NIODebugger: Repairing Non-Idempotent-Outcome Tests with LLM-Based Agent

Kaiyao Ke (kaiyaok2@illinois.edu)

University of Illinois Urbana-Champaign

ICSE 2025

Problem : Non-Idempotent-Outcome (NIO) Tests

- Tests that pass in 1st run and fail in subsequent runs
- First studied in by Wei et al. (ICSE ‘22)
- Uncovers polluted states
- Related to test order dependency (OD Tests)

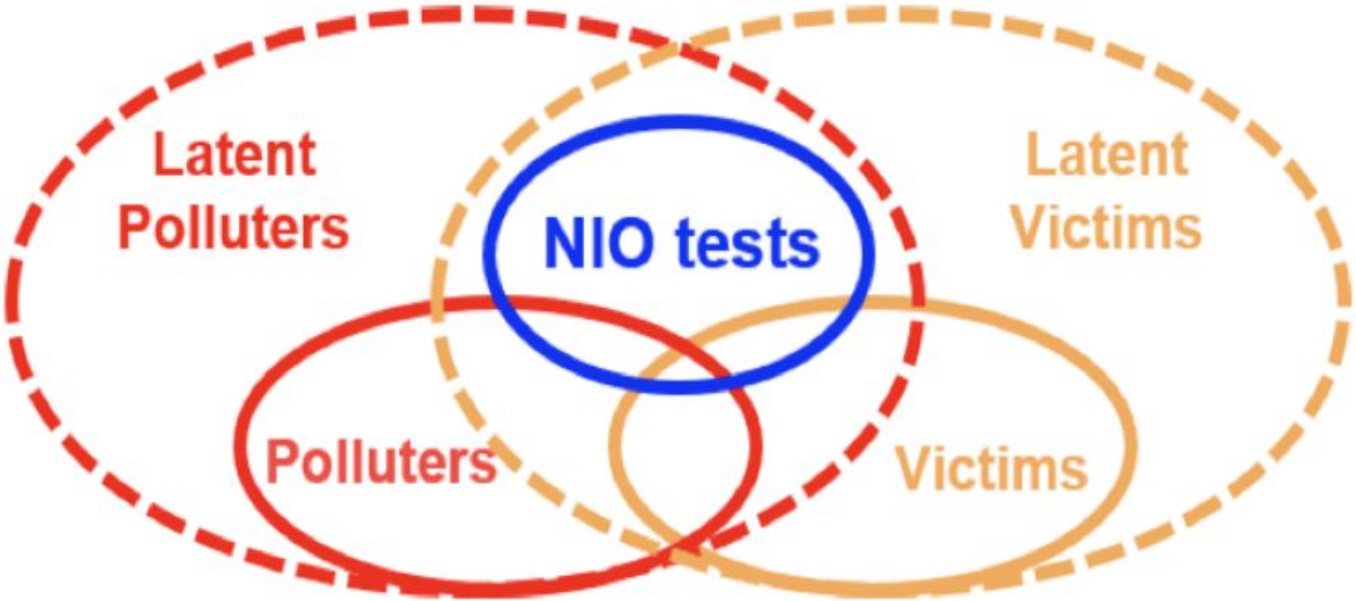
Example: NIO Test vs. OD-Victim Tests

An **OD-Victim** fails when running after a **Polluter**

```
1  // shared variables x, y, z, w are initialized to 0
2  void t1() { assert x == 0; } // victim
3  void t2() { x = 1; } // polluter
4  void t3() { assert z == 0; } // latent-victim
5  void t4() { y = 1; } // latent-polluter
6  void t5() { assert w == 0; w = 1; } // NIO
```

If Developer adds **void t6() { assert w == 0}** and **void t7() { w = 1}**…

Why Fixing NIO Tests?



