

Everyone wants to shift left...
but our tests are just too slow!



Jonas Bogenberger

- Master of Science in Informatics
- Responsible for applying the research results

CQSE

 Teamscale

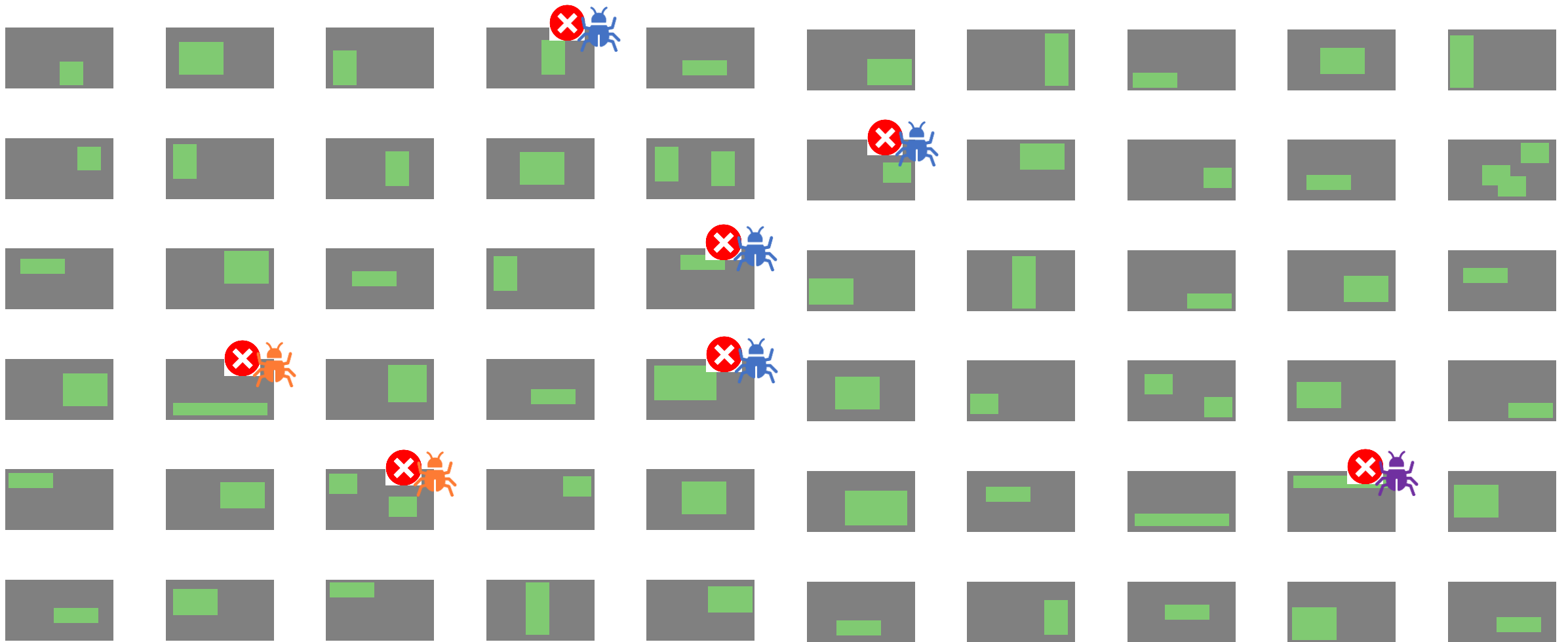


Hours, Days

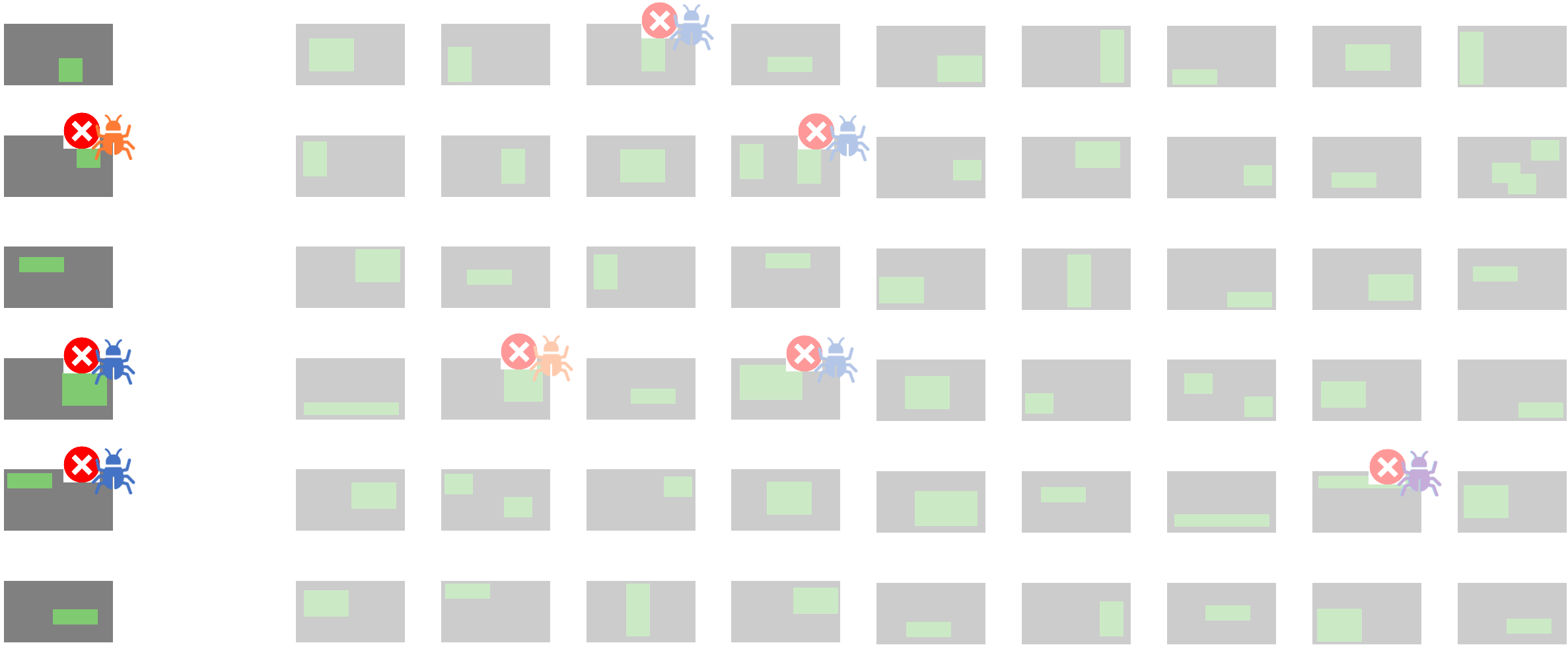


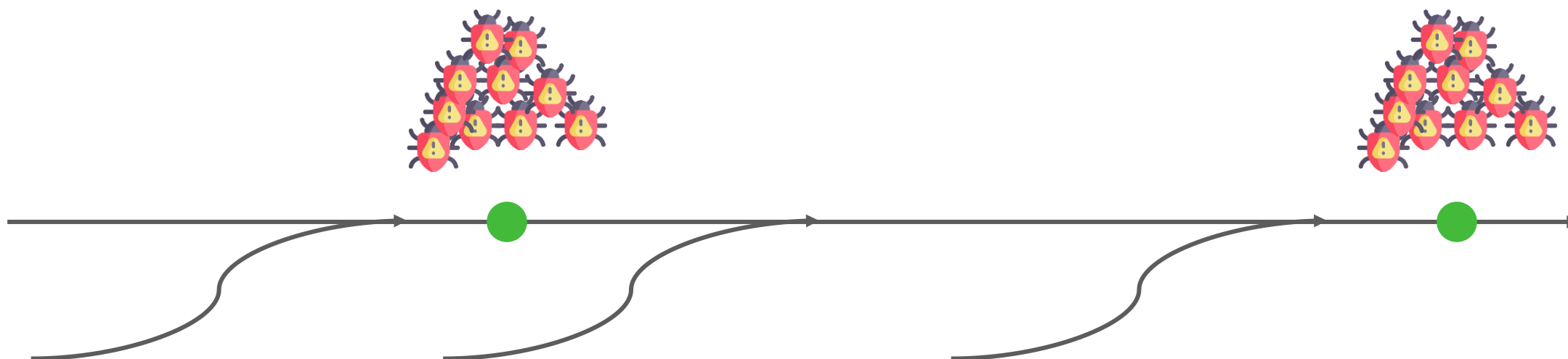
Minutes

manual, automated,
E2E, ...

