Supplemental Material: What we talk about when we talk about software test flakiness

Morena Barboni[1], Antonia Bertolino[2], and Guglielmo De Angelis[1]

[1] IASI-CNR, Rome, Italy
{morena.barboni, guglielmo.deangelis}@iasi.cnr.it
[2] ISTI-CNR, Pisa, Italy
antonia.bertolino@isti.cnr.it

Supplemental Material

This supplemental material details the definitions of the concepts that have been found by conducting the *scoping review* [3, 32] of both the white and grey literature introduced in Section 2 of the manuscript titled:"*What we talk about when we talk about software test flakiness*".

Reference	Title	Definition(s)
[34]	Test Flakiness – Methods for identifying and dealing with flaky tests	A flaky test is a test that both passes and fails periodically without any code changes. ...
[26]	We Have A Flaky Test Problem	A flaky test is a test that both passes and fails periodically without any code changes.[101] ...
[43]	Dealing with flaky tests	A flaky test is a test that's unreliable in behaviour, meaning that it yields different results inconsistently. They are sometimes referred to as "random failures", but in reality, it's often less about actual randomness than very reproducible edge cases that happen in a seemingly random fashion. The majority of the time, a test's flakiness is not due to randomness. If conditions can be reproduced accurately, then the test will always behave the same. Even though they appear to happen randomly, they're usually triggered by a very reproducible set of conditions.
[42]	What is flakiness and how we deal with it	Sometimes you run your tests multiple times in a row with no code change, and even then, the results are different. This instability is called flakiness. A flaky test is a test that can be failing or passing with no changes in the application or infrastructure.

		Two main reasons are standing behind test suite instability - bugs in your application or defects in your testing code. Here are some examples of failures in our app that have led to flakiness in our tests. If flakiness is caused by the instability of something bigger (like our infra or deployment process), we very rarely introduce mechanisms for automatic test reruns In most cases, flakiness is caused by issues in testing code. One of the reasons may be using the application in the wrong way. Flakiness caused by fails in testing code can also be coming from improper usage of the testing tool.
[10]	Flaky Tests (And How To Avoid Them)	A "flaky" test is one that has a non-deterministic outcome: it can pass sometimes and fail others, for the same code, running the same test. Flaky tests (sometimes also called "Flappers") their failure does not necessarily indicate a bug We then grepped through the log for keywords "flak" and "intermit" to catch variations of the words flaky and intermittent. These tests are usually flaky because the developer made an incorrect assumption about the ordering of operations being performed by different threads. The final category of flaky tests we looked at in detail are those that would pass or fail depending on which tests were executed before them.
[44]	Flaky Tests are Not Random Failures	same definitions as in [43]
[6]	Flaky tests caused by a production bug: fix the flakiness, not the bug	flaky tests, i.e. tests which fail randomly Despite the low probability, when run hundreds of times on the CI, flaky tests will cause the CI to fail regularly for no real reason at all.
[27]	A machine learning solution for detecting and mitigating flaky tests	A test which passes or fails in a nondeterministic way is referred to as flaky. There are two main types of flaky tests. Those that are flaky due to some external conditions, such as network issues, machine crashes, power outages etc. ...The second type of flakiness is due to defects in the test case's code or in the CUT (code under test), such as asynchronous waits, concurrency issues such as race conditions, priority inversion or incorrect assumptions about timezones or database ordering. Flaky tests pass and fail on successive git revisions over a long period of time

[16]	Dealing with the flakiness of UI Tests	no defs in this article
[33]	Flaky tests	Part of the test or production code has a non-deterministic outcome. The test is flaky because the code doesn't always return the same result. Flakiness in tests is caused by poor quality of test code or bug in production code. Researchers split the root causes of flakiness into 10 categories. The top three categories of flaky tests are Async Wait, Concurrency, and Test Order Dependency.[102]
[31]	Flaky Tests at Google and How We Mitigate Them	We define a "flaky" test result as a test that exhibits both a passing and a failing result with the same code.
[11]	Eradicating Non-Determinism in Tests	A test is non-deterministic when it passes sometimes and fails sometimes, without any noticeable change in the code, tests, or environment. Test failures for such tests are seemingly random.