Fix NIO Tests: Limitations of Existing Approaches

- **Symbolic approaches:**

  Hard to find/reset polluted states

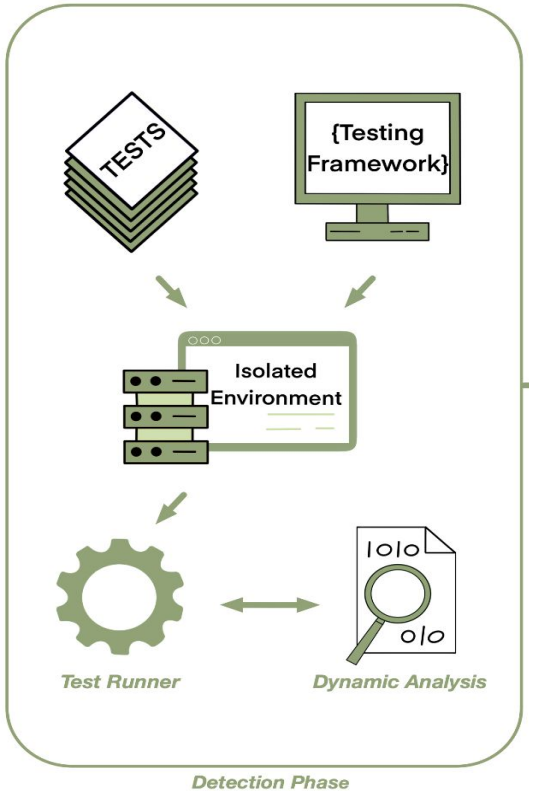
  (ODRepair, ICSE'22, Wei et al.)

- **Non-agentic LLM-based approaches:**

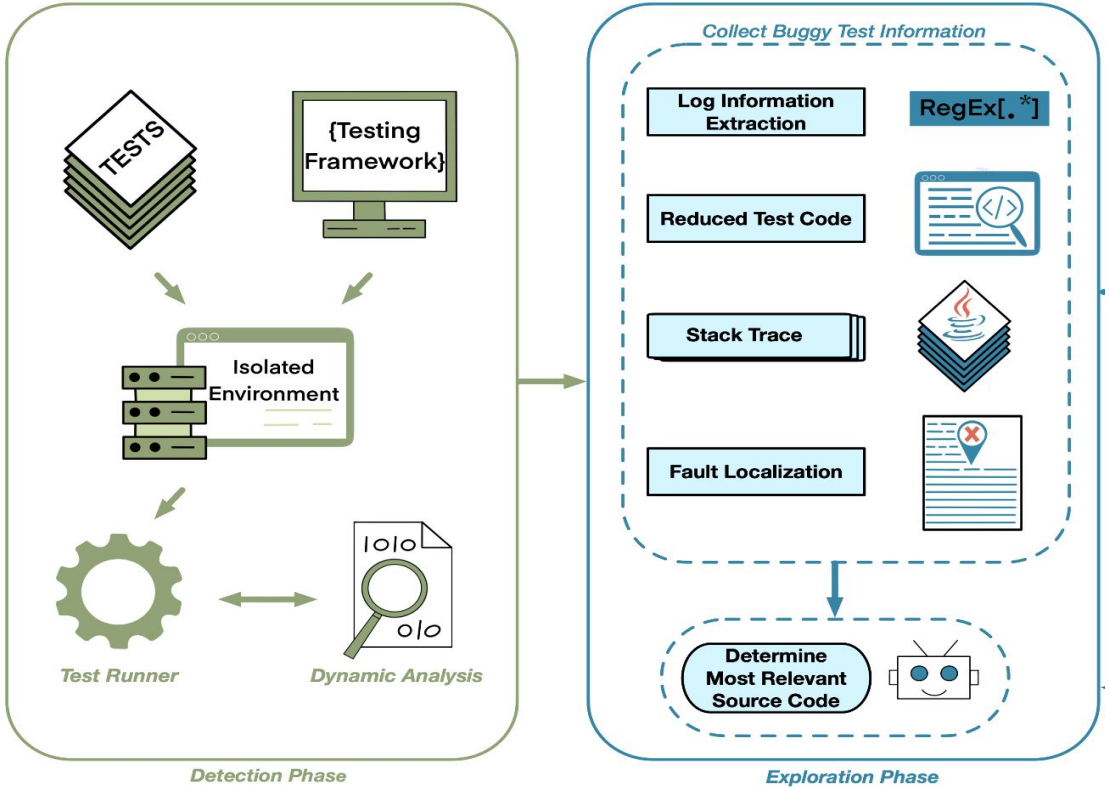
  Limited context window & Too much noise

  (FlakyDoctor, ISSTA'24, Chen et al.)

