

Flaky Test Detection

- [Introduction](#)
- [Flaky Score Instance Level Settings](#)
- [Flaky Score Project Level Settings](#)
- [Display of Flaky Score](#)
- [Filter Test Cases on Flaky Score](#)
- [View Flaky Score on Test Case List View](#)
- [View Flaky Score on Test Case Detail Page](#)
- [View Flaky Score on Test Case Link Screens](#)
- [View Flaky Score on Test Execution Screen](#)
- [How to Reduce Flakiness?](#)

Introduction

QTM has introduced an AI-powered, innovative approach to assess the flakiness of test cases by calculating a "Flaky Score" derived from their execution history; this insightful feature empowers testers with the ability to identify test cases whose future execution status is non-deterministic. A flaky test case refers to one that exhibits non-deterministic behavior when executed repeatedly within the same code and environment, resulting in intermittent successes and failures. The crucial first step towards gaining control over flaky tests is identifying them. Flaky tests have the potential to slow down testing pipelines and erode confidence in testing processes. Today determining test case flakiness requires a manual comparison of test results from multiple runs, which is time-consuming. However, with QMetry Intelligence, this process is now automated, saving valuable time and effort for testers and developers.

QA Managers can define and configure settings according to their specific testing processes for calculating the flaky score, ensuring its relevance to their testing methodologies.

For Example,

The following table shows the execution results of the test cases executed multiple times.

Test Case Name	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Flaky or Non-flaky?
Test Case A	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Non-flaky
Test Case B	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Non-flaky
Test Case C	Pass	Pass	Fail	Fail	Pass	Fail	Pass	Fail	Flaky

Use Cases:

- QA Managers and Testers can assess the risk probability linked to the test cases.
- The flaky score on the execution screen shows the tester the probability of risk while executing the test case.

i Flaky Score is calculated only for the executed test cases and is determined based on the latest X number of executions. The Flaky Score ranges between 0 (Not Flaky) and 1 (Flaky).

Notes:

- The final execution status assigned to the test case will only be considered.
- The Flaky Score is calculated only for the test executions of the same project.

Flaky Score Instance Level Settings

These are the settings for the QTM instance. To activate the QI feature for QMetry users, the QTM super administrator needs to enable it for the QMetry instance from Customization > General Settings & Audit > QI Configuration.

→ **Required Permission:** Only the QMetry super admin can access and configure QMetry Intelligence Configuration under Customization > General Settings & Audit > QI Configuration.

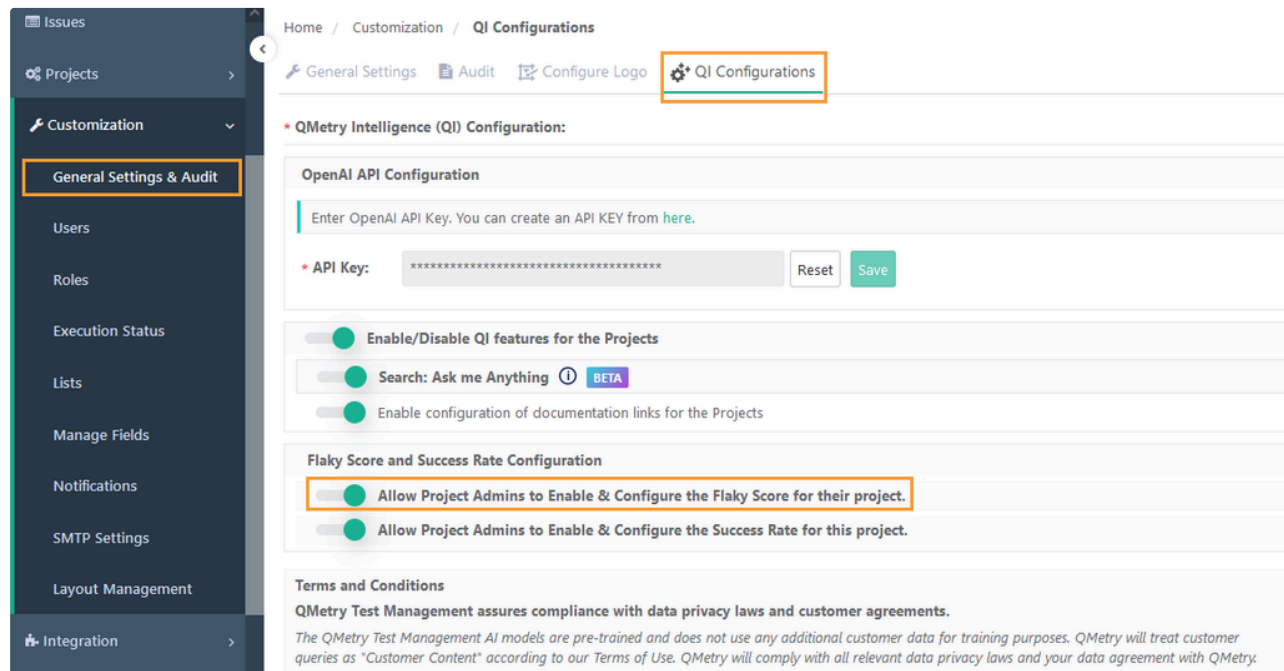
QI Configuration > Flaky Score and Success Rate Configuration (Default: Disabled)