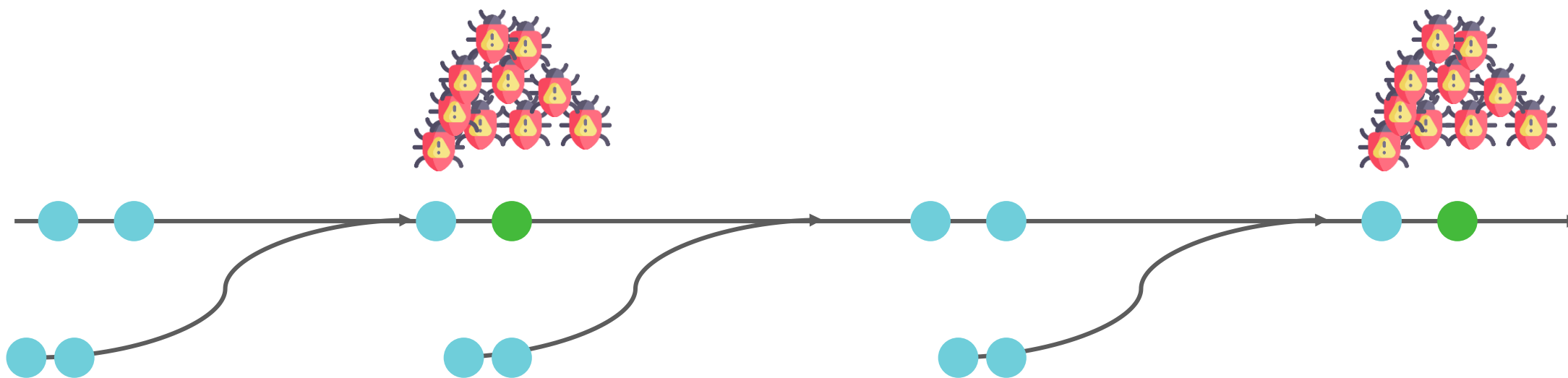


Test Selection



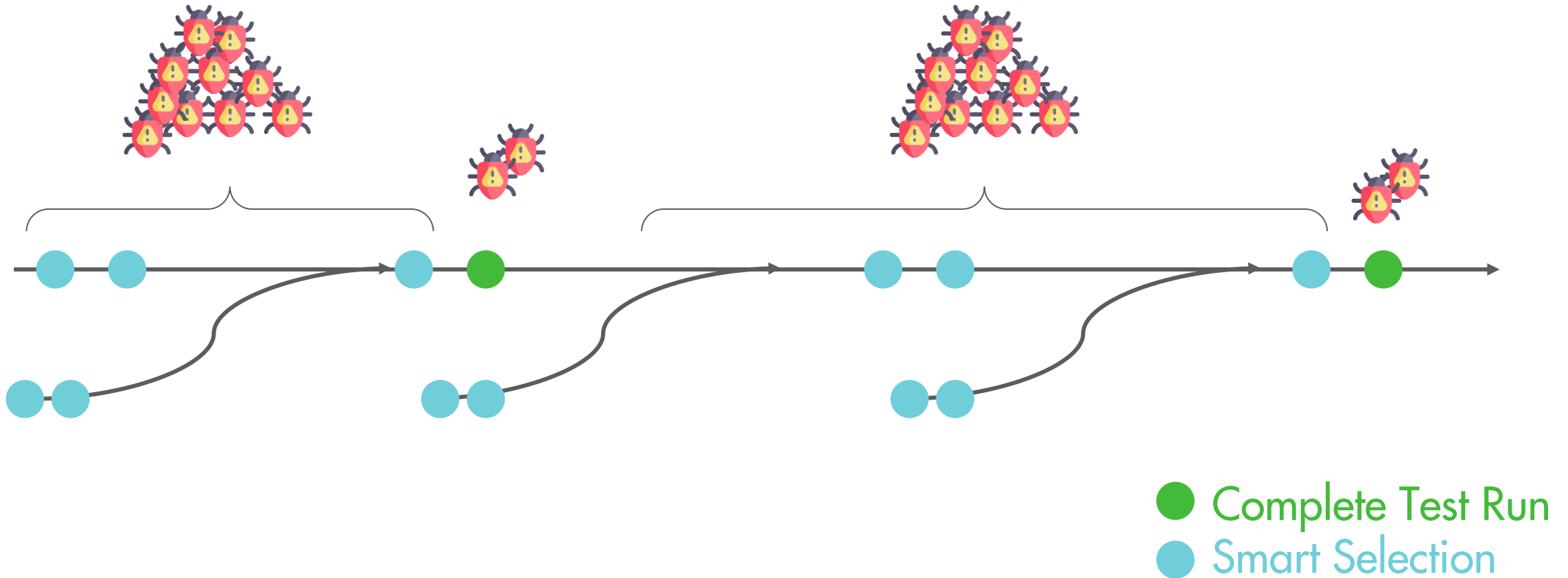


● Complete Test Run

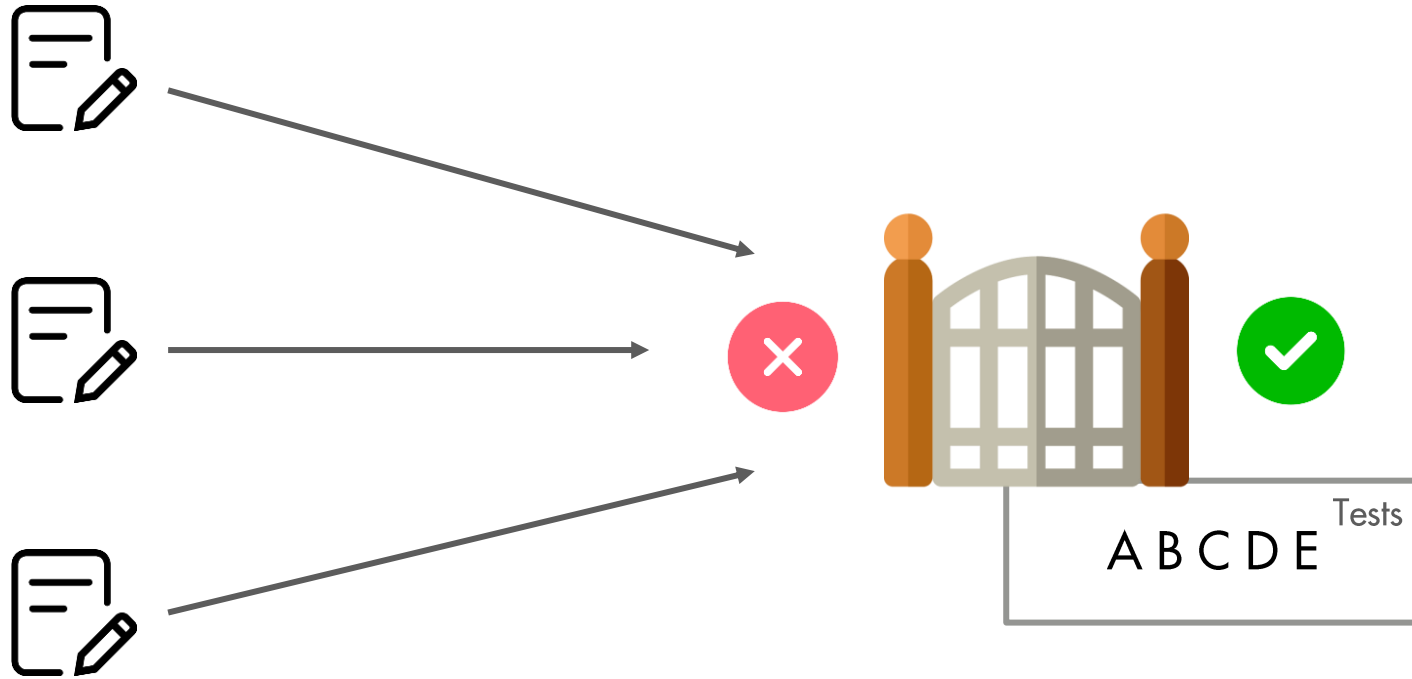


