

Answers for Mindset Exercise

Choose 3 of 5 instances below where you can provide specific examples.

1.A. Identified a problem or inefficiency in a process related to quality or development, and implemented a change to improve it

Problem faced:

Lot of repetitive test cases in release regression with high failure rate, list of 'known failures' and no emphasis on coverage.

Solution:

I suggested to implement the process of code coverage for functional tests and used necessary tools to instrument the code and to record coverage. Reduced the regression set of 5000 tests to 2000 by removing redundant meaningless tests and adding more tests to improve code coverage.

1.C. Made changes to prevent defects from occurring or improve the quality before reaching the testing stage

I included Static analysis execution and Unit test coverage execution in the pre-commit CI pipeline (pre-commit CI runs as soon as pull request is raised in git to merge the code). This ensured well tested code and clean code writing practices which in turn will improve code quality even before code is reviewed, accepted and tested.

1.D. Recognized a pattern of bugs repeating and implemented a change to stop the pattern from recurring.

Pattern of bugs: There was a database field inconsistency in bank account records of millions of customers due a migration effort in one of the past features. This inconsistency was always handled in code as a workaround due to it being known issue for a long time. New team members who are not familiar with these issues often mishandled this workaround or forgot to handle certain cases and this started creating issues in production.

Change implemented: Implemented an audit batch to correct the inconsistency in bank record states (state is used to indicate a particular customer's bank account lifecycle in PayPal) so that developers don't have to handle the state inconsistency. This batch audited the customer data and corrected the states of 100 million PayPal users.

2. When you are testing a new feature, under what circumstances would you deviate

Some instances where I deviated from scripted testcases are:

1. When it's a brand-new feature that never existed before, I would want to use the product as a customer to get a feel of it and to explore it before I run the automation.
2. When I give wrong inputs, which was not in plan and the application behaves in an unexpected way, I test more negative scenarios along those lines.

3. If you joined a team of 10 developers as the only Quality Engineer and could implement and change any process(es) you'd like, how would you ensure that the team delivers high quality software?

I would implement a set of robust smoke test which will be integrated to pull request's pre commit CI pipeline. So, whenever developers raise a pull request, their code will be automatically tested for release critical cases and the test results are published in the pull request for reviewer's attention. So even before code is reviewed and merged, developer would have ensured that the code quality is good.

I would use Data driven QA automation framework which will help to ease the automation effort and hence developers can quickly add their own functional tests.

I would include Static analysis execution and Unit test coverage execution in the pre-commit CI pipeline which will help with clean and well tested code writing practices which in turn will improve code quality even before code is reviewed and accepted.