

Ai for the Test case authorization and Patent study

I

Revolutionizing Software Testing

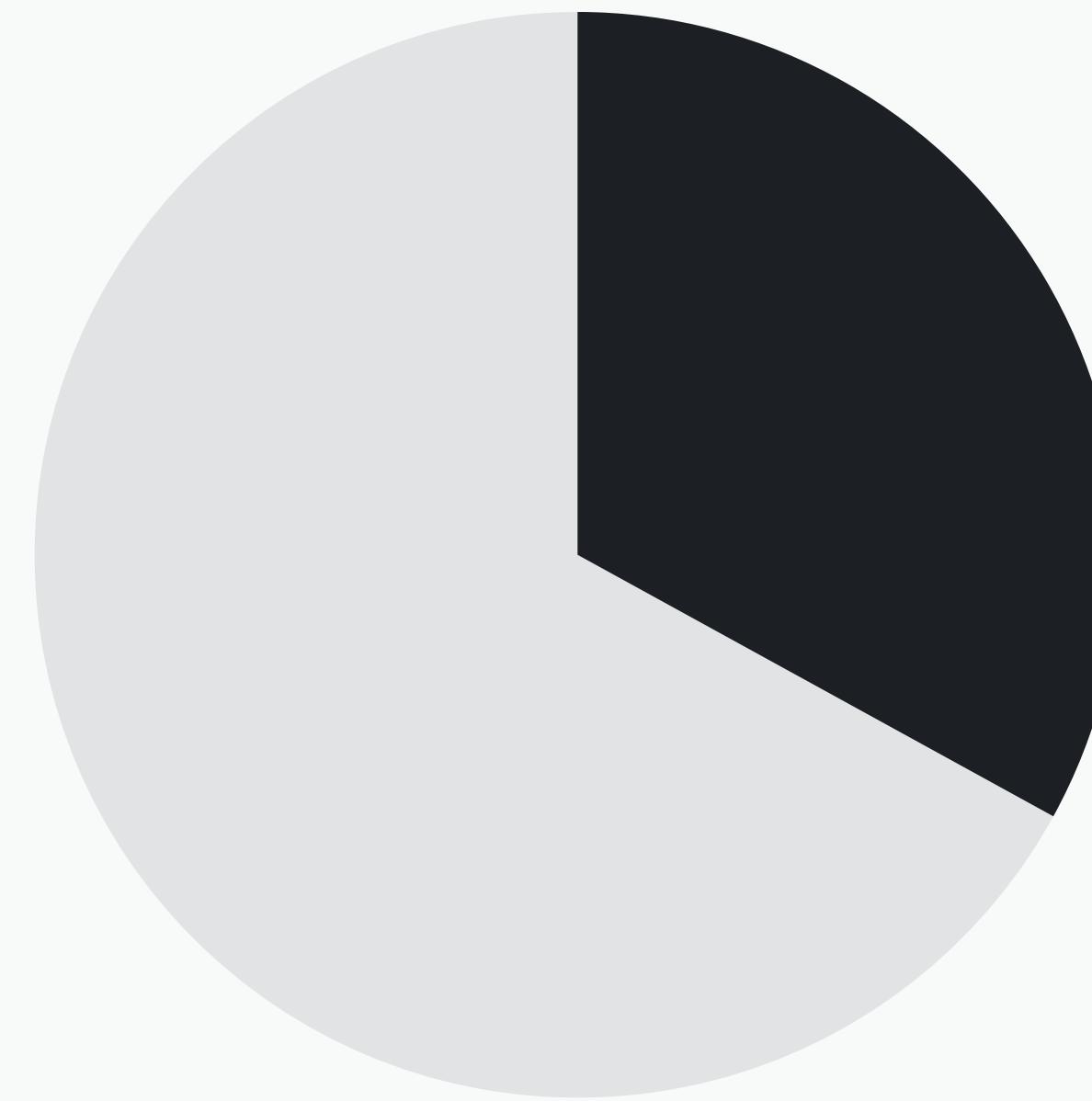
00

by Somnath Thander
for **Wipro**

01

The Bug Problem

High Costs & Lost Time



time and money
goes into the whole
software life cycle

32%

A recent study shows that 32% of time and cost is spent on bug localization during the software development life cycle [42]. Bug localization remains a challenging task due to the significant gap

