**Principles**

**Kindly share your views (not less than 100 words) on**

**Importance of quality in a fast-paced delivery environment**

* The quality is driven from testing in a fast-paced delivery environment where different approaches are required to deliver high quality of products to customers.
* Companies are competing to provide not only rich features but higher quality of products to end-users along with faster delivery to clients or customers to stay ahead in competition.
* In this environment, teams can present their quality products or new features to end-users quickly and can have more customer engagements.
* Companies are ready to adopt new approaches to stay ahead of competitors like continuous testing, shifting left testing, new tools, etc.
* Developers develop code quickly in a fast paced environment and prone for more bugs, testers can identify bugs at early stages and prevent those bugs from flowing to Production.

**What is shifting left - Pros and Cons of shifting left quality**

Shifting left testing means testing earlier in the software development process. It is the process where testers improve the quality of products by moving testing to the left of the software development lifecycle. For example, testers can work closely with developers to ask questions and create test ideas and 'what if' type scenarios. Testers can join design sessions to ask questions about how customers work, which ultimately leads to design changes.

Every approach has its pros and cons, some of the points I have mentioned below;

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Shifting left increases delivery speed by finding the defects in earlier development cycles and you can fix them a lot quicker, as a result the quality of software improves. | It is challenging to adopt new changes in culture as teams are usually set in their traditional ways of working. When the team considers shifting then they must consider how the methods, processes, skills, etc. will need to change. More importantly, what all the roles within a team are required to align with the new change. |
| It gives a greater ability to automate testing. | Approach is not so easy to implement. |
| Increasing the satisfaction of end-users by faster delivery of software with less defects is a major benefit of the shift-left approach. | Requires all developers to put an accent on testing as well, which causes a bit of an overhead. |
| Earlier the bug is found the cheaper it is to fix it and can lead to reduction in human error rate. |  |
| Increase test coverage as more tests can be run in the same amount of time. |  |
| More time for testers to focus on more challenging and rewarding tasks. |  |

**Exploratory testing and automation testing - Pros and Cons and how they can be synergized**

* **Exploratory testing** is an approach that is described as simultaneous learning, test design, and execution. Test cases are not created in advance for this type of testing as testers check the functionality through exploring it. This test requires cognitive skills as it requires thinking activity. It is a process of test design and test execution at the same time. In this testing, the tester has more personal freedom and responsibility to utilize their skills and knowledge to optimize the quality of their work. Exploratory testing is performed after functional and regression testing, at that stage, tester has sufficient knowledge on functions to perform this type of testing.

Let’s see some of the pros and cons of exploratory testing below;

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Directed from requirements and exploring during testing | There is no documentation for reference, so the testing cannot be reproduced |
| Determination of test cases during testing | Important bugs can be missed |
| Investigation of system or application | Insufficient knowledge on domain cannot perform this type of testing |
| Emphasizes adaptability and learning | There is no clear and measurable test coverage |
| Involves Investigation | The application is compared to the expectations and understanding of the tester on how the application is supposed to work |
| It is about Improvement of test design |  |
| The tester's mind is in control |  |
| It is not a technique but it is an approach |  |
| It is cognitively structured |  |
| it is an ad-hoc testing with a purpose of finding bugs |  |

* **Automation testing** is scripted testing which requires creating test cases in advance, executing later and comparing the actual results with expected outcomes. Automation testing saves testers time to run repetitive but necessary tasks and other testing tasks which are difficult to perform manually. It requires special software tools to perform automation testing. Test automation is critical for continuous delivery and continuous testing.

Below are the pros and cons of automation testing;

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Much fewer human errors | Proficiency is required to write the automation test scripts. |
| Increased test coverage (multiple tests can be conducted at the same time) | Debugging the test script is a major issue. If any error is present in the test script, then it may lead to deadly consequences. |
| Ability for testers to focus on more interesting and fulfilling tasks | Test maintenance is costly in case of playback methods. Even though a minor change can occur in the GUI, the test script has to be replaced by a new test script. |
| Fewer production issues |  |
| You can reuse tests on different versions of an application, even if the user interface changes. |  |
| Automated Tools run tests significantly faster than human users. |  |
| It is economical as the number of resources for regression testing are reduced. |  |
| You can build a suite of tests that covers every feature in your application. |  |

**How they can be synergized**

During exploratory testing, if any critical bug is found then we can use that scenario into a test script to automate it, to make sure this bug does not leak to the production environment.

**Checklist for an efficient automation framework**

Framework is the backbone of any automation testing as it provides an execution environment for test scripts. It allows the user to develop, execute and report on the automation test scripts in an efficient manner. Framework combines guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanisms, test data injections and more. Testers benefit from it by easily following standards while automating the application. My checklist are;

* Configurable
* Reporting
* Easy to maintain
* Reusable utilities
* Complete Test plan document
* Defined Business scenario
* Data driven tests to expand test coverage
* Verifications added
* Maximum coverage
* Recovery scenario
* Minimal manual intervention

**Share your experience where you have driven quality process improvement**

When I joined the current team there was no proper Quality Assurance process, Business Analyst used to verify the functions in one environment. I have introduced the quality process in a team where I have set three separate test environments, Build environment for smoke and function testing, SIT environment for automation testing of WEB UI applications which includes Scripts development, Scripts execution, Scripts maintenance. UAT environment for exploratory testing. I have implemented Selenium using Java as a programming language, TestNG as a Unit Testing framework, Page Object Model along with Page Factories as a Design Pattern. Further, I have implemented open source libraries including Log4J for Logging, ExtentReports for Reporting and Apache POI for Read/Write operations for Excel files. For the Version Control System, I have a git repository and for CI/CD, I am working with Jenkins.

I am leading the QA process by creating and maintaining test plans, test cases, test execution, bug reporting and test reports. I have introduced test metrics to identify the number of bugs and trace them to avoid in the future, created in JIRA as a dashboard. I am creating a new test metric for each quarter to make comparisons that allow my manager to analyse the QA progress. I have categorised the bugs based on functions and severity, where I can analyse the problematic areas in the product. Recently, I have introduced shifting left testing to the team where initially developers had some concerns but later I have cleared their doubts and started adopting the new approach. My manager has appreciated my approaches to bring new ideas and driving the quality process improvements.