[101] Adopted from the Google's Definition at `https://testing.googleblog.com/2016/05/flaky-tests-at-google-and-how-we.html`

[102] The blog posts in referring to [28]

Table 2: Details of Flaky Tests definitions in the White Literature

Paper	Title	Definition	Synonym(s)
[9]	Understanding Flaky Tests: The Developer's Perspective	Flaky tests are software tests that exhibit a seemingly random outcome (pass or fail) despite exercising unchanged code. . . . since flaky tests fail intermittently, their priority is often lower than those of permanent failures	Intermittently Failing Tests Non-Deterministic Tests -
[37]	The Impact of Failing, Flaky, and High Failure Tests on the Number of Crash Reports Associated with Firefox Builds	Flaky tests fail non-deterministically. For example, a test may both pass and fail on the same build.	Non-Deterministic Tests
[28]	An empirical analysis of flaky tests	Test outcomes are not reliable for tests that can intermittently pass or fail even for the same code version. Following practitioners, we call such tests flaky: their outcome is non-deterministic with respect to a given software version.	Intermittently Failing Tests Non-Deterministic Tests
[41]	Shake It! Detecting Flaky Tests Caused by Concurrency with Shaker	A test is said to be flaky when it non-deterministically passes or fails depending on the running environment	-
[47]	An Empirical Study of Flaky Tests in Android Apps	Flaky tests are the tests that terminate with nondeterministic outcomes given the same CUT (code under test)	-
[25]	A large-scale longitudinal study of flaky tests	A test that can both pass and fail in repeated runs, on the same SUT (even without new changes), is known as a flaky test.	-

Continued on next page

Table 2 – continued from previous page

Paper	Title	Definition	Synonym(s)
[20]	Root causing flaky tests in a large-scale industrial setting	are tests that that may pass and fail with the same version of source code and the same configuration.	Non-Deterministic Tests
[45]	Intermittently failing tests in the embedded systems domain	Flaky tests are tests that yield differing verdicts when nothing in the SW, HW or TW (TestWare) have been changed.	-
[30]	Automated Analysis of Flakiness-mitigating Delays	Such tests are commonly called flaky and can be described as a test that when applied to a system S yields different outcomes on different occasions in apparently identical test scenarios	-
[13]	Practical Automatic Lightweight Nondeterminism and Flaky Test Detection and Debugging for Python	. . . regression tests that do not behave reliably (known as flaky tests). Flaky tests are regression tests that fail in an intermittent, unreliable fashion. The essence of a flaky test is that, for the same snapshot of test code and code under test, it sometimes fails and sometimes passes.	-
[49]	Test Analysis: Searching for Faults in Tests	. . . this type of dependency has been identified as one of the main sources of flaky tests. . . . this type of pattern can lead to tests that fail intermittently.	Intermittently Failing Tests
[22]	IDFlakies: A framework for detecting and partially classifying flaky tests	Previous work defines flaky tests as tests that may non-deterministically pass or fail even on the same version of the code under test.	Non-Deterministic Tests

Continued on next page

Table 2 – continued from previous page

Paper	Title	Definition	Synonym(s)
[39]	Detecting Assumptions on Deterministic Implementations of Non-deterministic Specifications	Unexpected behavior of ADINS code can lead to flaky tests, which are tests that seem to non-deterministically pass or fail.	-
[40]	IFixFlakies: A framework for automatically fixing order-dependent flaky tests	Flaky tests can pass or fail even when run on the same code, without any changes.	-
[1]	Empirical analysis of factors and their effect on test flakiness – practitioners perceptions	Developers submit code changes and expect possible test failures to be connected with the submitted change. Unfortunately, some test failures are not due to the submitted changes but flaky tests. In addition to this, tests failing without any change in the code base (e.g., regression tests executing on the same build) are also called flaky tests.	-
[4]	DeFlaker: Automatically Detecting Flaky Tests	As in previous work, we define a flaky test as a test that can non-deterministically pass or fail when run on the same version of the code. ... recall that a test is flaky if it both passes and fails when the code that is executed by the test did not change;	Non-Deterministic Tests
[21]	A study on the lifecycle of flaky tests	Flaky Tests are tests that pass and fail non-deterministically on the same code.	Non-Deterministic Tests
[17]	Towards a Bayesian Network Model for Predicting Flaky Automated Tests	Flaky tests exhibit both passing and failing results although neither the code nor test has changed.	Non-Deterministic Tests