“Bug localization is
challenging due to the gap between
natural language
and
programming language”

02

Why It's Hard to Find bugs

Traditional Methods

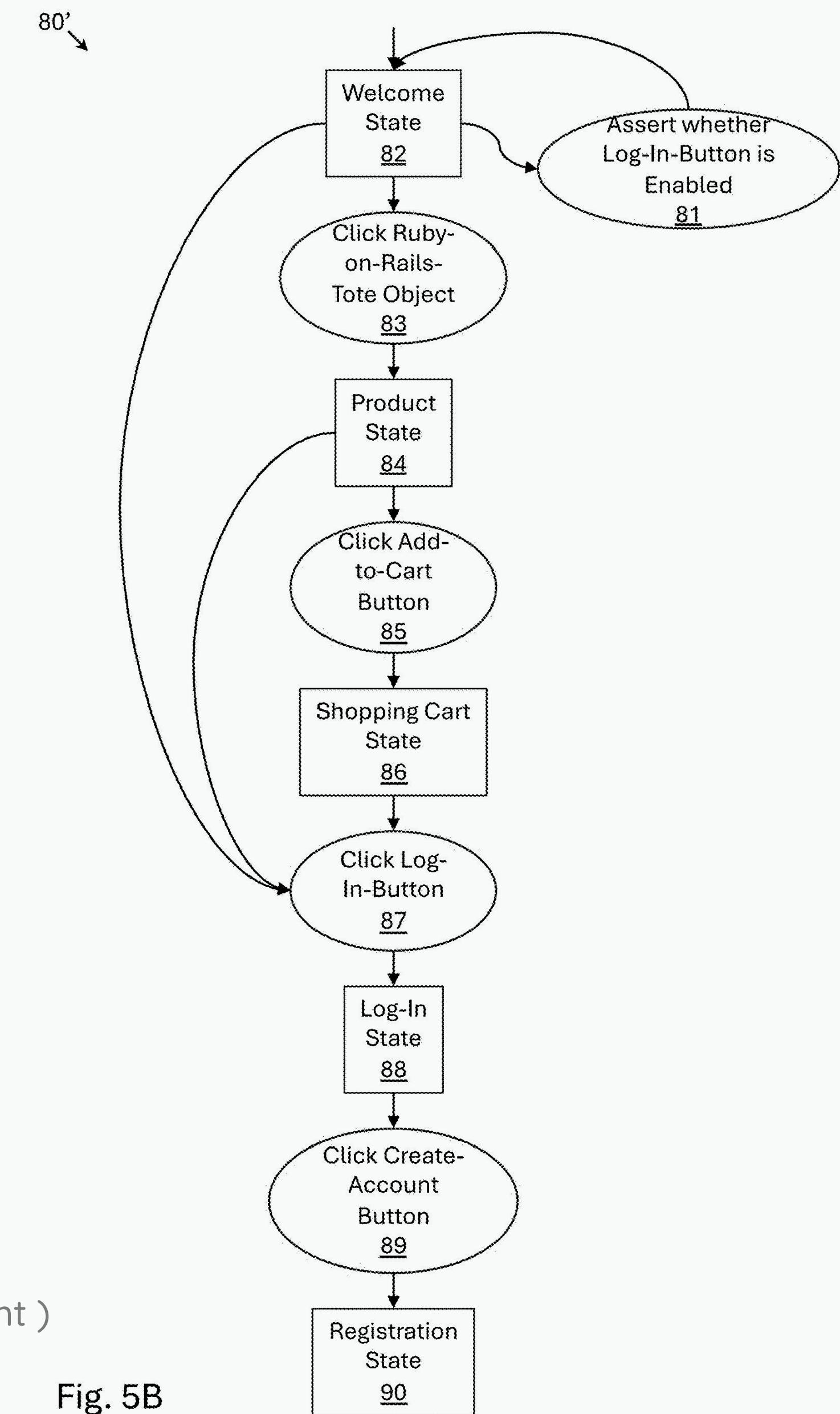
Manual Testing: Slow, Labor-intensive, and prone to human error. Difficult to Scale

Scripted Automation: Brittle, high-maintenance, and it struggles with dynamic software.