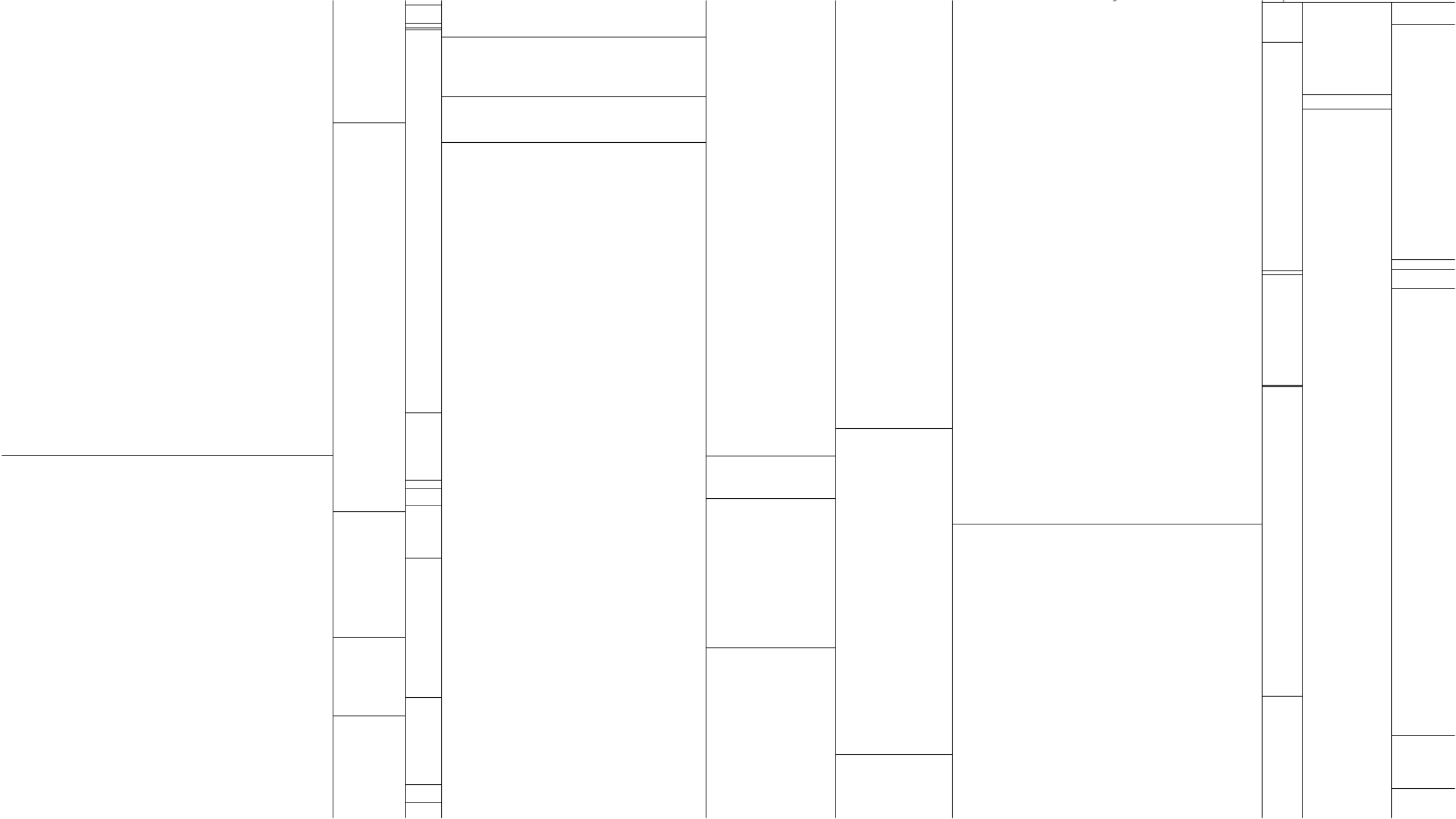
● Complete Test Run
● Smart Selection

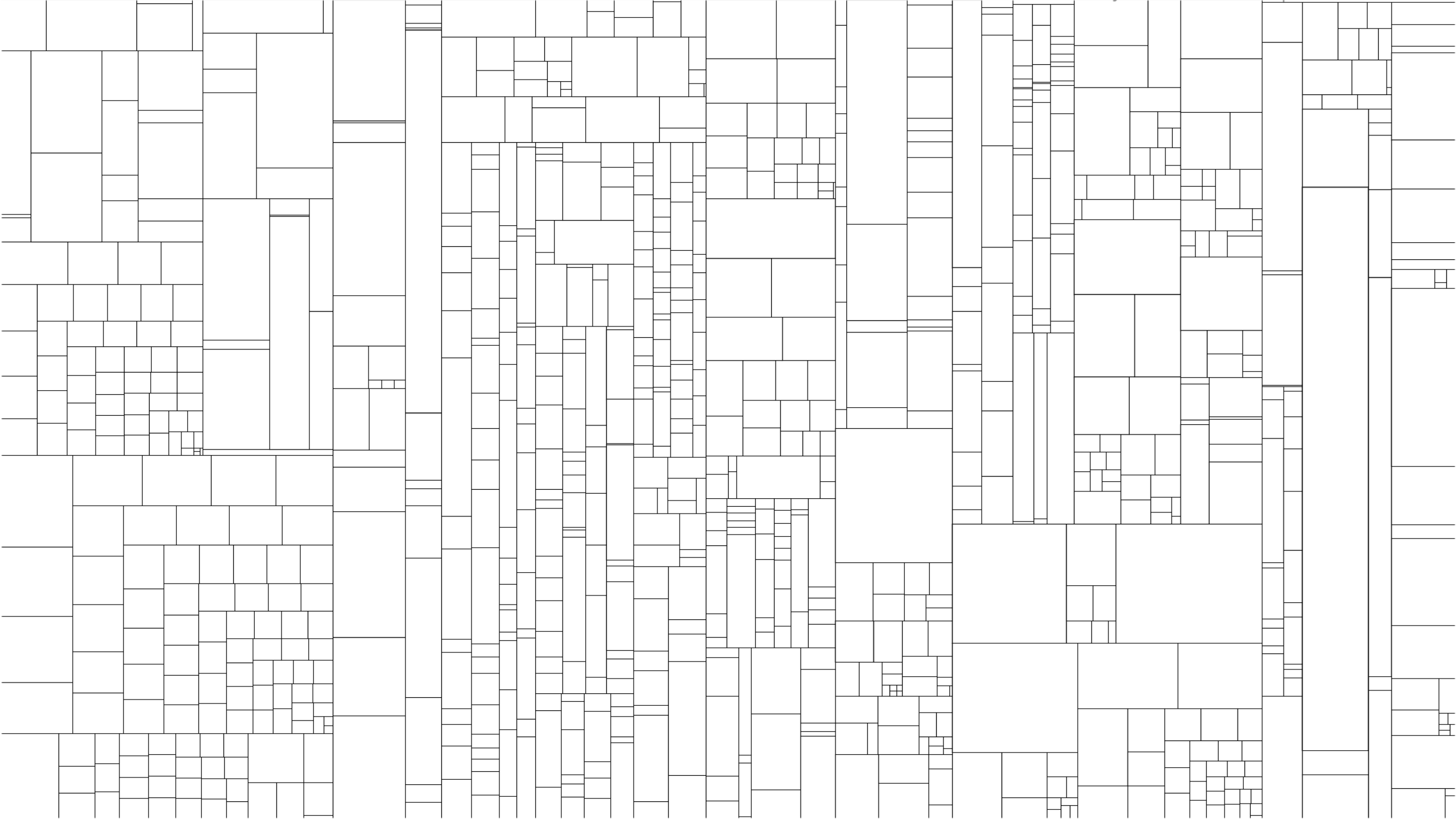
Shift Left

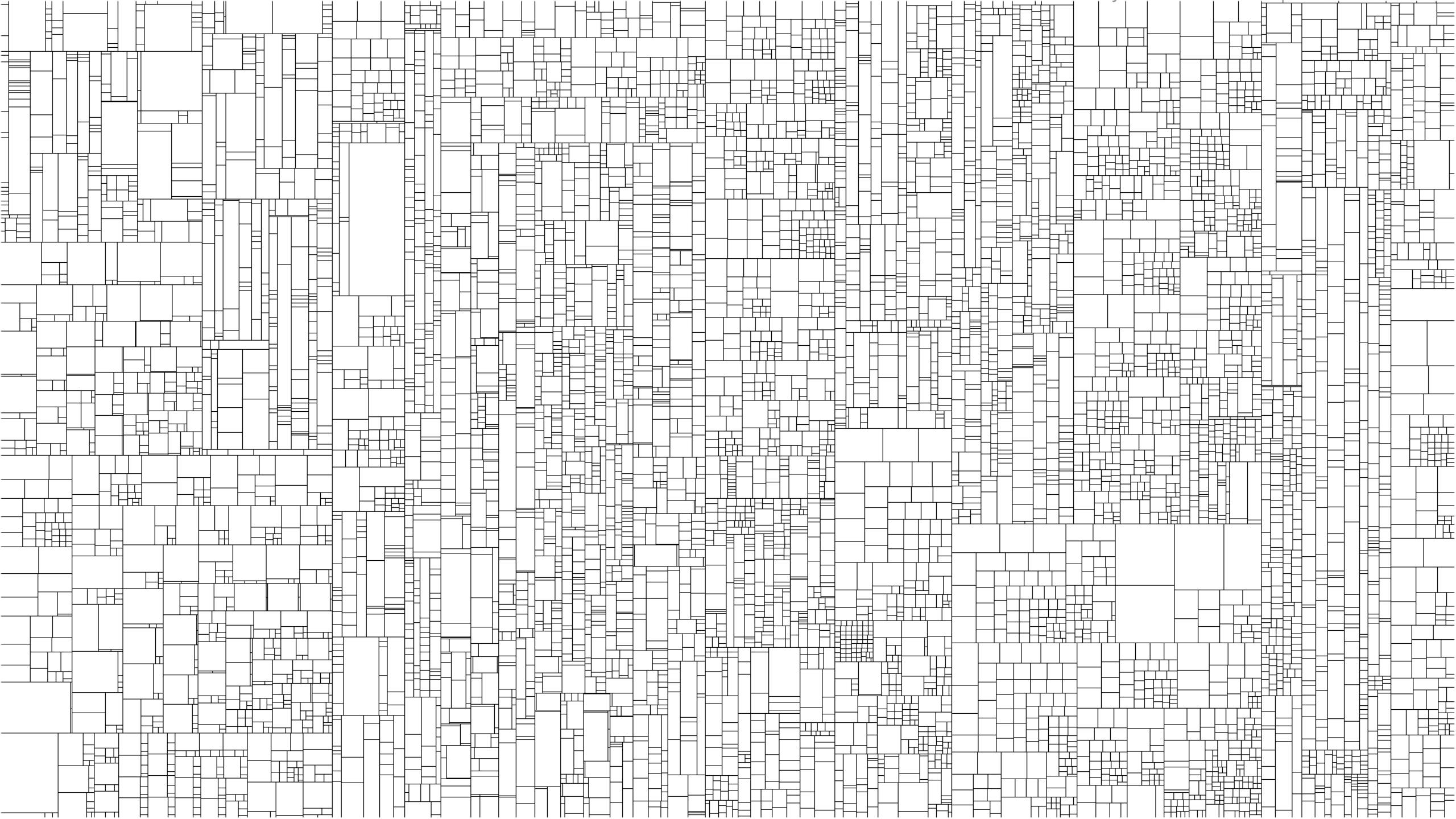