Continued on next page

Table 2 – continued from previous page

Paper	Title	Definition	Synonym(s)
[2]	FlakeFlagger: Predicting Flakiness Without Rerunning Tests	Flaky Tests are non-deterministic tests which pass and fail when run on the exact same version of a codebase	Non-Deterministic Tests
[23]	Dependent-test-aware regression testing techniques	Flaky tests are tests that can both pass and fail when run multiple times on the same version of code and tests.	-
[8]	Detecting Flaky Tests in Probabilistic and Machine Learning Applications	flaky tests – tests which fail non-deterministically when run on the same version of code	Non-Deterministic Tests
[36]	Wait Wait. No, Tell Me. Analyzing Selenium Configuration Effects on Test Flakiness.	A common issue is that Selenium tests, like other automated tests with a broad scope, are often non-deterministic (flaky).	Non-Deterministic Tests
[46]	A Container-Based Infrastructure for Fuzzy-Driven Root Causing of Flaky Tests	This kind of tests are called "flaky" (non-deterministic), that is, a test that passes or fails intermittently for the same code version, the same inputs, and the same configuration.	Non-Deterministic Tests
[51]	De-Flake Your Tests - Automatically Locating Root Causes of Flaky Tests in Code At Google	If the test suite is executed without any changes with the same configuration parameters, they should either always pass or always fail. Unfortunately, there might be non-deterministic,so called flaky.	Non-Deterministic Tests
[38]	Mitigating the Effects of Flaky Tests on Mutation Testing	...flaky tests, which can exhibit different behaviors (e.g., passing or failing) even with no changes to the code under test.	-

Continued on next page

Table 2 – continued from previous page

Paper	Title	Definition	Synonym(s)
[29]	Predictive Test Selection	Flakiness is the phenomenon whereby the same test produces different outcomes upon multiple independent trials. ... the non-determinism of test outcomes, also known as test flakiness.	Non-Deterministic Tests
[19]	Modeling and Ranking Flaky Tests at Apple	A flaky test is one that may fail or pass non-deterministically.	-
[24]	Understanding Reproducibility and Characteristics of Flaky Tests Through Test Reruns in Java Projects	Flaky tests are tests that can non-deterministically pass and fail in different test runs, even for the same code under test and the same test environment that the developers can easily control.	-
[52]	Root causing, detecting, and fixing flaky tests: State of the art and future roadmap	A flaky test is a test that exhibits both passing and failing results even though there is no code change in CUT (code under test) or test code whose outcome is non deterministic.	-
[35]	Flake It 'Till You Make It: Using Automated Repair to Induce and Fix Latent Test Flakiness	Flaky tests are software tests that appear to exhibit an element of randomness in their outcome despite covering code that has not changed.	-
[12]	Practical Test Dependency Detection	Rerunning tests on the same code should not cause the outcome of any test to change. However, in practice this is not always the case, and tests may be flaky, passing and failing non-deterministically.	

Continued on next page

Table 2 – continued from previous page

Paper	Title	Definition	Synonym(s)
[48]	An Empirical Study of Bugs in Test Code	Flaky Tests: These test bugs are caused by non-deterministic behaviour of test cases, which intermittently pass or fail.	

