

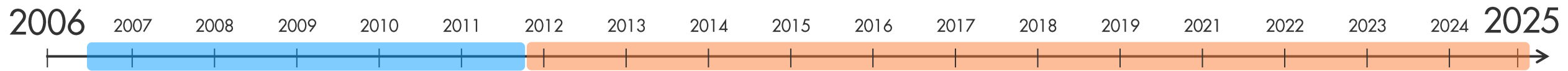
Dear AI, Which Tests Should We Run Now?

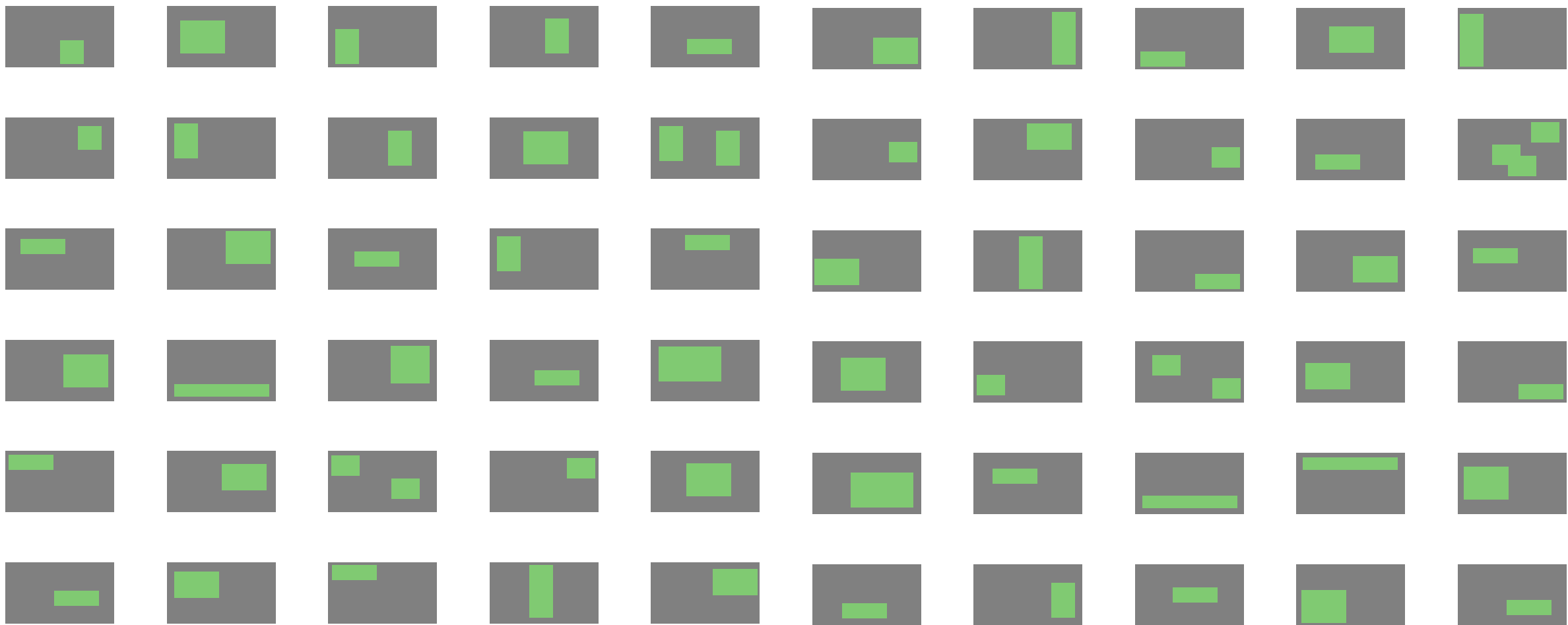


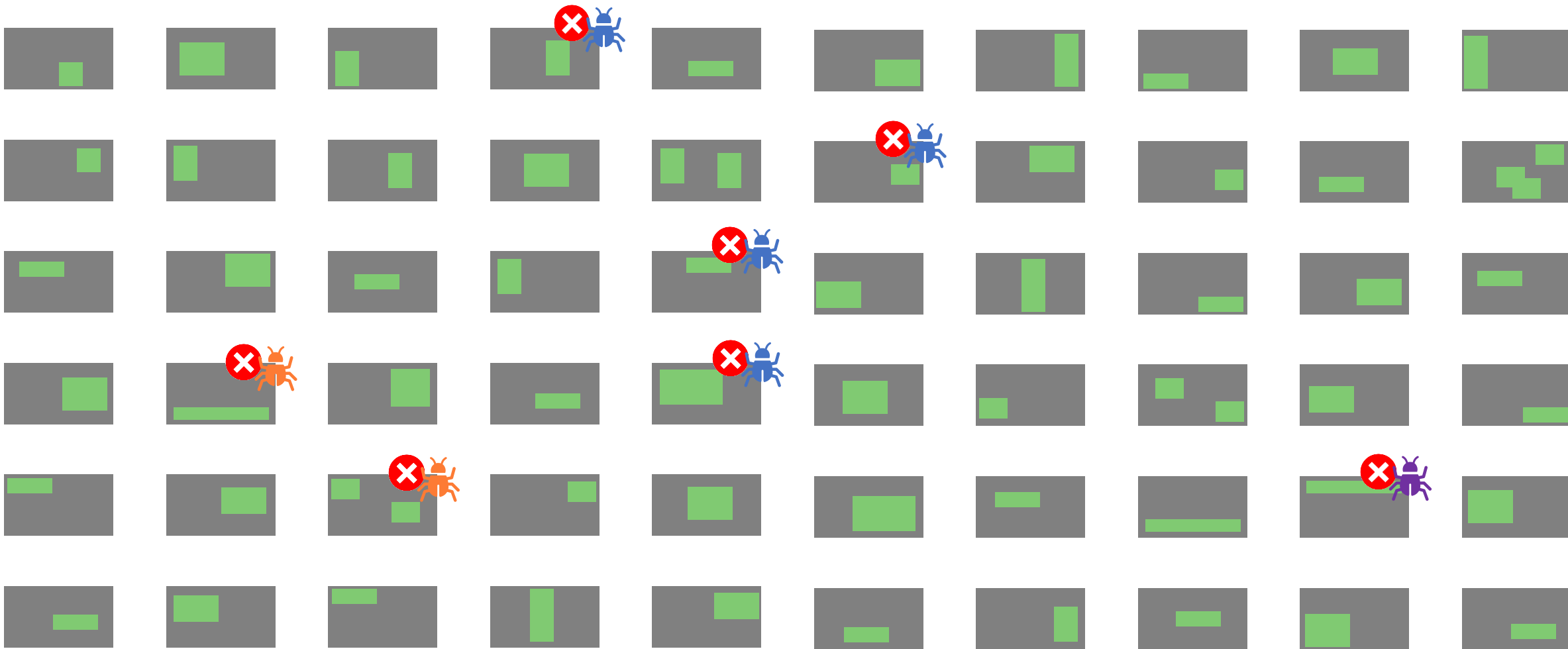
TUM



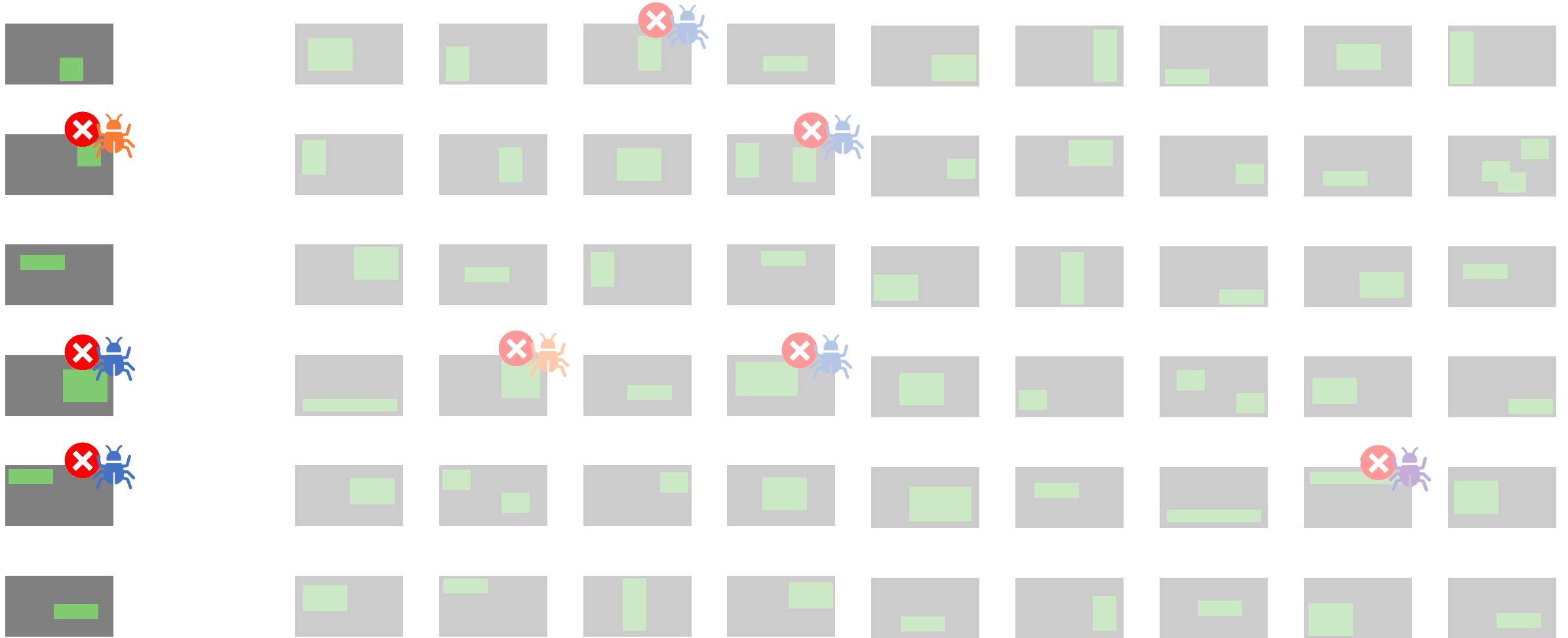
CQSE







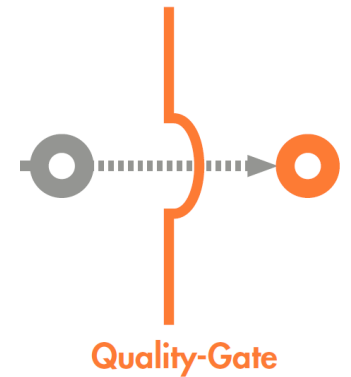
Test Selection



Use Cases for Test Selection

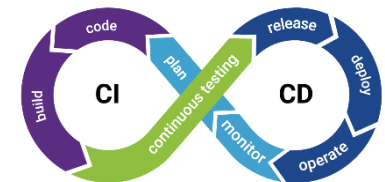
Quality Gate

- Before an expensive test run that executes many tests
- Makes sure that a bad software version does not ruin the run

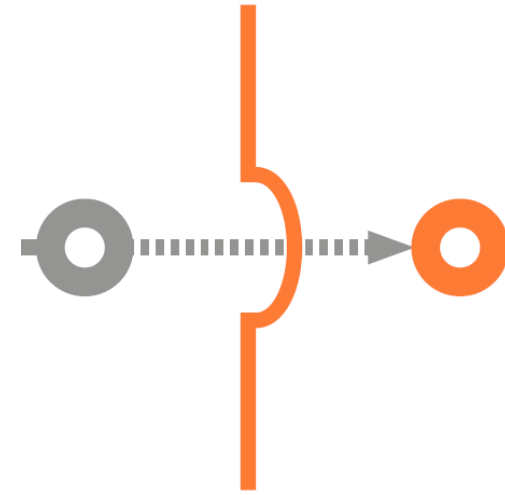


Continuous Integration

- I want Test-Feedback for each pull request!
- But unfortunately, my test suite is too slow
- Run the selected subset in the CI

