FLAKY MANAGER PROJECT REPORT

BACKGROUD

Flaky test is identifying a test that sometimes fails, but if you retry it enough times, it passes, eventually[1]. Flaky Tests occur in repeated tests. One of the characteristics of automated testing is that testing is easier to repeat. Therefore, compared to manual testing, flaky tests is more of a product of automated testing. In the Agile and DevOps era, automated testing is the main theme of software testing. Flaky Tests has become a common and prominent problem with the popularity of automated testing.

PERSPECTIVE

After investigation, I found that the test results of Flaky Tests are uncertain. Some people believe that Flaky Tests can not achieve the test objectives, and may even destroy the test value. As you can see from the **DeFlaker: Automatically Detecting Flaky Tests**[2],DeFlaker is very professional in finding flaky tests. I think finding them is the first step. We need to solve these flaky tests. So my idea is to build a tool to manage the life cycle of flaky tests, from discovery to record and then fix it.

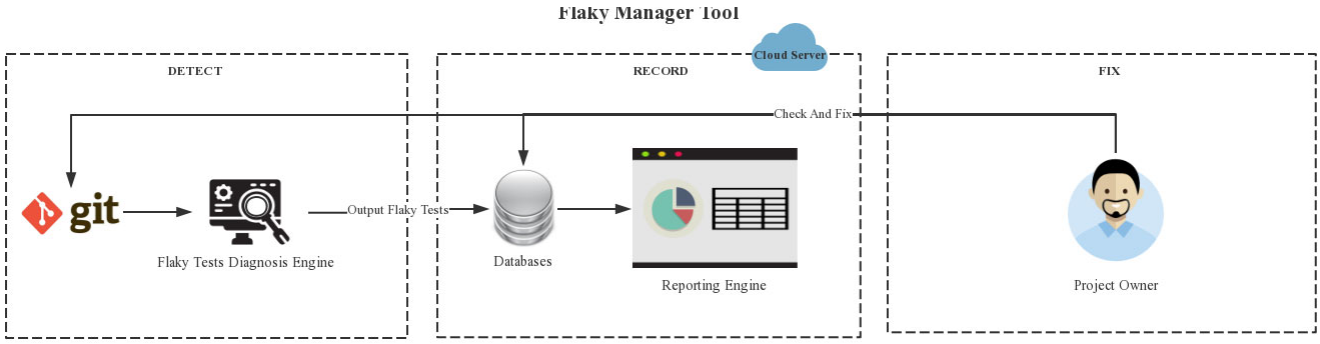


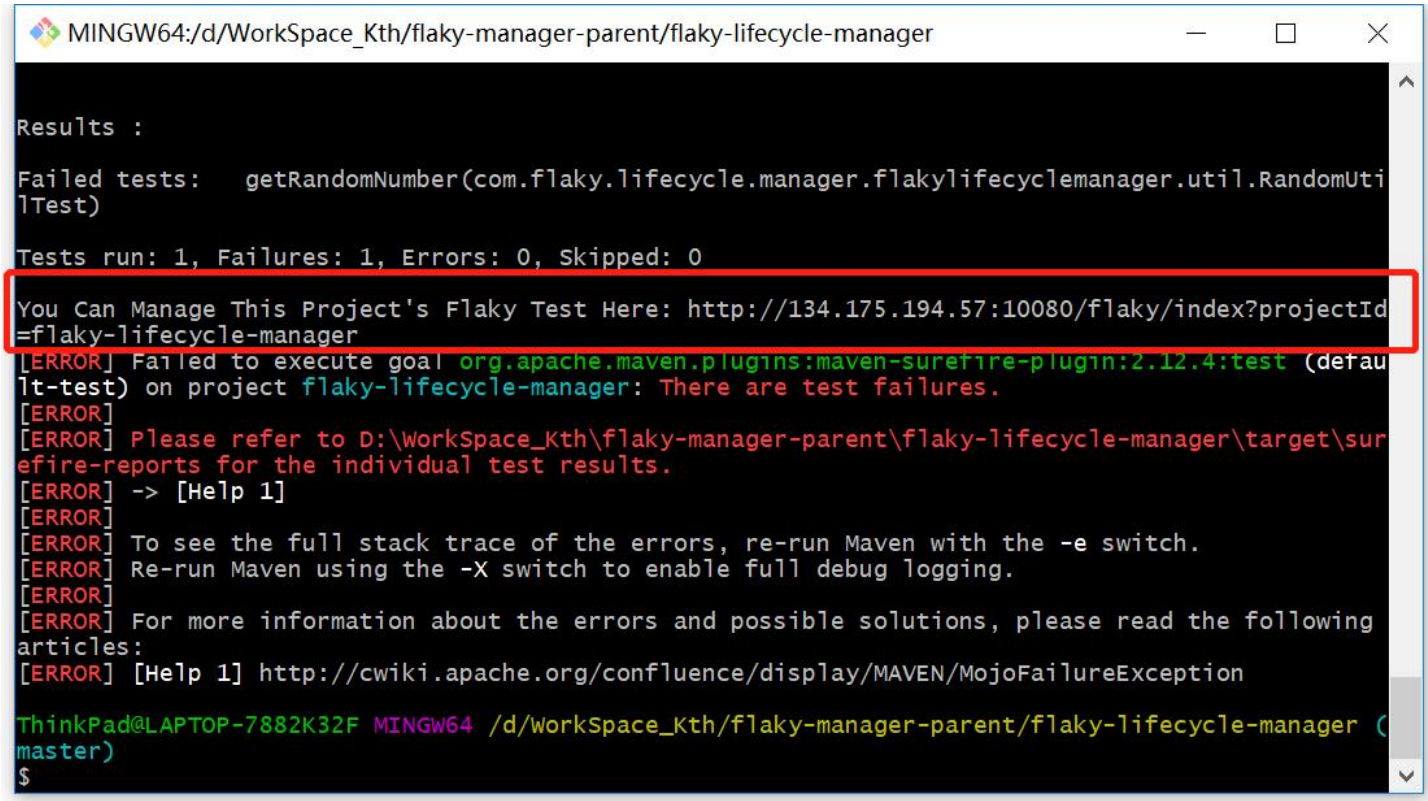
Figure 2 : High-level architecture of Flaky Manager, with three phases : detect, record and fix

High-level architecture of Flaky Manager

HIGH-LEVEL ARCHITECTURE

In my design, The tool is divided into three parts：

- **The first part** is used to discover flaky tests tools, need to be integrated into the target project, detects the flaky tests and show the information of the flaky manager tool.



Flaky Test Result

- **The second part** is to report the discovery of flaky tests to the cloud server, for the same test case, we upload data related to the operating environment each time,Because changes in the operating environment of the program often lead to the generation of flaky tests.