Table 3: Details of other definitions in the White Literature

Paper	Title	Definition	Synonym(s)
Latent Flaky Test			
[35]	Flake It 'Till You Make It: Using Automated Repair to Induce and Fix Latent Test Flakiness	We refer to tests that are not currently flaky, but that could become so, as having latent flakiness. There two most critical sources of latent flakiness are test order dependencies and test resource leaks.	-
Non-Flaky Test			
[9]	Understanding Reproducibility and Characteristics of Flaky Tests Through Test Reruns in Java Projects	Tests that are not flaky either always pass (all orders have 0% failure rate) or always fail (all orders have 100% failure rate)	-
Non-Hermetic Test			
[19]	Modeling and Ranking Flaky Tests at Apple	Quantifying flakiness is useful where all tests have some degree of flakiness, a situation not uncommon in practice for non-hermetic tests (i.e., tests not run in pure isolation), such as system tests.	-
ND (Non-Deterministic) Test			
[25]	A large-scale longitudinal study of flaky tests	tests that non-deterministically pass or fail with no changes to test execution order or implementation of test dependencies	-
ID (Implementation-Dependent) Test			

Continued on next page

Table 3 – continued from previous page

Paper	Title	Definition	Synonym(s)
[25]	A large-scale longitudinal study of flaky tests	Other flaky tests may be implementation-dependent, where a test is flaky due to an assumption that an API is deterministic, when that API is not (e.g., the order of iteration over a HashSet) ... flaky tests whose test result depends on the implementation of a non-deterministic specification;	-
Smelly Test			
[45]	Intermittently failing tests in the embedded systems domain	Tests can also be flaky because of poor design. These are sometimes called smelly tests.	-
[1]	Empirical analysis of factors and their effect on test flakiness – practitioners perceptions	Smells refer to any characteristic in the programming code that possibly indicate a problem. Code smells refer to smells in source code or system under test whereas test smells refer to smells in the test case code. The test smell is one of the factors that can affect test flakiness.	-
[2]	FlakeFlagger: Predicting Flakiness Without Rerunning Tests	A recent survey has found 23 factors that increase, decrease and otherwise affect the ability to identify flakiness in tests. These factors include features such as the presence of test smells.	-
Intermittently Failing Test			

Continued on next page

Table 3 – continued from previous page

Paper	Title	Definition	Synonym(s)
[45]	Intermittently failing tests in the embedded systems domain	We define an intermittently failing test to be a test case that has been executed repeatedly while there is a potential evolution in SW, HW or TW, and where the verdict changes over time. They are different from flaky tests in that they allow changes in the SW or HW of the ES under test, as well as in the TW used for testing.	-
		Consistently Failing Test	
[45]	Intermittently failing tests in the embedded systems domain	are tests that consistently cause failures.	-

Table 4: Additional definitions related to flaky test behaviour.

Paper	Title	Definition	Synonym(s)
Test Flakiness			
[46]	A Container-Based Infrastructure for Fuzzy-Driven Root Causing of Flaky Tests	Intermittent test failures	-
[19]	Modeling and ranking flaky tests at apple	Inability to reliably repeat a test's Pass/Fail outcome	-
[29]	Predictive Test Selection	Flakiness is the phenomenon whereby the same test produces different outcomes upon multiple independent trials. ... the non-determinism of test outcomes, also known as test flakiness.	-
[17]	Towards a Bayesian Network Model for Predicting Flaky Automated Tests	Maintaining automated test scripts at scale can be costly, especially if they become slow and unstable – a problem referred to as test flakiness [8], [17], [25]	-
[1]	Empirical analysis of factors and their effect on test flakiness – practitioners perceptions	Different participants provided different perception for what flakiness is and whether we should call it test flakiness, source code flakiness or environment flakiness.	-
False Alarm			
[15]	Empirically Detecting False Test Alarms Using Association Rules	A false test alarm is a test failure that is due to any other reason than a code defect. In most cases, such false alarms are caused by test and infrastructure issues.	
[48]	An Empirical Study of Bugs in Test Code	The majority of test bugs are false alarms, i.e., test fails while the production code is correct.	
Silent Horror			

Continued on next page

Table 4 – continued from previous page

Paper	Title	Definition	Synonym(s)
[48]	An Empirical Study of Bugs in Test Code	. . . a minority of these bugs result in silent horrors, i.e., test passes while the production code is incorrect	
Intermittent Test Failures			
[46]	A Container-Based Infrastructure for Fuzzy-Driven Root Causing of Flaky Tests	Intermittent test failures (test flakiness) is common during continuous integration as modern software systems have become inherently non-deterministic.	Test Flakiness