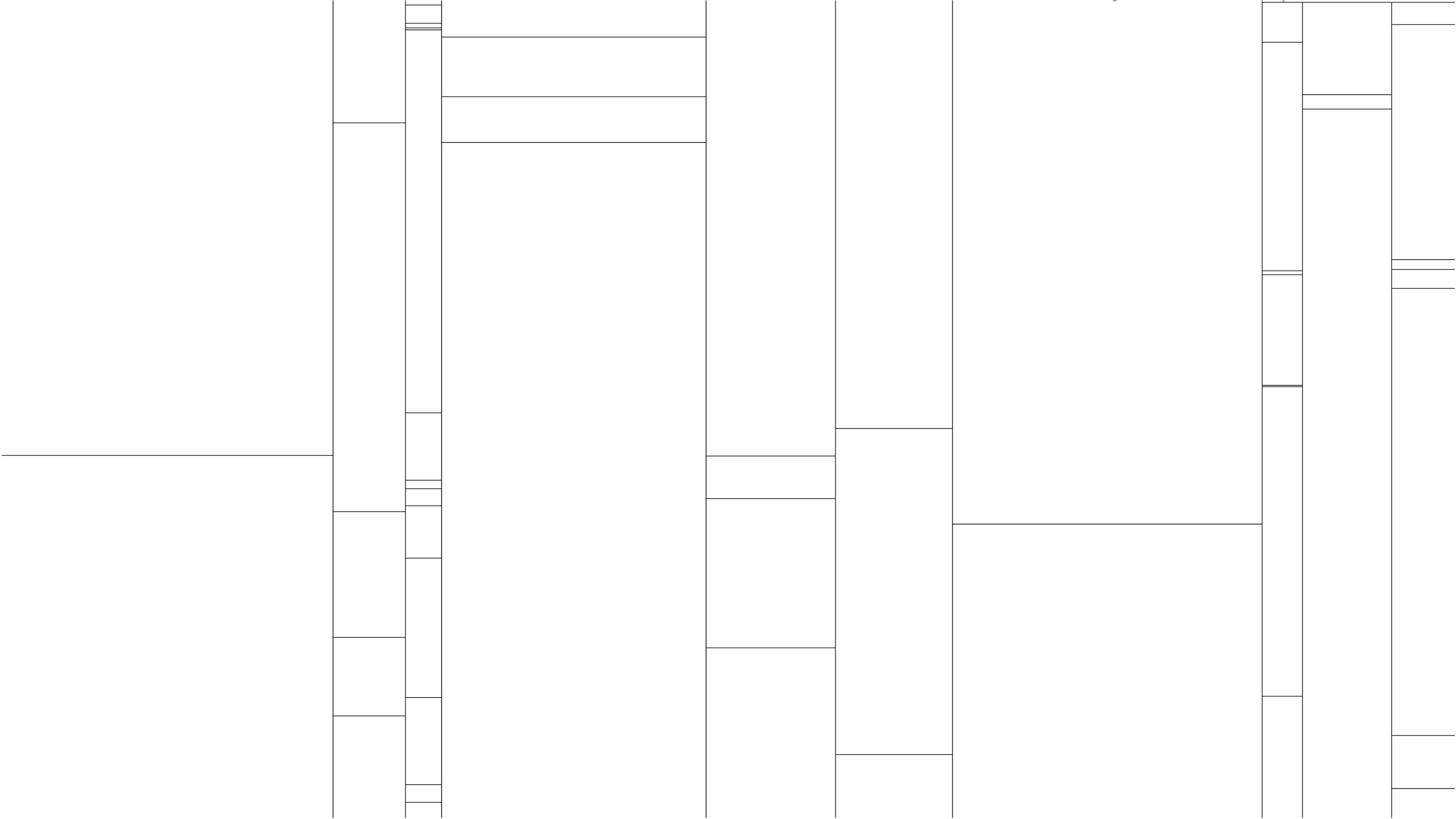


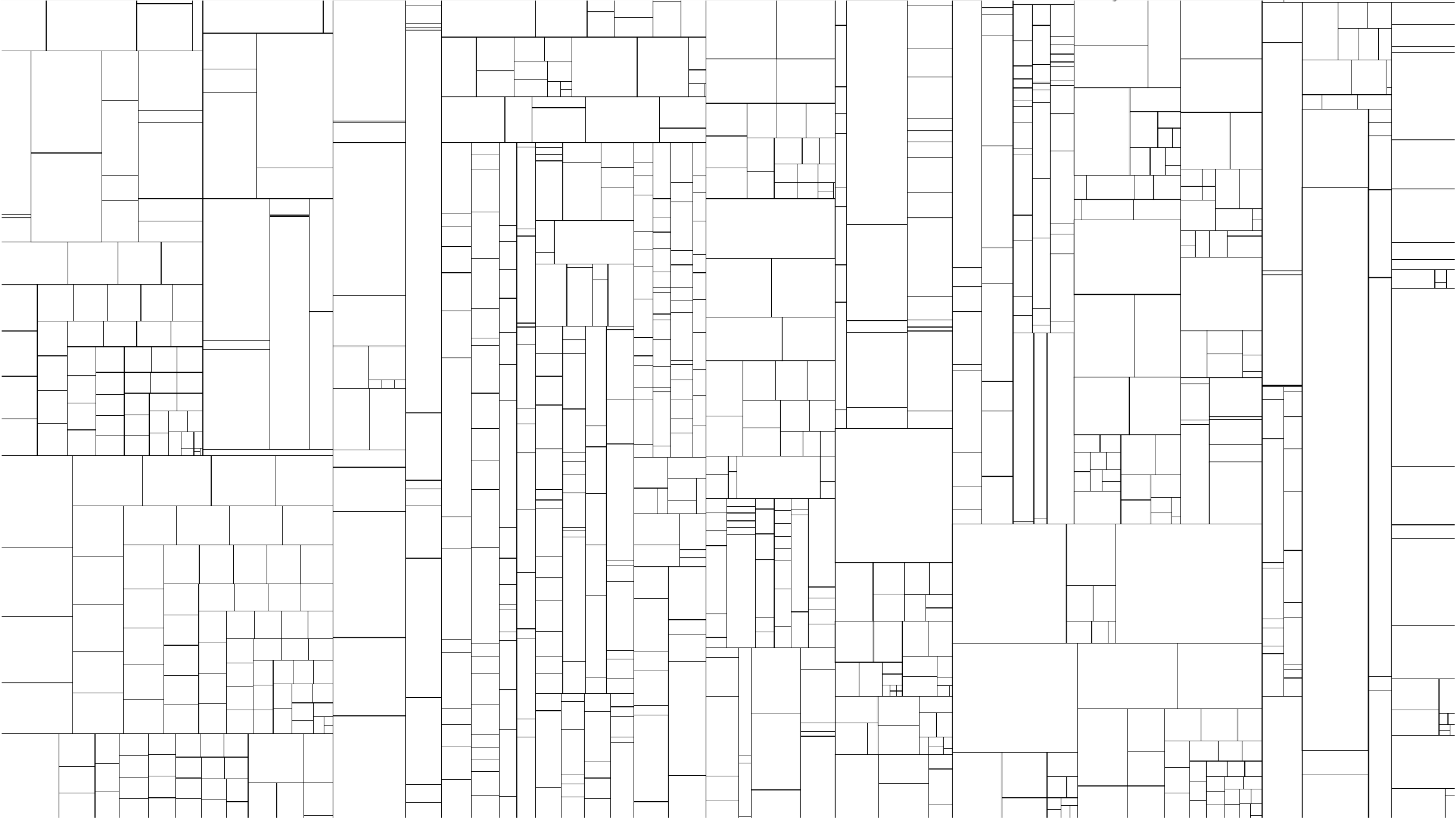
Test Selection for a Quality Gate

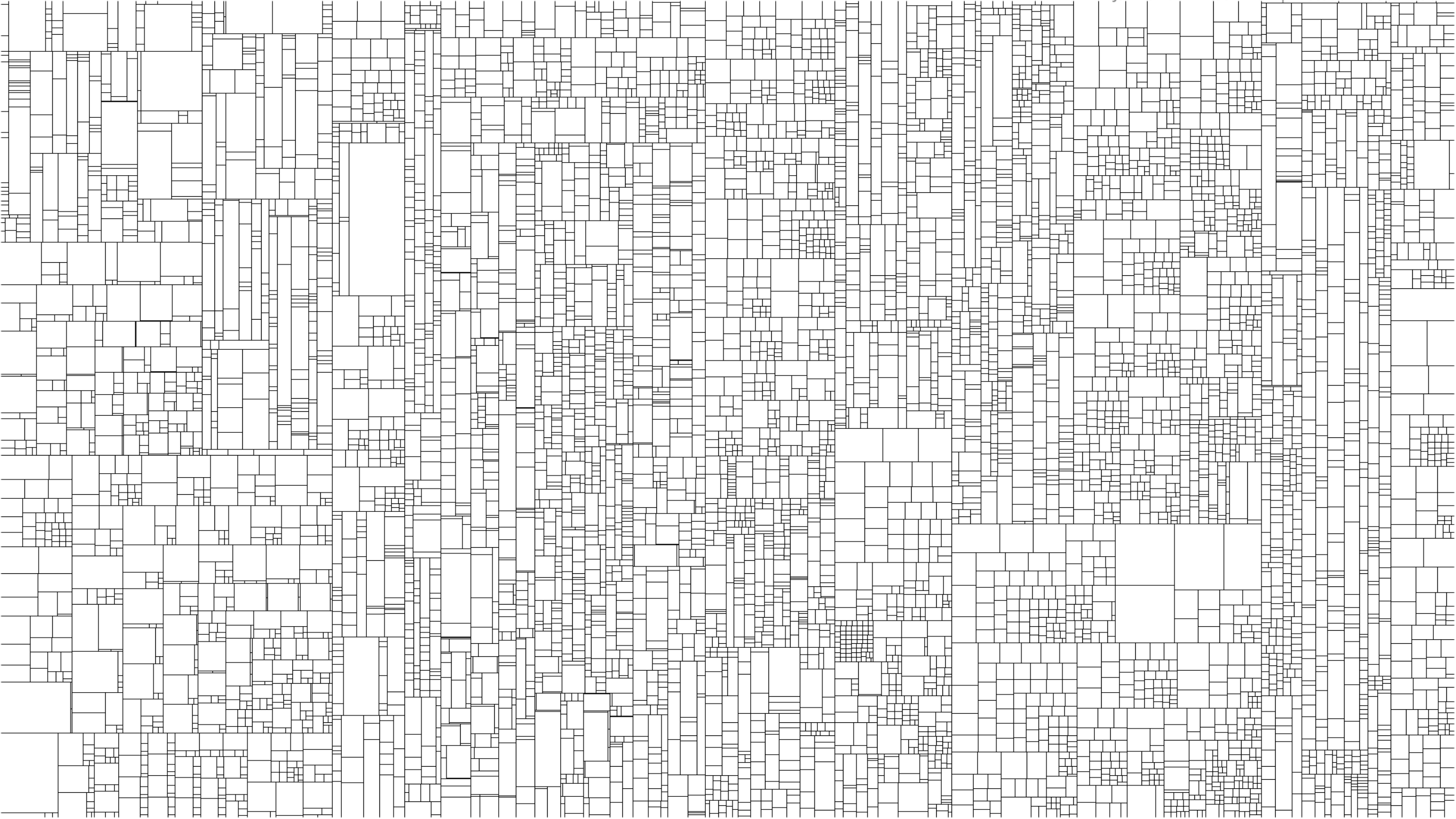


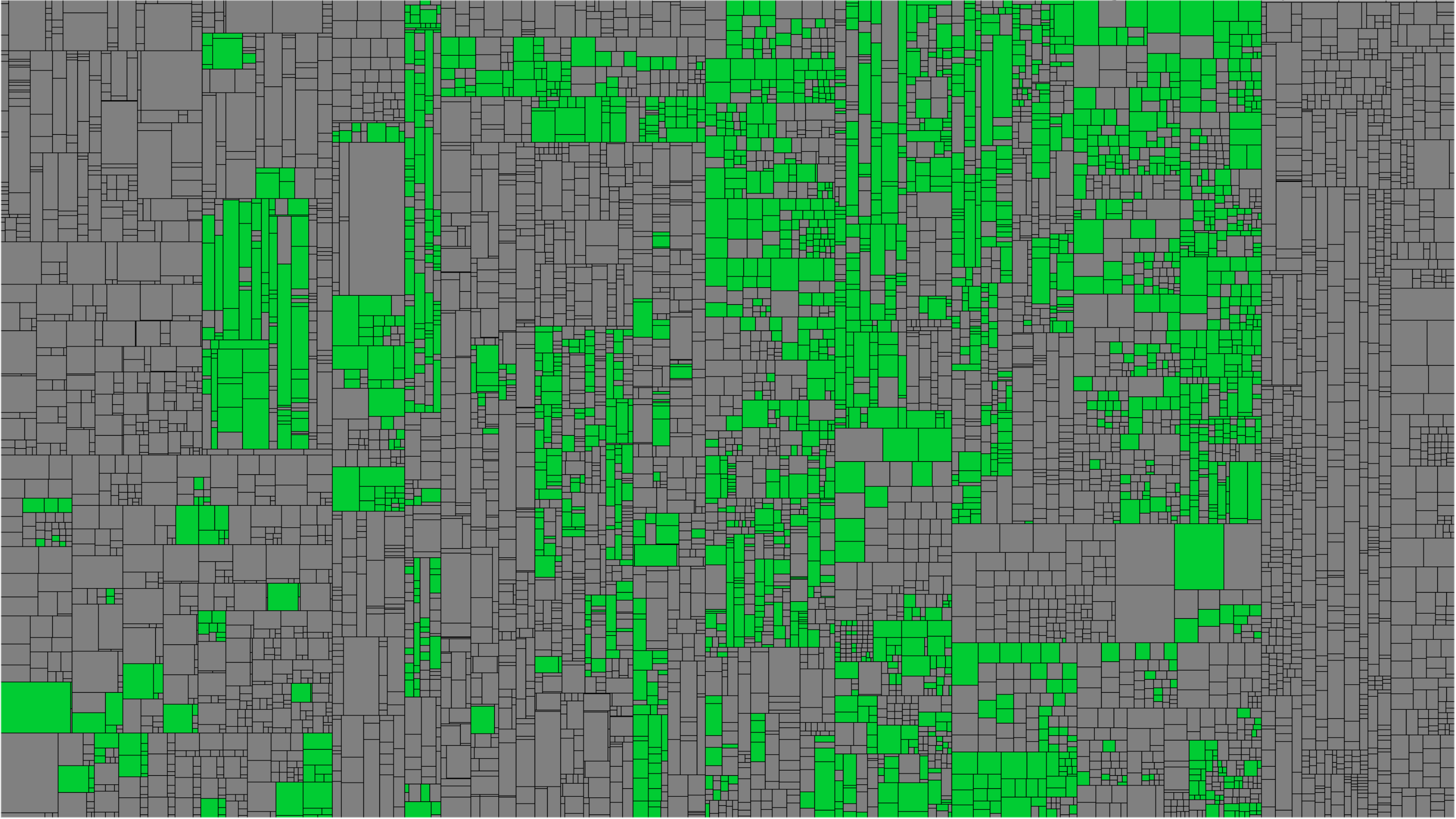


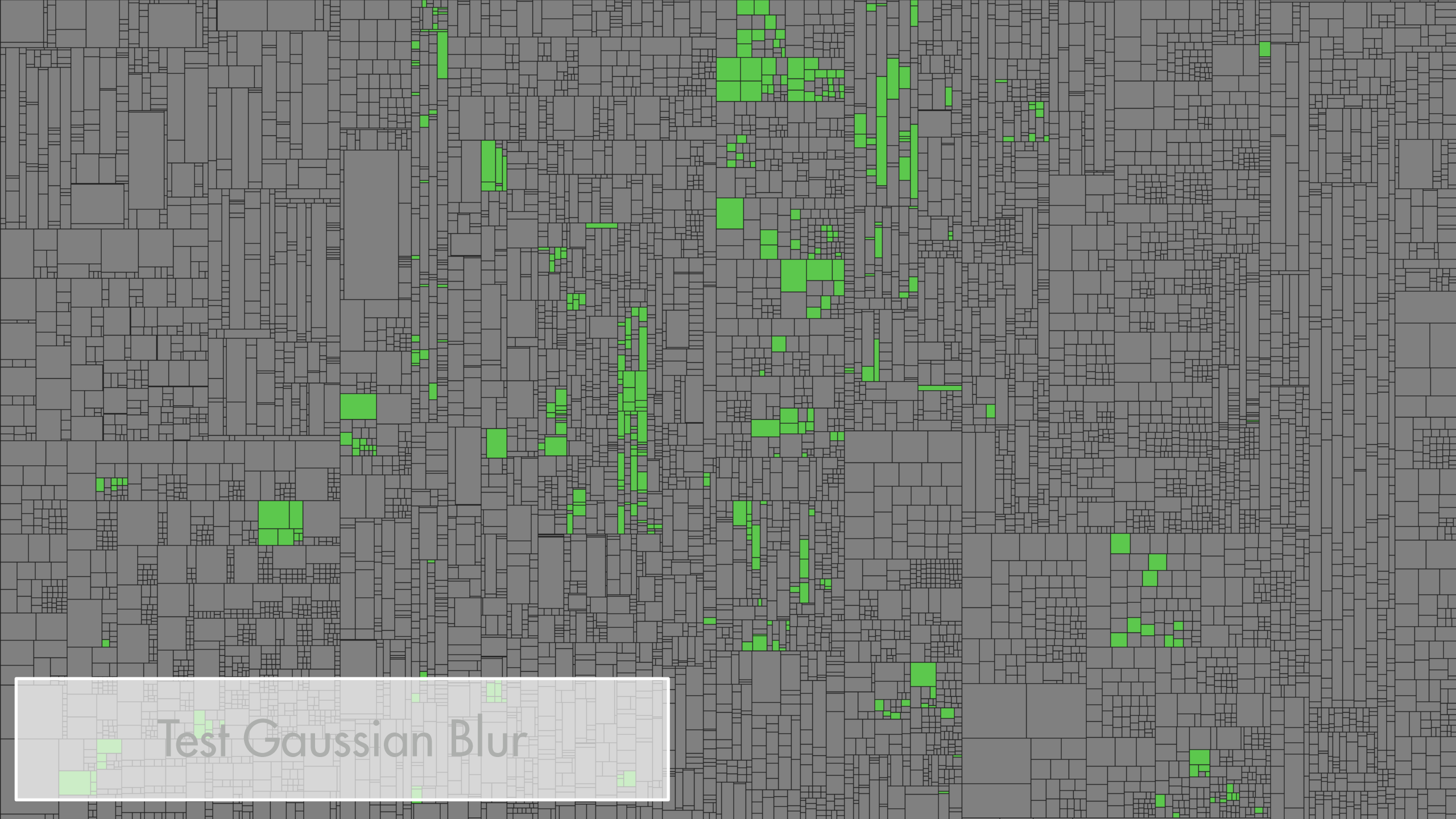

```
114
115    private static void createAndShowGUI(String[] args) {
116        assert calledOnEDT() : threadInfo();
117
118        Messages.setMsgHandler(new GUIMessageHandler());
119
120        // GlobalKeyboardWatch.showEventsSlowerThan(100, TimeUnit.MILLISECONDS);
121
122        Theme theme = Themes.DEFAULT;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











Test Gaussian Blur



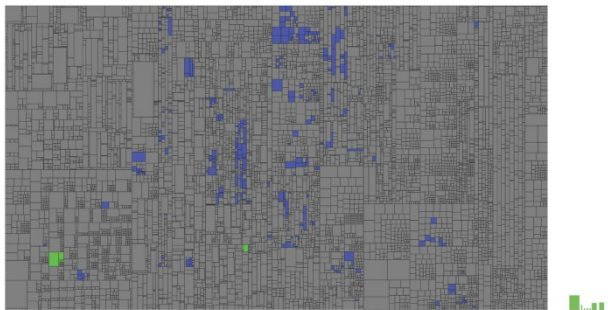
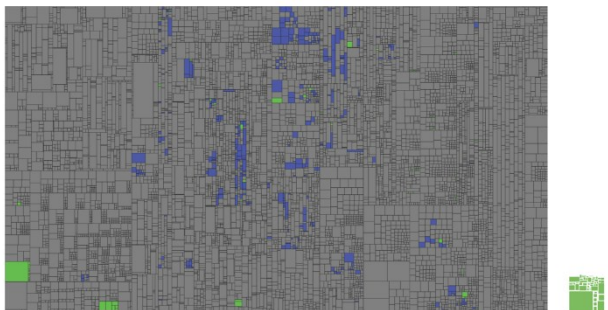
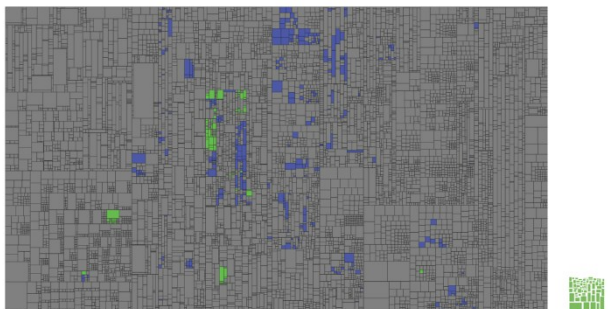