There are two settings for Flaky Score and Success Rate each.

(A) Allow Project Admins to Enable & Configure the Flaky Score for their project.

(B) Allow Project Admins to Enable & Configure the Success Rate for this project.

Enable the “Allow Project Admins to Enable & Configure the Flaky Score for their project” settings for Flaky Score calculations.



Flaky Score Project Level Settings

Users need to do these settings for individual projects in the QTM instance.

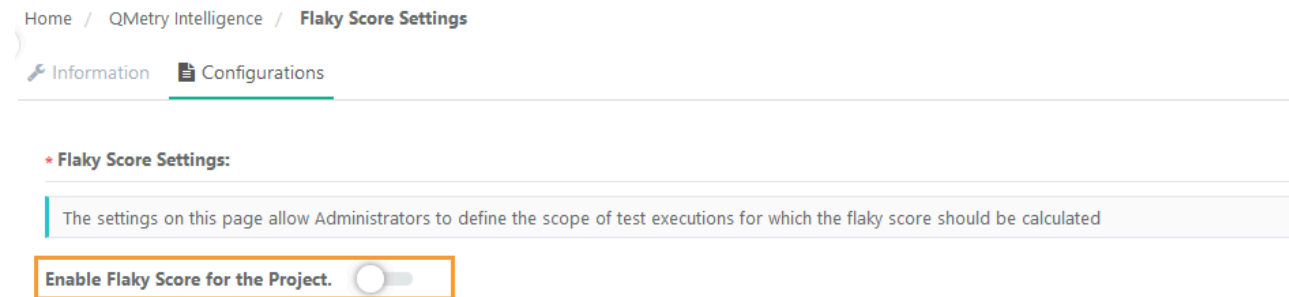
→ **Prerequisite:** Project Administrators can configure the Flaky Score settings if the option “Allow Project Admins to Enable & Configure the Flaky Score for their project.” is enabled in Customization > General Settings & Audit > QI Configuration.

Required Permissions:

- Project Admins and users with the **Flaky Score** “Modify” and “Generate” permissions can configure the settings for Flaky Score. Users with only “View” rights can view the configuration.

Steps

1. Go to QMetry Intelligence.
2. Select the **Flaky Score Settings** option.
3. Open the **Configurations** tab.
4. Enable Flaky Score for the Project.



Once the Flaky Score is enabled, the fields for Flaky Score Configuration appear editable.

Flaky Score Configuration:

The settings on this page allow Administrators to define the scope of test executions for which the flaky score will be calculated. The criteria should be refined to calculate the Flaky Score based on the QA reporting methodologies.

QI will calculate the Flaky Score based on the filtered test execution records according to the criteria. The existing Flaky Score will be reset and recalculated whenever the configuration is modified.