Ref : Methods for creating test scripts and/or updating a model of an application (US Patent) Filed March 7, 2024

Link : <https://patents.google.com/patent/US12353317B1/en?oq=US12353317B1>

A typical test case is a **sequence** of states and actions for a user journey.



Ref : Methods for creating test scripts and/or updating a model of an application (US Patent)

Link : <https://patents.google.com/patent/US12353317B1/en?oq=US12353317B1>

Fig. 5B

03

Bug Hunting

Power of Context



How can an Ai understand a project's unique context?

knowledge graphs, such as `testcase` `depends` `on` `bug`, `bug` `has` `suspect` `code`, to facilitate understanding of relationships between entities. Obviously, without contextual information on the bug report and suspicious code, it is difficult for the model to understand them. A number of researchers have argued that bug reports and source code are difficult to understand without contextual information in the literature [59].

Knowledge Graph

system of connected **entities** and **relationships** to maps
project-specific knowledge.

Ref : A Knowledge Enhanced Large Language Model for Bug Localization

Link : <https://dl.acm.org/doi/10.1145/3729356>

KEPT

Knowledge Enhanced Pre-Trained model

Text Knowledge Graph

Reads project documents
to learn project-specific
terms and concepts.



Code Knowledge Graph

Analyzes source code to
map how different code
components are related.