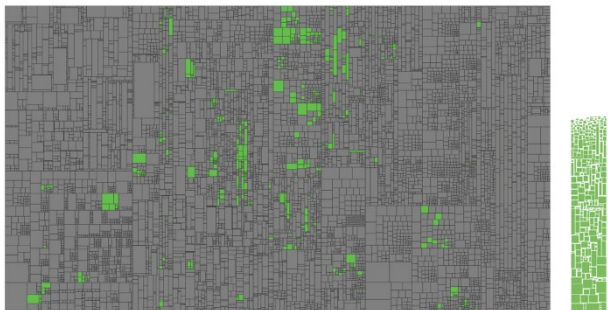
Test Motion Blur



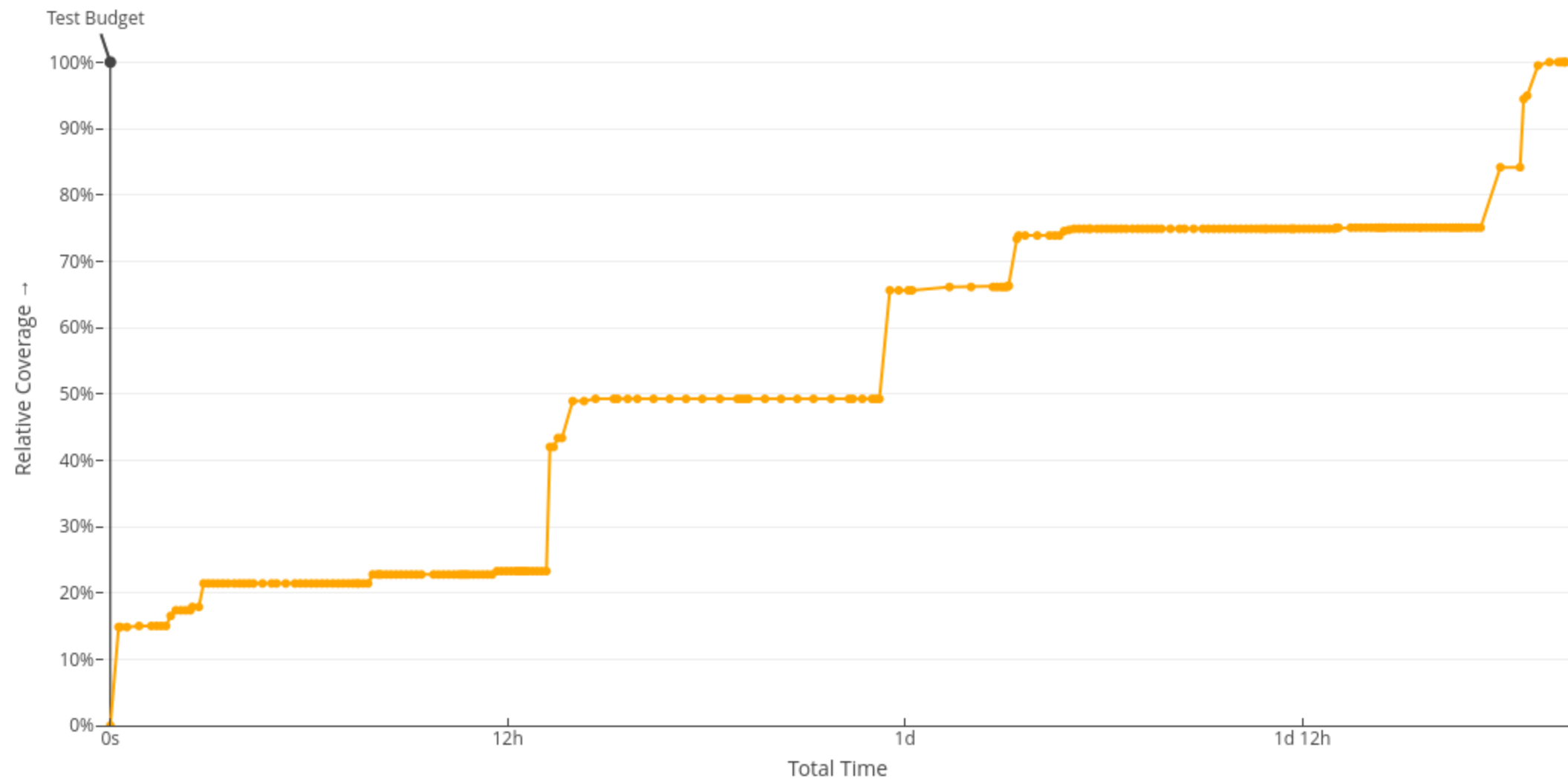
Test Lens Blur



Test Smart Blur



Coverage over Time ?



Results for Test Query & Budget Restriction

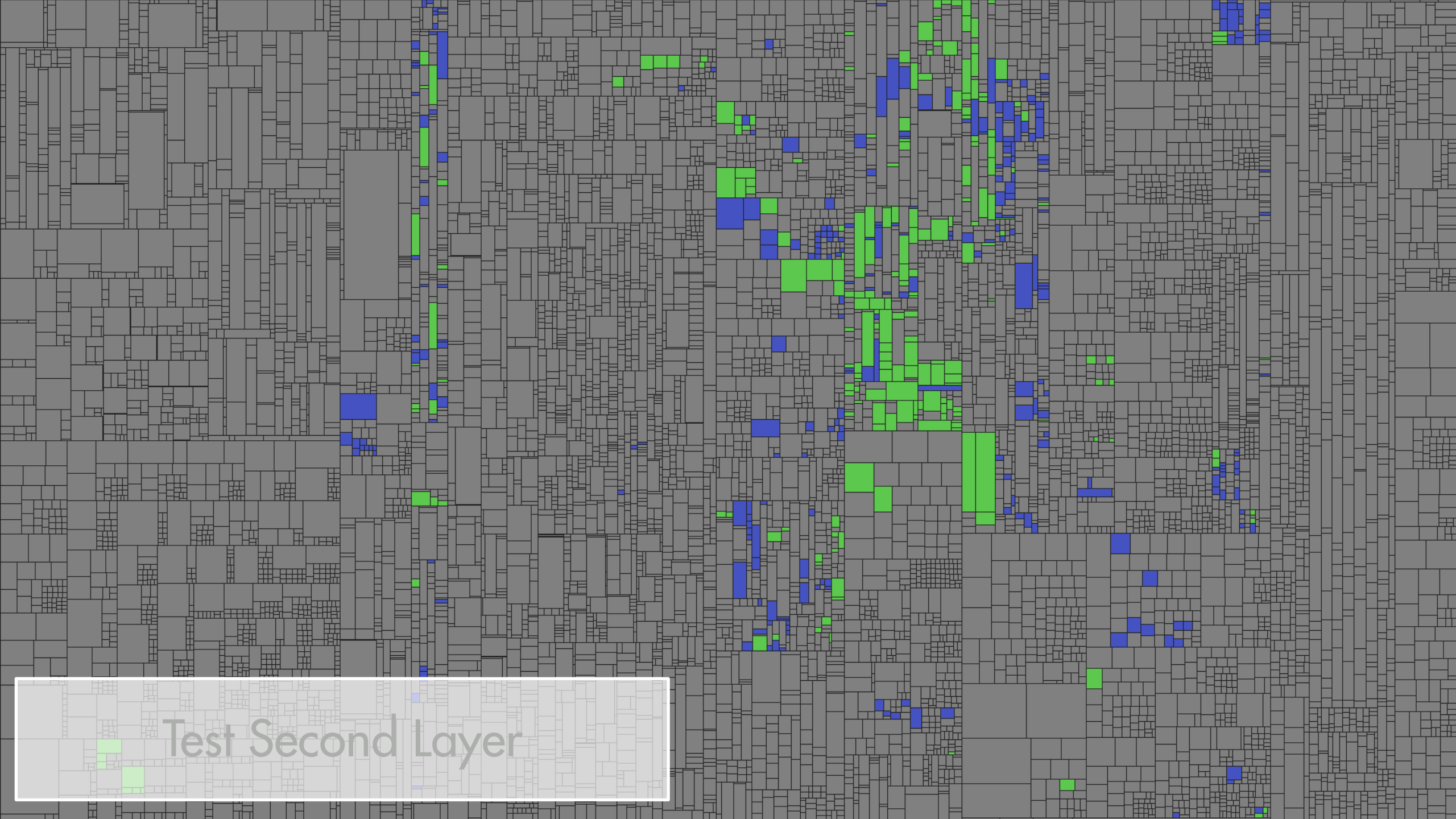
Relative Coverage: 0%, Selected Tests: 0 out of 236 (0%)



Test Create and Modify
Selection



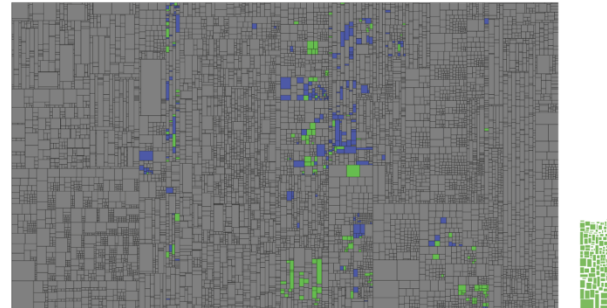
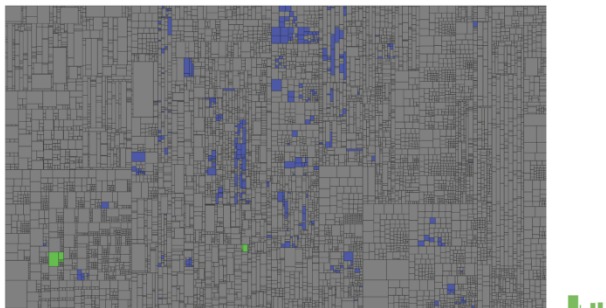
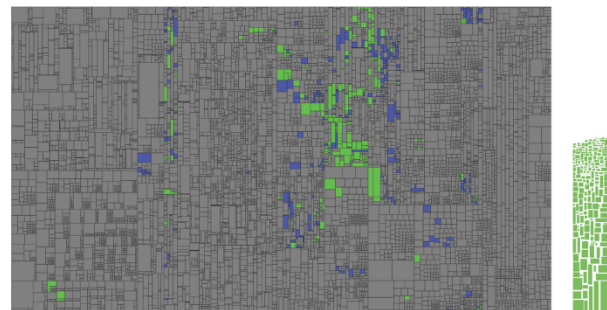
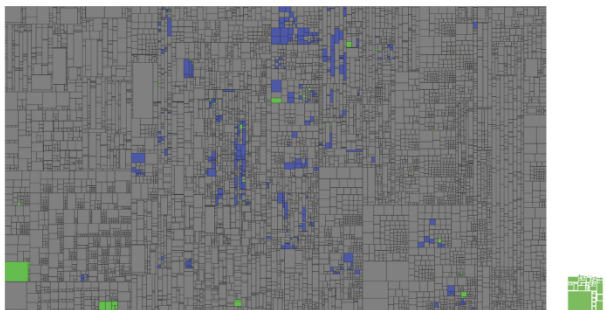
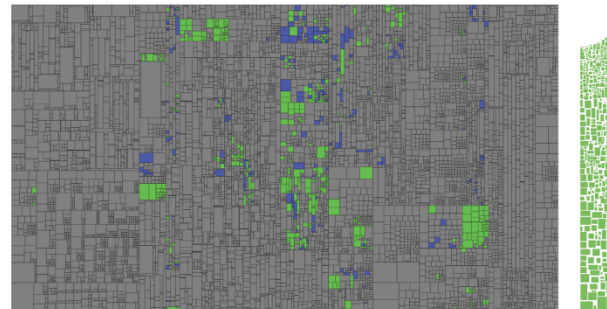
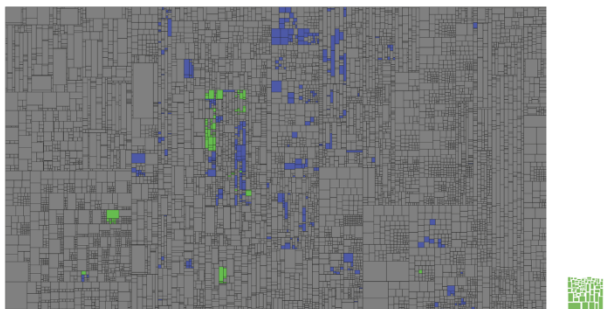
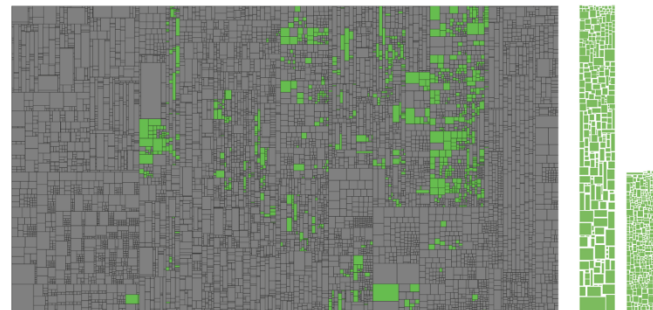
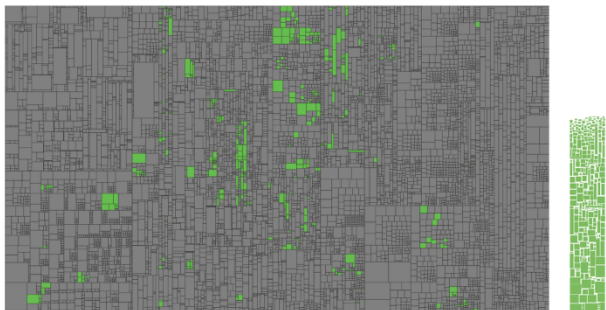
Test Change View Settings



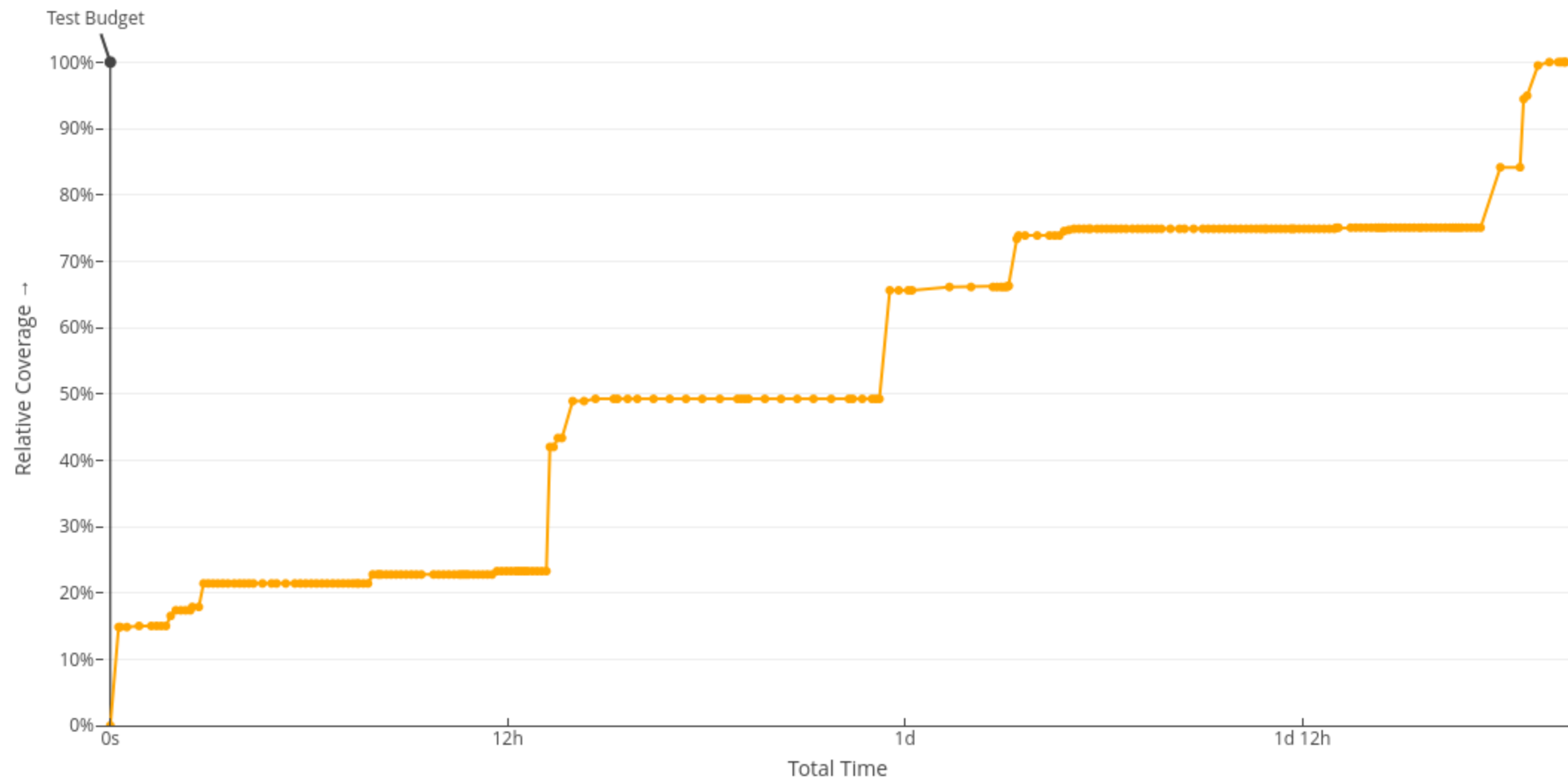
Test Second Layer



