# Everyone Wants to Shift Left

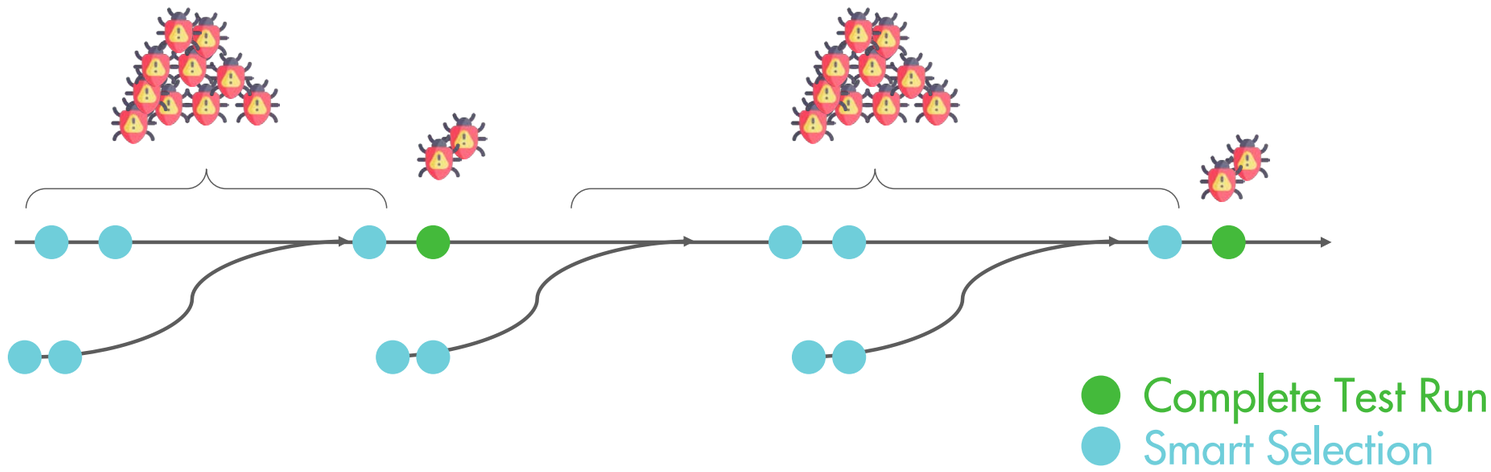
but our tests are just too slow...

# How to Shift Left Despite Long-running Tests

Running tests as often and as early as possible is the dream of all testers.

But what if our tests take hours or even days? What if we conduct manual testing?
For many, this dream seems unattainable or at least impractical.

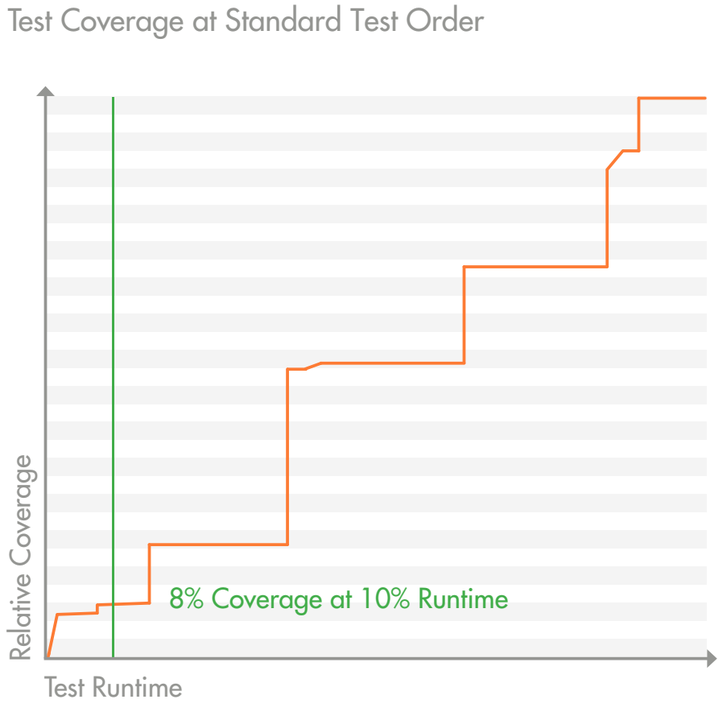
Research shows a possible solution called »Test Selection«: **Running a small subset that finds most bugs** in a fraction of the total runtime, yielding fast feedback even from long-running tests.

