

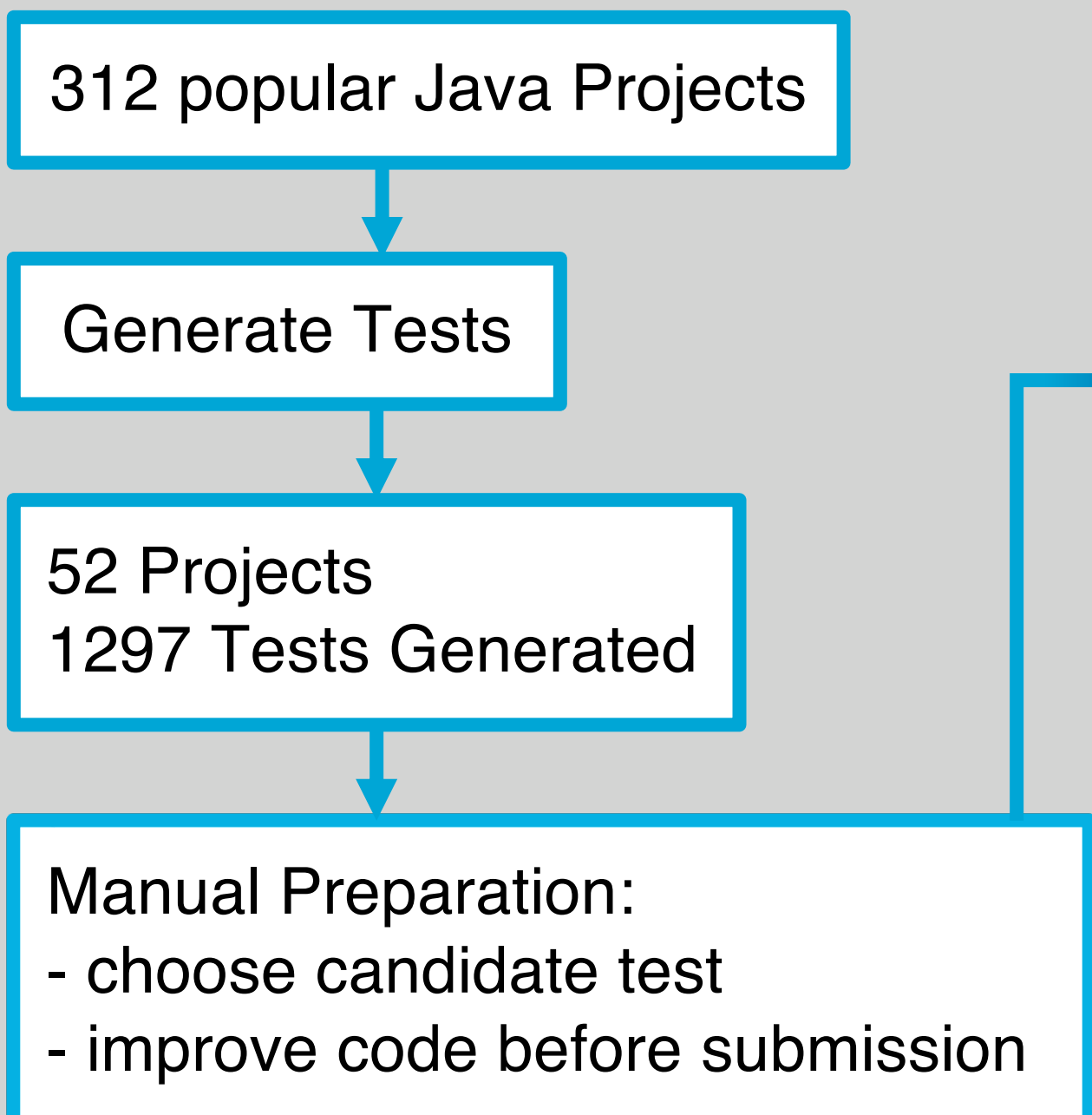
Tests: Small programs, used by developers to check whether their code behaves as they expect.

Seen as tedious and unproductive
 → Too little testing is done
 → Research: let's try to generate them!

But: output complex, difficult to understand
 → not directly usable

What would developers change before accepting an automatically generated test into their test suite?

Let's explore this with open source Java projects & their maintainers!



RQ1: Preparing Tests For a Pull Request

RQ1.1: Criteria to Select a Test

<ul style="list-style-type: none"> Coverage False Positive Simple G/Setters with Non-Standard Names Assertion Unrelated to New Coverage No Explicitly Thrown Exception Readability and Understandability Unclear Link between Test and Coverage 	X	<ul style="list-style-type: none"> Coverage Understandability ✓ Curious about reaction
---	---	---

RQ1.2: Manual Edits to Improve

Assertion Style	IDE Recommendation
Unnecessary Code	Formatting / Linters
Adapt To Other Edits	Move Test

Two kinds of selection / change tasks:
 (1) could be **automated by customizing** setup to project,
 (2) highly benefit from the **developer's comprehension**.

Open 39 Pull Request & Analyze the Maintainer's Reaction

lacinoire commented on 15 Jun · edited · Contributor

Hey 😊
I want to contribute the following test:

Test that a `org.eclipse.lemminx.commons.BadLocationException` is thrown when the parameter `line` of `Position.<init>` is set to `-1`. This tests the method `TreeLineTracker.getLineInformation`. This test is based on the test `testEmptyDocumentInc`.

Curious to hear what you think!

(I wrote this test as part of a research study at TU Delft. [Find out more](#))

```

}

@Test
public void testGetLineInformation() throws BadLocationException {
    assertThrows(BadLocationException.class, () -> {
        TextDocument document = new TextDocument("", "");
        document.setIncremental(true);
        Position position = new Position(-1, 0);
        document.offsetAt(position);
    });
}

@Test
public void testPositionAtInc() throws BadLocationException {

```

Outcome: 15 accepted, 9 closed

RQ2: Feedback From Maintainers

RQ2.1: Proposed Changes

- Style Conventions
- Unnecessary Code
- Defensive Programming
- Simple / Meaningful Scenario
- Merge / Extend Test
- Test All Scenarios
- Move Test / Data

RQ2.2: Information Requested

- Purpose, Added Value
- Coverage Increase
- Description of Test
- Compare to Other Tests
- Curiosity

RQ2.3: Judgment Rationale

- Incomplete Contribution
- Would Not Test
- Test Untested Scenarios
- Code Quality, Clarity

Implications

For Practitioners
 Informed decision on whether to adopt test generation

For Researchers and Tool Designers
 Collaborating with developer effective alternative to automating further?
 Providing information to judge and change is key.

Should we focus on **automating further** or **leave edits to developers** and support them in understanding the tests?