18      @Test
17      public void transferBetweenSameAccountShouldNotBePossible() {
16        Account account = new Account("savings", AccountType.SAVING
15        AccountService service = new AccountService();
14        AccountTransaction transaction =
13          MockAccountTransaction.createForAmount(100);
12        service.transfer(account, account, transaction);
11
10        assertThat(this.getErrors()).contains(
9           new Error("Transfer between same account is not possible.
8
7         Account databaseAccount = this.findAccountById(account.getI
6         AccountTransactionList transactionList = this.getTransactio
5         assertThat(transactionList).isEmpty()
4       }
3
2     }
1
21
```

Changed Code

```
43
42      LOG.debug("Debit Transaction from Account: Account Updated.");
41
40  }
39
38  /*
37   * Transfer amount between two accounts
36   *
35   * Accounts should be full objects. With that said, the objects are f
34   *
33   * AccountTransaction can be a partial object but must contain the tra
32   */
31  public void transfer(Account fromAccount, Account toAccount, AccountT
30
29      LOG.debug("Transfer Between Accounts:");
28
27      // From Transaction
26      fromAccount = this.getAccountById(fromAccount.getId());
25      AccountTransaction fromAt = new AccountTransaction();
24      fromAt.setAmount(accountTransaction.getAmount());
23      fromAt.setTransactionDate(accountTransaction.getTransactionDate());
22      fromAt.setDescription("Transfer to Account (" + toAccount.getAccoun
21      fromAt.setTransactionType(transactionTypeRepository.findByCode(Cons
20      debitTransaction(fromAccount, fromAt);
19
18      // To Transaction
17      toAccount = this.getAccountById(toAccount.getId());
16      AccountTransaction toAt = new AccountTransaction();
15      toAt.setAmount(accountTransaction.getAmount());
14      toAt.setTransactionDate(accountTransaction.getTransactionDate());
13      toAt.setDescription("Transfer from Account (" + fromAccount.getAcco
12      toAt.setTransactionType(transactionTypeRepository.findByCode(Consta
11      creditTransaction(toAccount, toAt);
10
9       LOG.debug("Transfer Between Accounts: Accounts Updated.");
8   }
7
6   /*
5    * Get Account object by Id
4    */
3   public Account getAccountById(Long id) {
2       Optional<Account> act = accountRepository.findById(id);
```

Cucumber Test

```
22  @ui @account @savings
21  Feature: Transfer Money (UI)
20    As a DigitalBank user
19    I want to transfer money between accounts
18    So I can change how much is in each account
17
16
15  @negative
14  Scenario: Transfer between the same account is not possible
13    Given Carleen is logged into the application with Carleen6231@gmail.com
12    And they attempt to open a new 'Savings Account'
11      When Carleen enters 'Tangerine Savings' into the Account Name field
10      And they select 'Individual' from the Ownership radio button
9       And they select 'Money Market' from the Account Type radio button
8       And they enter '2500' into the Money Market Initial Deposit field
7       And they click the Submit button
6     And they attempt to transfer money
5     When Carleen selects account number '1' as the from account
4     And they select account number '1' as the to account
3     And they enter '11' into the amount field
2     And they submit the form
1     Then Carleen verifies the transfer failed
23
```

Changed Code

```
43
42      LOG.debug("Debit Transaction from Account: Account Updated.");
41
40  }
39
38  /*
37   * Transfer amount between two accounts
36   *
35   * Accounts should be full objects. With that said, the objects are fe
34   *
33   * AccountTransaction can be a partial object but must contain the tra
32   */
31  public void transfer(Account fromAccount, Account toAccount, AccountTr
30
29      LOG.debug("Transfer Between Accounts:");
28
27      // From Transaction
26      fromAccount = this.getAccountById(fromAccount.getId());
25      AccountTransaction fromAt = new AccountTransaction();
24      fromAt.setAmount(accountTransaction.getAmount());
23      fromAt.setTransactionDate(accountTransaction.getTransactionDate());
22      fromAt.setDescription("Transfer to Account (" + toAccount.getAccount
21      fromAt.setTransactionType(transactionTypeRepository.findByCode(Const
20      debitTransaction(fromAccount, fromAt);
19
18      // To Transaction
17      toAccount = this.getAccountById(toAccount.getId());
16      AccountTransaction toAt = new AccountTransaction();
15      toAt.setAmount(accountTransaction.getAmount());
14      toAt.setTransactionDate(accountTransaction.getTransactionDate());
13      toAt.setDescription("Transfer from Account (" + fromAccount.getAccou
12      toAt.setTransactionType(transactionTypeRepository.findByCode(Constan
11      creditTransaction(toAccount, toAt);
10
9       LOG.debug("Transfer Between Accounts: Accounts Updated.");
8   }
7
6   /*
5    * Get Account object by Id
4    */
3   public Account getAccountById(Long id) {
2       Optional<Account> act = accountRepository.findById(id);
```

Robot Test