Test Save Image



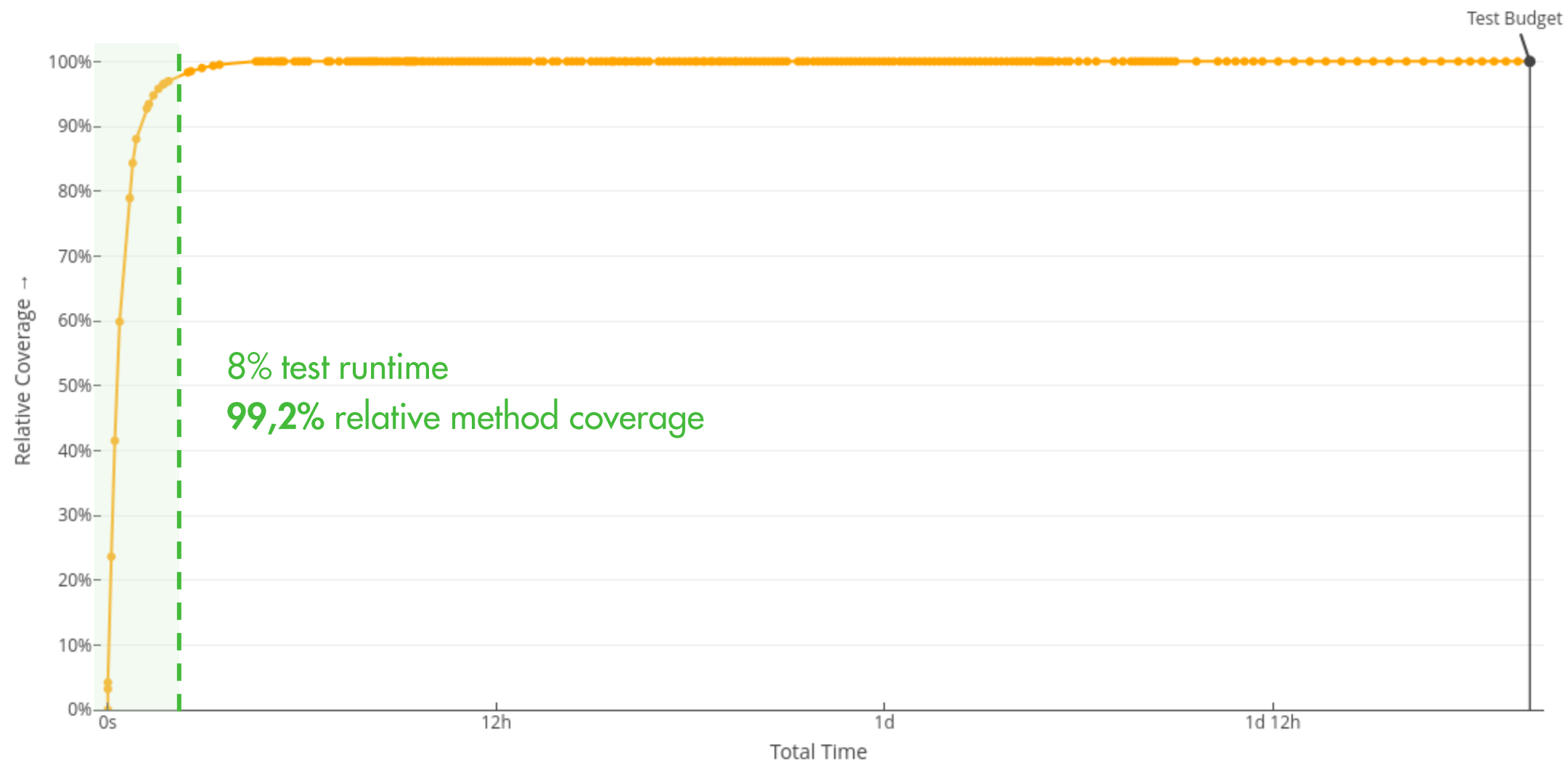
Coverage over Time ?



Results for Test Query & Budget Restriction

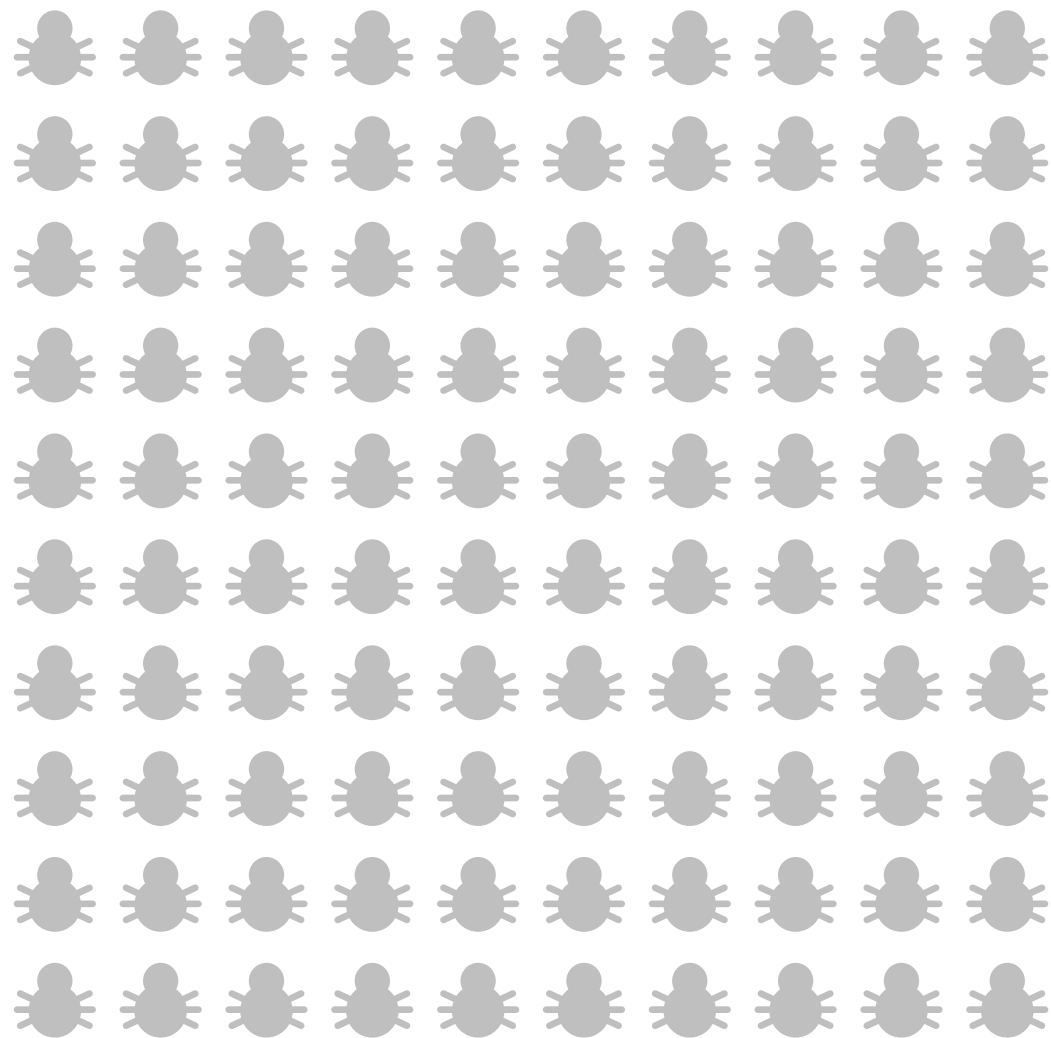
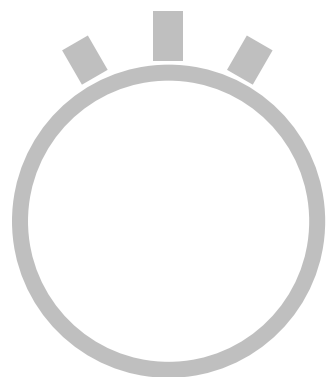
Relative Coverage: 0%, Selected Tests: 0 out of 236 (0%)

Coverage over Time ?

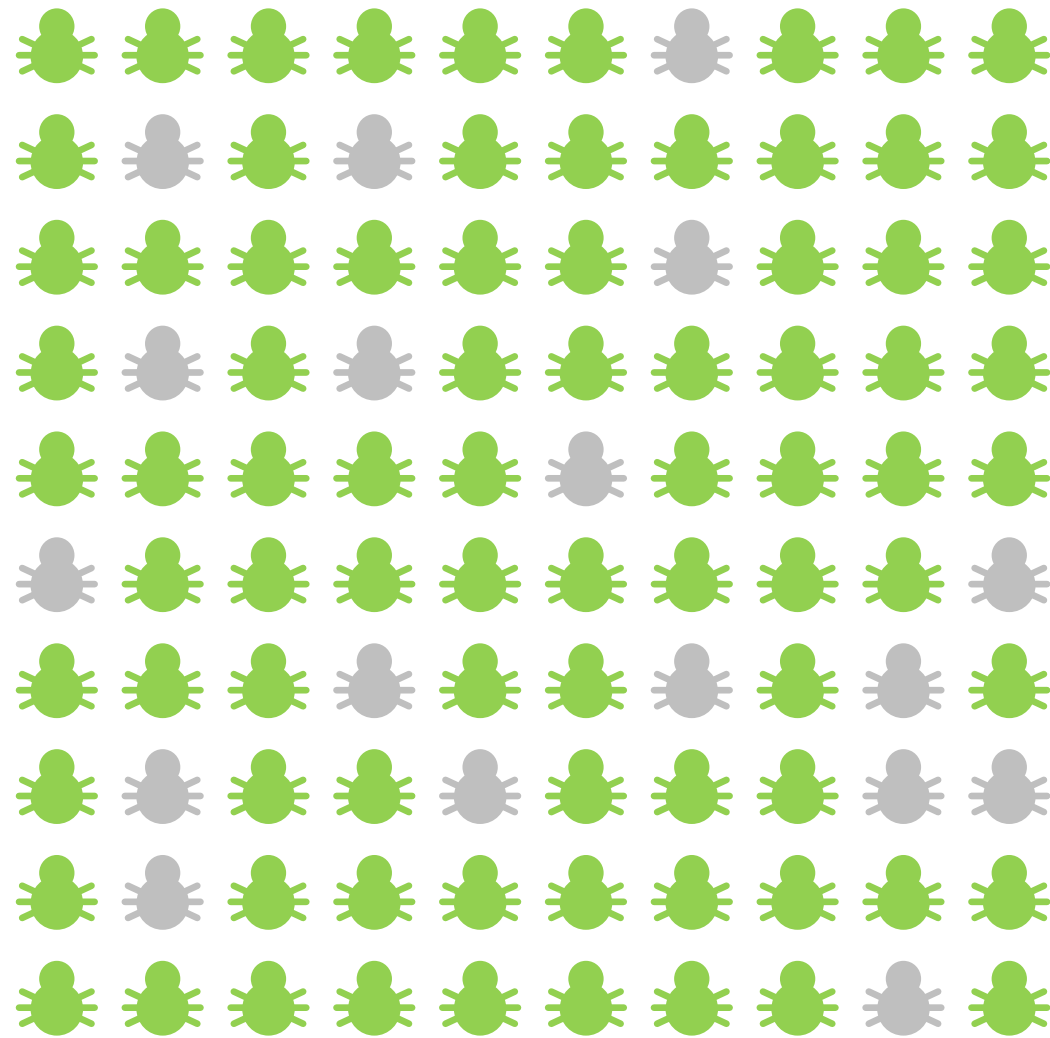


Results for Test Query & Budget Restriction

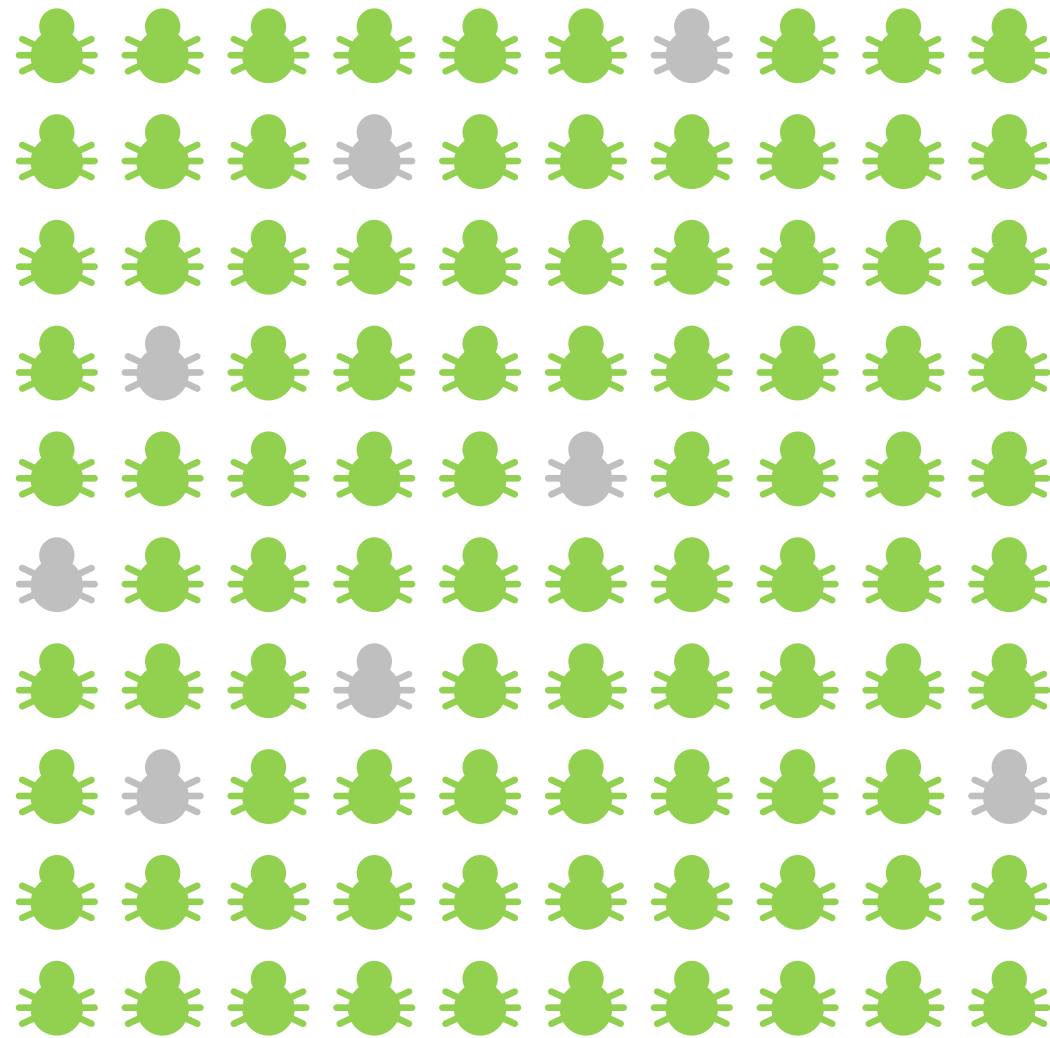
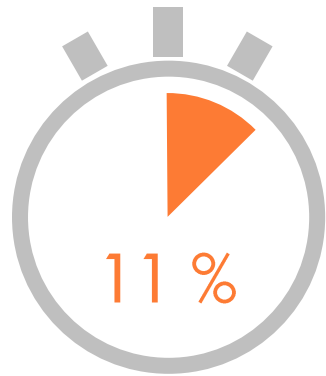
Relative Coverage: 100%, Selected Tests: 236 out of 236 (100%)



Pareto Test List



Pareto Test List



Industry Example

Before: nightly „Re-run all“

Feedback late next day



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



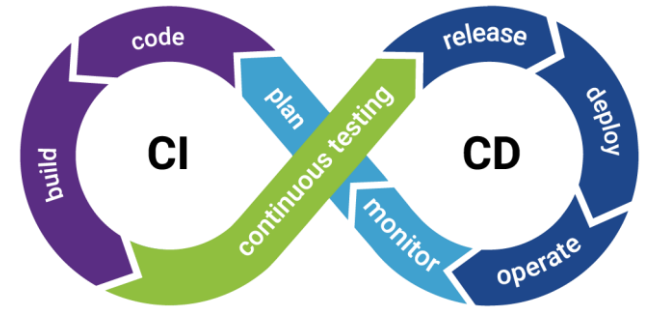
Nightly: „Re-run all“