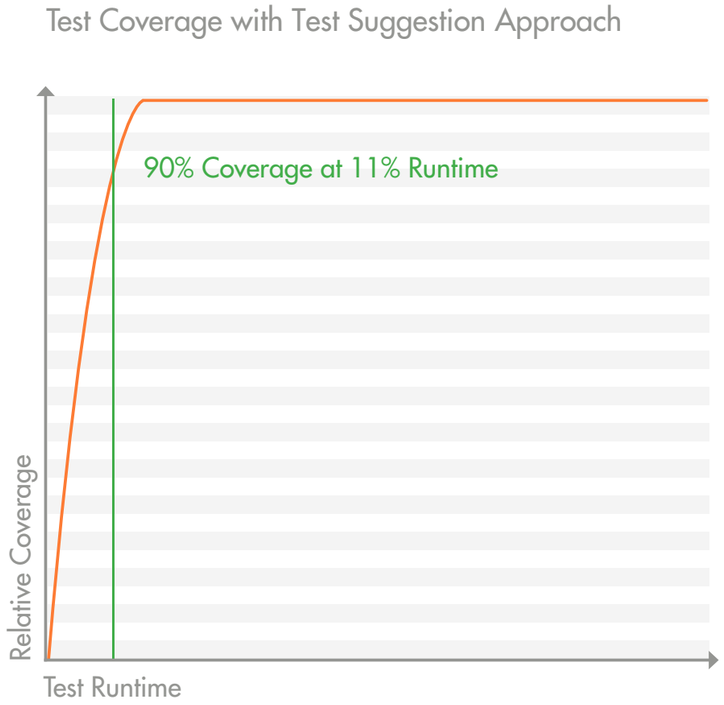


# Option 1: Change-independent Test Suggestion

Intuition: Cover the entire codebase as quickly as possible (think: smoke tests), by running a subset of **the most dissimilar tests**.

Pro: Easy setup for decent speed up.