```
17  *** Settings ***
16  Resource        ../keywords/digibank_keywords.robot
15
14  *** Test Cases ***
13  Transfer between the same account is not possible
12      Log in  Carleen6231@gmail.com
11      Open new account    Savings Account     Individual  Money Market    2
10      Open transfer page
9       Select from account number   1
8       Select to account number   1
7       Enter amount    11
6       Submit transfer form
5       Transfer failed message should be displayed
4
3
2
1
18
```

Changed Code

```
43
42      LOG.debug("Debit Transaction from Account: Account Updated.");
41
40    }
39
38    /*
37     * Transfer amount between two accounts
36     *
35     * Accounts should be full objects. With that said, the objects are f
34     *
33     * AccountTransaction can be a partial object but must contain the tra
32     */
31    public void transfer(Account fromAccount, Account toAccount, AccountT
30
29      LOG.debug("Transfer Between Accounts:");
28
27      // From Transaction
26      fromAccount = this.getAccountById(fromAccount.getId());
25      AccountTransaction fromAt = new AccountTransaction();
24      fromAt.setAmount(accountTransaction.getAmount());
23      fromAt.setTransactionDate(accountTransaction.getTransactionDate());
22      fromAt.setDescription("Transfer to Account (" + toAccount.getAccount
21      fromAt.setTransactionType(transactionTypeRepository.findByCode(Const
20      debitTransaction(fromAccount, fromAt);
19
18      // To Transaction
17      toAccount = this.getAccountById(toAccount.getId());
16      AccountTransaction toAt = new AccountTransaction();
15      toAt.setAmount(accountTransaction.getAmount());
14      toAt.setTransactionDate(accountTransaction.getTransactionDate());
13      toAt.setDescription("Transfer from Account (" + fromAccount.getAccou
12      toAt.setTransactionType(transactionTypeRepository.findByCode(Constan
11      creditTransaction(toAccount, toAt);
10
9       LOG.debug("Transfer Between Accounts: Accounts Updated.");
8     }
7
6     /*
5      * Get Account object by Id
4      */
3     public Account getAccountById(Long id) {
2       Optional<Account> act = accountRepository.findById(id);
```

Manual Test

Action	Check
Log in as Carleen6231@gmail.com	
Open a new account: Type Savings Account, Individual In the Money Market Start deposit: 2500	Account was created as specified.
Open the transfer page.	
Select the account from step 2 as both from and to account.	
Enter amount: 11	
Submit the form	Transfer should fail with a message that transfers between the same account are prohibited

Google

Test cases for Feature 12345

#1

Test Suite 1
https://www.atlassian.com › jira-... · Diese Seite übersetzen