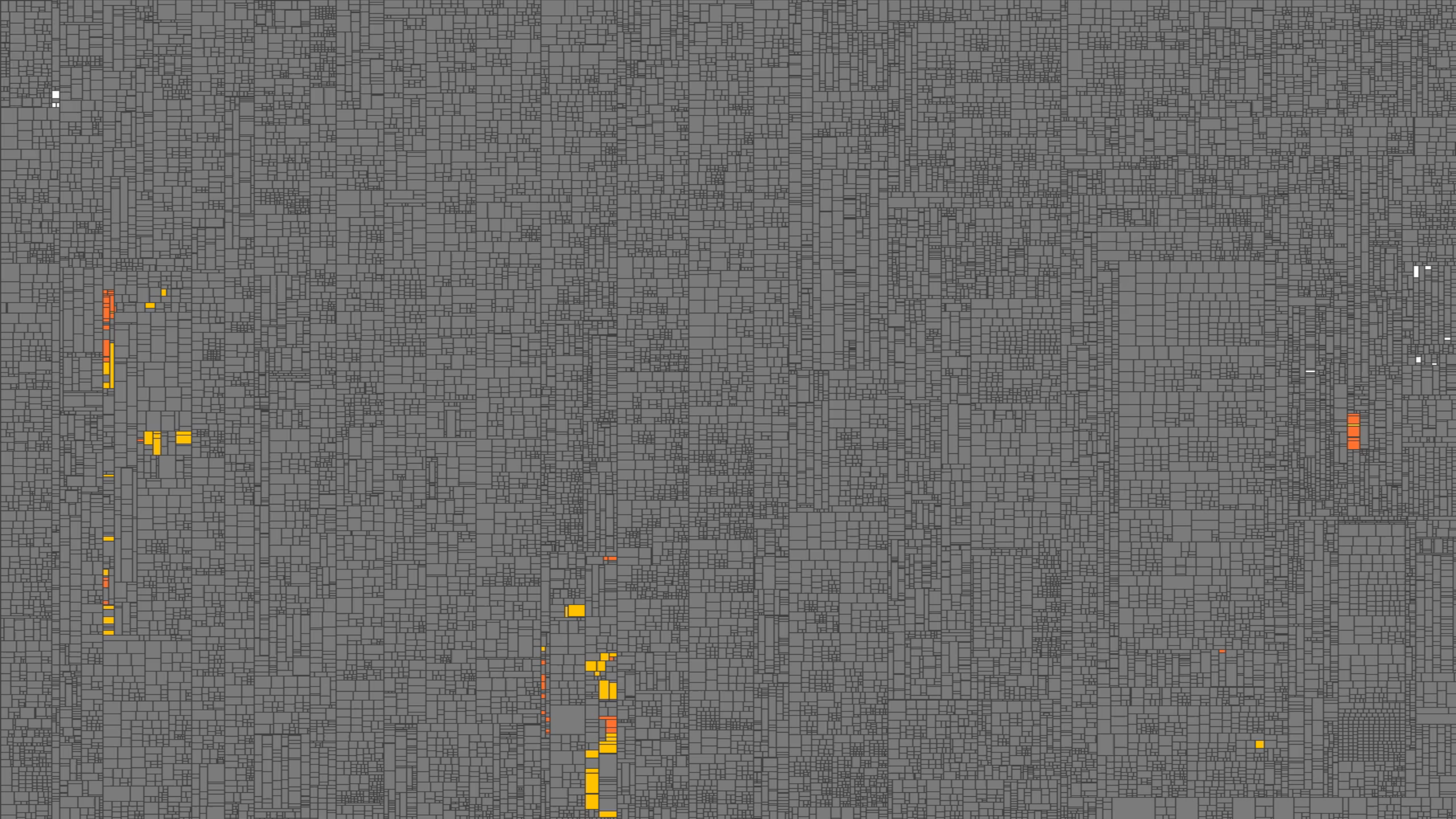


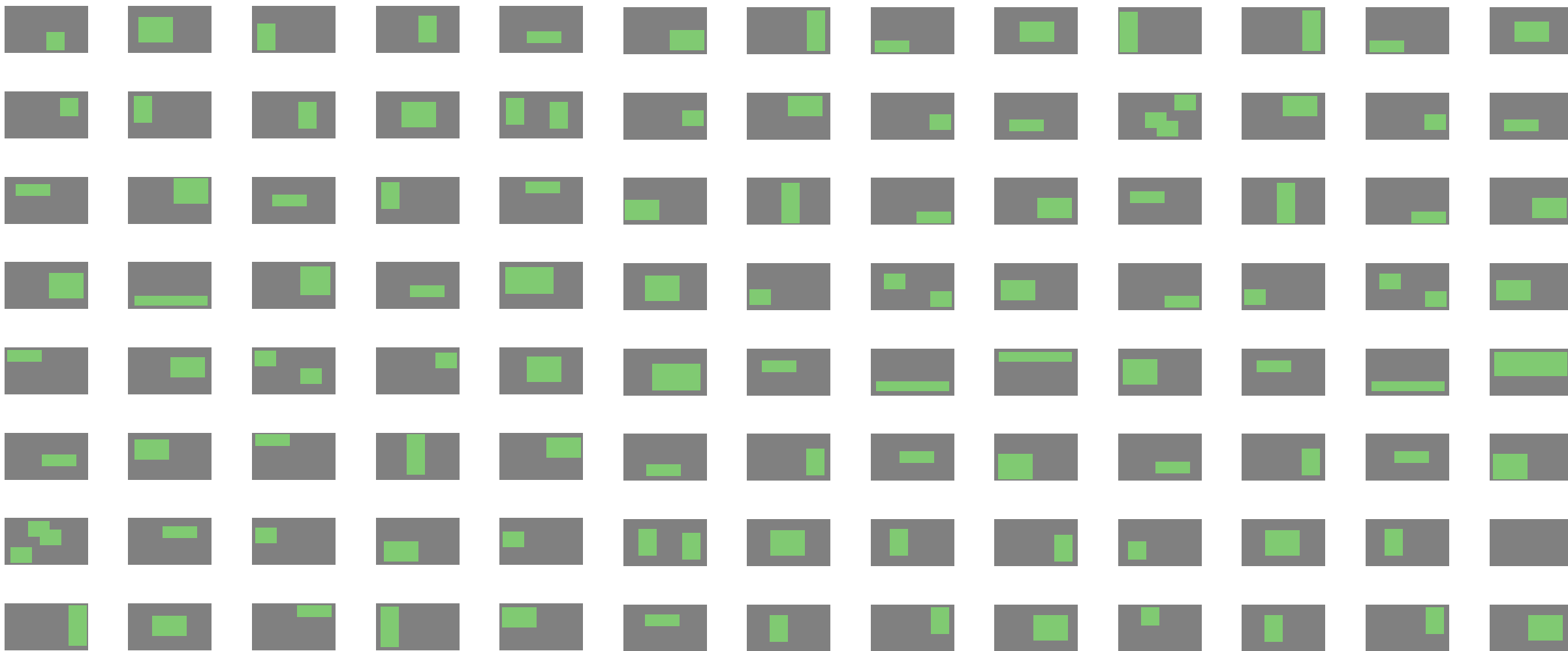
With only 5% test
time budget, we find
**70% of failing
commits**



Test Selection for Continuous Integration



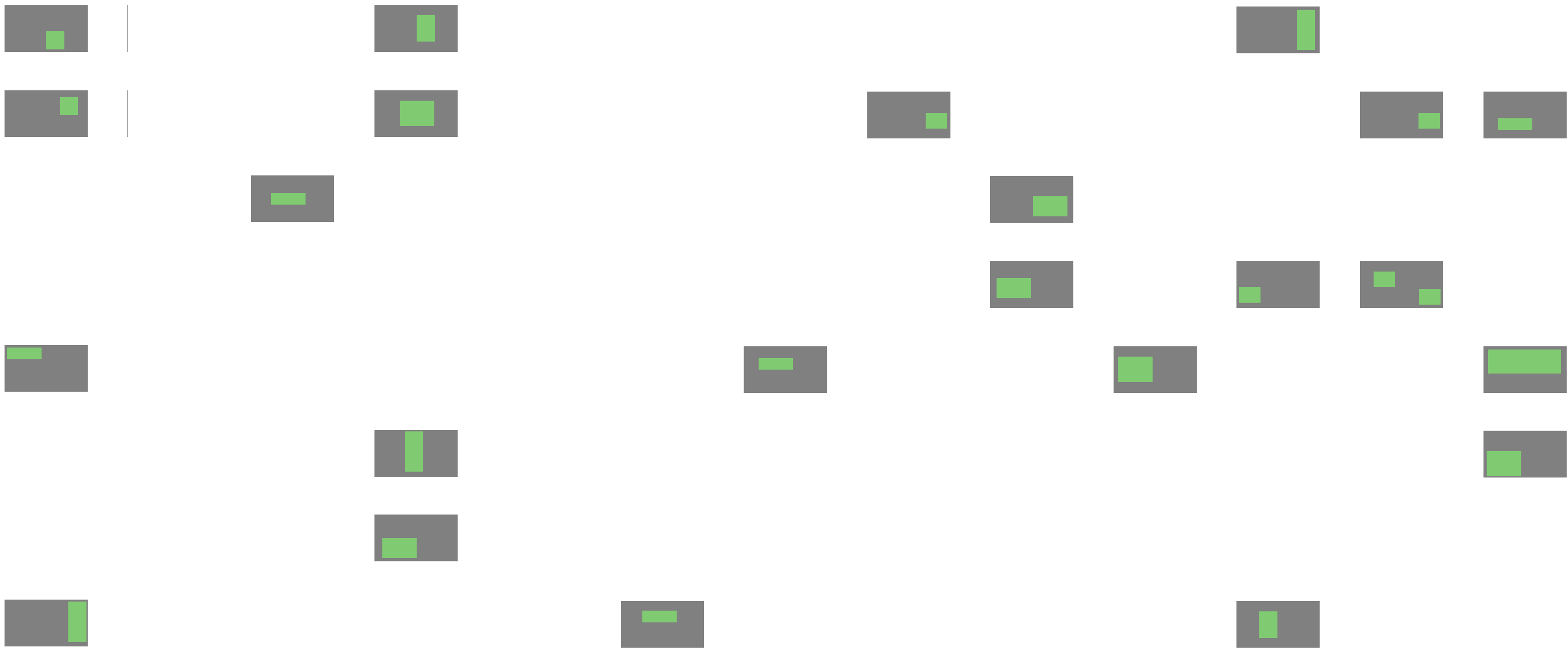




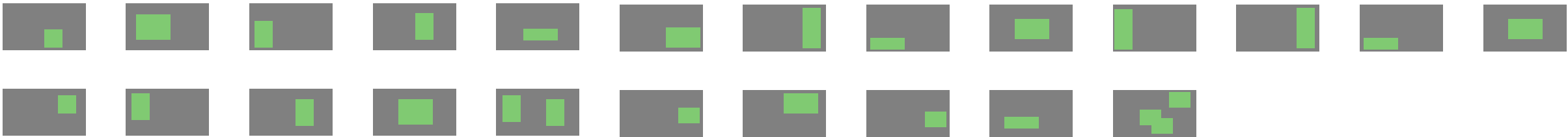
Step 1: Selection of Impacted Tests



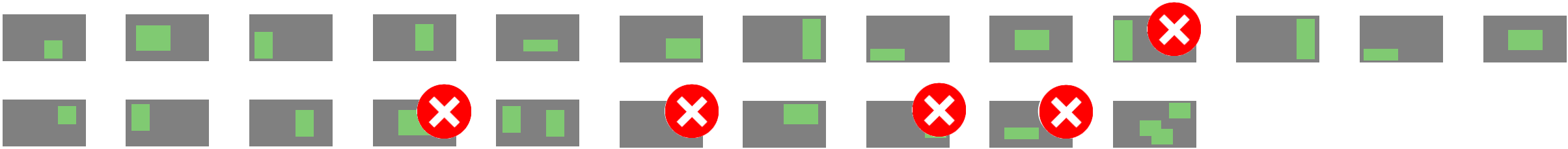
Step 1: Selection of Impacted Tests



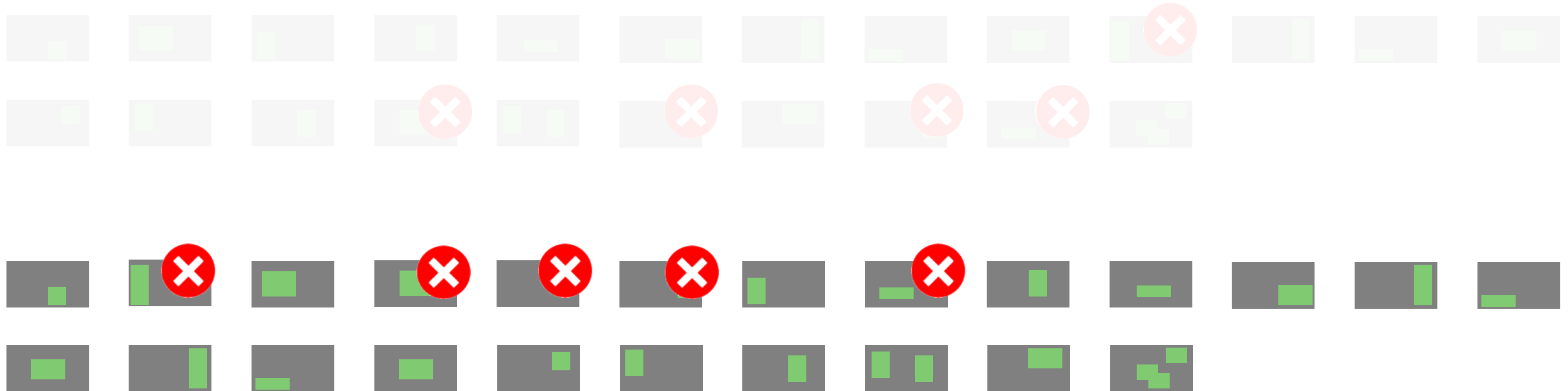
Step 1: Selection of Impacted Tests

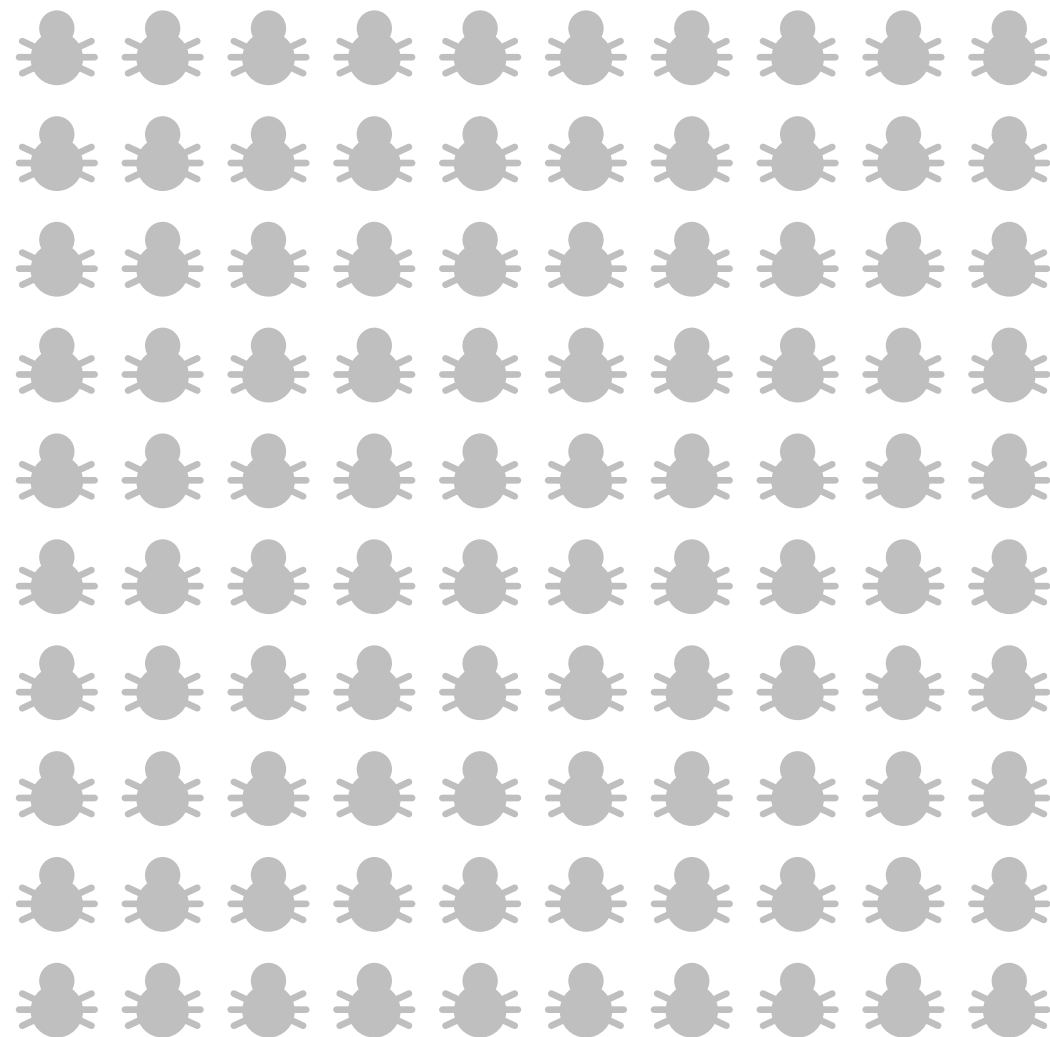
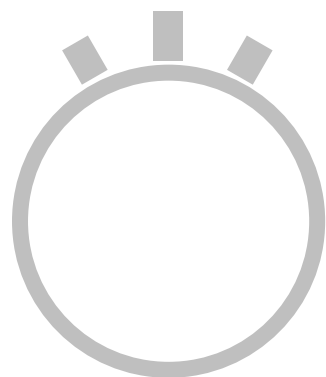


Step 2: Prioritization of Impacted Tests



Step 2: Prioritization of Impacted Tests





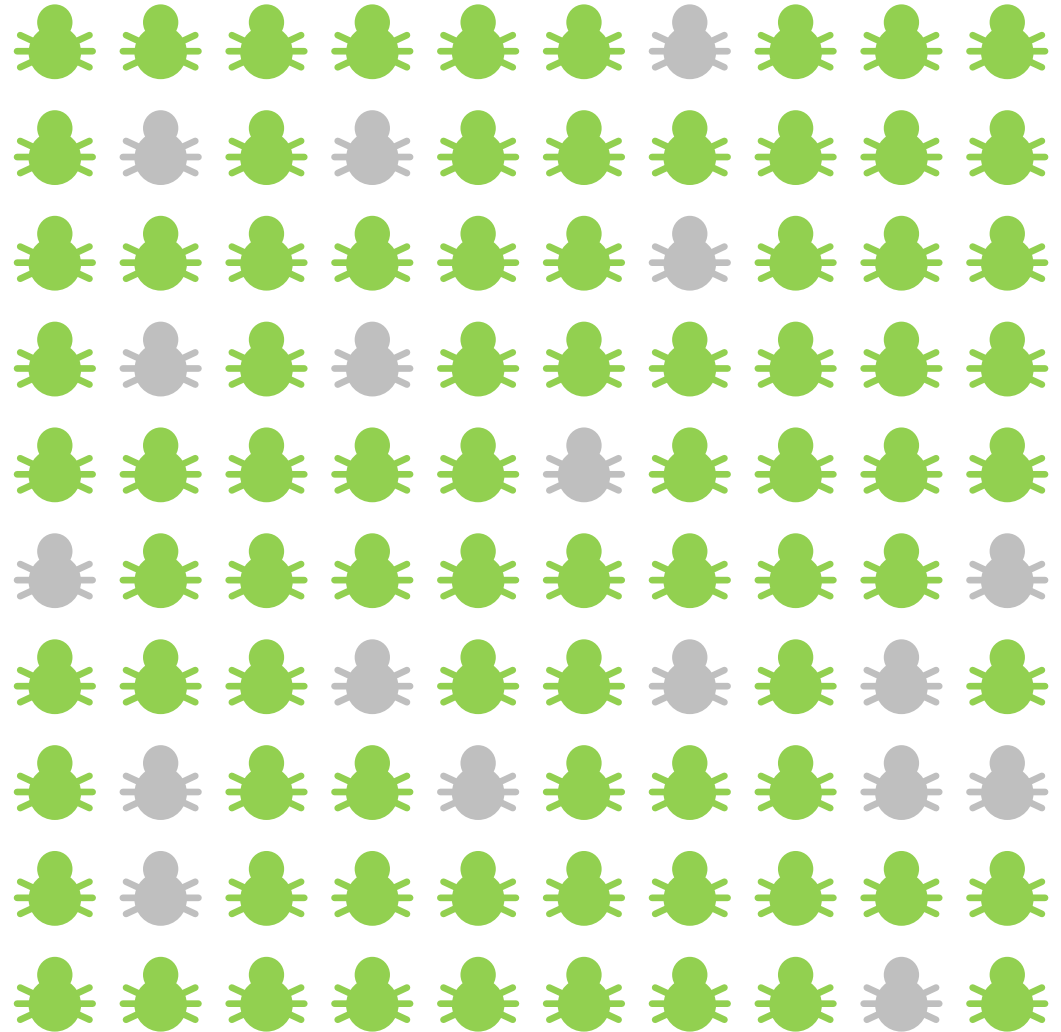
Test Impact Analysis



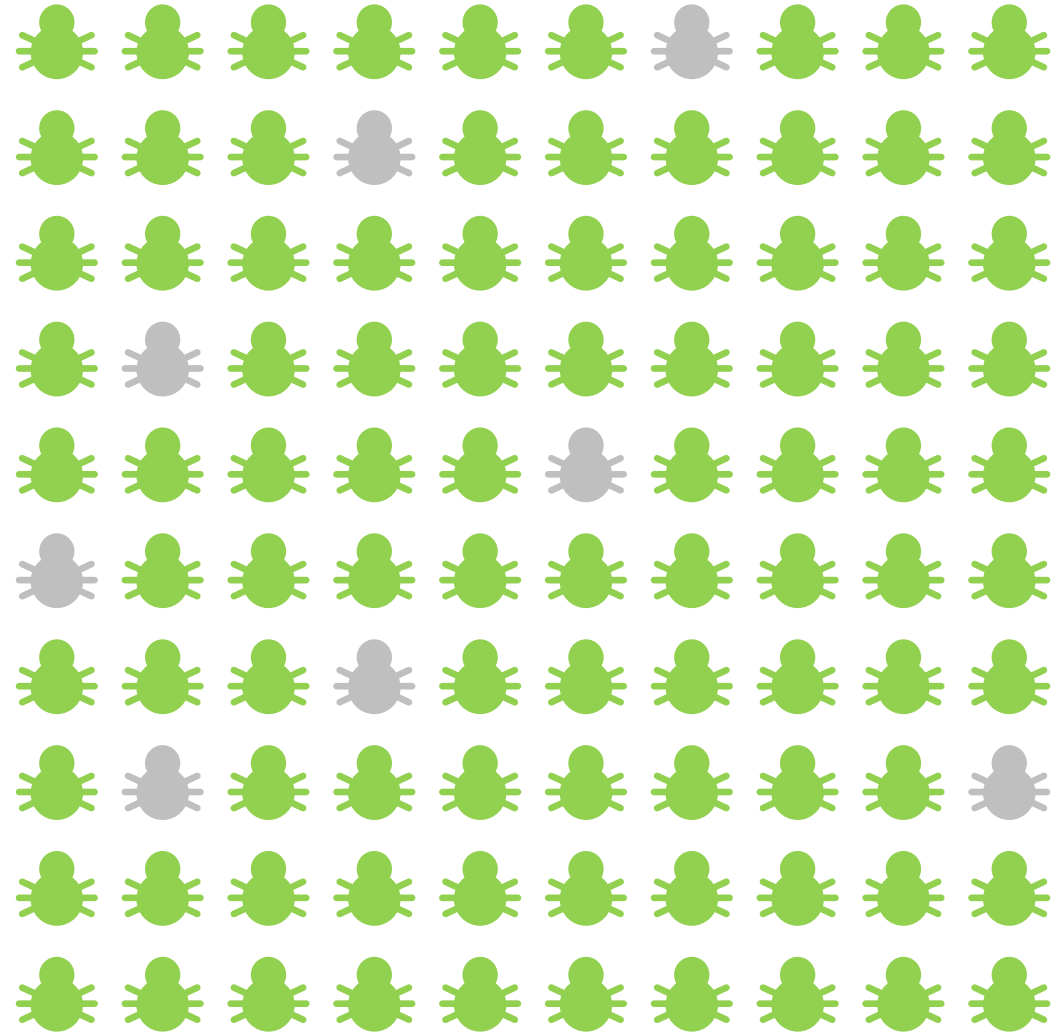
1 %