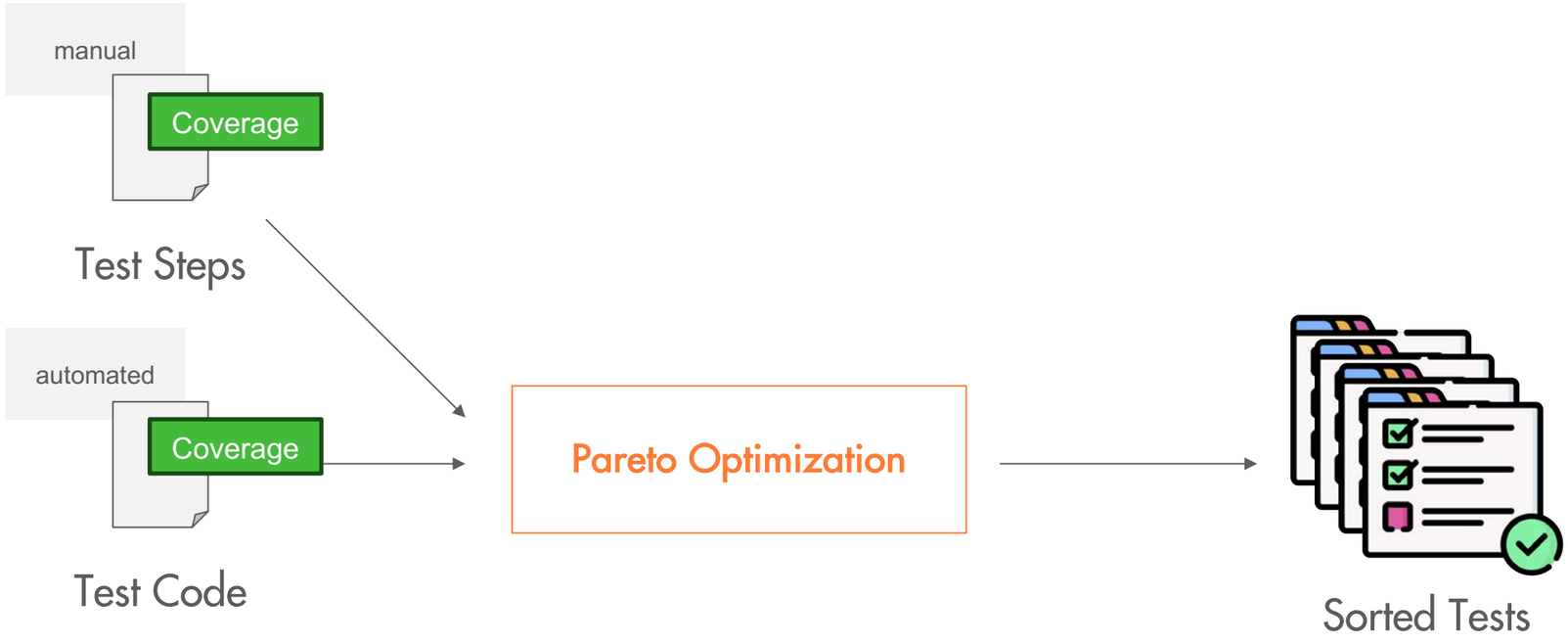
# Pareto Optimization

90% bugs

11% time

Clusters and suggests the tests with the most dissimilar code coverage.

- **Data input**: Testwise code coverage data from your tests (be they automated or manual).
- **Outcome**: Finds 90% of the bugs in 11% of the total runtime.



# AI Test Clustering

90% bugs

13% time

Uses a large language model to cluster similar tests based on their content.

- **Data input**: Automated test code and/or descriptions of manual test steps.
- **Outcome**: Finds 90% of the bugs in 13% of the total runtime.

# Option 2: Change-based Test Suggestion

Intuition: Focus on recent changes, by running a subset of the tests **most relevant for the changes**.

Pro: More speed up from more input data.

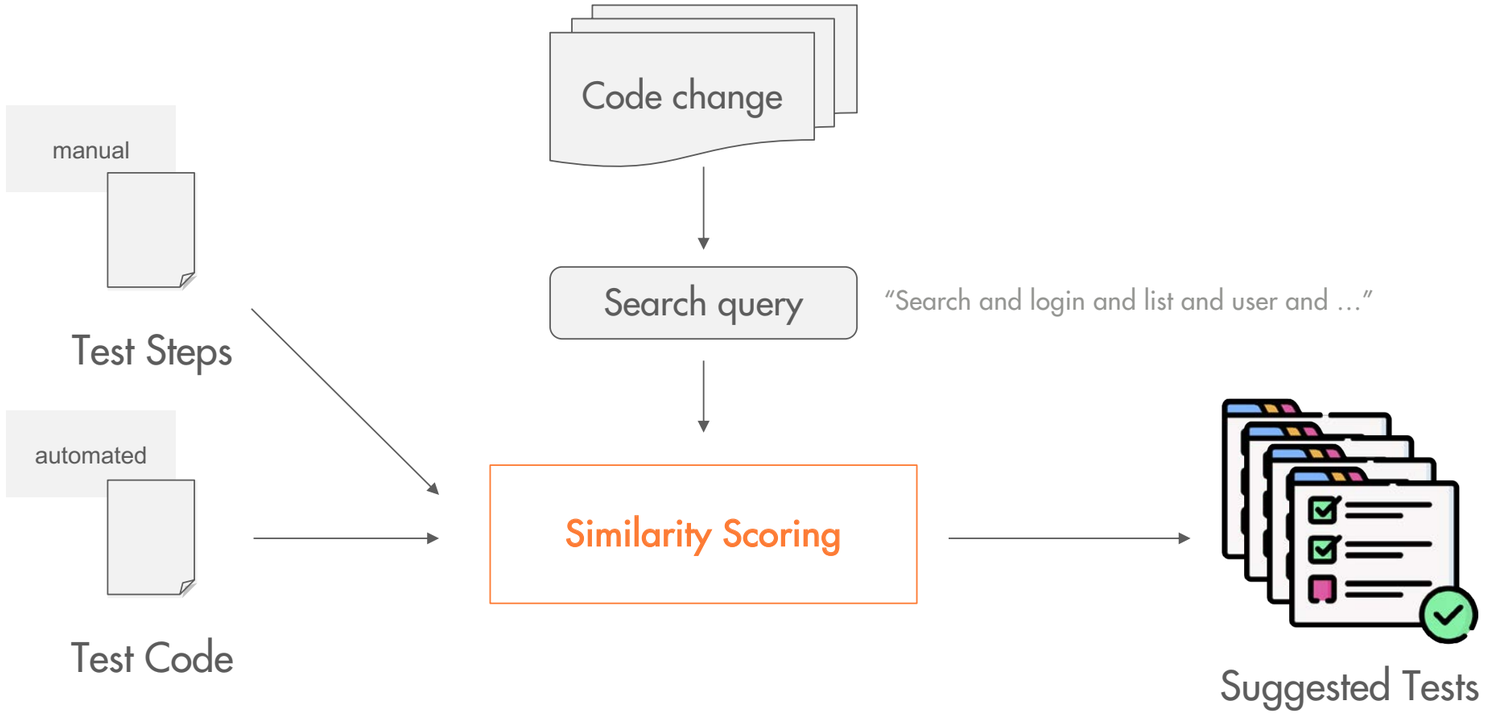
Con: More effort to set up than change-independent approaches.