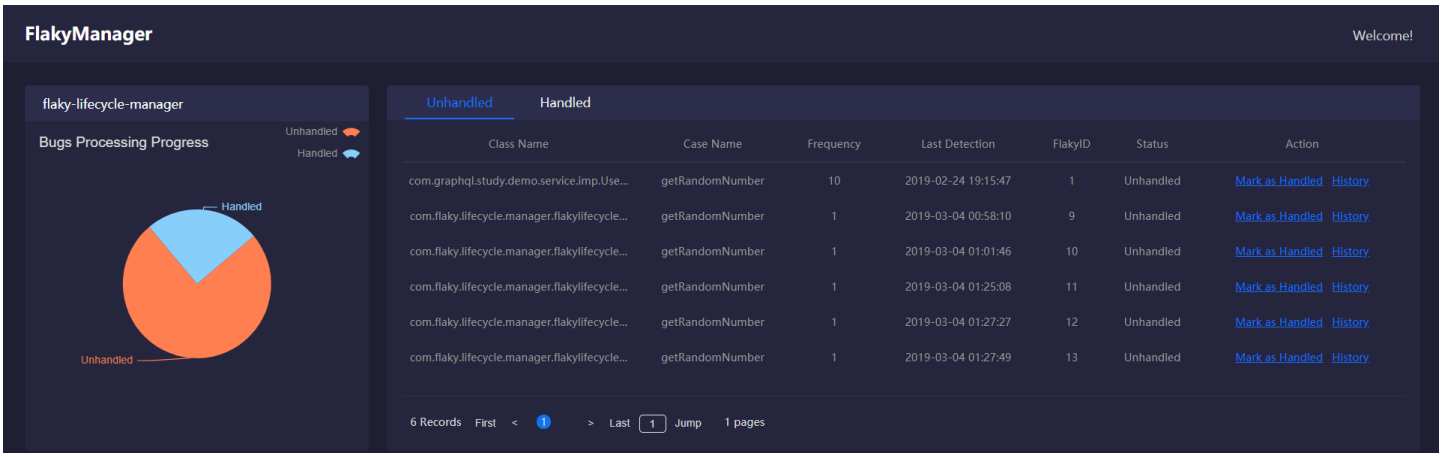
```
mysql&gt; desc flaky;
+------------------+-------------+------+-----+---------+----------------+
| Field            | Type        | Null | Key | Default | Extra          |
+------------------+-------------+------+-----+---------+----------------+
| flaky_id         | int(13)     | NO   | PRI | NULL    | auto_increment |
| flaky_status     | int(2)      | YES  |     | NULL    |                |
| last_detect_time | datetime    | YES  |     | NULL    |                |
| last_sha_1       | varchar(255) | YES  |     | NULL    |                |
| class_name       | varchar(255) | YES  |     | NULL    |                |
| unit_test_name   | varchar(255) | YES  |     | NULL    |                |
| detect_count     | int(100)    | YES  |     | NULL    |                |
| project_id       | varchar(255) | YES  |     | NULL    |                |
+------------------+-------------+------+-----+---------+----------------+
8 rows in set (0.04 sec)

mysql&gt; desc flaky_history;
+-------------------+-------------+------+-----+---------+----------------+
| Field             | Type        | Null | Key | Default | Extra          |
+-------------------+-------------+------+-----+---------+----------------+
| flaky_history_id  | int(13)     | NO   | PRI | NULL    | auto_increment |
| flaky_id          | int(13)     | NO   |     | NULL    |                |
| flaky_status      | int(2)      | YES  |     | NULL    |                |
```

```
| detect_time        | datetime     | YES  |     | NULL    |                |
| sha_1              | varchar(255) | YES  |     | NULL    |                |
| class_name         | varchar(255) | YES  |     | NULL    |                |
| unit_test_name     | varchar(255) | YES  |     | NULL    |                |
| detect_count       | int(100)     | YES  |     | NULL    |                |
| project_id         | varchar(255) | YES  |     | NULL    |                |
| environment_detail | text         | YES  |     | NULL    |                |
+--------------------+--------------+------+-----+---------+----------------+
10 rows in set (0.05 sec)

mysql&gt;
```

- **The third part** is the cloud server used to manage flaky tests, and we can see the reports related to the target project at a glance and manage their lifecycle by flaky tests status tags.



Flaky Manager Page Capture

WORKFLOW

In the workflow design of tools, the integration phase of tools should be simplified or even imperceptible. Developers should focus on solving these problems, clearly know the current situation of their projects, charts can do this job well, cloud servers can ensure that records can be well stored. By separating the data for each project in the project ID area, you can focus the developer's attention.Finally, the program can't confirm the uncertain results, and it will be a compromise for the developer to confirm that flaky tests have been repaired.

