

# Improving C# Pipeline Debugging Information And Tool Documentation

Ishaan Sehgal

Mentors: Wing Lam, Liia Butler

## Background

- Software testing is used to evaluate whether software is defect free
- Flaky tests
  - Flaky tests are tests that can both fail and pass on the same version of code
  - Issues with flaky tests [3]:
    - Reduces developer’s trust in results
    - May only be reproducible in specific test scenarios
    - May require significant time and resources to manually debug
- C# Pipeline Overview [1]
  - Detects flaky tests from C# projects and provides debug information for each flaky test found
  - Input
    - Project name and directory
  - Output
    - Name of flaky test & its failing frequency

```
1 [TestMethod()]
2 public void randomGenerator(){
3     Random r = new Random();
4     int rand = r.Next(1,10);
5     Assert.AreEqual(5, rand);
6 }
7
```

Example 1. A random number between 1 and 10 is generated. Theoretically, the test will pass 1/10 runs and fail 9/10. Unpredictable, the test is considered flaky due to randomness. [2]

## References

1. Wing, L. & Liia, B. (2018). iC#Flakies: A Toolset for Detecting and Debugging Flaky Tests in .NET Projects. Unpublished manuscript.
2. Q. Luo, F. Hariri, L. Eloussi, and D. Marinov. An empirical analysis of flaky tests. In Proc. of the 22nd ACM SIGSOFT International Symposium on the Foundations of Software Engineering, pages 643–653, Hong Kong, China, Nov. 2014.
3. W. Lam, R. Oei, A. Shi, D. Marinov, and T. Xie. iDFlakies: A framework for detecting and partially classifying flaky tests. In Proc. of the 12th IEEE International Conference on Software Testing, Verification and Validation, pages to–appear, Xi’an, China, Apr. 2019.

## Debugging Information

- Goal: improve comprehensibility of the pipeline’s output, to help developers debug flaky tests
- Added the solution file path (file executing C# project) corresponding to each test in the project’s output
- Prevented users from generating extraneous files by preventing unnecessary repeated pipeline executions
- Included the total number of times all tests passed and failed across multiple executions of the pipeline

## Tool Documentation

- Goal: document tool setup information and thorough fixes for tool run issues
- Troubleshooting and setup
  - Added tool’s machine compatibility information
  - Added steps for fixing missing tool environment variable error
  - Added steps for fixing script execution policy errors
  - Added steps for including missing dependencies

## Benefits

- New pipeline debugging information allows users to identify the solution file corresponding to each flaky test
- Preventing multiple executions of the same project on the pipeline saves users time and reduces extraneous files generated
- Pipeline users can fix common troubleshooting errors found during pipeline execution using documentation