Table 5: Order dependent test definitions in the White Literature

Paper	Title	Definition	Synonym(s)
OD (Order-Dependent) Test			
[9]	Understanding Flaky Tests: The Developer's Perspective	Test Order Dependency: This class is characterized by the result of the test run depending on the execution order of the tests.	
[28]	An empirical analysis of flaky tests	Test Order Dependency: We classify a commit into this category when the test outcome depends on the order in which the tests are run.	
[22]	IDFlakies: A framework for detecting and partially classifying flaky tests	Following prior work, we refer to flaky tests whose only source of non-determinism is order dependencies as order-dependent (OD) tests. OD tests can deterministically pass or fail depending on the order in which the tests are run.	
[40]	IFixFlakies: A framework for automatically fixing order-dependent flaky tests	A common kind of flaky tests are order-dependent tests, which pass or fail depending on the order in which the tests are run. We classify an order-dependent test into one of two types: victim or brittle.	

Continued on next page

Table 5 – continued from previous page

Paper	Title	Definition	Synonym(s)
[24]	Understanding Reproducibility and Characteristics of Flaky Tests Through Test Reruns in Java Projects	Order-dependent (OD) tests can deterministically pass or fail based on the order in which the tests are run. OD tests deterministically fail for some order of tests in a test suite but deterministically pass for some other orders. Such tests are deterministic in that their failure rates are either 0% or 100% for each order, and they have at least two orders whose failure rates differ.	-
[2]	FlakeFlagger:Predicting Flakiness Without Rerunning Tests	Other flaky tests may be order dependent, which means that when they run in a different order than is expected, they can fail (be flaky)	
[50]	Empirically Revisiting the Test Independence Assumption	We call A an order-dependent test, since its result depends on whether it runs after B or not. Manifest test dependence requires a concrete order of the test suite that produces different results than expected.	Dependent Tests
[23]	Dependent-test-aware regression testing techniques	One prominent type of flaky tests is order-dependent (OD) tests. An OD test is a test that passes or fails depending only on the order in which the test is run.	

Continued on next page

Table 5 – continued from previous page

Paper	Title	Definition	Synonym(s)
[14]	Reliable Testing: Detecting State-Polluting Tests to Prevent Test Dependency	. . . even for the same version of the code under test, the tests could pass when executed in one order but fail when executed in another order.	Dependent Tests
[38]	Mitigating the Effects of Flaky Tests on Mutation Testing	When multiple tests share resources, they may be subject to flakiness due to test-order dependencies: the behavior of a test might change based on which tests had run previously.	
[52]	Root causing, detecting, and fixing flaky tests: State of the art and future roadmap	Order-dependent tests produce flaky tests due to the order of sequence in which the tests run. If there were test A and test B, changing the order of the tests on the CUT may result in different outcomes.	-
[25]	A large-scale longitudinal study of flaky tests	flaky tests whose test result depends on the order the tests are run.	-
OD Vic (Order-Dependent Victim)			
[22]	iFixFlakies: A Framework for Automatically Fixing Order-Dependent Flaky Tests	A victim is an order-dependent test that consistently passes when run by itself in isolation from other tests (but fails when run with some other tests).	-
[25]	A large-scale longitudinal study of flaky tests	Order-dependent victim (OD Vic) are tests that pass when run in isolation but fail when run after some specific tests;	
OD Brit (Order-Dependent Brittle)			

Continued on next page

Table 5 – continued from previous page

Paper	Title	Definition	Synonym(s)
[22]	iFixFlakies: A Framework for Automatically Fixing Order-Dependent Flaky Tests	A brittle is an order-dependent test that consistently fails when run by itself in isolation (but passes when run with some other test(s))	-
[25]	A large-scale longitudinal study of flaky tests	Order-dependent brittle (OD Brit) are tests that fail when run in isolation but pass when run after some specific tests;	