Ref : A Knowledge Enhanced Large Language Model for Bug Localization

Link : <https://dl.acm.org/doi/10.1145/3729356>

KEPT : Knowledge Enhanced Pre-Trained model

A Knowledge Enhanced Large Language Model for Bug Localization

FSE086:5

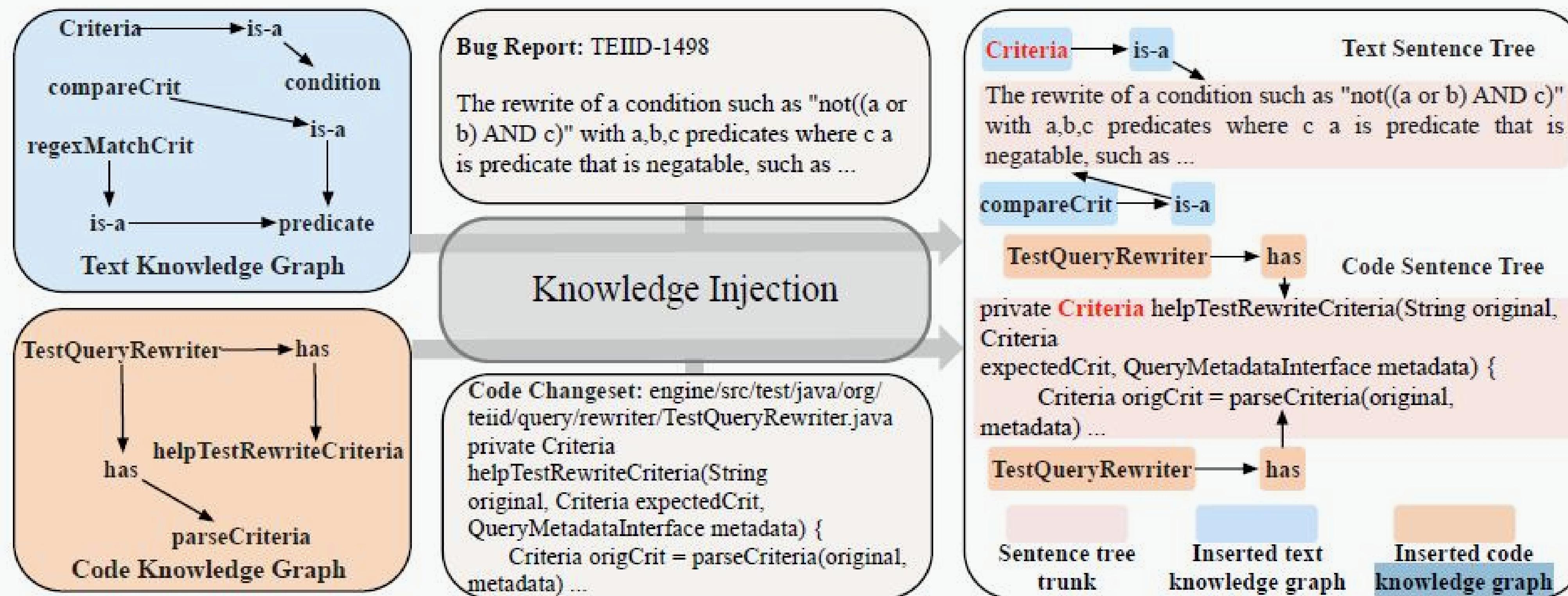
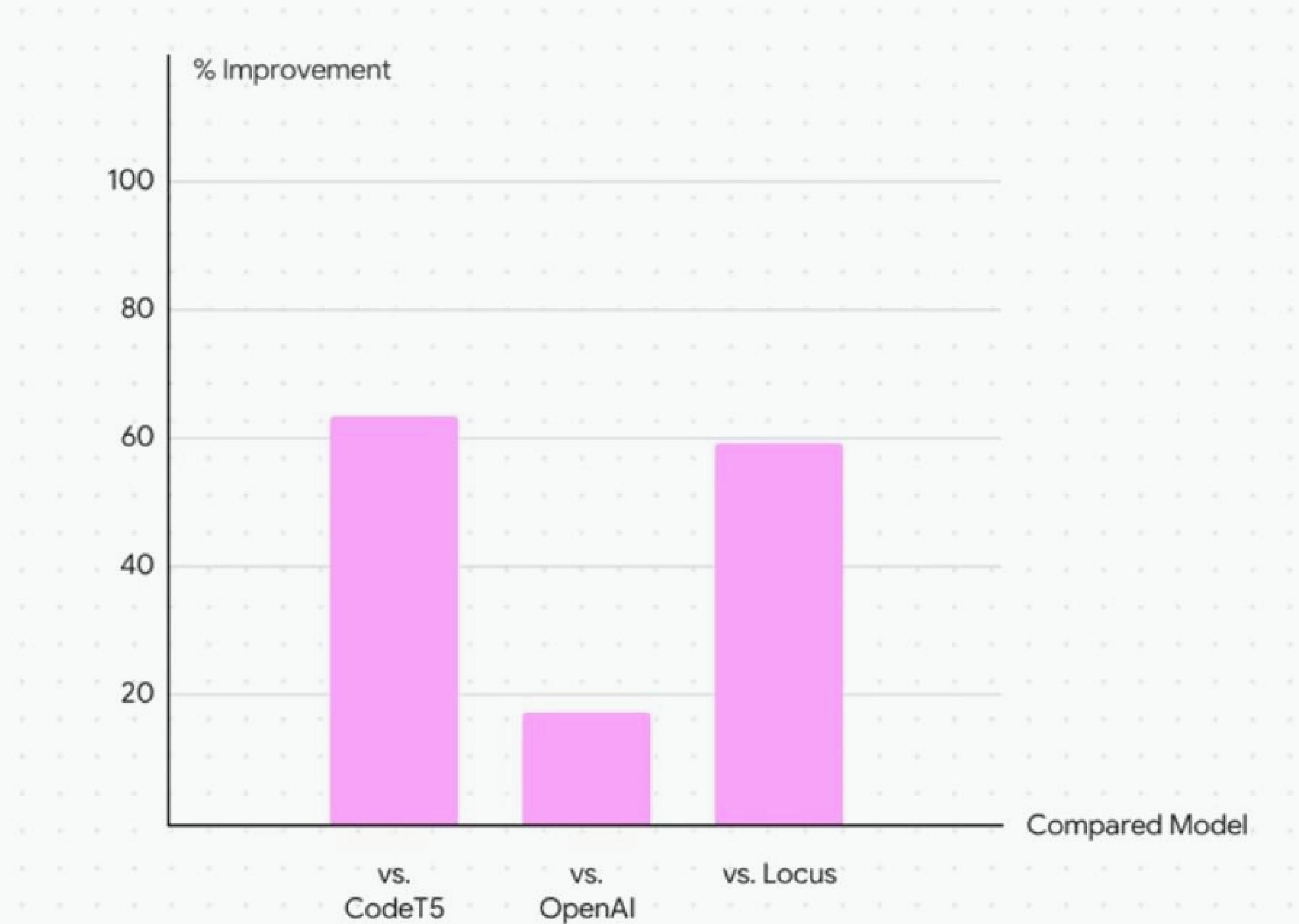