Provide the following details for the Flaky Score rate configuration:

- Test Case Executions: Mention the number of the latest test executions for a test case you want to consider while calculating the flaky score. 10 test executions are selected by default. The Test Case Execution value should be from 1 to 1000.
- Test Case Versions: Select the executions for either the "Latest Version" or "All Version" test case versions to consider while calculating the flaky score. The "Latest Version" test case version is selected by default.
 - Latest Version: The test executions of only the latest test case versions will be considered while calculating the flaky score.
 - All Version: The latest test executions of all test case versions will be considered while calculating the flaky score.
- Execution Timeframe (days): Mention the days to define the timeframe in which the test executions were executed in the last specified days. 90 days are selected by default, which indicates the test executions carried out during the previous 90 days will be considered while calculating the flaky score. If left blank, it will be considered as all the days.
- Release: Select Release(s) to consider test executions for those releases. "All" releases are selected by default.
- Cycle: Select Cycle(s) to consider test executions for those cycles. "All" cycles are selected by default.
- Platform: Select Platform(s) to consider test executions for those platforms. "All" platforms are selected by default.
- Build: Select build(s) to consider the test executions pertaining to that particular build(s). "All" builds are selected by default.
- Test Case Execution Status (*All execution statuses other than selected as passed and failed will be considered as Skipped. Skipped status will not be considered for calculation*).
 - As Passed: The test execution with these statuses will be considered as "Passed" while calculating the Flaky Score. The default value is "Passed".
 - As Failed: The test executions with these statuses will be considered as "Failed" while calculating the Flaky Score. The default values are "Failed" and "Blocked".

Click **Update** to save the Flaky Score Settings.

To reset the settings, click on the **Reset** button.

To generate the flaky score, click on the **Generate** button.

The success message pops up.

Note: The Flaky Score can be calculated once in 24 hours for each project. Once you generate the Flaky Score, the **Generate** button will remain disabled for the next 24 hours.

When you hover over the **Generate** button, it shows the details of the user who generated Flaky Score last with the timestamp.

Information Configurations

Flaky Score Settings:

The settings on this page allow Administrators to define the scope of test executions for which the flaky score should be calculated

Enable Flaky Score for the Project. ☒

Test Case Executions ⓘ
20

Test Case Version ⓘ
☐ Latest Version ☒ All Version

Execution Time Frame (in Days) ⓘ
90

Release ⓘ
Macro, Pro

Cycle ⓘ
MacroQ1 (Macro), ...

Platform ⓘ
Google Chrome, Fir...

Build ⓘ
Build 4.5.4, Build 4.5.5

Test Case Execution Status ⓘ (All execution statuses other than selected as passed and failed will be considered as Skipped. Skipped status will not be considered for calculation)

As Passed ⓘ
Passed

As Failed ⓘ
Failed, Blocked

Tip: QI will calculate the Flaky Score based on the filtered test execution records according to the above criteria. The existing Flaky Score will be recalculated when the configuration is modified.

Last Sync : COMPLETED Reset Update Generate

Flaky Score last generated by Liza Mathew at 05-29-2024 09:36:22
It can be generated again after 05-30-2024 09:36:22

Once you generate the Flaky Score on the Flaky Score Settings screen, you can view the Flaky Score on the test case list view, test case details screen, test case link screens, and test execution screens. You can show or hide the Flaky Score column on the test case list view, test case details screen, test case link screen, and test execution screens.

Display of Flaky Score

The lesser the value (closer to 0), the test case has less flakiness. It means the behavior is deterministic.