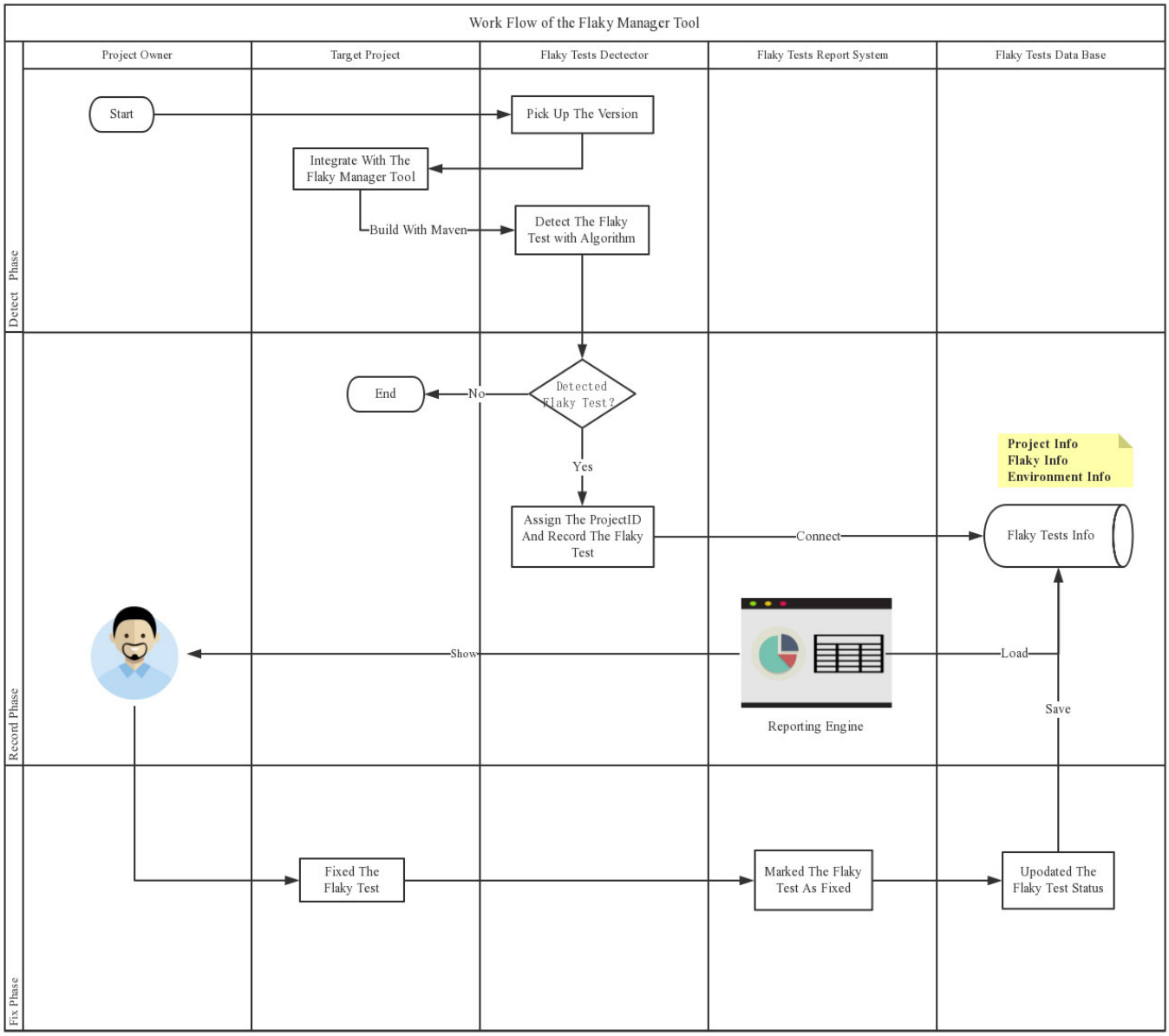


Figure 3 : Work Flow of the Flaky Manager Tool

FlakyManager Workflow

IMPLEMENTATION

On the flaky test discovery standard, the tool determines whether it is a flaky test in a relatively simple way, and the related code is not changed on the same test case, and the result of the test and the last one is judged to be faky test. If there is no

previous test result, we will not mark it as flaky test for the sake of prudence.

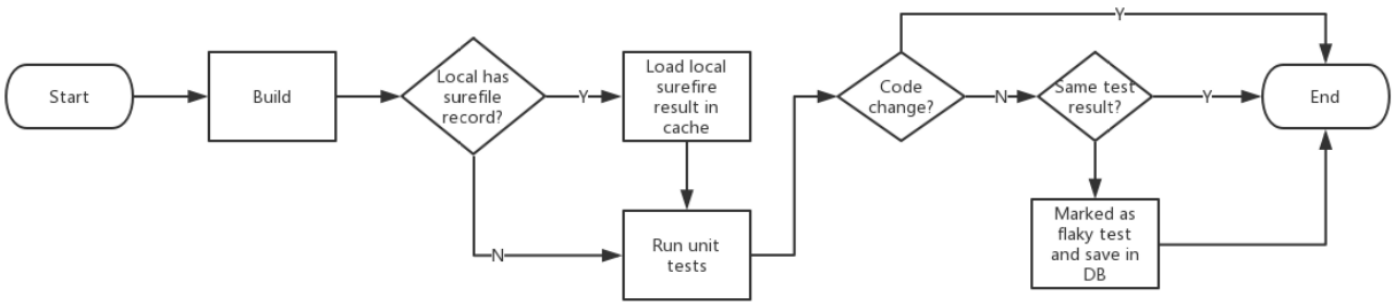


Figure 4 : Work Flow of the Flaky Tests Clue Provider

In implementation, we refer to the implementation of Deflaker [3]. By integrating plug-ins in to the tested project in a simple way, Maven 3 life cycle extension [4] is a good way to achieve this goal. Using Maven 3 life cycle extension to build our own extension, make sure our extension will be triggered in each maven build.