Fig. 1. An example of knowledge injection

Ref : A Knowledge Enhanced Large Language Model for Bug Localization

Link : <https://dl.acm.org/doi/10.1145/3729356>



Ref : A Knowledge Enhanced Large Language Model for Bug Localization
Link : <https://dl.acm.org/doi/10.1145/3729356>

04

The Automated Tester

Test orchestration

Applicant: **INFOSYS LIMITED**, Bangalore (IN)

Inventors: **BASAVARAJ VEERAPPA SOMAWAGOL**, Bangalore (IN);
BALAJI THANGAVELU, Bangalore (IN); **SREEKANTH SREEDEVI SASIDHARAN**, Bangalore (IN)

01

Receive Input

User provides test data via a web application.

03

Select Module

model choose the correct model option from the plurality of options

02

Parse Data

Sys analyze the Input to understand the request

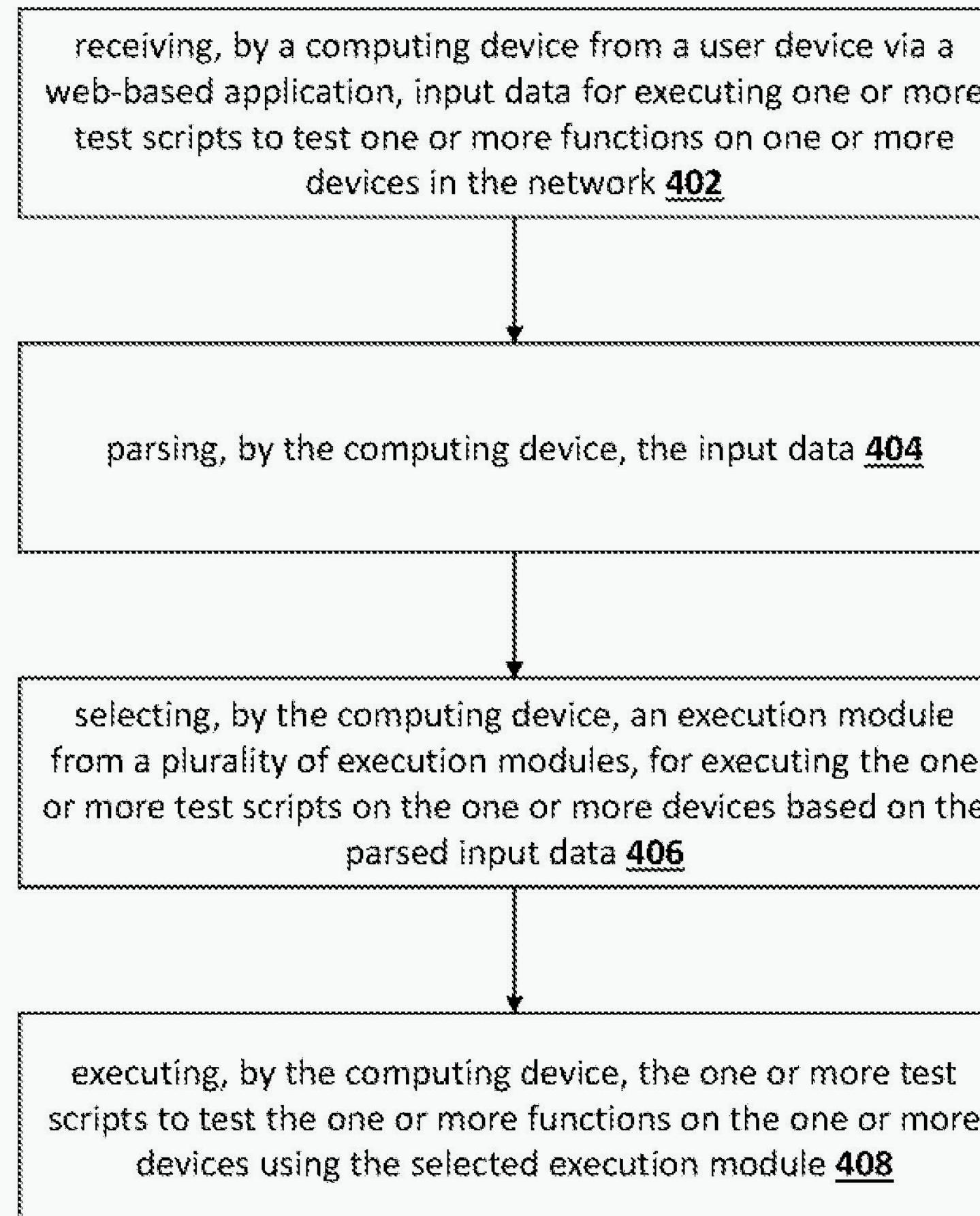
04

Execute Scripts

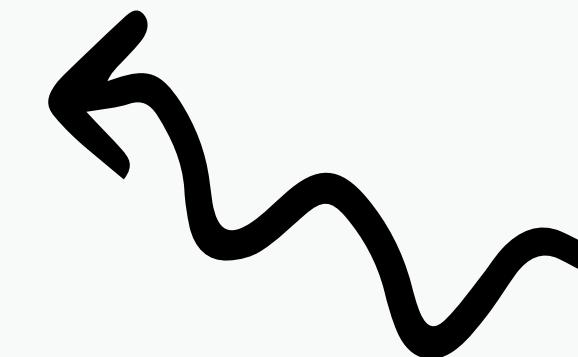
run the selected tests on one or more devices in the network

Ref : Method and system for framework agnostic smart test orchestration in network test automation (US Patent) Apr 3, 2025

Link : <https://patents.google.com/patent/US20250112848A1/en?oq=US20250112848A1>



400



A high-level view of an automated test orchestration process.

Applicant: **INFOSYS LIMITED**, Bangalore (IN)

Inventors: **BASAVARAJ VEERAPPA SOMAWAGOL**, Bangalore (IN);
BALAJI THANGAVELU, Bangalore (IN); **SREEKANTH SREEDEVI SASIDHARAN**, Bangalore (IN)

01

Detect Failure

The system identifies that at least one test script has failed.

02

Determine the Rerun

ML model analyzes historical data to decide if a re-run is useful.

03

Execute again

If valuable, the system automatically **re-runs** the failed test.

Ref : Method and system for framework agnostic smart test orchestration in network test automation (US Patent) Apr 3, 2025

Link : <https://patents.google.com/patent/US20250112848A1/en?oq=US20250112848A1>

Dashboard/Test Scheduler

JOB-1 VALID JOB-2 VALID

TestBed_2

DUT Model : XYZ Mini PC Windows

1.2.2

CMTS **Server 1** **WIRELESS CLIENT**

Test Suite 2

Test Suites
 Test case 1 - Description
 Test case 2 - Description
 Test case 3 - Description

Total Test Cases: 3

ID	Name	Delete
TC 1	Test case 1 description	<input type="checkbox"/>
TC 2	Test case 2 description	<input type="checkbox"/>
TC 3	Test case 3 description	<input type="checkbox"/>