Table 6: Related concepts to order dependent tests in the White Literature

Paper	Title	Definition	Synonym(s)
Helper			
[22]	iFixFlakies: A Framework for Automatically Fixing Order-Dependent Flaky Tests	Helpers are tests whose logic (re)sets the state required for order-dependent tests to pass. Both cleaners (for victims) and state-setters (for brittles) help make order-dependent tests pass when they run in certain test orders. Hence, we refer to cleaners and state-setters as helpers.	-
Polluter			
[22]	iFixFlakies: A Framework for Automatically Fixing Order-Dependent Flaky Tests	These tests pollute the state (e.g., global variable, file system, network) on which the victim depends. A polluter can consist of multiple tests, where the combination of running those tests in a certain order leads to the victim failing.	-
[14]	Reliable Testing: Detecting State-Polluting Tests to Prevent Test Dependency	Polluters are tests that pollute the shared state. These are tests that modify some location on the heap shared across tests or on the file system; a subsequent test could fail if it assumes the shared location to have the initial value before the state was modified.	State-Polluting Tests
[12]	Practical Test Dependency Detection	In this paper, we consider the problem caused by state polluting tests: tests that leave the environment in a different state than they found it in	State-Polluting Tests
Cleaner			

Continued on next page

Table 6 – continued from previous page

Paper	Title	Definition	Synonym(s)
[22]	iFixFlakies: A Framework for Automatically Fixing Order-Dependent Flaky Tests	A cleaner is a test order that resets the state polluted by a polluter;	-
State-Setter			
[22]	iFixFlakies: A Framework for Automatically Fixing Order-Dependent Flaky Tests	A state-setter is a test order that sets up the state for a brittle.	-

Table 7: Non-order dependent test definitions in the White Literature

Paper	Title	Definition	Synonym(s)
NOD (Non-Order-Dependent) Test			
[22]	IDFlakies: A framework for detecting and partially classifying flaky tests	We refer to all other types of flaky tests, which are not OD tests, as non-order-dependent (NOD) tests.	-
[2]	FlakeFlagger: Predicting Flakiness Without Rerunning Tests	Tests that are flaky regardless of execution order.	-
[24]	Understanding Reproducibility and Characteristics of Flaky Tests Through Test Reruns in Java Projects	Non-order-dependent (NOD) tests are flaky but not OD. NOD tests can non-deterministically pass and fail even for the same order of tests. Such tests have at least one order where the test fails non-deterministically (failure rate is neither 0% nor 100%).	-
NDOD (Non-Deterministic Order-Dependent) Test			
[24]	Understanding Reproducibility and Characteristics of Flaky Tests Through Test Reruns in Java Projects	NDOD tests are NOD tests where at least one order's failure rate significantly differs from other orders' failure rates. e.g., a test that has a 99% failure rate in one order but 0% in another.	-
NDOI (Non-Deterministic Order-Independent) Test			
[24]	Understanding Reproducibility and Characteristics of Flaky Tests Through Test Reruns in Java Projects	NDOI tests are NOD tests where all failure rates do not significantly differ.	-
ND (Non-Deterministic) Test			

Continued on next page

Table 7 – continued from previous page

Paper	Title	Definition	Synonym(s)
[25]	A large-scale longitudinal study of flaky tests	tests that non-deterministically pass or fail with no changes to test execution order or implementation of test dependencies	-