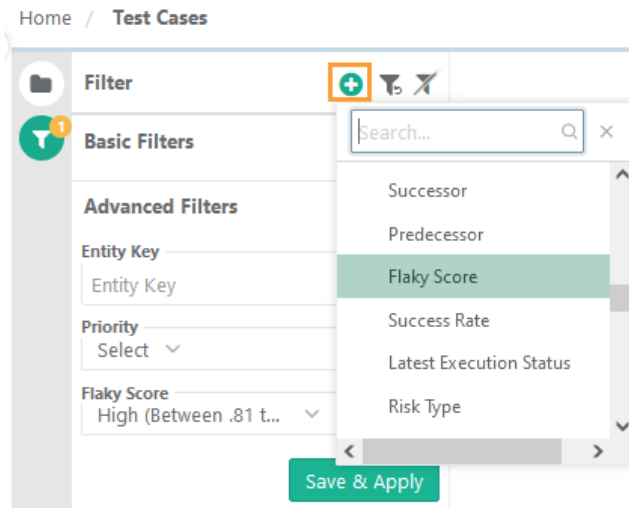
The larger the value (closer to 1), the test case has more flakiness. It means the behavior is non-deterministic.

The following table interprets the intensity of the flaky score in accordance with its derived calculation and color code.

Flaky Score Range	Intensity of Flakiness
Between 0.81-1	High
Between 0.41-0.80	Medium
Between 0-0.40	Low

Filter Test Cases on Flaky Score

The test cases can be filtered on Flaky Score. You can find the Flaky Score under the Advanced Filters.

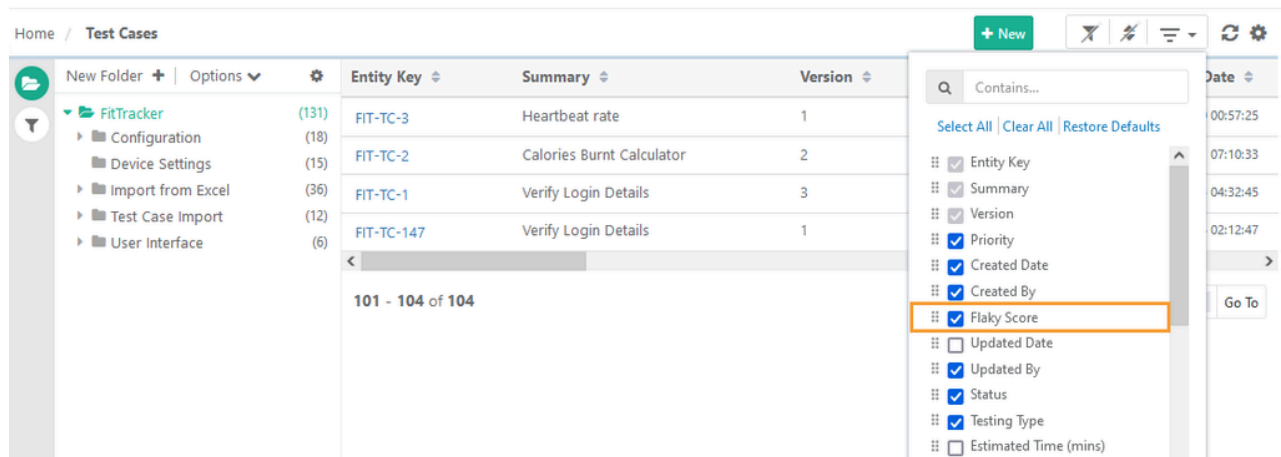


View Flaky Score on Test Case List View

As per the Flaky Score Settings done in the Configuration, the Flaky Score is calculated and displayed on the test case list view. It allows QA managers and Testers to view the risk probability and test execution history.

Go to the Test Cases module and make the Flaky Score column visible for the list view.

- **Flaky Score:** It is calculated based on pass or fail results frequency.



You can see the **Flaky Score** column with corresponding statistics. You can sort on the column to view test cases with higher flaky scores.

Entity Key	Summary	Priority	F.S.	S.R.
FIT-TC-147	Verify Login Details	Major	0.02	50%
FIT-TC-146	Calories Burnt Calculator	Critical	0.15	50%
FIT-TC-145	Heartbeat rate	Major	0.46	75%
FIT-TC-144	Set device to walk mode	Critical	0.00	100%
FIT-TC-143	Set device to Run mode	Critical	0.96	75%
FIT-TC-142	User Login	-	0.34	80%

Click on the Flaky Score to view the traceability report for the test case, which shows the test case is associated with which test executions, issues, and execution results. The report helps you analyze the Flaky Score further.