Rerun on Fail Iterations: **1** DUT Telemetry

Automated Mail Test Software Update

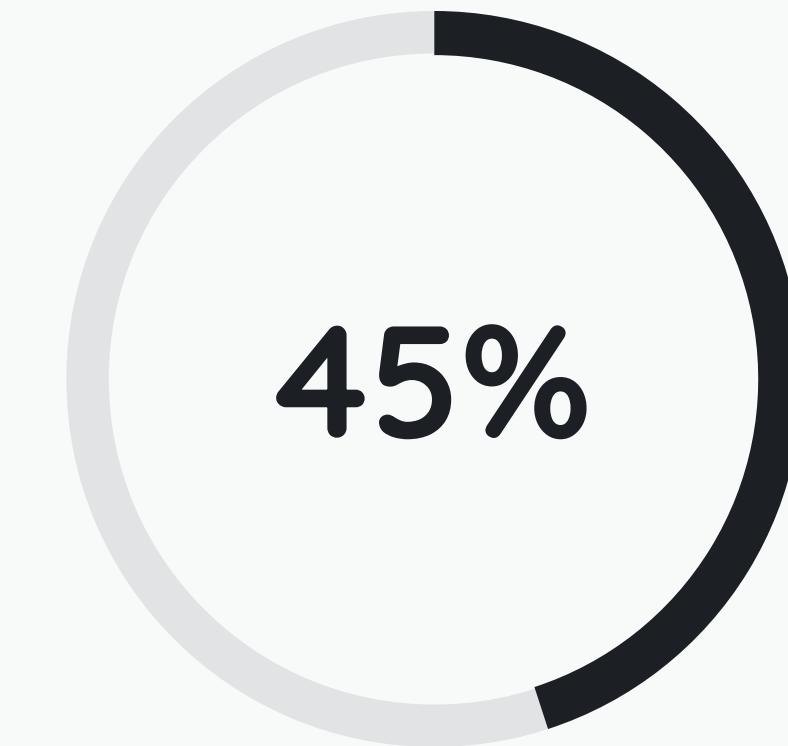
FIG. 7

[0018] FIG. 7 is another screenshot of a user interface depicting a test scheduler, in accordance with an exemplary embodiment of the present disclosure;