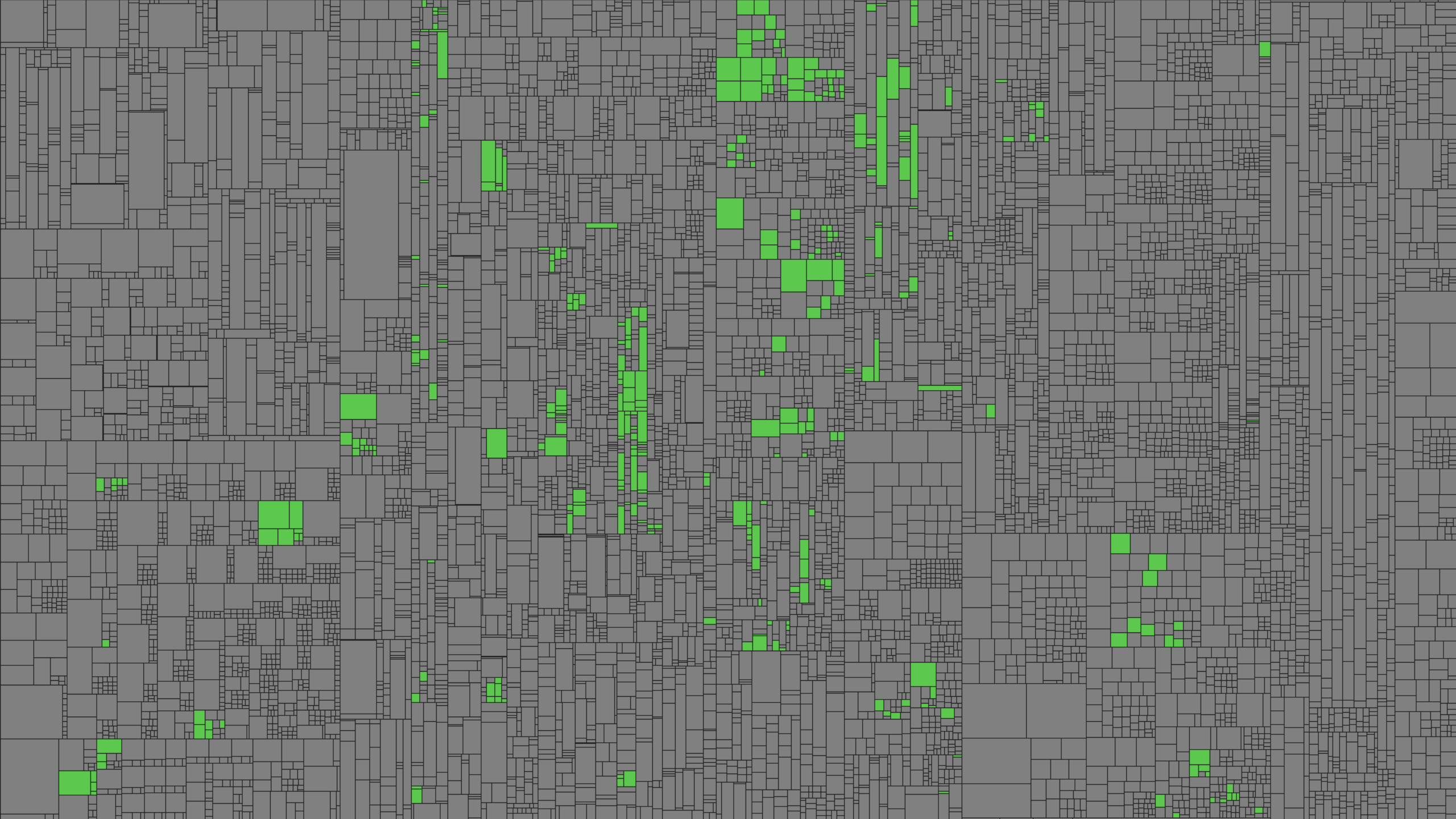
Quality Gate

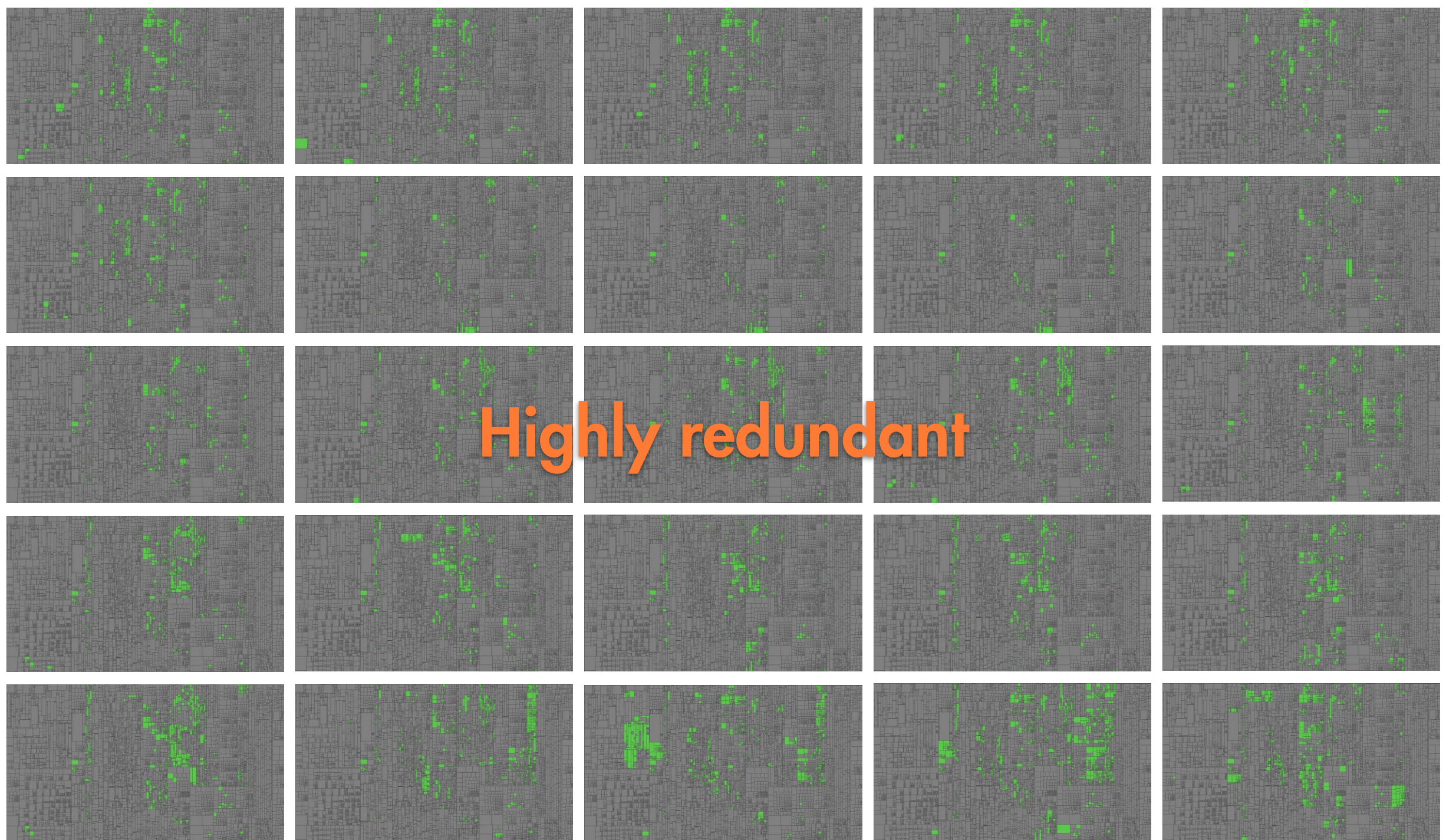






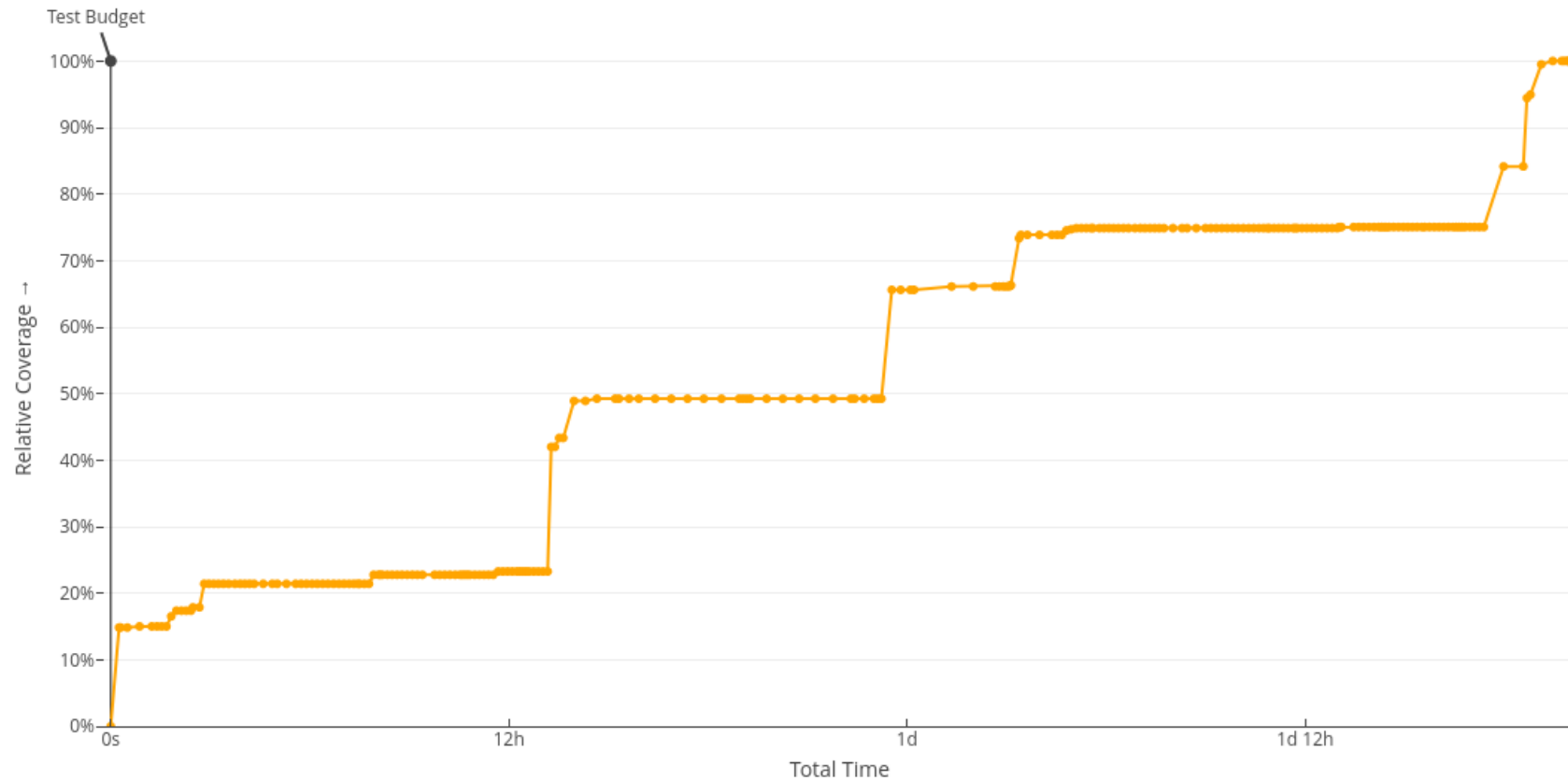






Quality Gate

Coverage over Time ?