Test Executions and Issues

Test Executions	Issues
<div><div>FIT-TS-34Failed</div><div>Summary: Test Suite for Flaky</div><div>Status: -</div><div>Release: Macro</div><div>Cycle: MacroQ2</div><div>Platform: Firefox</div><div>Build: -</div><div>TC Executed Version: 1</div><div>Executed On: 05-18-2024 07:50:32</div><div>Executed By: mrunalini</div></div>	<div><div>FIT-IS-726</div><div>Issue Status: Done</div><div>External Key: FTR-3</div><div>Issue Summary: Error on Landing Page</div><div>Issue Type: Bug</div><div>Issue Priority: High</div><div>Resolution: Done</div></div>
<div><div>FIT-TS-34Passed</div><div>Summary: Test Suite for Flaky</div><div>Status: -</div><div>Release: Release 1</div><div>Cycle: R1C2</div><div>Platform: Firefox</div><div>Build: -</div></div>	

1 - 20 of 20

If the Flaky Score Settings are changed, the existing Flaky Score will get reset.

The info icon beside the Flaky Score column displays the details of when the score was last generated and by whom.

Home / Test Cases

+ New

	Entity Key	Summary	Version	Latest Executi...	F.S.	S.R.
<div>FitTracker</div>	FIT-TC-147	Verify Login Details	1	Failed	<div>Flaky Score last generated by Mathew at 05-29-2024 09:36:22. It can be generated again after 05-30-2024 09:36:22. Navigate to QMetry Intelligence Flaky Score Settings Page to regenerate score or manage the configuration settings.</div>	
<div>Configuration</div>	FIT-TC-146	Calories Burnt Calculator	1	Not Run		
<div>Device Settings</div>	FIT-TC-145	Heartbeat rate	1	Failed		
<div>Import from Excel</div>	FIT-TC-144	Set device to walk mode	1	Passed		
<div>Test Case Import</div>	FIT-TC-143	Set device to Run mode	1	UnCovered		
<div>User Interface</div>						

View Flaky Score on Test Case Detail Page

The generated Flaky Score is displayed beside the Test Case Key at the top of the screen.

Home / Test Cases / FIT-TC-147

F.S. 0.0250%

Verify Login Details

Version 1 (mrunalini | 05-07-2024 02:12:47)

Steps

Dependent Test Cases

Details

Requirements

Release & Cycles

Test Executions

Issues

Version

Attachments

Comments

Edit

Create Shareable Test Case

#	Step Description	Input Data	Expected Outcome
1	Enter username	admin	-
2	Enter password	solarEclipse	-
3	Enter @Employee ID, @User Type, @Designation	-	-

View Flaky Score on Test Case Link Screens

You can show or hide the Flaky Score column on the Link Test Cases screen.

Test Execution Default View

You can show or hide the Flaky Score column on the execution screen.

The screenshot shows the 'Test Execution' screen for '[FIT-TS-2]Regression'. A modal window titled 'Test Cases' is open, displaying a list of columns to be shown or hidden. The 'Flaky Score' column is highlighted with an orange box. The modal also includes a search bar, 'Select All', 'Clear All', and 'Restore Defaults' options. The background table lists test cases with columns for Test Case Entity, Summary, E.V., and a status icon.