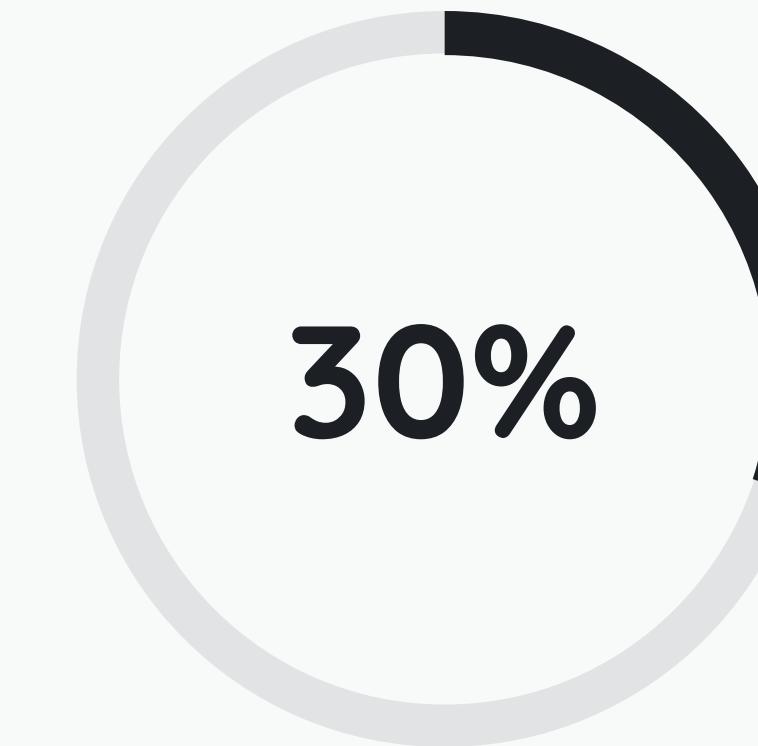
05

The Future of testing

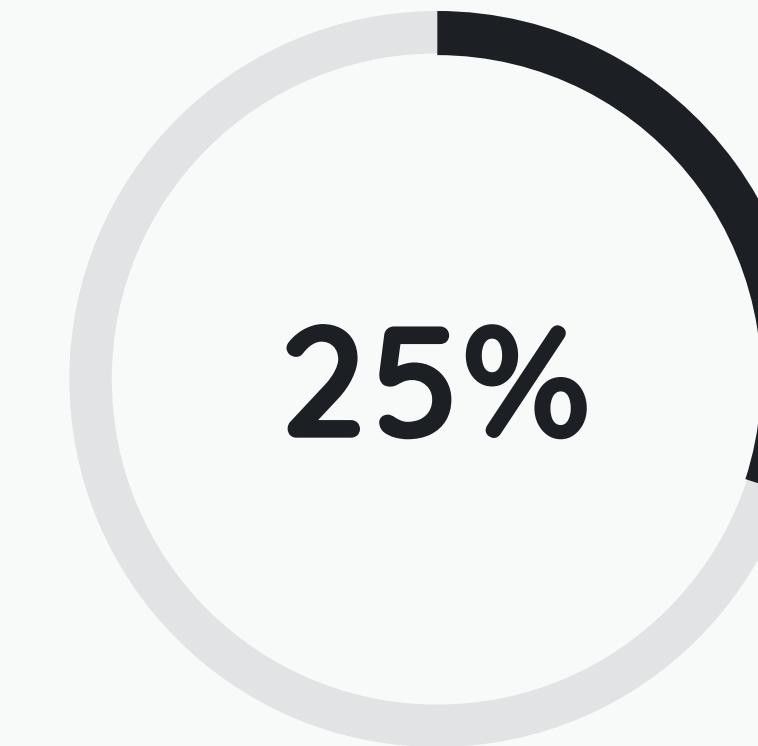
New Challenges ????



Test Case Preparation



Program Debugging



Auto Bug Repair

survey of 102 studies
shows where LLMs are
used most in software
testing.

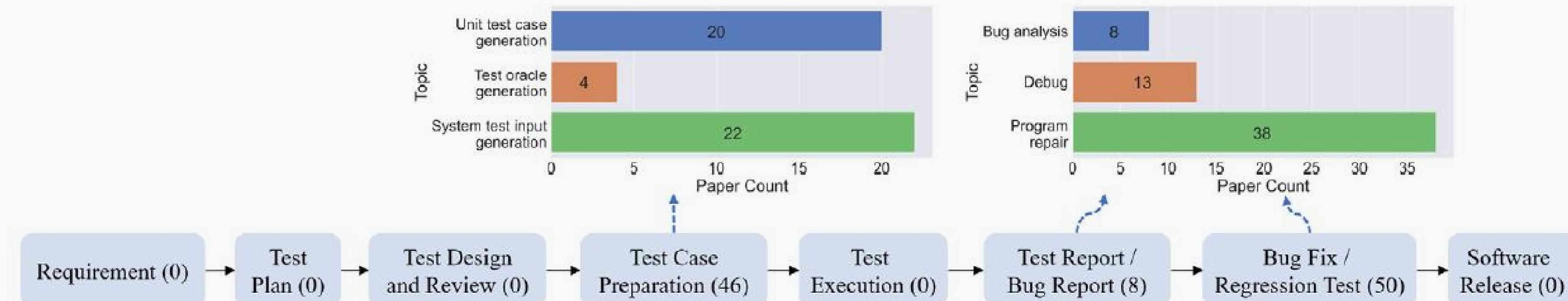


Fig. 2. Distribution of testing tasks with LLMs [56].

Challenges ???

- Testing the **ai models** themselves
- **Quality assurance** for auto-generated code
- Defining **Human-ai** collaboration
- Ensuring generated code is **secure** and **efficient**

Thank you !!