Software Quality Assurance (SQA): Software Metrics

Software quality metrics are tools to help development teams improve software quality by allowing them to identify issues. There are many different software metrics, I will explain some of their uses below.

Defect Density

Defect density helps to identify high risk modules by measuring defects per unit size of software, this is called KLOC or function points. By monitoring defect density improves reliability as it focuses testing on defect prone areas, therefore it helps reliability as it addresses defect prone areas of code before release.

Test Coverage

Test coverage reduces the likelihood of undetected bugs, as higher coverage means critical parts of the codebase are tested. This helps teams prioritise testing efforts.

Defect Leakage

Defect leakage measures the percentage of defects that have escaped testing and are discovered after the software has been released. If a high leakage percentage is found, this indicates that testing improvements are needed, which helps teams

Mean Time to Detect (MTTD) & Mean Time to Repair (MTTR)

MTTD and MTTR are used to track on average how quickly defects are detected and how quickly the defects are being fixed. A low score demonstrates quick identification of defects and that they are fixed quickly, this improves software reliability.

Defect Removal Efficiency (DRE)

DRE measures the percentage of defects before the release of the software, which means a lower percentage of defect reports after release. The higher the DRE the better as this indicates that most defects are being caught pre-release.

Customer-Reported Defects

Customer-Reported Defects tracks the percentage of defects that are found by customers after release, which can damage user trust. A high rate of customer reported defects implies that there has been inadequate pre release testing.

Test Execution Efficiency

Test Execution Efficiency ensures that testing stays on schedule by measuring the percentage of planned test cases executed. This metric helps avoid last-minute quality issues.

Escaped Defects

Escaped Defects count issues that were found in production. A high number of escaped defects indicate that there are issues in the testing process as the defects weren't found during testing. Analysing this metric helps to refine test coverage and improve automation strategies.

Automated Test Pass Rate

Automated Test Pass Rate measures the proportion of automated tests that pass. Low pass rate indicates bad tests.

Practical Scenario – Food Ordering App

After releasing the latest version of the app, the team has noticed an increase in customer complaints about performance issues and unexpected crashes. Below is an example of how each metric could be used to investigate and resolve performance issues in the app.

Defect Density

By identifying high risk modules using the defect density metric, this will help the development team pinpoint the areas with the highest defect rates and ensures that problematic code is addressed before release.

Test Coverage

The test coverage metric will help the development team know if there are functionalities that havent been tested properly (if the test coverage is low) and allows them to ensure they expand tests and improve coverage. If the test coverage **is low then the team will know to expand tests to improve coverage.**

Defect Leakage

By monitoring defect leakage the team can see if their pre release testing strategy needs improvement. If the leakage is high, then the team will need to introduce more real world testing scenarios.

MTTD & MTTR

By using MTTD and MTTR, this can help the team maintain their software

reliability as these metrics can indicate the teams response times to defects and how quickly the defects are fixed. If these scores are high then the team will know to implement better monitoring and debugging tools to ensure quicker detection and resolution times.

Customer-Reported Defects
By tracking the percentage of customer reported defects this conveys to the team that there has been poor pre release testing and that test case coverage needs to be increased.

Escaped Defects
By counting defects found in production the team can make a judgement on how well they are managing to catch issues before release, a high number indicates to the team that they need to refine test case design and improve automated tests in order to maintain software quality.

Reflection on Software Metrics

Software testing metrics are essential for maintaining software quality. These metrics help software development teams understand how well their current testing and debugging strategies are performing and what areas need more work. This ensures that software quality and reliability is maintained which is important as it could lead to poor user experience, revenue loss, security risks and reputational damage.