Test Case Entity	Test Case Summary	E.V.	Status
FIT-TC-2	Calories Burnt Calculator	1	Pass
FIT-TC-1	Verify Login Details	1	Pass
FIT-TC-17	Set Pagination	1	Pass
FIT-TC-16	Filter Grid option	1	Pass
FIT-TC-15	Add Column to Grid Option	1	Pass
FIT-TC-14	Sort Grid Option	1	Pass
HTC-TC-1	Verify Login Details	1	Pass
BVP-TC-204	Admin - Roles List View	1	Pass
FIT-TC-5	Set device to Run mode	1	Pass
FIT-TC-132	Verify Login Details	1	Pass

The screenshot shows the 'Test Execution' screen for '[FIT-TS-2]Regression' with the 'Flaky Score' column visible. The table lists test cases with columns for Test Case Entity, Summary, E.V., Steps, Flaky Score (F.S.), Success Rate (S.R.), and Execution Type. The 'Flaky Score' column is highlighted with an orange box.

Test Case Entity	Test Case Summary	E.V.	Status	Steps	F.S.	S.R.	Execution Type
FIT-TC-2	Calories Burnt Calculator	1	Pass	3	0.46	75%	Manual
FIT-TC-1	Verify Login Details	1	Pass	2	0.02	50%	Manual
FIT-TC-17	Set Pagination	1	Pass	11	0.96	75%	Manual
FIT-TC-16	Filter Grid option	1	Pass	5	-	-	Manual
FIT-TC-15	Add Column to Grid Option	1	Pass	2	0.00	100%	Manual
FIT-TC-14	Sort Grid Option	1	Pass	3	0.02	50%	Manual
HTC-TC-1	Verify Login Details	1	Pass	2	0.34	80%	Manual

Click on the Flaky Score to drill down to view the test executions and defects associated with the test case.

How to Reduce Flakiness?

Flaky tests can occur due to various reasons. Testers should work with their development teams to find the exact reason for the cause. Here are the 10 common causes of flaky tests:

1. Async wait:

Some tests are written in a way that requires waiting for something else to complete. Many flaky tests use sleep statements for this purpose. However, sleep statements are imprecise, and the test may fail if the waiting time exceeds expectations due to variations in processing time.

2. Concurrency issues:

Flaky tests can result from concurrency issues such as data races, atomicity violations, or deadlocks. These tests often make incorrect assumptions about the ordering of operations performed by different threads. To address this, synchronization blocks can be added or the test can be modified to accommodate a wider range of behaviors.

3. Test order dependency:

Certain tests may pass or fail depending on the order in which preceding tests were executed. A good test should be independent and able to run in any order with other tests. It should be properly isolated and set up its own expected state.

4. Timing issues:

Flaky tests can arise from timing inconsistencies when the test code relies on specific event timings. For example, if a test checks for a particular webpage element after a specific delay, network issues or differences in CPU performance between test runs can lead to intermittent failures.

5. Element locator issues:

Many automation tools use XPath to locate webpage elements, but XPath can be unstable as it is sensitive to changes in the page's DOM. Self-healing AI techniques can address challenging testing scenarios involving dynamic elements, iFrames, and pop-ups. This involves using multiple locator strategies to find an element, and switching to a backup strategy if the primary one fails. Modifications to an element's properties or the addition of similar elements can render the initial XPath invalid, resulting in false positives or negatives.

6. Test code issues:

Poorly written or ambiguous test code can contribute to flaky tests. If the test code lacks clarity regarding the expected application behavior, the test may inconsistently fail or pass. Additionally, complex test code or code relying on external dependencies may be more prone to failure.

7. Test data issues:

Tests that depend on inconsistent test data can become flaky. Corrupted test data or different test runs using the same data can lead to inconsistent results. Tests utilizing random data generators without considering the full range of possible results can also introduce flakiness. It is advisable to control the seed for random data generation and carefully consider all possible values.

8. Test configuration issues:

Inconsistent test configurations between runs can cause flaky tests. Incorrect test parameters or improper test settings setup can result in test failures.

9. Environment issues:

Flaky tests can be attributed to the execution environment. Network connectivity problems, improper handling of I/O operations, hardware differences between test runs, or variations in test environments can introduce non-determinism, leading to flaky tests.

10. Resource leaks: Tests can become flaky if the application or tests do not adequately acquire and release resources, such as memory or database connections. To avoid such issues, it is recommended to use resource pools and ensure that resources are properly returned to the pool when no longer needed.