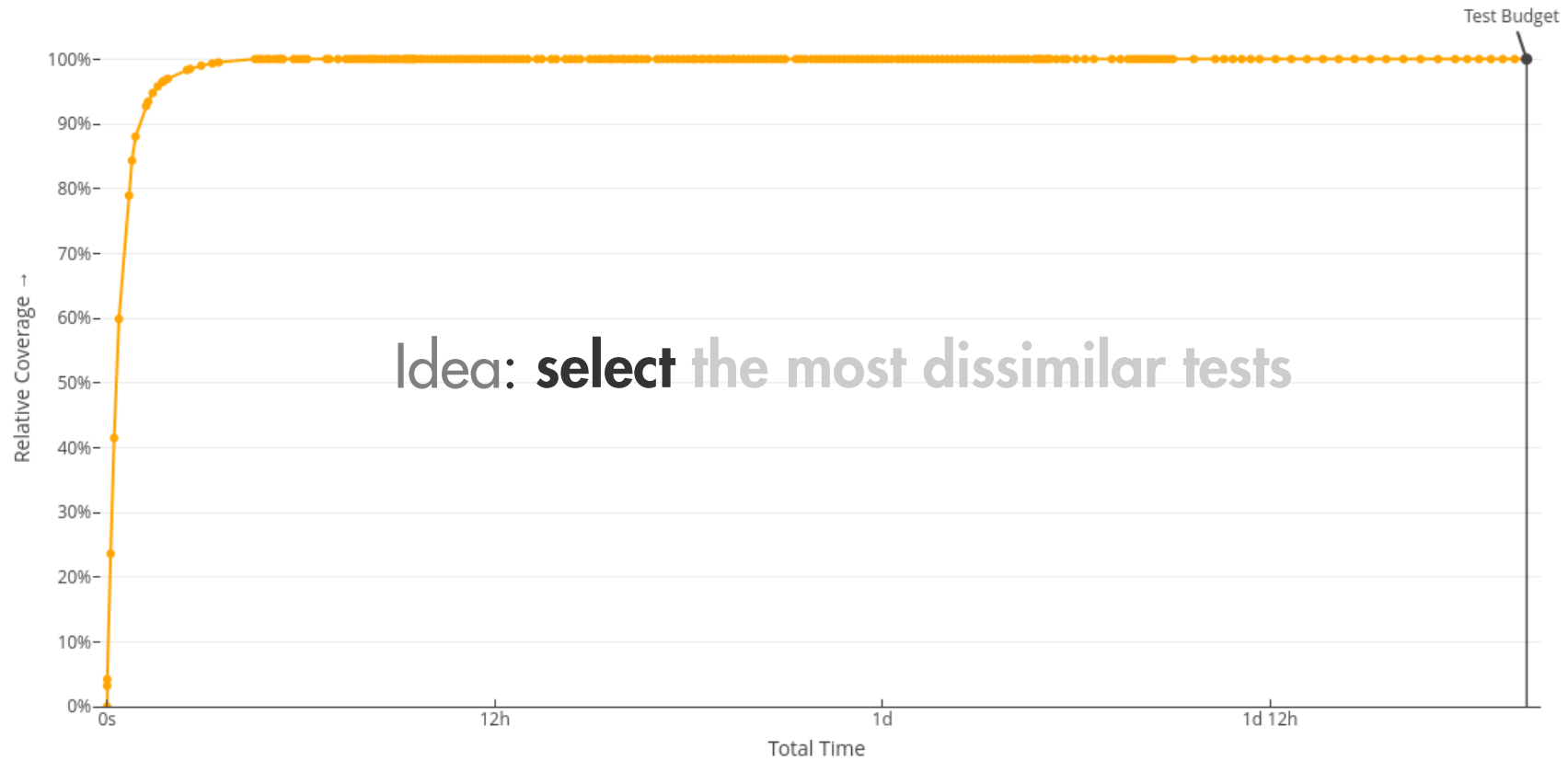


Results for Test Query & Budget Restriction

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

Quality Gate

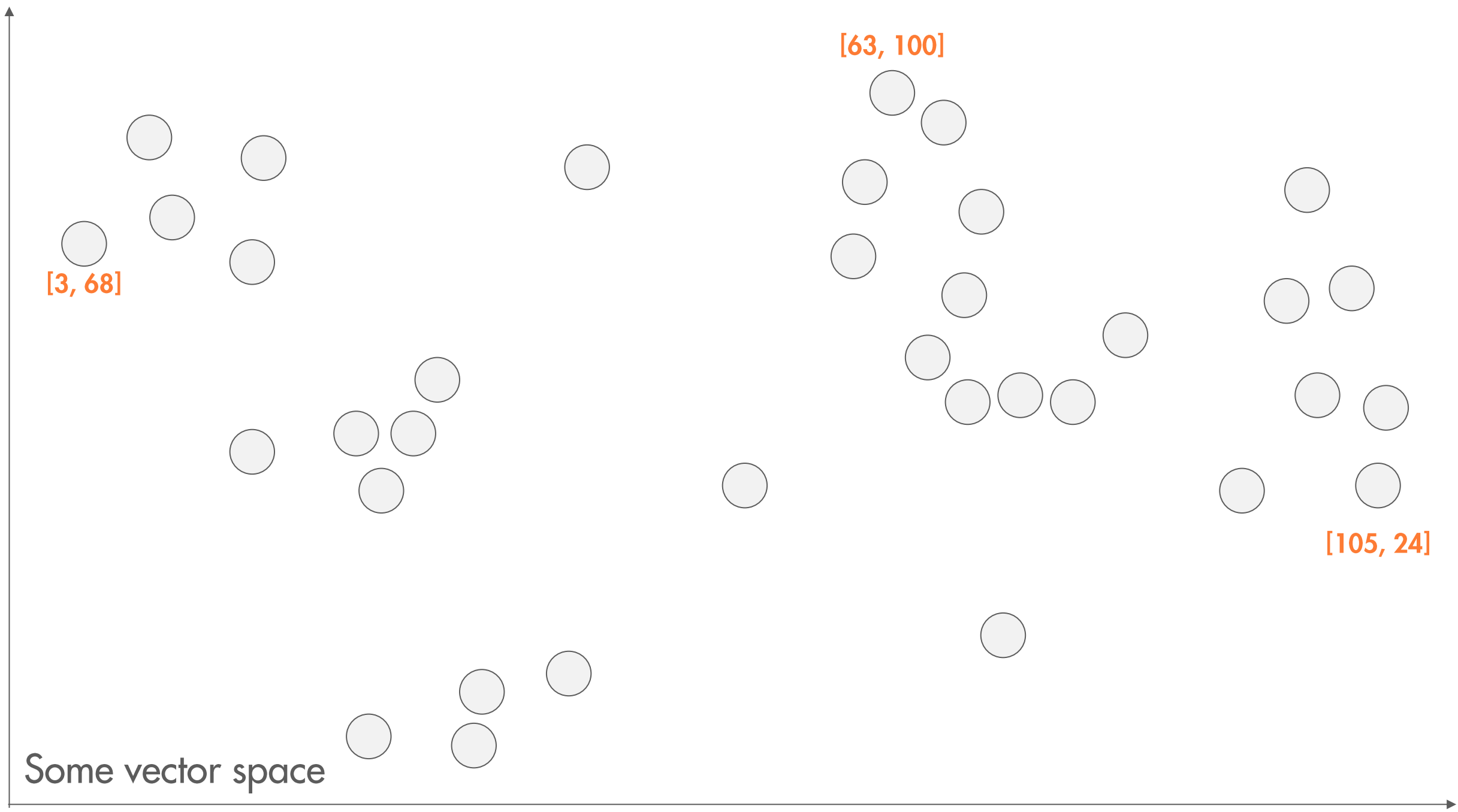
Coverage over Time ?



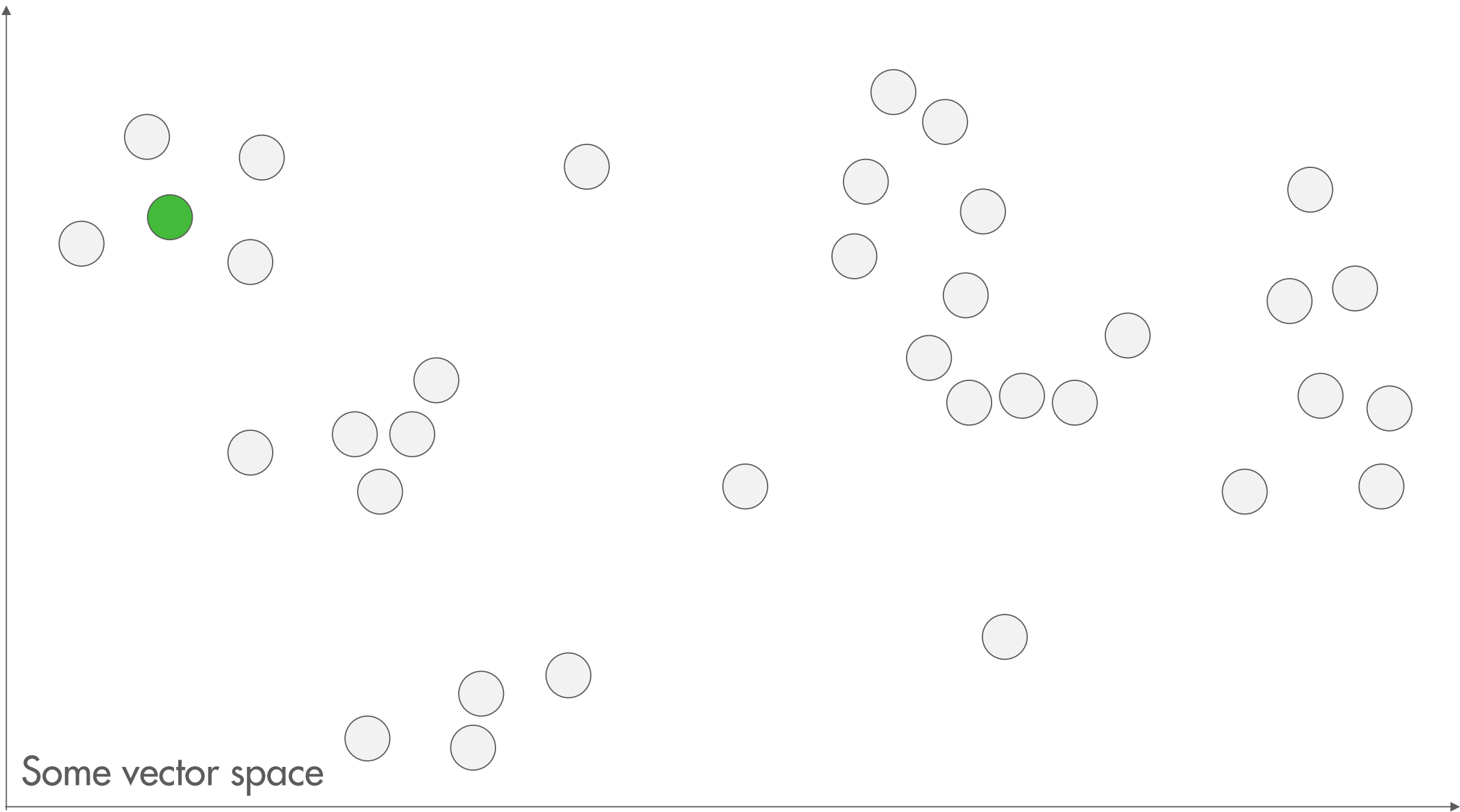
Idea: **select** the most dissimilar tests

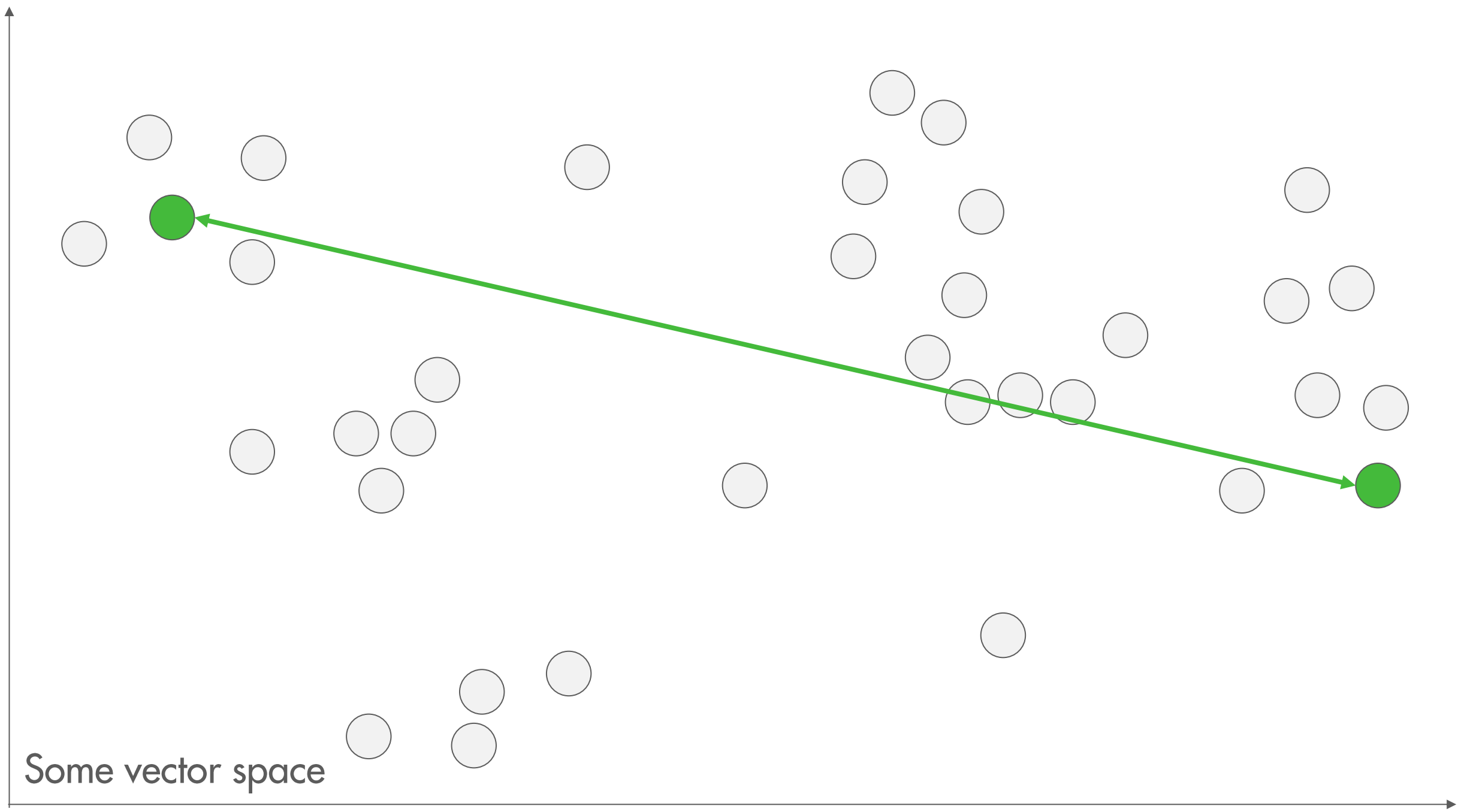
Results for Test Query & Budget Restriction