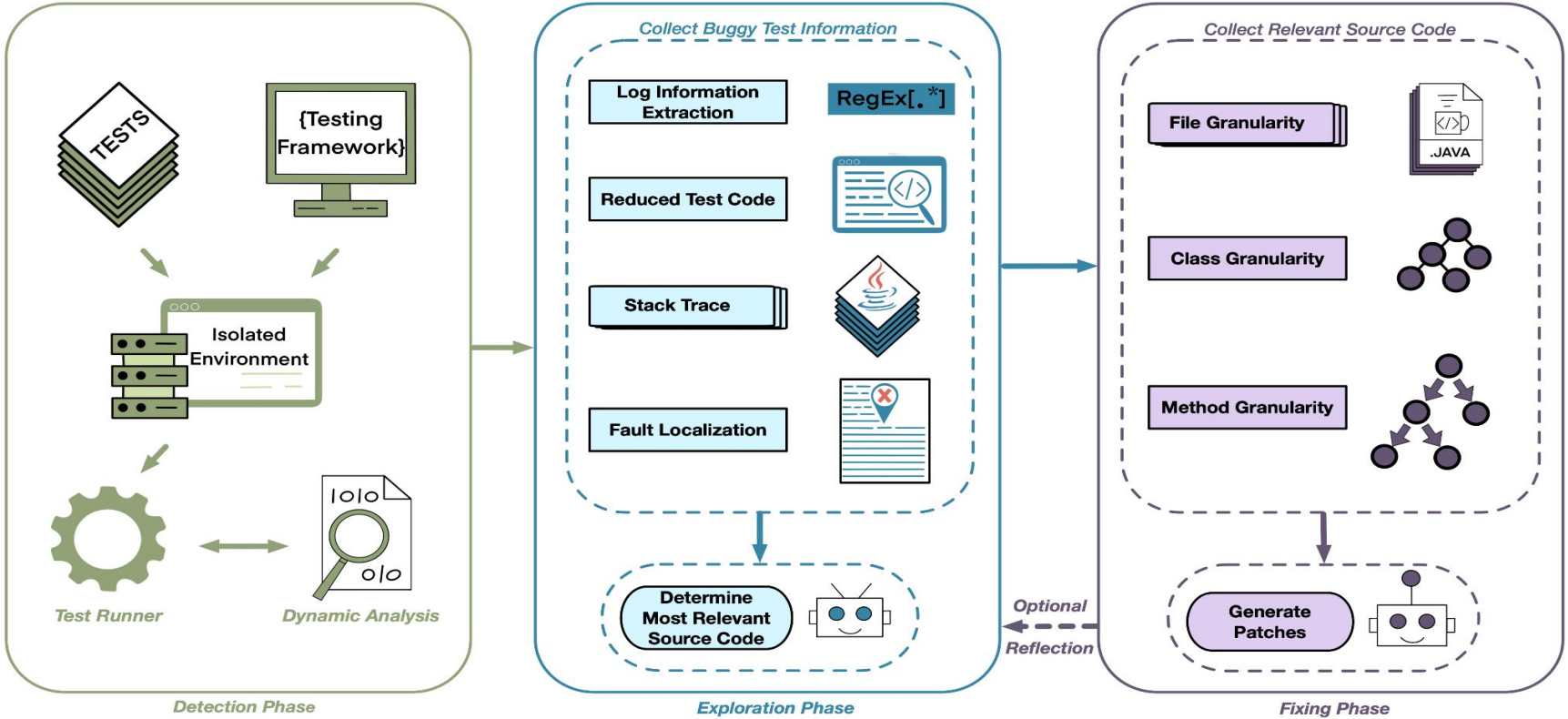
Solution: NIODebugger



NIODebugger - Agentic Workflow



NIODebugger - Agentic Workflow



Sample Patch by NIODebugger (from eclipse-vertx/vert.x)

```
1   public void testDeployClass() {
2   +    ReferenceSavingMyVerticle.myVerticles.clear();
3        vertx.deployVerticle(//...).onComplete(onSuccess(
             deploymentId -> {
4          ReferenceSavingMyVerticle.myVerticles.forEach(
               myVerticle -> {
5            assertEquals(deploymentId, myVerticle.deploymentID)
                 ;
6            assertEquals(config, myVerticle.config);
7            assertTrue(myVerticle.startCalled);
8          });
9        }));
10       // ...
11  }
```