80 %



Test Impact Analysis



Industry Example



Before

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



Goal

Test feedback from the CI in less than 10 minutes.

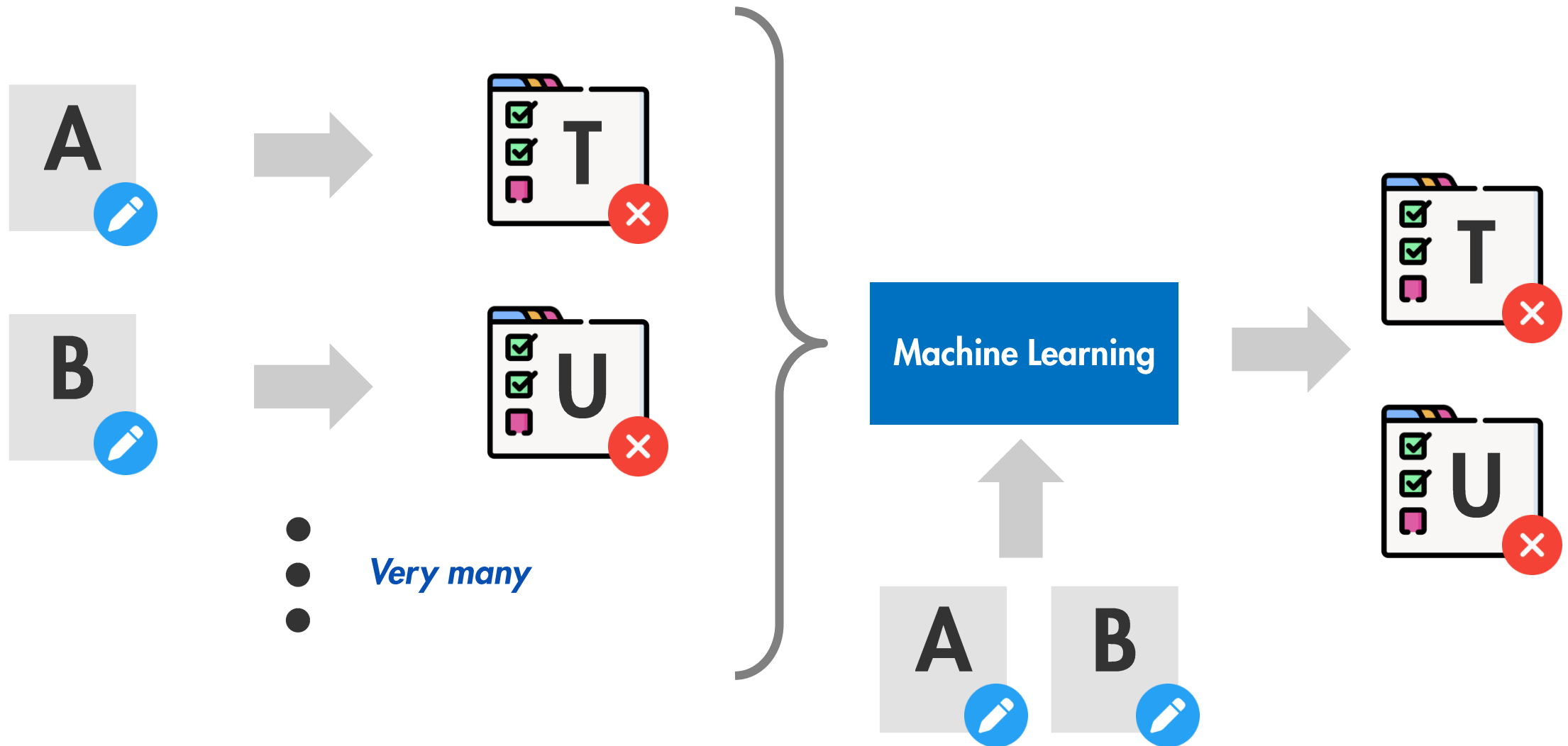


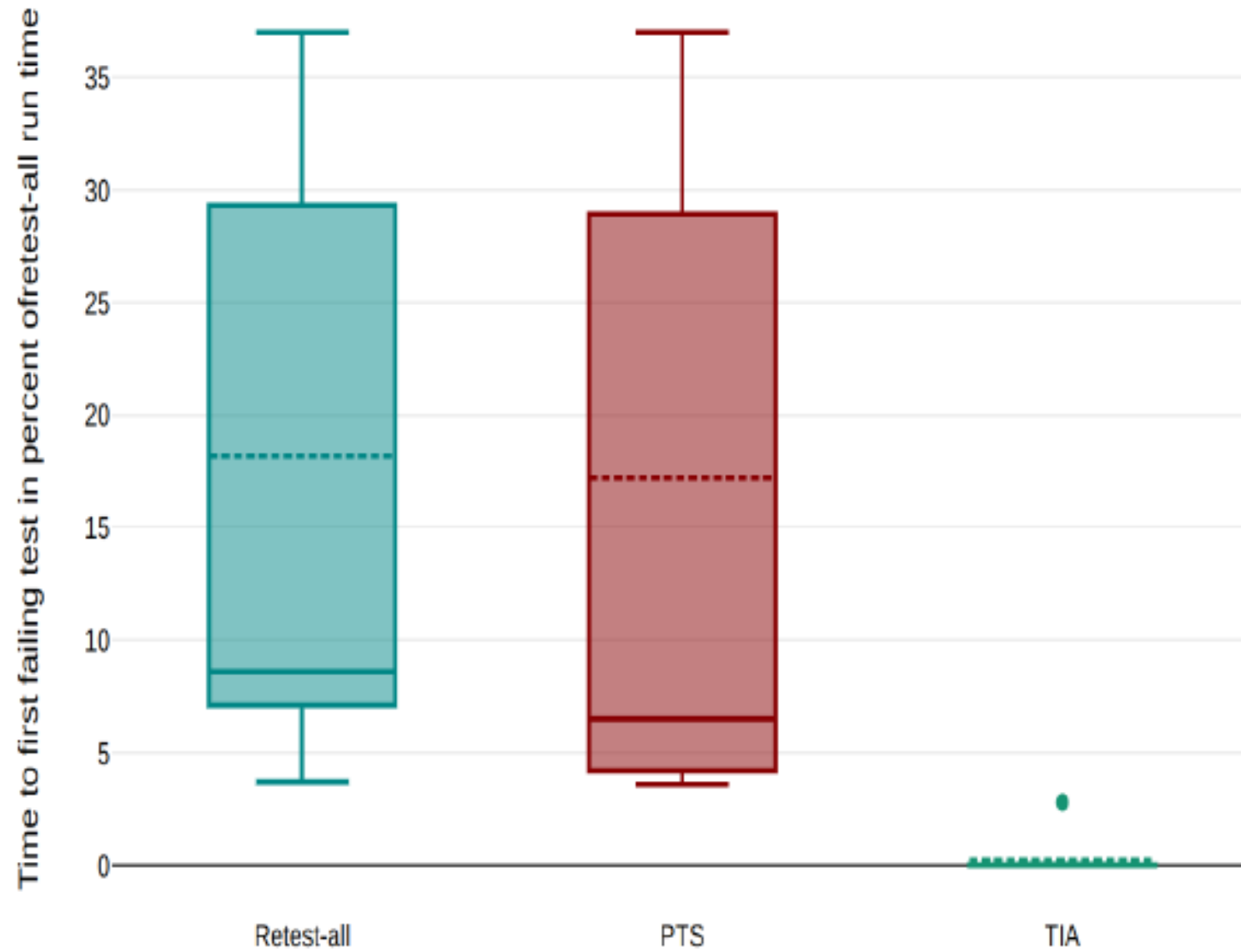
With TIA

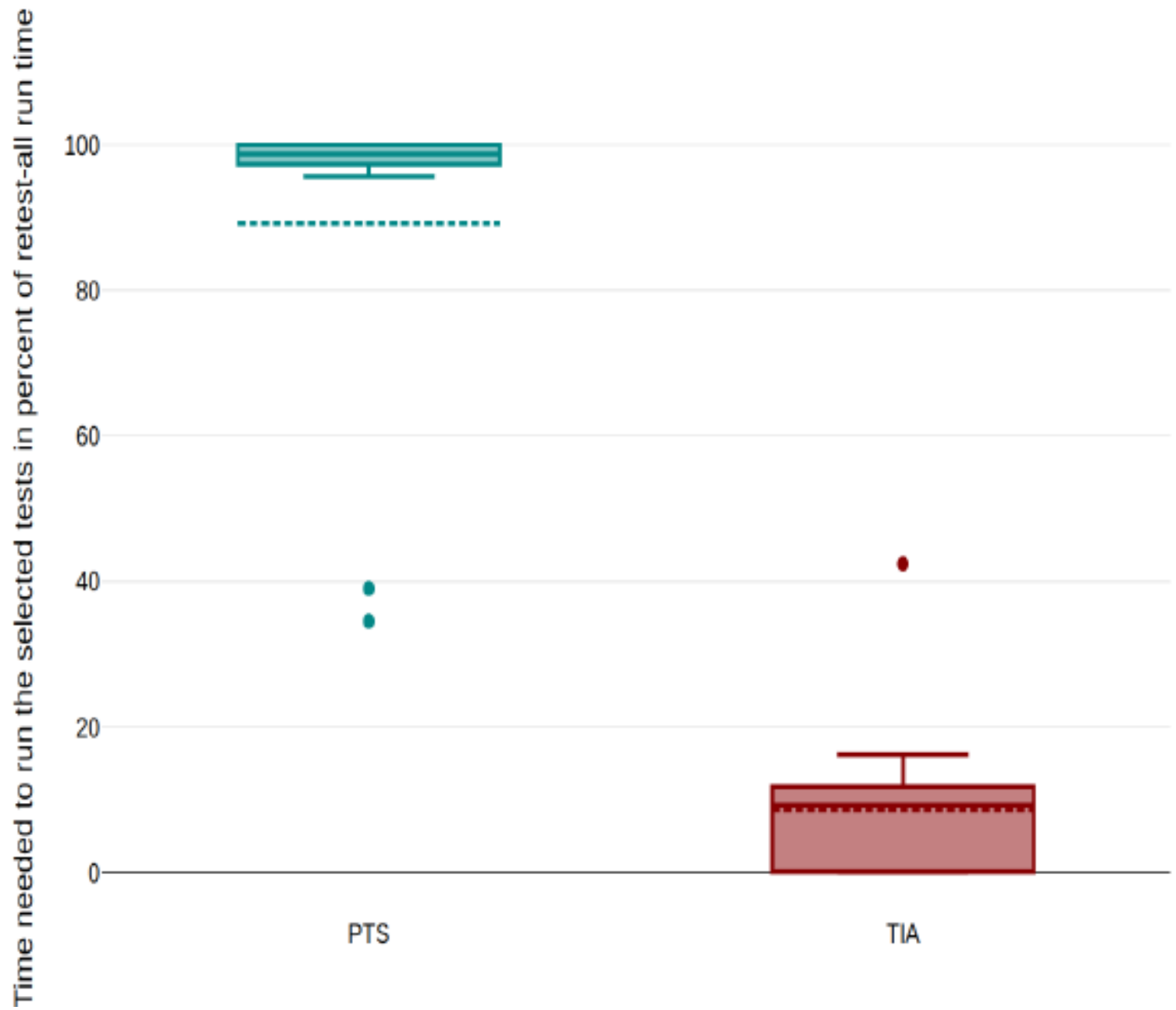
Since the introduction of test impact analysis years ago, test feedback always arrives within 10 minutes, although the number of tests increased 57% since.

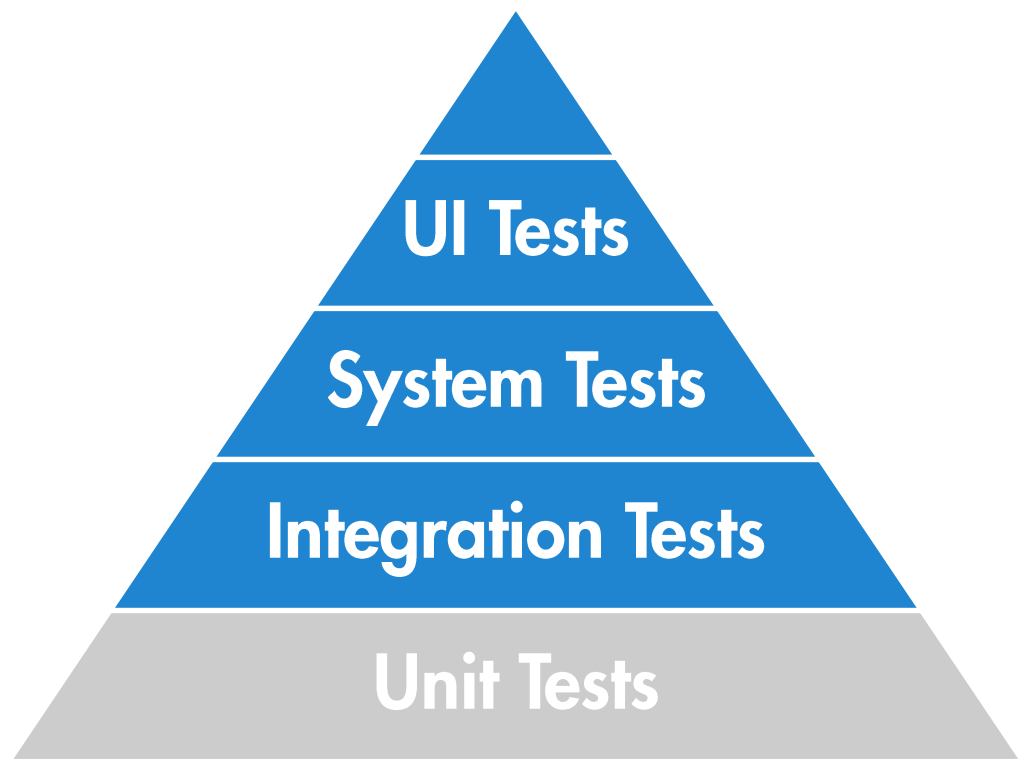
How about
without Test-Coverage?

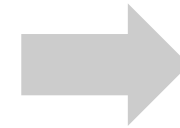
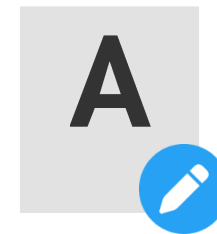
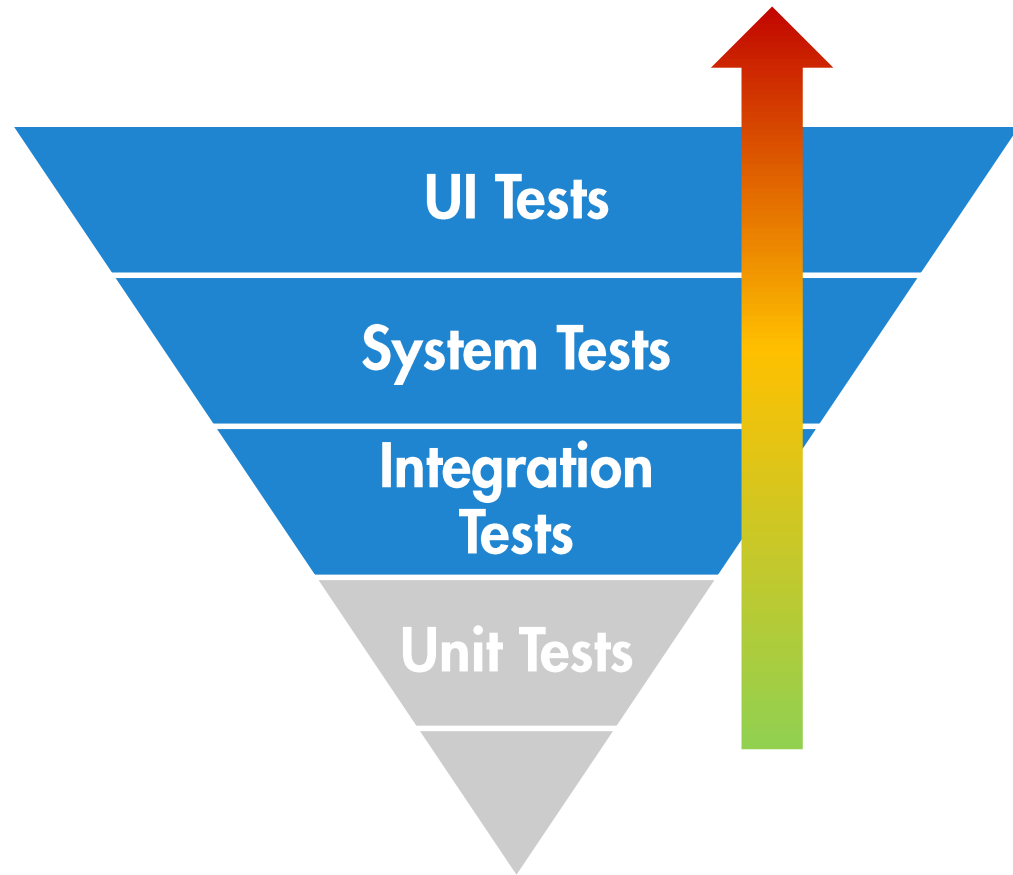
Predictive Test Selection











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);
```