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

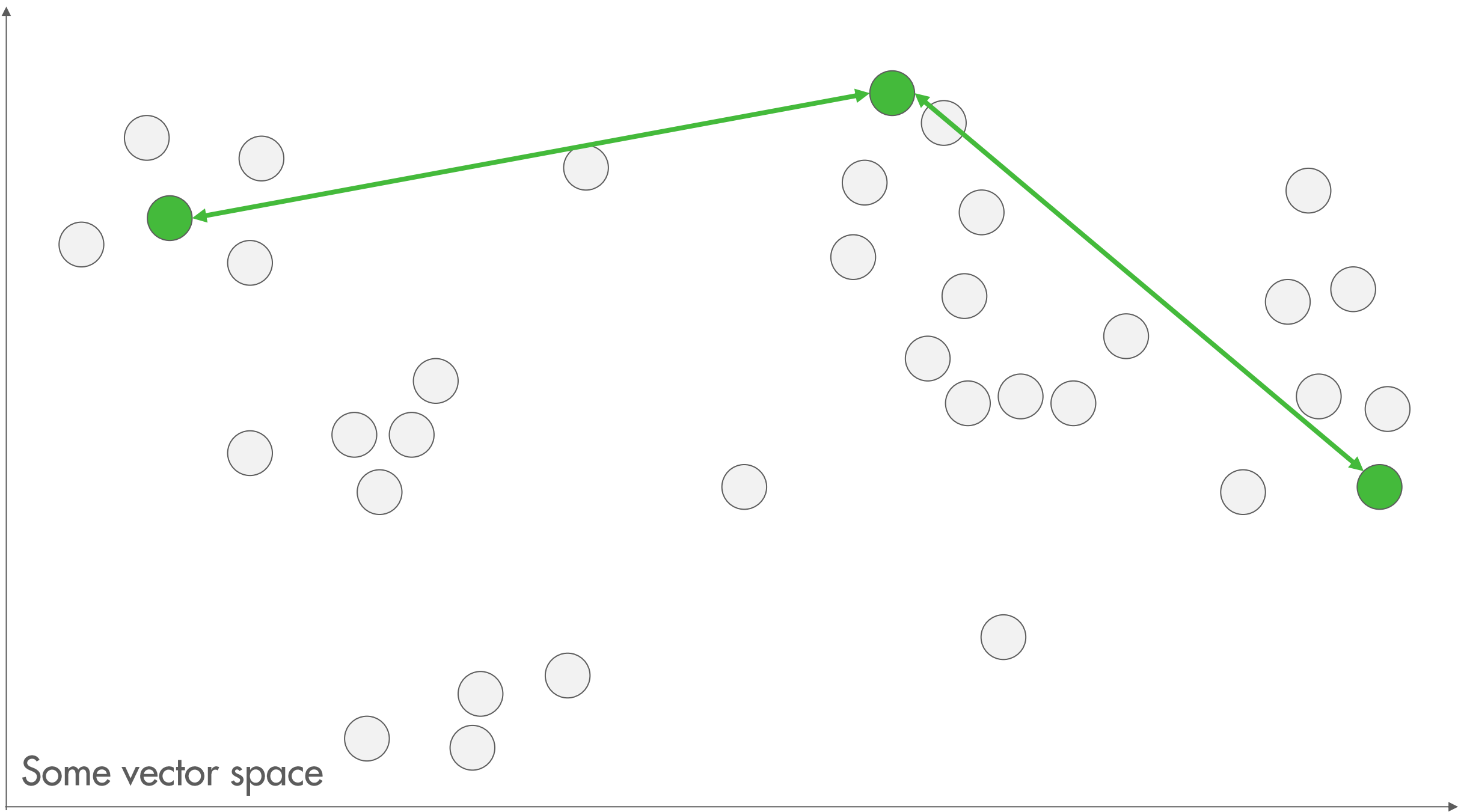


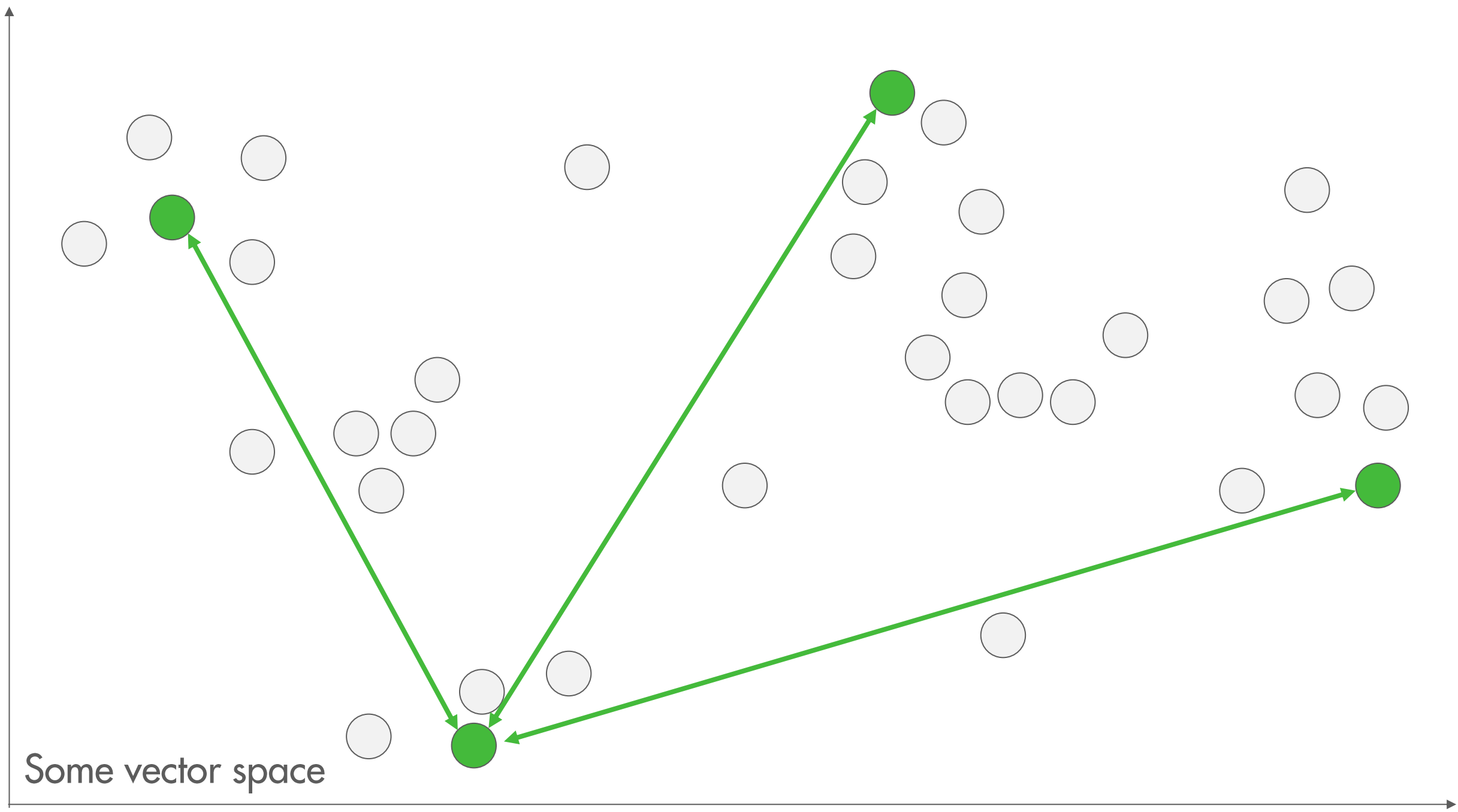
Some vector space



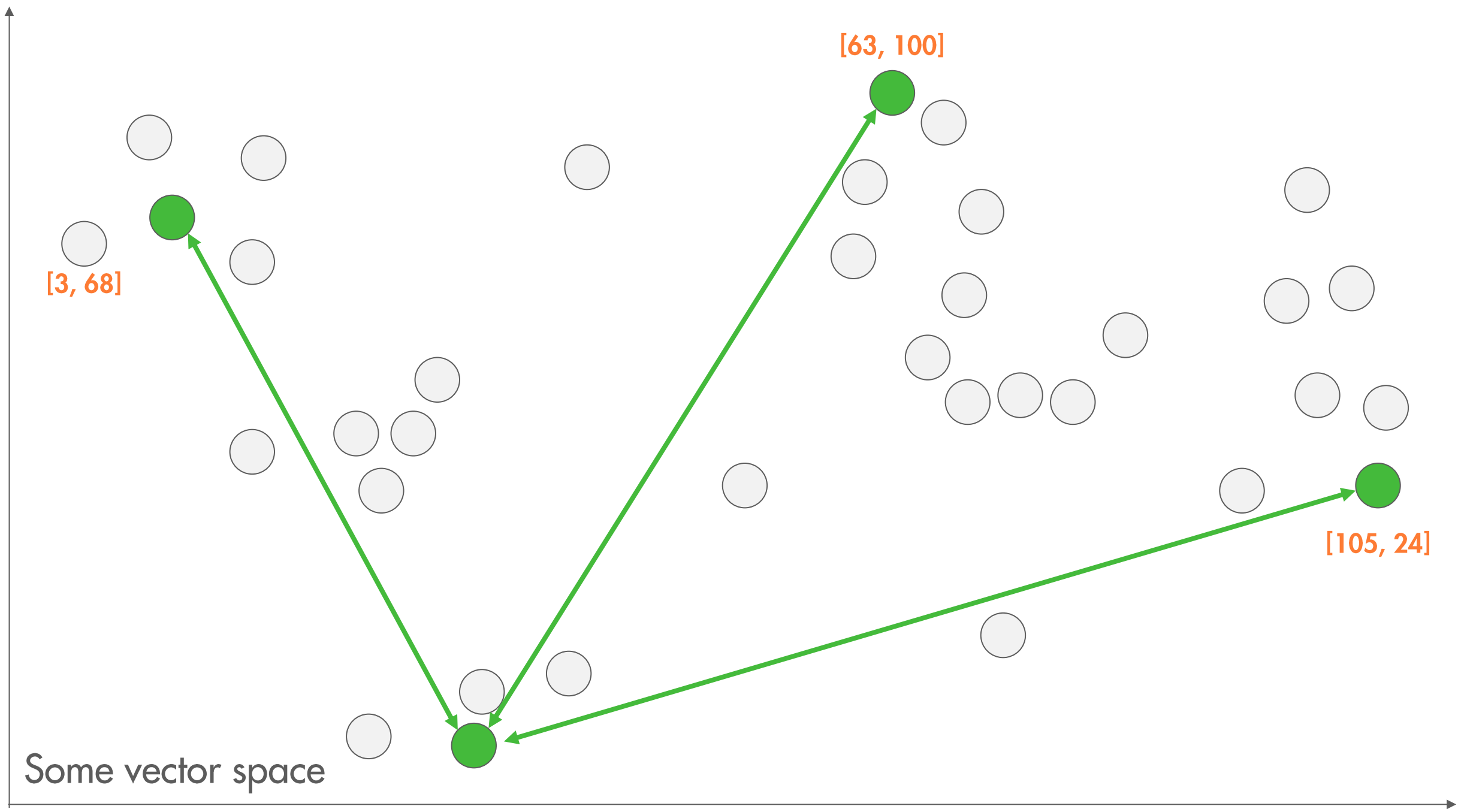


Some vector space





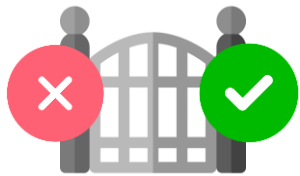
Some vector space



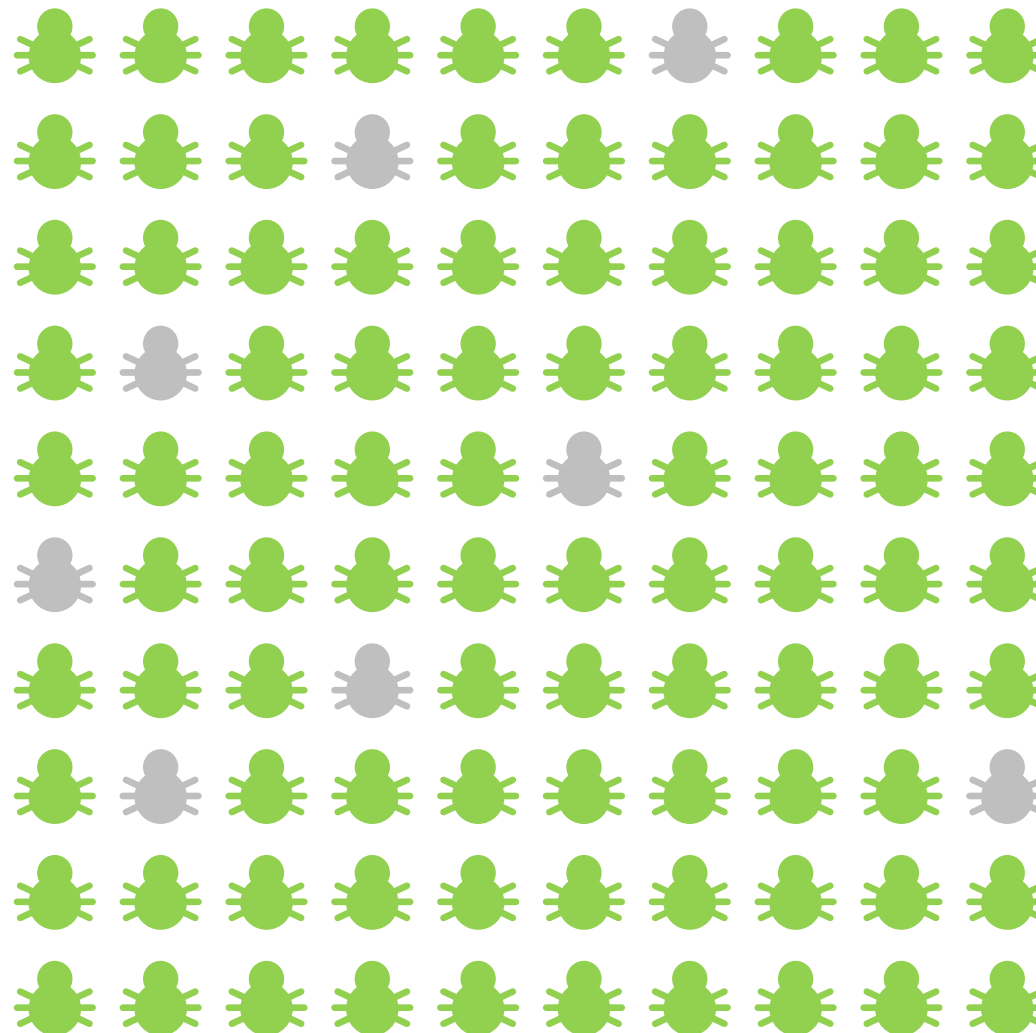
Large Language Models (AI)