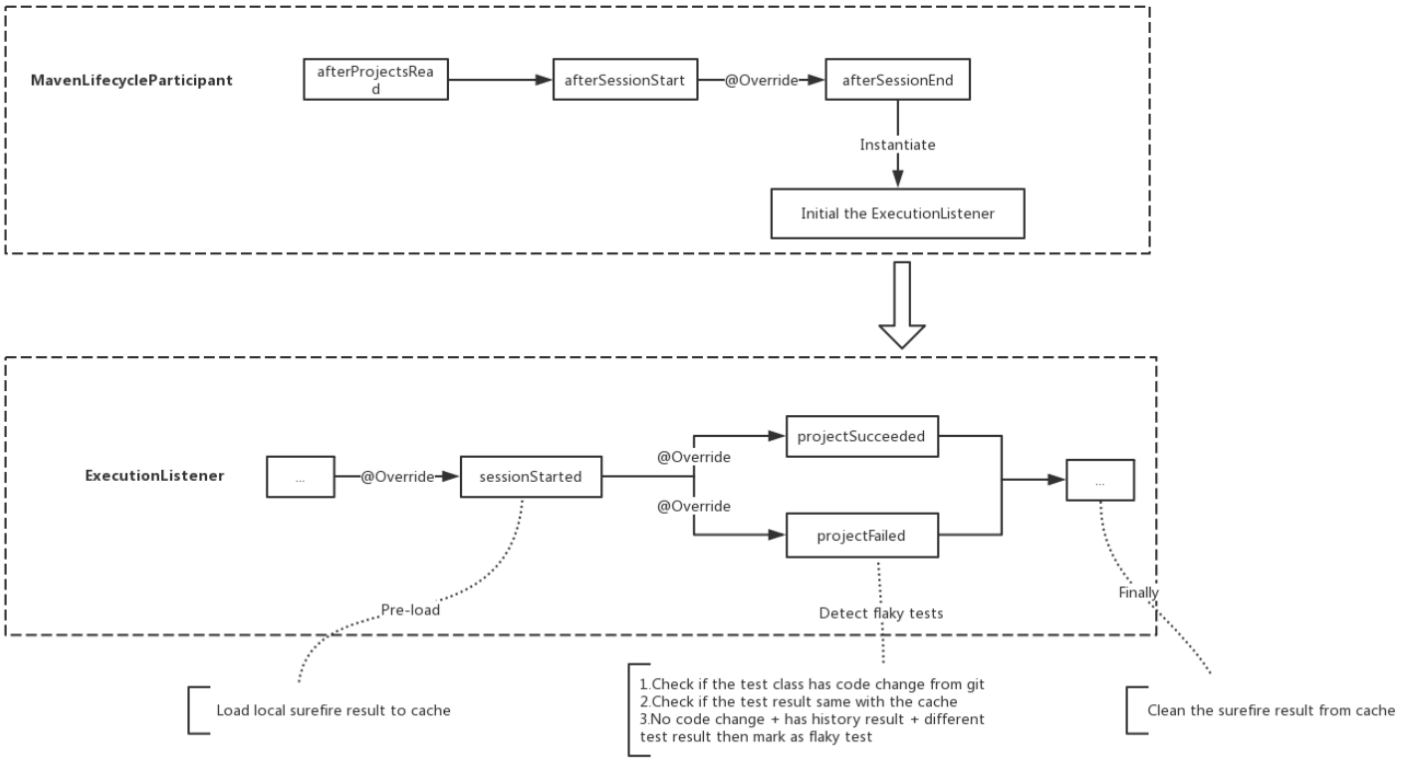


Figure 5 : Implement Details For Detecting The Flaky Test

Implement Details For Detecting The Flaky Test 2

Finally, we put the collected flaky data on the cloud server to manage. The cloud server uses spring boot to build quickly. With the increase of data volume, we can consider introducing ELK (Elastic search, Logstash, Kibana) to improve our storage pressure and access speed. In the future, there will be more reports based on these basic data, so that developers can have better data for decision-making.

REFERENCES

[1] What's a flaky test?https://docs.gitlab.com/ee/development/testing_guide/flaky_tests.html#whats-a-flaky-test

[2] DeFlaker: Automatically Detecting Flaky Tests https://www.jonbell.net/icse18-deflaker.pdf
[3] DeFlaker source code in github. https://github.com/gmu-swe/deflaker
[4] Maven 3 lifecycle extension. http://maven.apache.org/examples/maven-3-lifecycle-extensions.html