1.4 Test Execution and Defeat Management:
- Learning how to identify, report and manage defects?

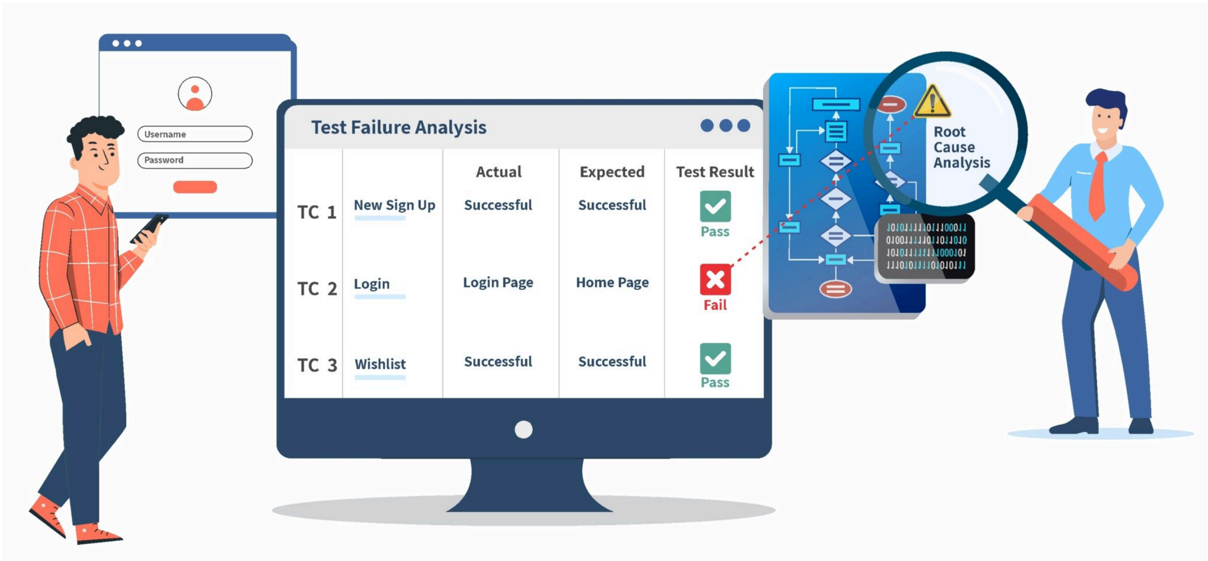
Ans: A defect in software testing is an error, flaw or anomaly in the software that can lead to unexpected behaviour or failure to meet requirements. It includes coding errors, design flaws, which coils hinder the smooth, uninterrupted functioning of a software system.

The aim is to find and fix defects, ensuring the software meets specifications and works correctly.

What is Defect Tracking?

Defect tracking is the process of identifying, documenting, and managing Defects in software during its development or maintenance.

It involves systematically recording information about a Defect define, prioritising, analysing, and tracking its status from discovery through resolution to ensure a more reliable and high-quality end product.



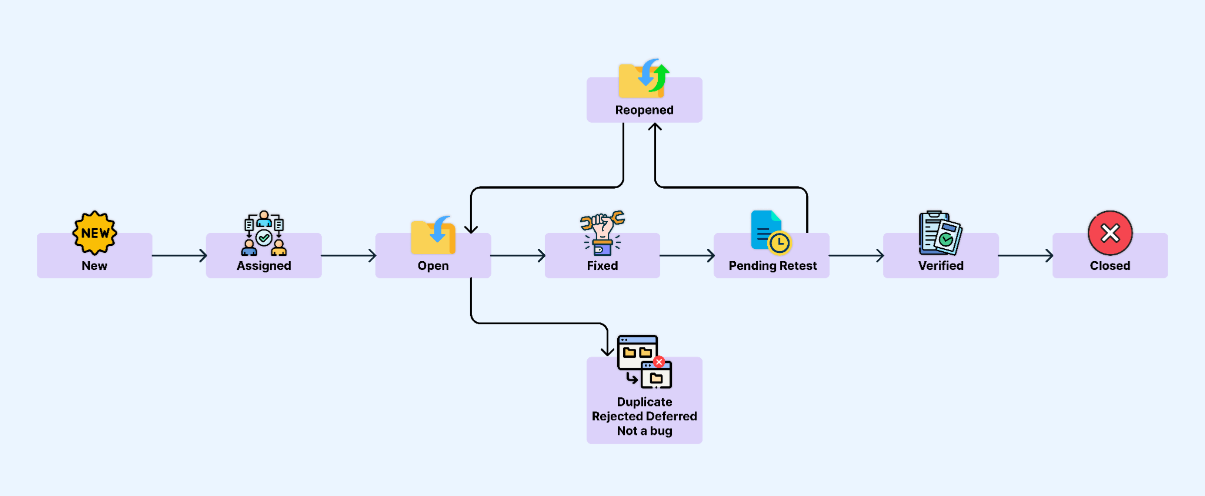
		Actual	Expected	Test Result
TC 1	New Sign Up	Successful	Successful	✓ Pass
TC 2	Login	Login Page	Home Page	☒ Fail
TC 3	Wishlist	Successful	Successful	✓ Pass

The primary goal of defect tracking is to improve the software development process by addressing and fixing issues promptly.

This is why a specialised team for QA and testing is required to manage this defect-tracking process. The main goal for test engineers is to identify, log, and track defects from the inception to the completion of the software project.

Different Phases of Defect Tracking

Defect Tracking process is a cycle that comprises different phases.