5 Actions for Source Code Exploration

1. **Find Method:** *${method_name}*
2. **Find Class:** *${class_name}*
3. **“Hypothesized” method:** *${method_name}*
4. **Explore Similar Files**
5. **Directly Fixable**

RQ1: Performance of NIODebugger

NIODebugger Variant or Baseline	Correct Patches
NIODebugger-GPT-4	101 (58.72%)
NIODebugger-GPT-3.5-Turbo	68 (39.53%)
NIODebugger-Qwen2.5-Coder-32B-Instruct	27 (15.69%)
NIODebugger-DeepSeek-Coder-33B-Instruct	20 (11.63%)
ODRepair	59 (34.30%)
iFixFlakies	0 (0%)

RQ2: Contribution to Real World Software

Project	SHA	NCNB Lines	Total Tests	Test Assertions	NIO Tests	Correct Patches	Accepted Patches
apache/dubbo	20f252d	287009	6415	13555	34	19	19
apache/hadoop	ecf665c	1929672	10874	78983	31	14	7
kiegroup/jbpm	1558f0d	295347	3675	19965	23	11	0
sismics/docs	afa7885	24600	74	999	21	17	17
apache/cxf	eea3c9b	695144	9528	30777	15	3	3
alibaba/COLA	1a8c433	11654	69	167	12	11	0
brianfrankcooper/YCSB	ce3eb9c	25434	76	733	12	12	0
apache/wicket	58d953e	220299	2811	10727	6	5	5
spring-cloud/spring-cloud-netflix	2a8b7ed	12259	235	461	4	1	1
ebean-orm/ebean	d034821	222188	1069	11791	2	2	2
stleary/JSON-java	8983ca6	14263	647	1561	2	2	2
apache/rocketmq	b37d283	242774	1676	7652	2	1	1
apache/tika	f78dc99	173227	1982	10305	1	1	0
apache/tinkerpop	8bb5d16	172540	25269	14855	1	1	1
eclipse-vertx/vert.x	0eb288b	140880	4849	9596	1	1	0
apache/incubator-kie-optaplanner	8c2fb1e	208565	3963	10910	1	0	N/A
Red5/red5-server	eb75c16	64046	170	434	1	0	N/A
spring-projects/spring-retry	9442435	11468	387	903	1	0	N/A
stanfordnlp/CoreNLP	2460079	619842	1459	7035	1	0	N/A
winder/Universal-G-Code-Sender	445cd19	92947	750	2514	1	0	N/A

RQ3: Ablation Study

Technique	Correct Patches
NIODebugger-GPT-4 without RSCE & DA	26 (15.12%)
NIODebugger-GPT-4 without RSCE	42 (24.42%)
NIODebugger-GPT-4 without DA	55 (31.98%)
NIODebugger-GPT-4	101 (58.72%)

RSCE = Relevant Source Code Extraction **DA** = Dynamic Analysis

```
1   public void testRegister() {
2       FilterFactory.INSTANCE.register(new NothingFilter());
3       // Other logic
4   +   FilterFactory.INSTANCE.unRegister("Nothing");
5   }
```

Contributions

- First automatic approach to fix NIO tests
- First LLM-based agent for flaky test fix
- Detecting and fixing flaky tests in 1 lifecycle
- A dataset comprising zero-day NIO tests

Future Work

- **Agentic Based Flaky Test Fixing:**

  Multi-Patch Exploration, Progress Tracking, …
- **Open-source (code) LLMs:**

  Generalize to agentic workflow