Changed Code

Code Test

```

43     LOG.debug("Debit Transaction from Account: Account Updated.");
42 }
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);

```

```

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.getId()
6         AccountTransactionList transactionList = this.getTransaction
5         assertThat(transactionList).isEmpty()
4     }
3
2 }
1
21

```

Changed Code

Cucumber Test

```
43 LOG.debug("Debit Transaction from Account: Account Updated.");
42 }
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.getAccount
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.getAccount
12 toAt.setTransactionType(transactionTypeRepository.findByCode(Consta
11 creditTransaction(toAccount, toAt);
10
9 LOG.debug("Transfer Between Accounts: Accounts Updated.");
8 }
7
6
5 /*
4  * Get Account object by Id
3  */
2 public Account getAccountById(Long id) {
1 Optional<Account> act = accountRepository.findById(id);
```

```
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

Robot Test

```

43     LOG.debug("Debit Transaction from Account: Account Updated.");
42 }
41
40 }
39
38 /*
37  * Transfer amount between two accounts
36  *
35  * Accounts should be full objects. With that said, the objects are full
34  *
33  * AccountTransaction can be a partial object but must contain the transaction
32  */
31 public void transfer(Account fromAccount, Account toAccount, 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.getAccountName()
21     fromAt.setTransactionType(transactionTypeRepository.findByCode(Constants
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.getAccountName()
12     toAt.setTransactionType(transactionTypeRepository.findByCode(Constants
11     creditTransaction(toAccount, toAt);
10
9     LOG.debug("Transfer Between Accounts: Accounts Updated.");
8 }
7
6
5 /*
4  * Get Account object by Id
3  */
2 public Account getAccountById(Long id) {
1     Optional<Account> acct = accountRepository.findById(id);

```