The different phases of Defect Tracking are:

1. New: Defects are initially identified and reported, starting in the "New" state.
2. Assigned: A team lead or manager assigns the defect to the appropriate developer or development team.
3. Open: The assigned developer works on fixing the defect, moving it to the "Open" state.
4. Fixed: Once the developer has implemented a solution, the defect is marked as "Fixed."
5. Verified: The testing team verifies the fix; if successful, the defect is marked as "Verified." If not, it may be reopened or assigned back to the developer.
6. Closed: After successful verification, the defect is marked as "Closed," indicating that it has been resolved, and the code is ready for release.

7. Deferred: The decision is made to postpone the resolution of the defect to a future release or update.

QAs take on the responsibility of evaluating the defects found and prioritising them for the developers, and they manage this process throughout the software development lifecycle. Further, the corrections made are also tested to ensure the defect is fixed, and the fix hasn't given rise to any other defects.

Why is Defect Tracking important?

Poor software quality cost US companies upwards of $2.08 trillion in 2020 alone. Any team juggling multiple projects will have to deal with finding and fixing defects among hundreds of thousands of lines of code. Missing any of these defects can cause serious issues with software functionality, which almost always leads to revenue loss.

Who needs Defect Tracking?

1. Developers who verify that their code is clean, functional, elegant (always a bonus) and efficient. Every defect leads to a drop in code and software quality, and devs must deal with each to ensure that the final product is pristine in usability and aesthetics.
2. QAs to identify and report defects to stakeholders. An automated and structured defect-tracking system helps QAs track defects seamlessly and check if they are resolved.
3. Product Managers evaluate product quality and its market-readiness with the help of defect tracking.
4. Business Analysts align future business requirements and feasibility based on the past defect history. Defect Tracking helps them understand the alignment of software development with the expected business requirements.

Defect Tracking Parameters

The most commonly used test tracking parameters are:

1. ID: Uniquely identified each defect.
2. Title: Name of the defect that is also usually descriptive.
3. Description: A larger description of the defect, as well as the step required to reproduce the defect.
4. Severity: Marking the intensity of the defect – critical, major, minor.
5. Priority: Defining how important it is to fix the defect, depending on how much it impacts the software system – high, medium, low.
6. Status: The current status of the defect – open, closed, deferred.
7. Assigned: The dev to whom the defect is assigned for resolution.
8. Due Date: The date by which the defect must be resolved.
9. Comments: Comments providing further information about the defect.

How to Design a Defect Tracking System/Process

1. A Defect tracking system is a process created to manage and deal with any and all defects found during software development. The process generally consists of the following:
2. Defect Detection

3. Defect Categorization and Assessment
4. Defect Fixing
5. Verifying Defect Resolution
6. Defect Reports
7. End of Project Defect Metrics

Defect Detection

Defect detection is the first step to defect tracking. Defects can be found by developers, test engineers, and product testers or during the beta testing phase. Sometimes defects are found following release as well. Once a defect is found, log it using a defect tracking tool.

Defect Categorization and Assessment

A report including the specifics of the defects is provided to the developer once they have been logged. The developer then determines the severity and scale of impact represented by each defect. Following this evaluation, the defects are segmented into categories such as critical, high severity, medium severity, and low severity, helping developers prioritise and fix defects according to this scale.

Defect Fixing

Defect resolution is performed by developers, and it is a multi-step process to evaluate and fix defects, where:

1. Defects are first assigned to developers with relevant knowledge.
2. Developers then perform a root cause analysis on the defects, based on their priority level. This helps them understand why the defect happened and how it can be addressed.
3. The developer then solves the defect and writes a comprehensive defect report explaining the solution and resolution process.

4. This report is once again added to the bug tracker and sent to the software testing team so they can carry out additional testing.

Verifying Defect Resolution

The software testing team will check the resolution report and retest the solution to ensure that the defect is resolved. If further defects have arisen downstream or the defect is not fixed, then the team would once again share the defects on the defect tracker and share the issues with the developers with relevant knowledge. On the other hand, if the defect is truly resolved, then the defect status will be updated as closed/resolved in the defect tracker.

Defect Reports

The software testing team creates reports for defects. This report explains:

1. Defects found in specific modules,
2. Why they happened,
3. How they were fixed,
4. Analysing the trends of the defects, and
5. How they were traced.

Test managers then write and deliver defect reports to the management team in order to receive feedback on the defect management procedure and the status of the faults. The management team then goes over the report and provides feedback, assistance, or further instructions as needed.

Defect Metrics

The development team learns through the process of tracking defects. Metrics of the defect finding and fixing process are calculated to evaluate performance, and data from the defect tracking process is recorded for future reference and to prevent recurrences.

Best Practices for efficient Defect Tracking

1. Some of the best practices for efficient defect tracking:
2. Clearly identify defects and document sufficient details like description, steps to reproduce, expected behaviour, actual behaviour, environment details, and artifacts. to understand the issue.
3. Follow a unified approach using the same techniques and tools for defect tracking.
4. Establish a standardised classification system for defects based on severity, priority, and type.
5. Assign prioritise correctly for effective defect tracking and troubleshooting.
6. Conduct regular defect triage meetings or sessions to review and prioritise incoming defects.
7. Track and analyse key metrics related to defect tracking, such as defect density, ageing, resolution time, reopen rate, and customer satisfaction.
8. Foster collaboration and communication among team members involved in defect tracking.
9. Continuously evaluate and improve the defect tracking process based on feedback, lessons learned, and evolving project requirements.

Regularly review and refine the workflow, tools, and practices to optimise efficiency and effectiveness.

Conclusion

Defect Tracking makes it easier to systematically debug a defect. It helps identify and resolve defects in a structured manner based on priority. Defect Tracking is critical in ensuring the delivery of high-quality software.

Using tools, such as BrowserStack Test Observability, you can get a unified view of the effectiveness of your defect-tracking processes for an enhanced debugging experience.