References

1. Ahmad, A., Leifler, O., Sandahl, K.: Empirical analysis of factors and their effect on test flakiness-practitioners' perceptions. arXiv preprint arXiv:1906.00673 (2019)
2. Alshammari, A., Morris, C., Hilton, M., Bell, J.: FlakeFlagger: Predicting flakiness without rerunning tests. In: Proc. ICSE Art. Ev. track. IEEE (2021)
3. Arksey, H., O'Malley, L.: Scoping studies: towards a methodological framework. International Journal of Social Research Methodology **8**(1), 19–32 (2005)
4. Bell, J., Legunsen, O., Hilton, M., Eloussi, L., Yung, T., Marinov, D.: DeFlaker: Automatically detecting flaky tests. In: Proc. ICSE. pp. 433–444. ACM (2018)
5. Carver, R.H., Tai, K.C.: Replay and testing for concurrent programs. IEEE Software **8**(2), 66–74 (1991)
6. Champier, C.: Flaky tests caused by a production bug: fix the flakiness, not the bug. Online on `medium.com` (Feb 2019)
7. Cotroneo, D., Grottke, M., Natella, R., Pietrantuono, R., Trivedi, K.S.: Fault triggers in open-source software: An experience report. In: Proc. ISSRE. pp. 178–187. IEEE (2013)
8. Dutta, S., Shi, A., Choudhary, R., Zhang, Z., Jain, A., Misailovic, S.: Detecting flaky tests in probabilistic and machine learning applications. In: Proc. ISSTA. pp. 211–224. ACM (2020)
9. Eck, M., Palomba, F., Castelluccio, M., Bacchelli, A.: Understanding flaky tests: The developer's perspective. In: Proc. ESEC/FSE. pp. 830–840. ACM (2019)
10. Eloussi, L.: Flaky tests (and how to avoid them). Online on `medium.com` (Sep 2016)
11. Fowler, M.: Eradicating non-determinism in tests (Apr 2011)
12. Gambi, A., Bell, J., Zeller, A.: Practical test dependency detection. In: Proc. ICST. pp. 1–11. IEEE (2018)
13. Groce, A., Holmes, J.: Practical automatic lightweight nondeterminism and flaky test detection and debugging for Python. In: Proc. QRS. pp. 188–195. IEEE (2020)
14. Gyori, A., Shi, A., Hariri, F., Marinov, D.: Reliable testing: Detecting state-polluting tests to prevent test dependency. In: Proc. ISSTA. pp. 223–233. ACM (2015)
15. Herzig, K., Nagappan, N.: Empirically detecting false test alarms using association rules. In: Proc. ICSE. pp. 39–48. IEEE (2015)
16. Jacob, J.: Dealing with the flakiness of UI tests. Online on `medium.com` (Mar 2020)
17. King, T.M., Santiago, D., Phillips, J., Clarke, P.J.: Towards a bayesian network model for predicting flaky automated tests. In: Proc. QRS-C. pp. 100–107. IEEE (2018)
18. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004), 1–26 (2004)

19. Kowalczyk, E., Nair, K., Gao, Z., Silberstein, L., Long, T., Memon, A.: Modeling and ranking flaky tests at Apple. In: Proc. ICSE-SEIP. pp. 110–119. ACM (2020)
20. Lam, W., Godefroid, P., Nath, S., Santhiar, A., Thummalapenta, S.: Root causing flaky tests in a large-scale industrial setting. In: Proc. ISSTA. pp. 101–111. ACM (2019)
21. Lam, W., Muşlu, K., Sajnani, H., Thummalapenta, S.: A study on the lifecycle of flaky tests. In: Proc. ICSE. pp. 1471–1482. ACM (2020)
22. Lam, W., Oei, R., Shi, A., Marinov, D., Xie, T.: iDFlakies: A framework for detecting and partially classifying flaky tests. In: Proc. ICST. pp. 312–322. IEEE (2019)
23. Lam, W., Shi, A., Oei, R., Zhang, S., Ernst, M.D., Xie, T.: Dependent-test-aware regression testing techniques. In: Proc. ISSTA. pp. 298–311. ACM (2020)
24. Lam, W., Winter, S., Astorga, A., Stodden, V., Marinov, D.: Understanding reproducibility and characteristics of flaky tests through test reruns in Java projects. In: Proc. ISSRE. pp. 403–413. IEEE (2020)
25. Lam, W., Winter, S., Wei, A., Xie, T., Marinov, D., Bell, J.: A large-scale longitudinal study of flaky tests. Proc. ACM on Programming Languages **4**(OOPSLA), 1–29 (2020)
26. Lee, B.: We have a flaky test problem. Online on `medium.com` (Nov 2019)
27. Liviu, S.: A machine learning solution for detecting and mitigating flaky tests. Online on `medium.com` (Oct 2019)
28. Luo, Q., Hariri, F., Eloussi, L., Marinov, D.: An empirical analysis of flaky tests. In: Proc. FSE. pp. 643–653. ACM (2014)
29. Machalica, M., Samylkin, A., Porth, M., Chandra, S.: Predictive test selection. In: Proc. ICSE-SEIP. pp. 91–100. IEEE (2019)
30. Malm, J., Causevic, A., Lisper, B., Eldh, S.: Automated analysis of flakiness-mitigating delays. In: Proc. AST. pp. 81–84. IEEE (2020)
31. Micco, J.: Flaky tests at Google and how we mitigate them (May 2016)
32. Munn, Z., Peters, M.D., Stern, C., Tufanaru, C., McArthur, A., Aromataris, E.: Systematic review or scoping review? guidance for authors when choosing between a systematic or scoping review approach. BMC medical research methodology **18**(1), 1–7 (2018)
33. Otrebski, K.: Flaky tests. Online on `medium.com` (Apr 2018)
34. Palmer, J.: Test flakiness – methods for identifying and dealing with flaky tests. Online on `medium.com` (Nov 2019)
35. Parry, O., Kapfhammer, G.M., Hilton, M., McMinn, P.: Flake it'till you make it: Using automated repair to induce and fix latent test flakiness. In: Proc. ICSE Workshops. pp. 11–12. ACM (2020)
36. Presler-Marshall, K., Horton, E., Heckman, S., Stolee, K.: Wait, wait. no, tell me. analyzing selenium configuration effects on test flakiness. In: Proc. Wksp AST. pp. 7–13. IEEE (2019)
37. Rahman, M.T., Rigby, P.C.: The impact of failing, flaky, and high failure tests on the number of crash reports associated with Firefox builds. In: Proc. ESEC/FSE. pp. 857–862. ACM (2018)
38. Shi, A., Bell, J., Marinov, D.: Mitigating the effects of flaky tests on mutation testing. In: Proc. ISSTA. pp. 112–122. ACM (2019)
39. Shi, A., Gyori, A., Legunsen, O., Marinov, D.: Detecting assumptions on deterministic implementations of non-deterministic specifications. In: Proc. ICST. pp. 80–90. IEEE (2016)