# Similarity Scoring

90% bugs  4% time

Suggests tests that fit the changes best, based on a search query derived from the code changes.

- **Data input**: Automated test code and/or descriptions of manual test steps, and changed code.
- **Outcome**: Finds 90% of the bugs in 4% of the total runtime

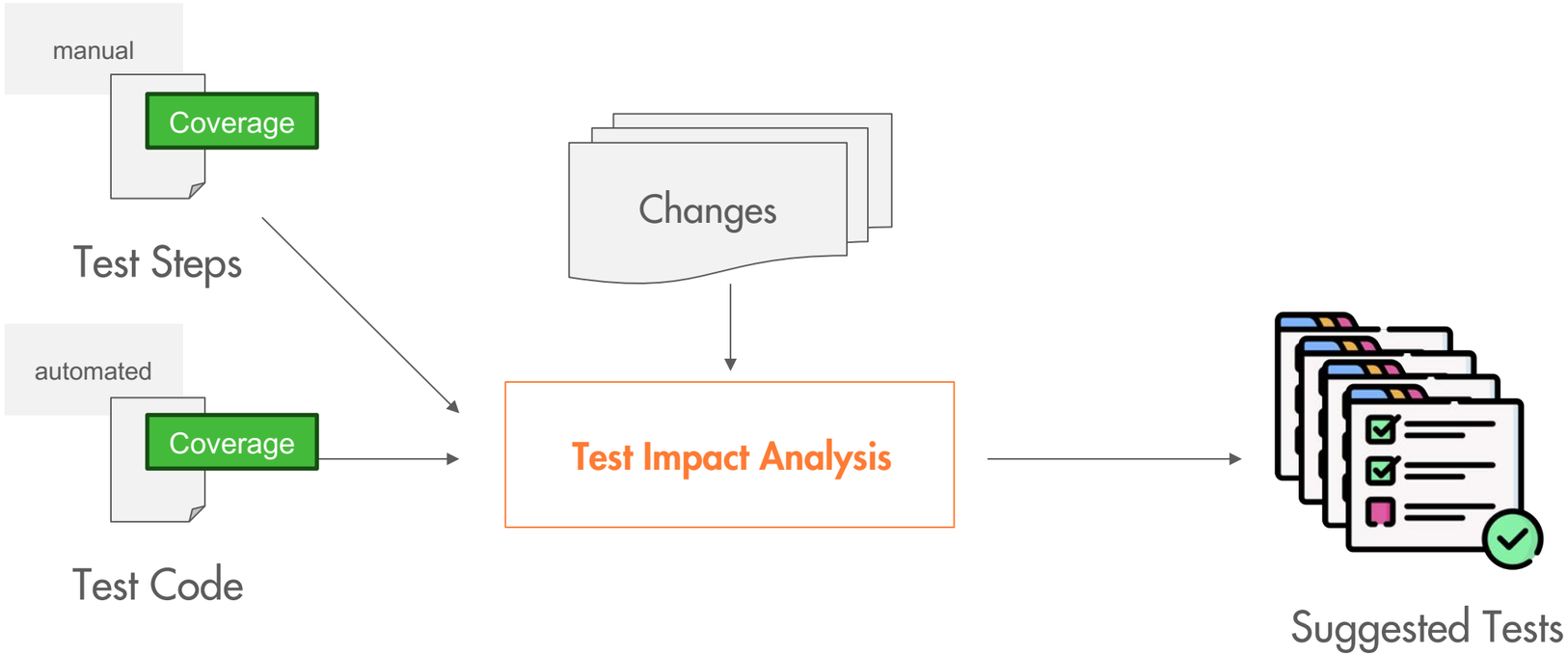


# Test Impact Analysis

90% bugs  2% time

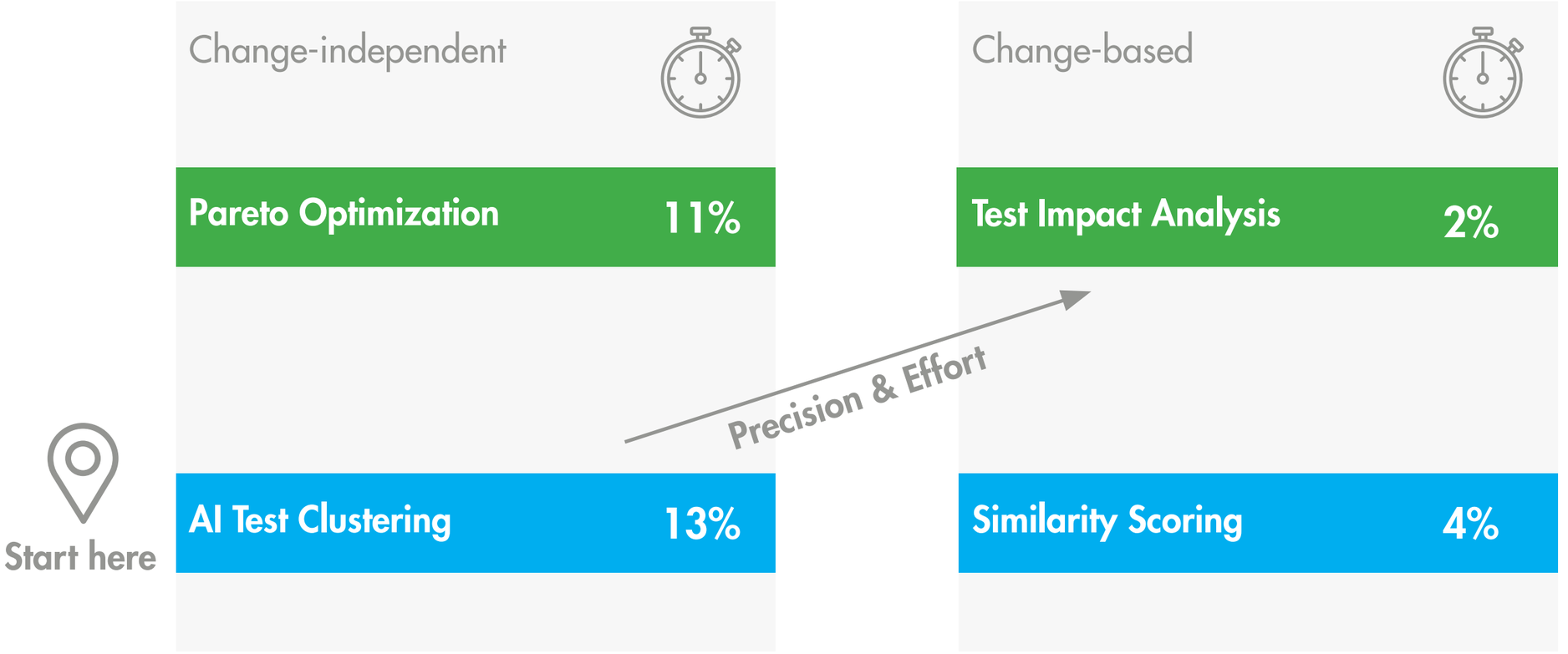
Suggests only tests that run through the changed code, based on test coverage.

- **Data input**: Automated test code and/or descriptions of manual test steps, code coverage, and changed code
- **Outcome**: Finds 90% of the bugs in 2% of the total runtime



# Which Method Fits Best?

Which method fits best depends on how slow your test suite runs and how much pain this causes.
All approaches find 90% of the bugs in X% of the total runtime.




**Test Coverage**

- ⊕ Less time needed to find a large number of bugs
- ⊖ Requires much more effort to set-up and maintain

**Test Content**

- ⊕ Low effort
- ⊖ Need to run more tests to achieve the same result

## Would you like to find out which of the approaches is the best fit for your tests?

## Contact us!

**Fabian Streitel**
streitel@cqse.eu
+49 159 04046270

**Johannes Veihelmann**
jveihelmann@cqse.eu
+49 152 02150022

## Want to dive deeper on your own first?

## Watch our free talk on all options and live demos!