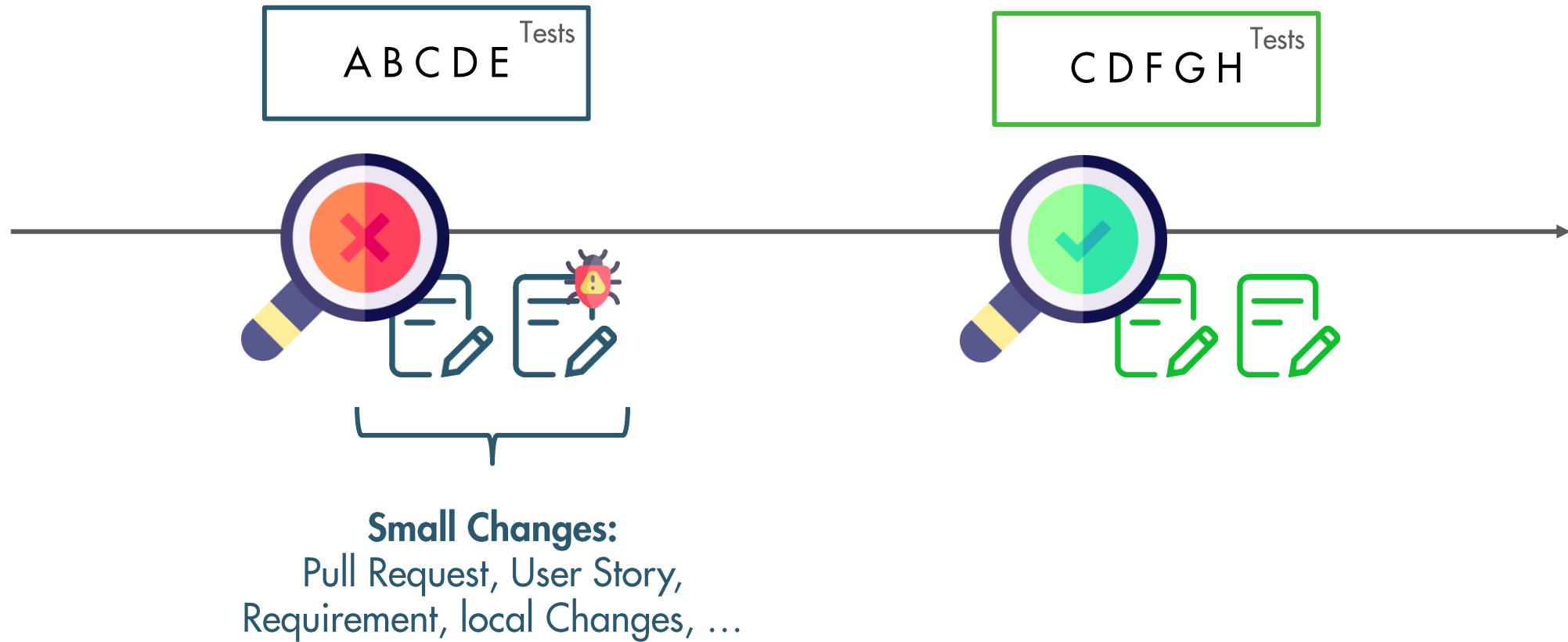


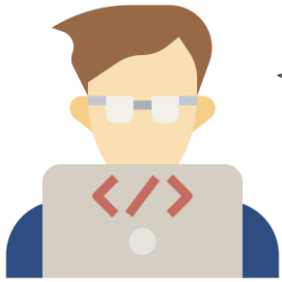


AI Test Clustering



Change-based Testing



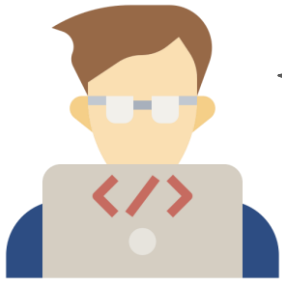


We changed **login, accounting
and search.**

(user story, pull request, release)



Tests for the impacted functionality



We changed **login, accounting
and search.**

(user story, pull request, release)