40. Shi, A., Lam, W., Oei, R., Xie, T., Marinov, D.: iFixFlakies: A framework for automatically fixing order-dependent flaky tests. In: Proc. ESEC/FSE. pp. 545–555. ACM (2019)
41. Silva, D., Teixeira, L., d'Amorim, M.: Shake it! detecting flaky tests caused by concurrency with Shaker. In: Proc. ICSME. pp. 301–311. IEEE (2020)
42. Słapiński, M.: What is flakiness and how we deal with it. Online on `medium.com` (Feb 2020)
43. Stosik, D.: Dealing with flaky tests. Online on `medium.com` (Nov 2019)
44. Stosik, D.: Flaky tests are not random failures. Online on `medium.com` (Nov 2019)
45. Strandberg, P.E., Ostrand, T.J., Weyuker, E.J., Afzal, W., Sundmark, D.: Intermittently failing tests in the embedded systems domain. In: Proc. ISSTA. pp. 337–348. ACM (2020)
46. Terragni, V., Salza, P., Ferrucci, F.: A container-based infrastructure for fuzzy-driven root causing of flaky tests. In: Proc. ICSE-NIER. pp. 69–72. IEEE (2020)
47. Thorve, S., Sreshtha, C., Meng, N.: An empirical study of flaky tests in android apps. In: Proc. ICSME. pp. 534–538. IEEE (2018)
48. Vahabzadeh, A., Fard, A.M., Mesbah, A.: An empirical study of bugs in test code. In: Proc. ICSME. pp. 101–110. IEEE (2015)
49. Waterloo, M., Person, S., Elbaum, S.: Test analysis: Searching for faults in tests (n). In: Proc. ASE. IEEE (Nov 2015)
50. Zhang, S., Jalali, D., Wuttke, J., Muşlu, K., Lam, W., Ernst, M.D., Notkin, D.: Empirically revisiting the test independence assumption. In: Proc. ISSTA. pp. 385–396. ACM (2014)
51. Ziftci, C., Cavalcanti, D.: De-Flake your tests: Automatically locating root causes of flaky tests in code at Google. In: Proc. ICSME. pp. 736–745. IEEE (2020)
52. Zolfaghari, B., Parizi, R.M., Srivastava, G., Hailemariam, Y.: Root causing, detecting, and fixing flaky tests: State of the art and future roadmap. Software: Practice and Experience (2020)