```

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
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 /*
36  * Transfer amount between two accounts
35  *
34  * Accounts should be full objects. With that said, the objects are full
33  *
32  * AccountTransaction can be a partial object but must contain the transaction
31  */
30 public void transfer(Account fromAccount, Account toAccount, AccountTransaction accountTransaction) {
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.getAccountName() + ")");
21     fromAt.setTransactionType(transactionTypeRepository.findByCode(Constants.CREDIT_TRANSACTION_TYPE));
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.getAccountName() + ")");
12     toAt.setTransactionType(transactionTypeRepository.findByCode(Constants.DEBIT_TRANSACTION_TYPE));
11     creditTransaction(toAccount, toAt);
10
9     LOG.debug("Transfer Between Accounts: Accounts Updated.");
8 }
7
6
5 /*
4  * Get Account object by Id
3  */
2 public Account getAccountById(Long id) {
1     Optional<Account> acct = accountRepository.findById(id);
```

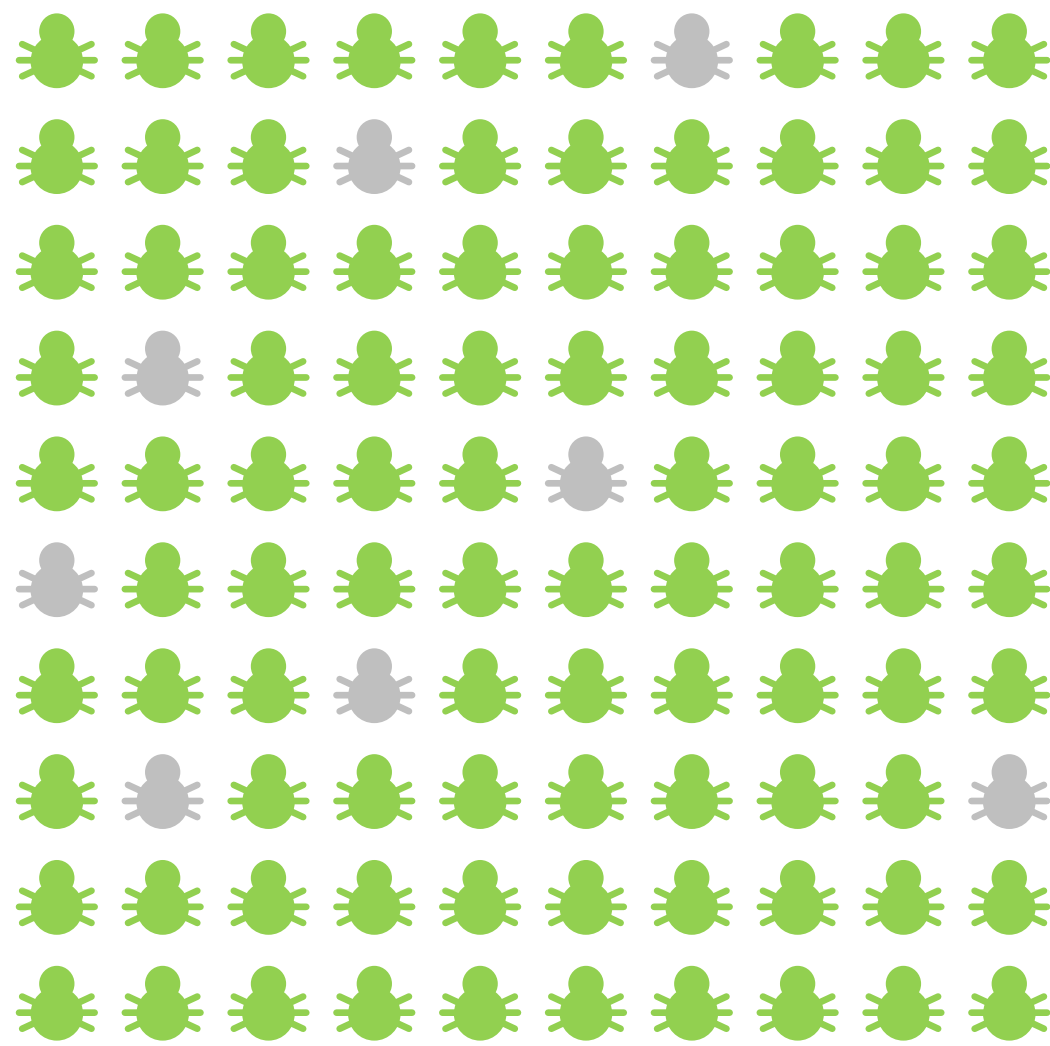
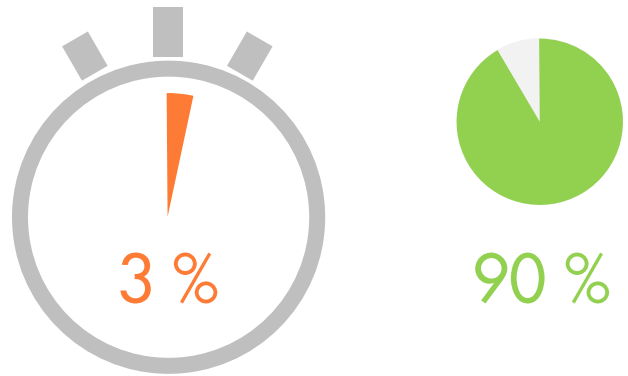
Manual Test	
Action	Check
Log in as Carleen6231@gmail.com	
Open a new account: Type Savings Account, Individual In the Money Market Start deposit: 2500	Account was created as specified.
Open the transfer page.	
Select the account from step 2 as both from and to account.	
Enter amount: 11	
Submit the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

Score of a Test for a search term =
term frequency * inverse **document frequency**

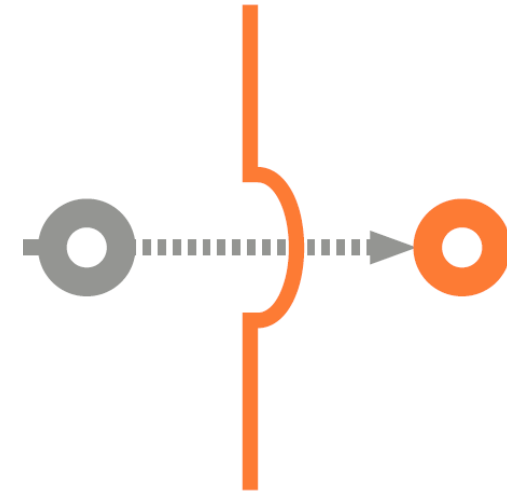
How often does the term occur in a test?
The more often, the higher the score

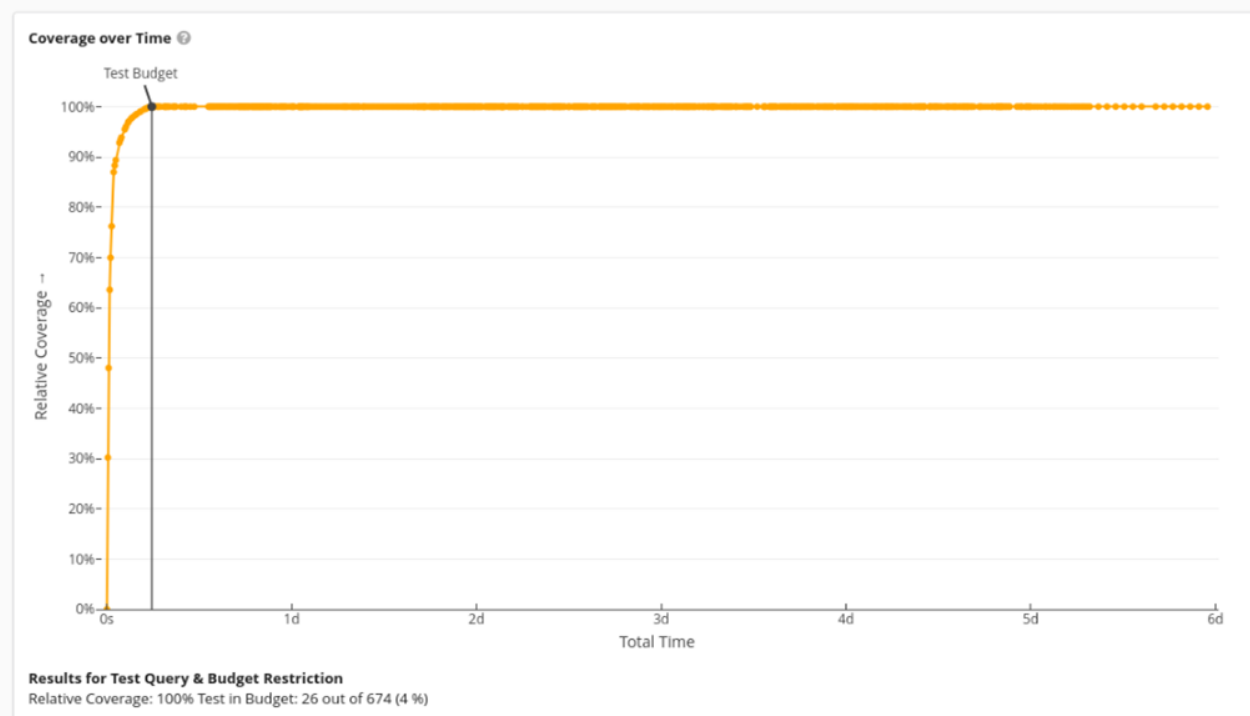
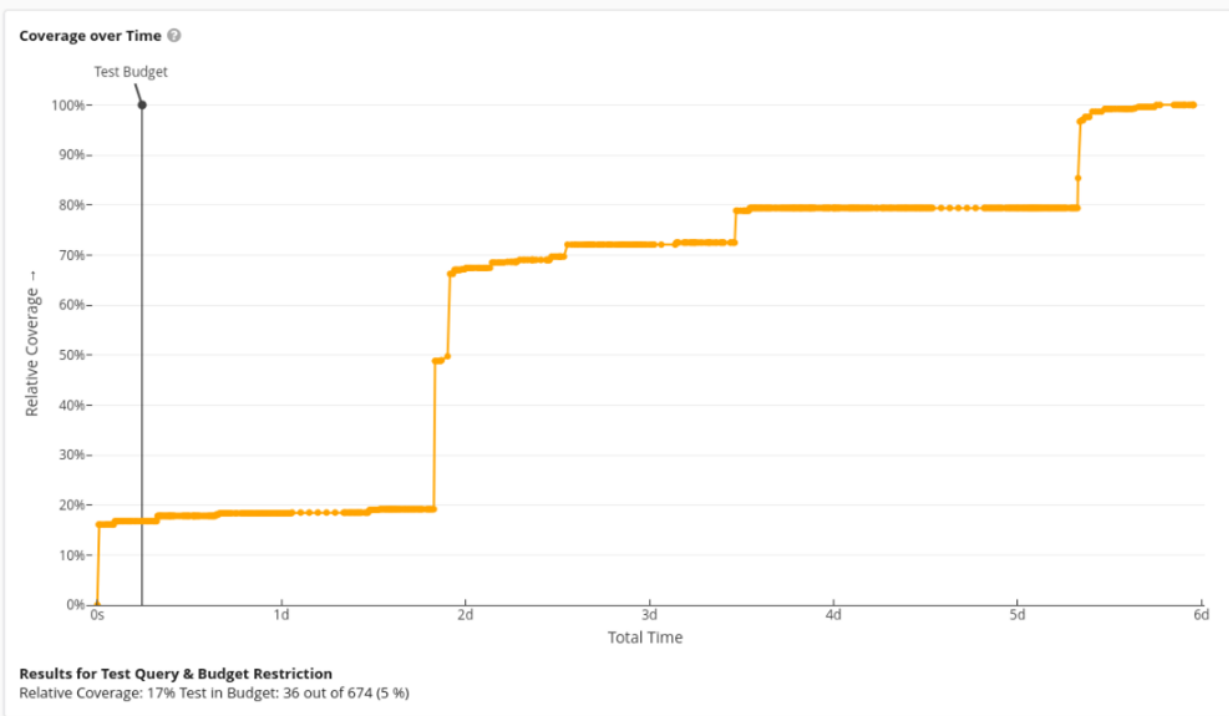
How many tests contain this term?
The more specific, the higher the score.

Information Retrieval



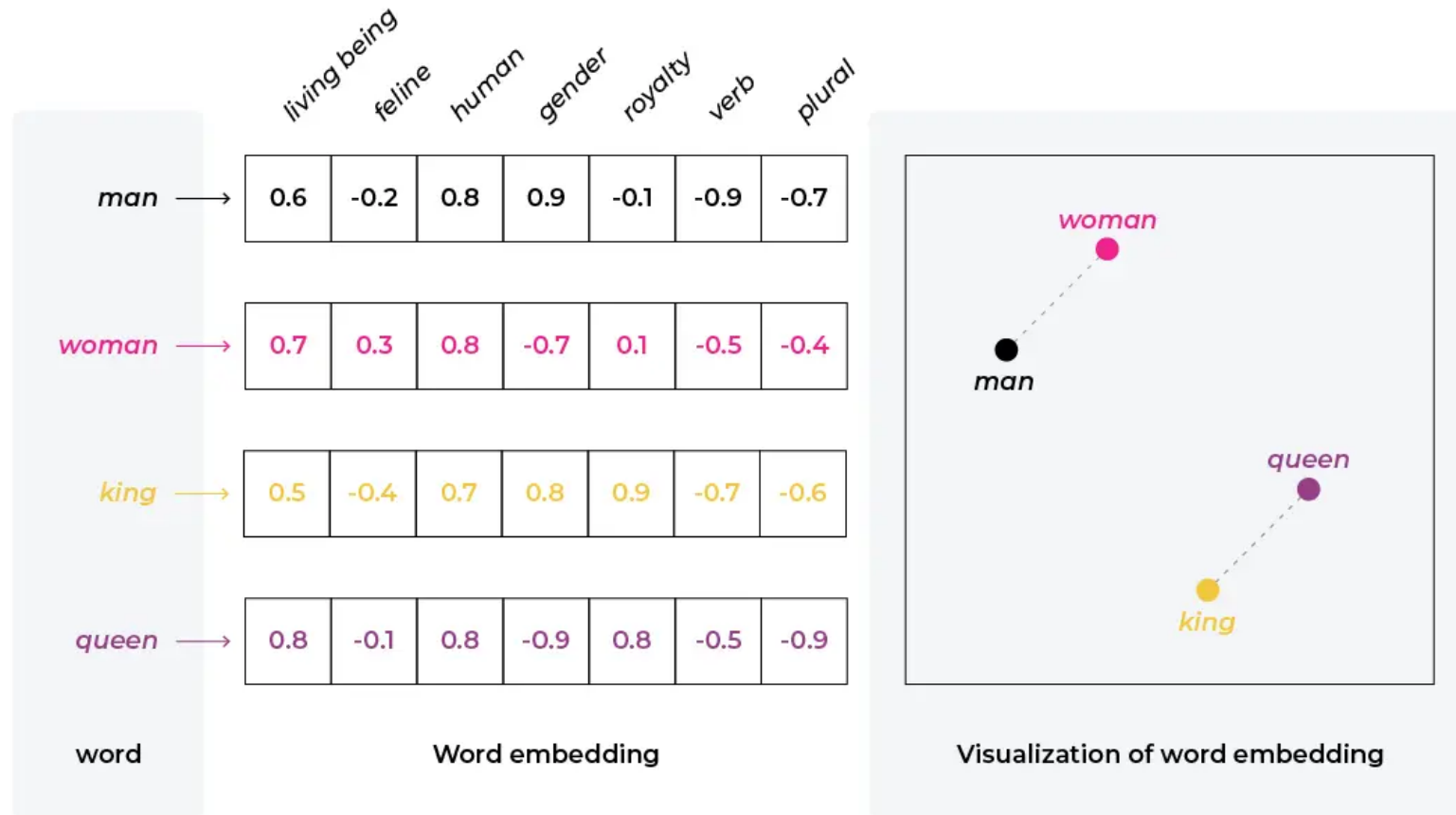
Test Selection for a Quality Gate **without Coverage**

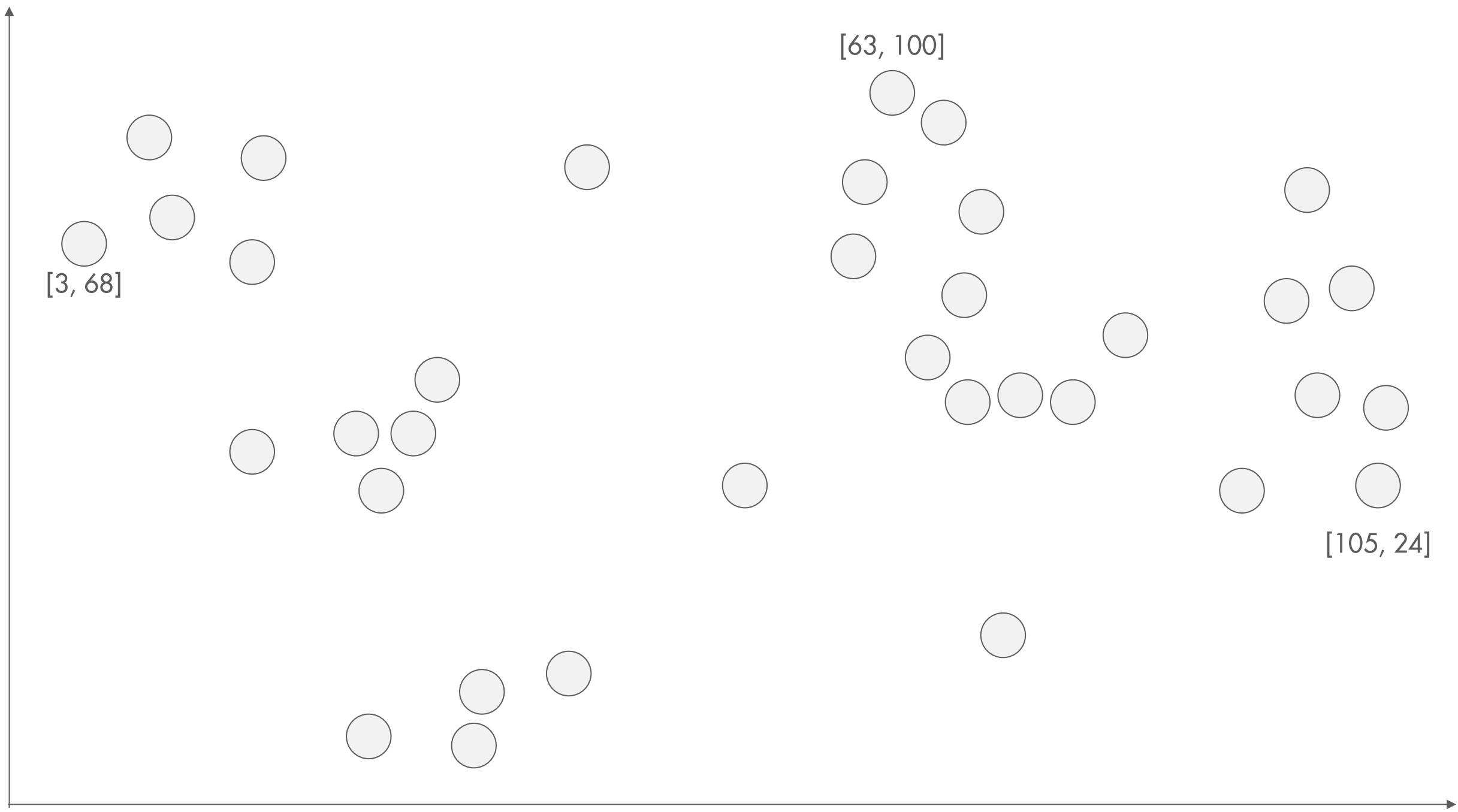


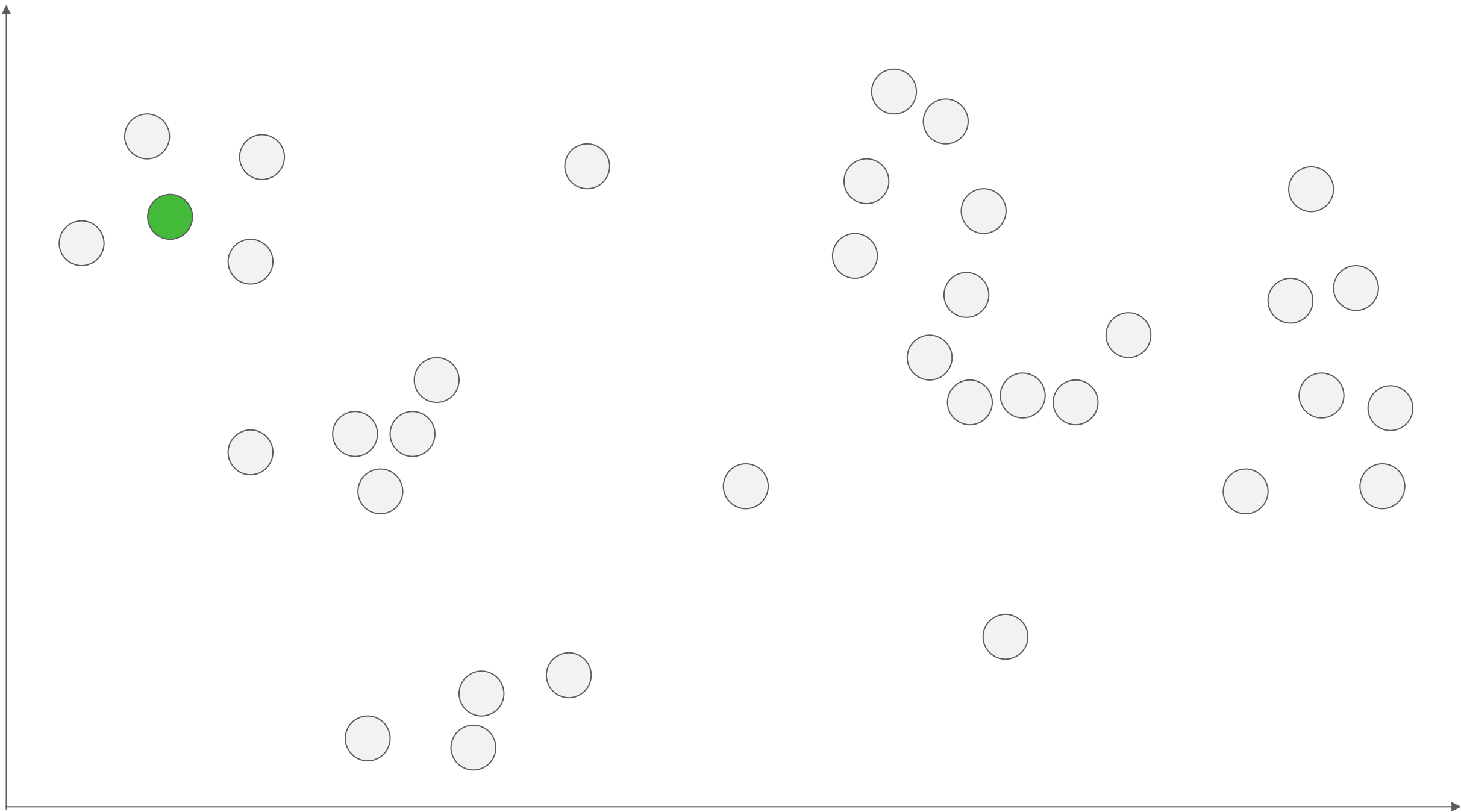


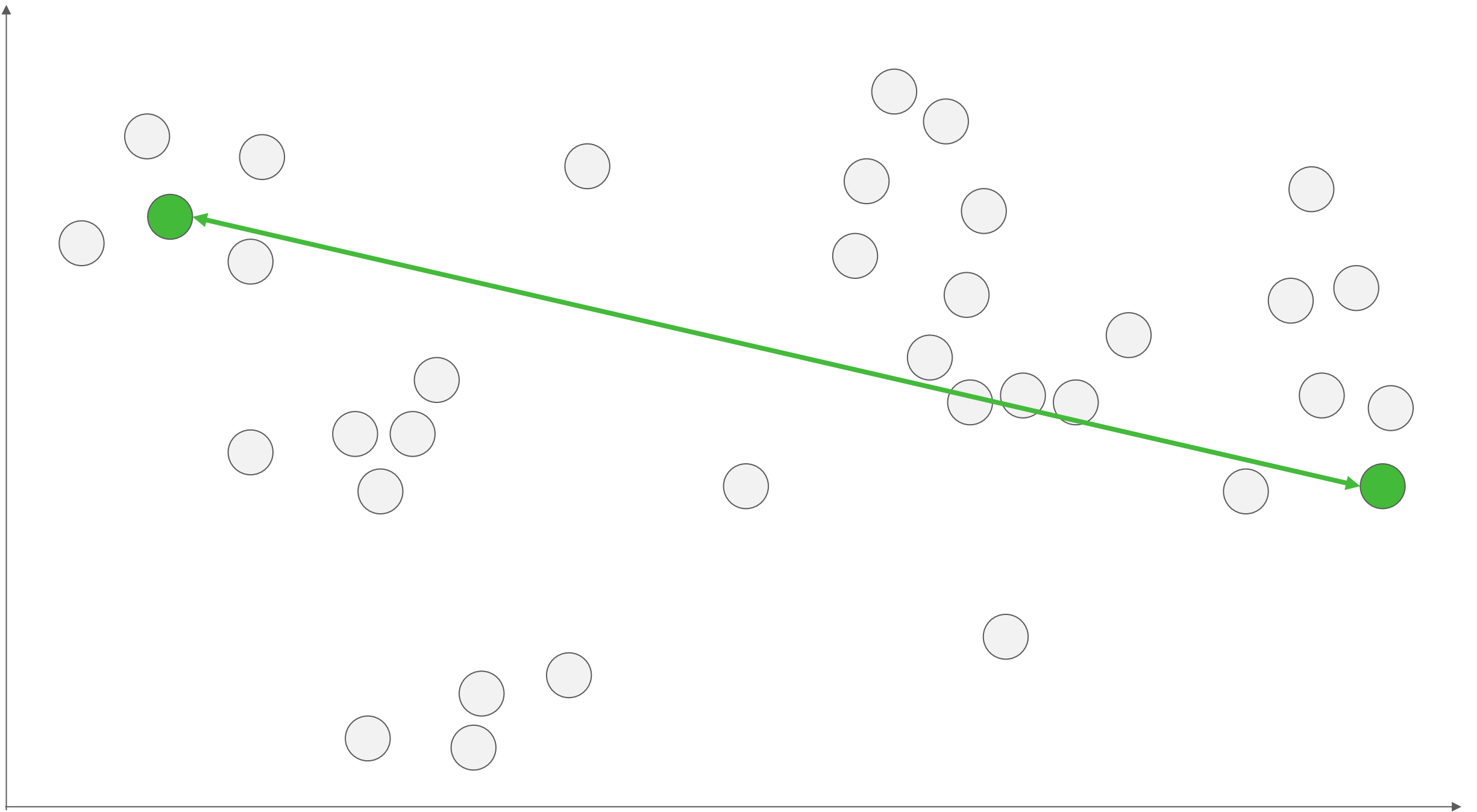
Idea: Tests with **very different content**

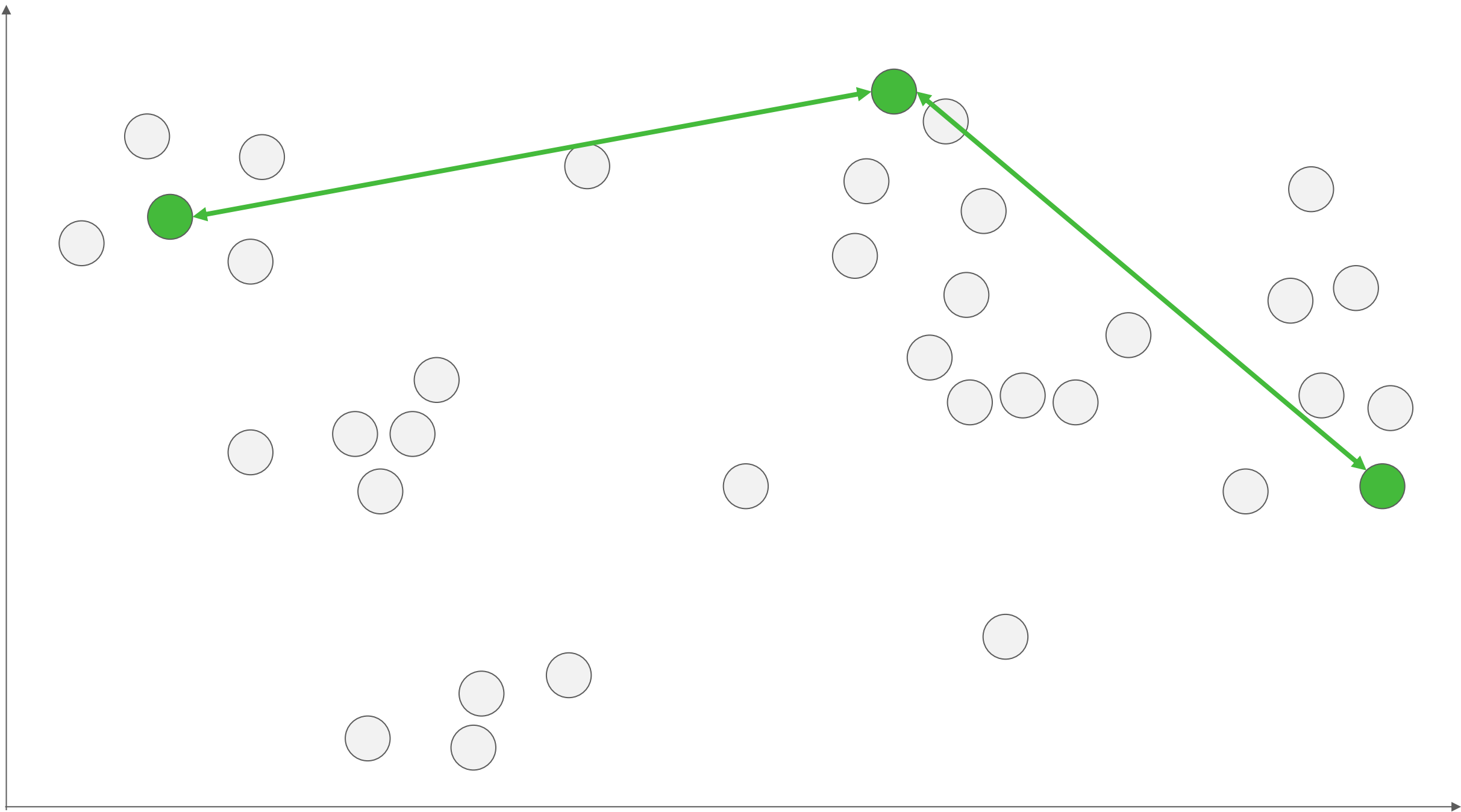
Large Language Models (AI)

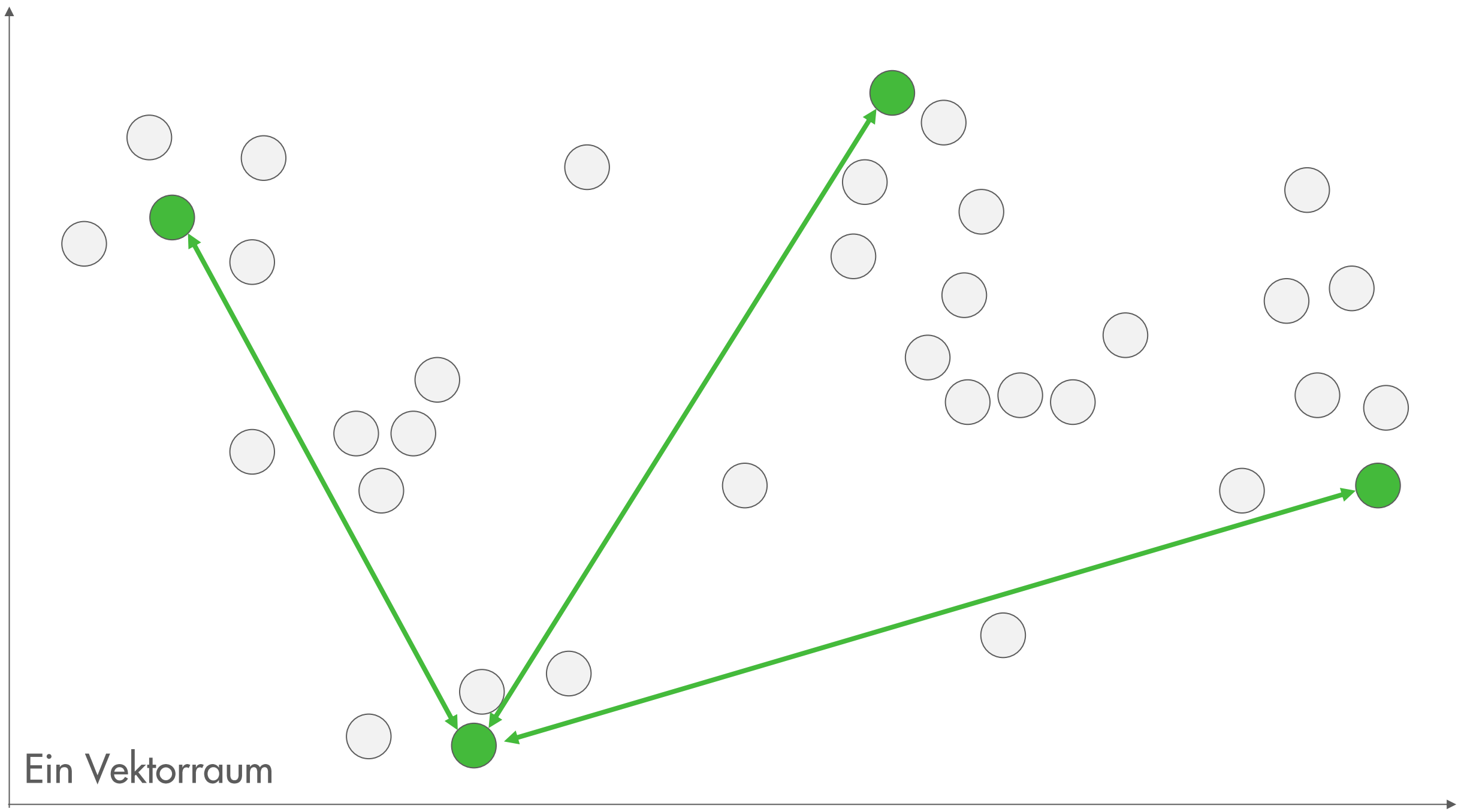






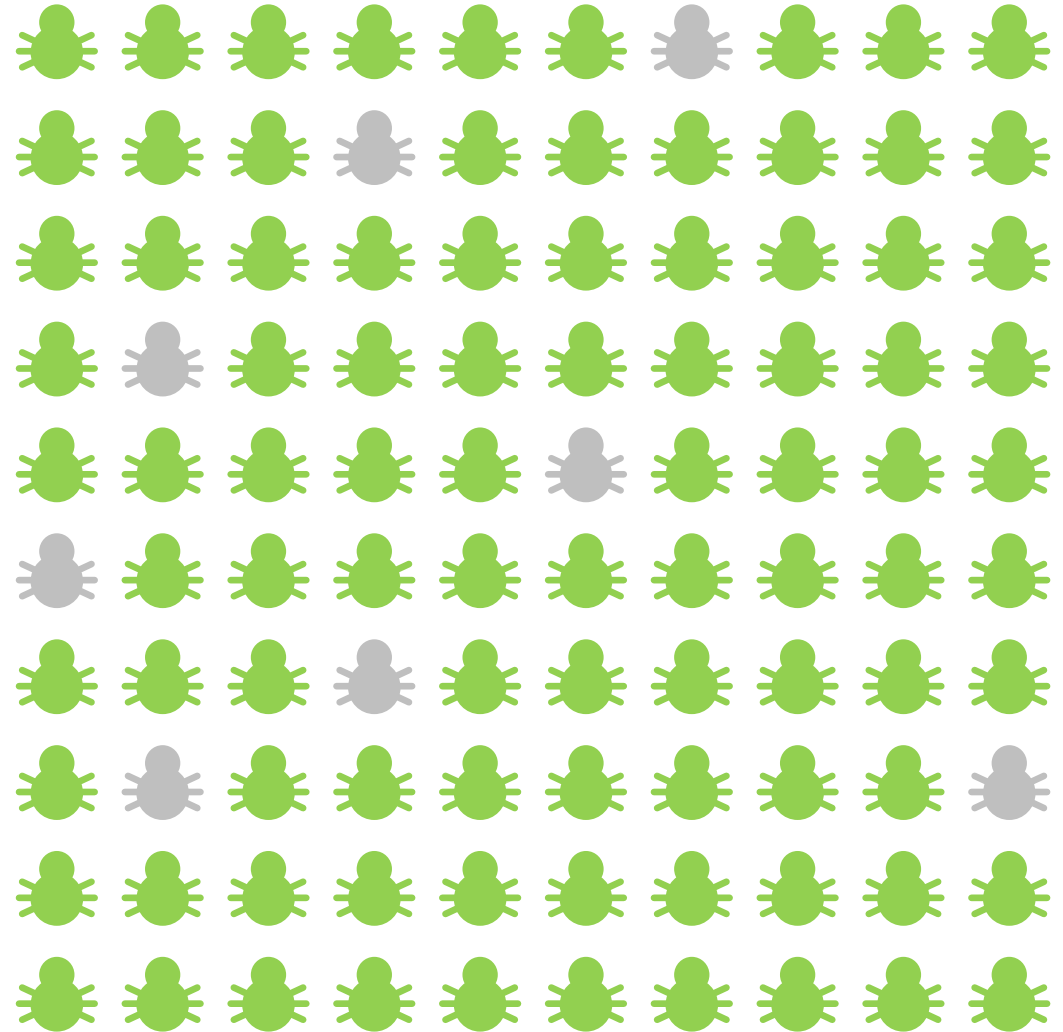






Ein Vektorraum

AI-Selected Tests



Quality Gate

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

11%

Pareto-Analysis

13%

AI-selektiert Tests

Continuous Integration

2%

Test-Impact-Analysis

3%

Information Retrieval

Precision & Effort

Test Coverage

Test Content

Start here!



Summary

Don't always run all tests (if it takes too long).

There are many test selection approaches that are fit for use in practice!

Talk to us. We are happy to discuss what works best in your context 😊

Contact – Looking forward to Diskussions 😊



Dr. Elmar Jürgens · juergens@cqse.eu



[Link to Slides](#)

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