Google

Google

Google

test cases for login



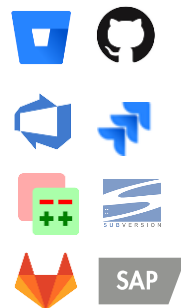
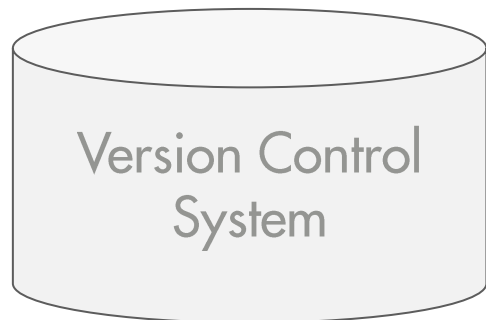
test cases for accounting



test cases for search



Tests for the impacted functionality



```
124 + agentOptions = AgentOptionsParser.parse(options, environmentConfigId, environmentConfigFile, credentials);
118 125 } catch (AgentOptionParseException e) {
119 - try (LoggingUtils.LoggingResources ignored = initializeFallbackLogging(options, delayedLogger)) {
120 - delayedLogger.errorAndStdErr("Failed to parse agent options: " + e.getMessage(), e);
121 - attemptLogAndThrow(delayedLogger);
126 + try (LoggingUtils.LoggingResources ignored = initializeFallbackLogging(options)) {
127 + String message = "Failed to parse agent options: " + e.getMessage();
128 + logAndPrintError(e, message);
122 129 throw e;
123 130 }
124 131 } catch (AgentOptionReceiveException e) {
125 - try (LoggingUtils.LoggingResources ignored = initializeFallbackLogging(options, delayedLogger)) {
126 - delayedLogger.errorAndStdErr(
127 - e.getMessage() + " The application should start up normally, but NO coverage will be collected!",
128 - e);
129 - attemptLogAndThrow(delayedLogger);
132 + try (LoggingUtils.LoggingResources ignored = initializeFallbackLogging(options)) {
133 + String message = e.getMessage() + " The application should start up normally, but NO coverage will be collected!";
134 + logAndPrintError(e, message);
130 135 throw e;
131 136 }
132 137 }
133 138
134 - initializeLogging(agentOptions, delayedLogger);
135 - Logger logger = LoggingUtils.getLogger(Agent.class);
136 - delayedLogger.logTo(logger);
139 + initializeLogging(agentOptions);
137 140 HttpUtils.setShouldValidateSsl(agentOptions.shouldValidateSsl());
138 141 return agentOptions;
139 142 }
140 143
141 - private static void attemptLogAndThrow(DelayedLogger delayedLogger) {
142 - // We perform actual logging output after writing to console to
143 - // ensure the console is reached even in case of logging issues
```



Google

Google

Google

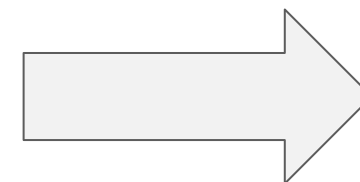
test cases for login



test cases for accounting



test cases for search



Tests for the impacted functionality

```
125 - try (LoggingUtils.LoggingResources ignored = initializeFallbackLogging(options, delayedLogger)) {
126 -     delayedLogger.errorAndStdErr(
127 -         e.getMessage() + " The application should start up normally, but NO coverage will be collected!",
128 -         e);
129 -     attemptLogAndThrow(delayedLogger);
130 + try (LoggingUtils.LoggingResources ignored = initializeFallbackLogging(options)) {
131 +     String message = e.getMessage() + " The application should start up normally, but NO coverage will be collected!";
132 +     logAndPrintError(e, message);
133 +     throw e;
134 + }
135
136 - }
137
138 - initializeLogging(
139 -     Logger logger =
140 -     delayedLogger.l
141 + initializeLoggin
142 + HttpUtils.setSh
143 + return agentOpt
144
145 - private static void att
146 - // We perform at
147 - // ensure the c
```

Code-changes

Search Query

"Search and login and list and user and ..."

Similarity Scoring



Tests to run

manual



Test Steps

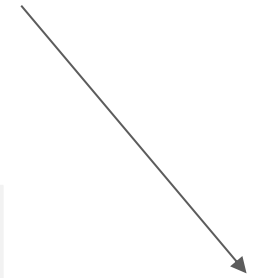
automated



Test Code



Document-database



Score of one test for one search term =
term frequency * inverse **document frequency**

How often the term appears in this test.
We reward repetition of terms in the
test.

How many of the tests contain the term.
We penalize terms that appear in many
tests.



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](#)

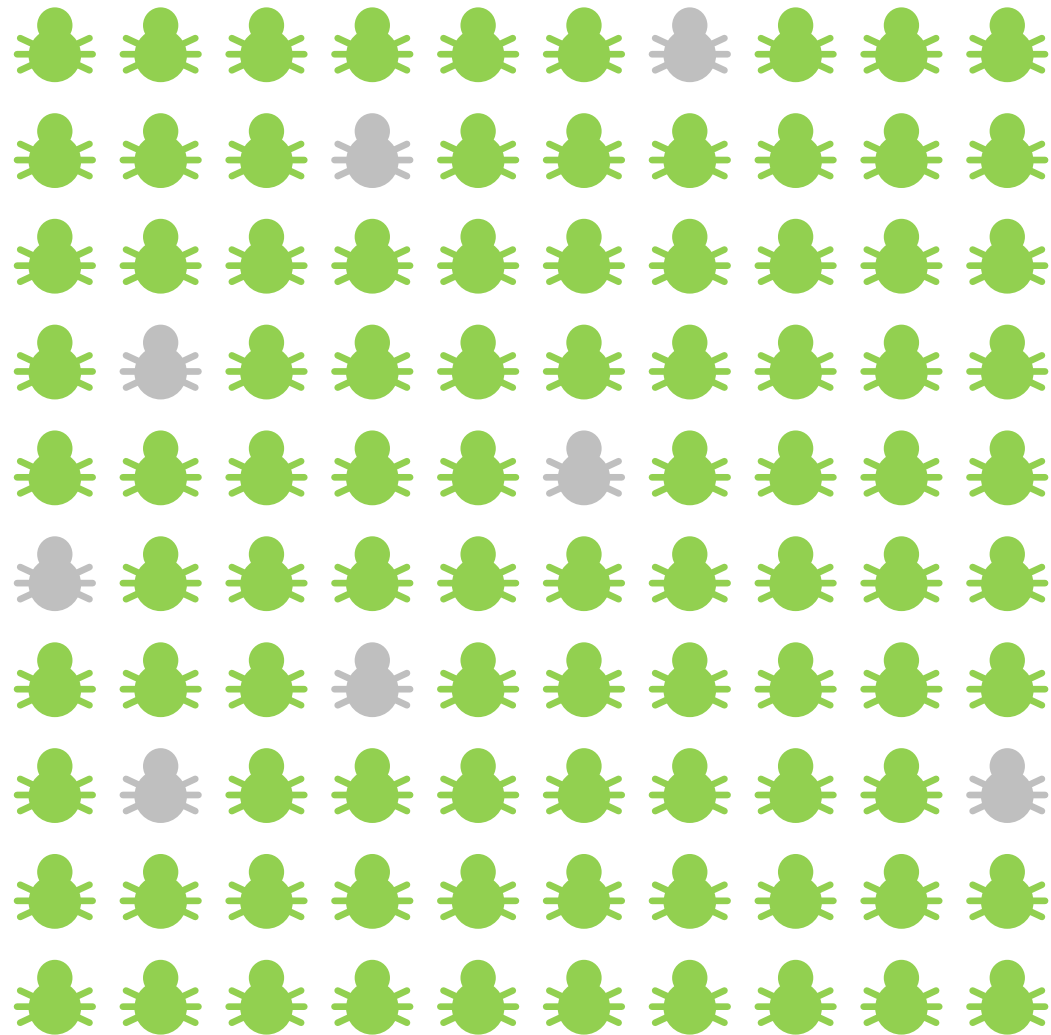
Test Case 3

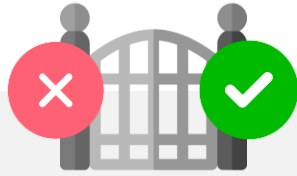
27.11.2020 — It's a full-featured tool that lives inside, and seamlessly integrates with Jira. Xray aims to help companies improve the quality of their ...





90 %





Quality Gate

90% of bugs founds in X% of the time

13%

AI Test Clustering



Change-based
Testing

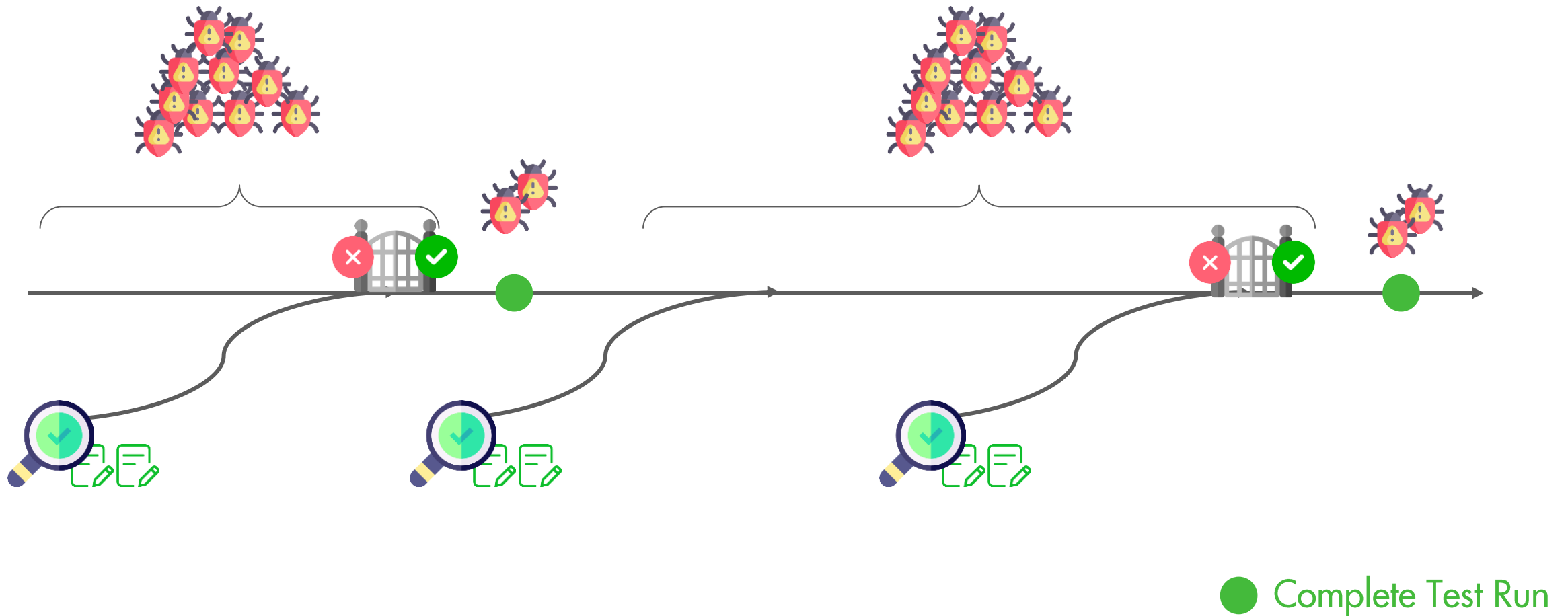
4%

Similarity Scoring

Precision and Effort



Shift Left



Thanks for Listening

Get my slides and future talks



Jonas Bogenberger
bogenberger@cqse.eu

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

Code Test

```

43 LOG.debug("Debit Transaction from Account: Account Updated.");
41 }
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.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.CREDIT_TRANSACTION_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.getAccountName() + ")");
12 toAt.setTransactionType(transactionTypeRepository.findByCode(Constants.DEBIT_TRANSACTION_CODE));
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  * 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(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.getAcco
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);
```

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