Test Case 1

Xray allows you to plan, design, and execute tests, as well as generate test reports. Xray uses specific Jira issues types for this process.

#2

Test Suite 2
https://www.atlassian.com › jira-... · Diese Seite übersetzen

Test Case 2

A step-by-step tutorial on how to use Xray Cloud, a continuous integration tool that triggers automated tests and provides results through an Xray Test Plan.

#3

Test Suite 1
https://www.getxray.app › blog · Diese Seite übersetzen

Test Case 3

27.11.2020 — It's a full-featured tool that lives inside, and seamlessly integrates with **Jira. Xray** aims to help companies improve the quality of their ...

Goooooooooooogle ›
1 2 3 4 5 6 7 8 9 10 Weiter

Score of a Test fos a search term =

term frequency * inverse **document frequency**

How often does the term occur in a test?
The more often, the higher the score

How many tests contain this term?
The more specific, the higher the score.

Evaluating Information Retrieval for the use in Regression Test Selection

Case Study

Author: Majd Akleh

Supervisors: **Prof. Dr. Ben Hermann**
TU Dortmund
Raphael Nömmer
CQSE GmbH

Date: September 2023

technische universität dortmund

Master Thesis

Optimization and Evaluation of an Information Retrieval Based Test Selection Approach

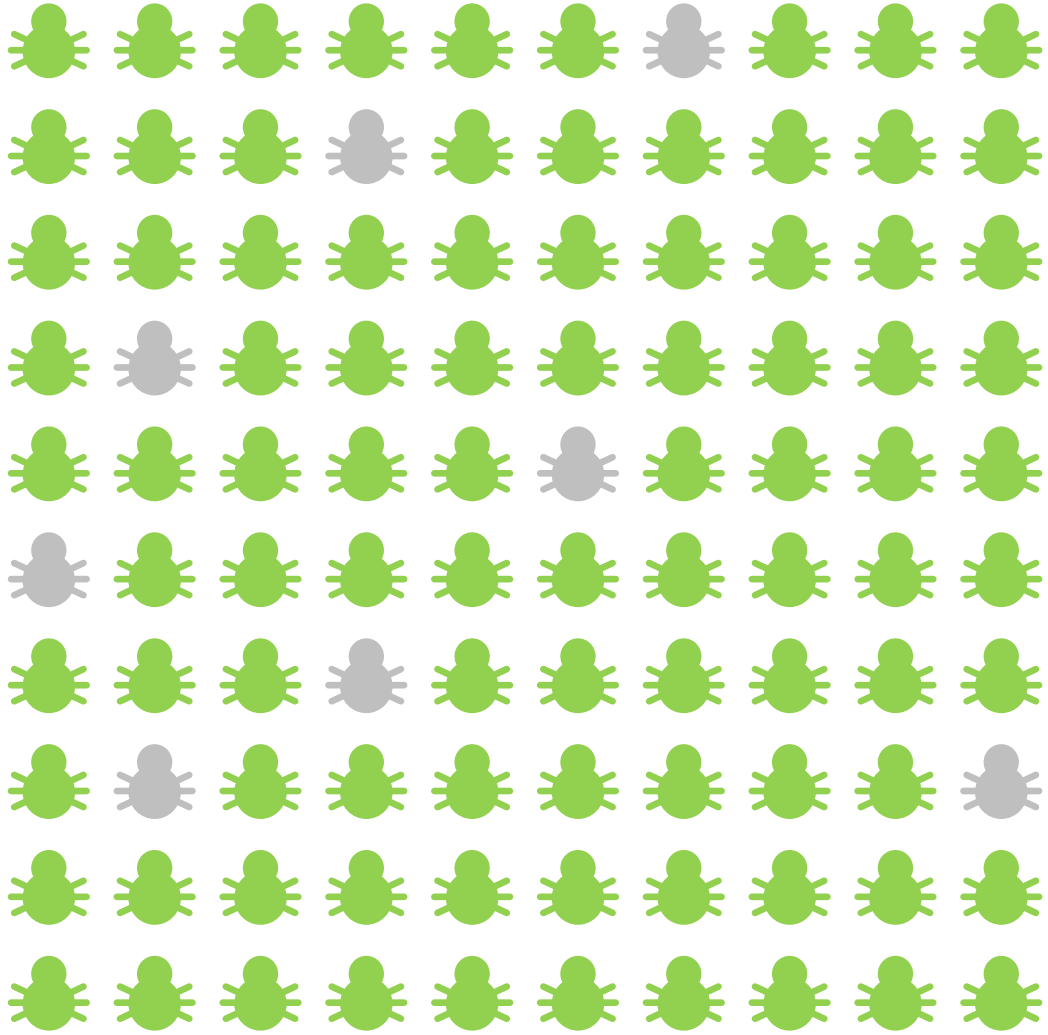
Majd Akleh

June 3, 2024

Reviewer:
JProf. Dr.-Ing. Ben Hermann
Dr. Elmar Jürgens

sse
Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl V - Programmiersysteme
Fachgruppe Softwaretechnik sicherer Systeme
https://sse.cs.tu-dortmund.de

Information Retrieval



Quality Gate

Continuous Integration

We find 90% of the bugs in X% of the time

11%

Pareto-Testliste

2%

Test-Impact-Analyse

Test Coverage

Start HERE

13%

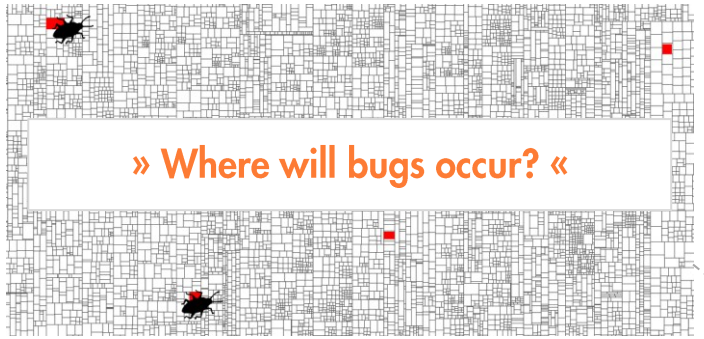
AI Test Clustering

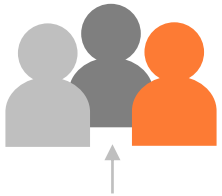
Precision
&
Effort

4%

Similarity Scoring

Test Content





Test Intelligence

Version History

Tickets

Test Coverage

Summary

The data in your repos can help us find more bugs in less time.

Test intelligence analyses are a toolbox. There are many useful tools, but they must be actively integrated into your processes.

I am happy to talk about if and how they could fit into your context.

Please Rate This Session

in the Conference App

Use the  Session feedback in this session's listing on the Agenda

This is how we select the EuroSTAR Awards
and speakers love to get your feedback!

Questions?

Email juergens@cqse.eu

LinkedIn @ElmarJuergens

Contact – Looking Forward to Discussions ☺

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

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