

Tester's and Developer's carry different mindsets, they often think differently. The developers think: "**How can I make the application?**" whereas the tester's perspective is: "**How can I break the application?**" But one can achieve the desired result only when both Testers and Developers work together collaboratively.

In this article, we will discuss the different perspectives of testers and developers and how they can work together to achieve efficiency and success.

Tester's and Developer's Mindsets

Tester's and Developer's Perspective at different phases of Software Development Life Cycle (SDLC)

Comparison of Tester's and Developer's Mindsets

Tester's and Developer's Mindsets

A mindset is a belief that guides the way we handle situations, how we solve what is happening, and what we should do. Different people around us have different mindsets. Likewise, developers and testers have a different way of thinking too. A developer will ask: "**What do I need to build, how should I do it?**" The tester will ask: "**What can go wrong? What can I do to break the application or find the weaknesses?**"

By saying, '**How can I break the application?**' it does not mean that the motto of a tester is to spoil the work done by the developers. It means that the tester should place himself in the customer's shoes and test the application for all possible scenarios. Which, in turn, ensures that the application does not break when it is in the production environment.

TESTER'S AND DEVELOPER'S MINDSETS

TOOLS  QA



Tester's and Developer's Perspective at different phases of Software Development Life Cycle (SDLC):

Software Development Life Cycle (SDLC) plays a very important role in any software application development. Previously, software testing happened in the last stages of development. However, fixing errors in the last stage usually, turned out to be very difficult and expensive. Therefore, now, software testing happens in every phase of SDLC. It means that the testing starts right from the requirement phase. Both tester and developer become an integral part of the development process right from the beginning.

Let us discuss and get a brief idea of testers and developers involvement at different phases of SDLC:

1. Requirement Gathering and Analysis: *The preparation of requirement documents happens in this phase, as stated by the customer.*

Developer's role: *After getting the requirement document, they will analyze the requirements and start finalizing the technology stack.*

Tester's role: *After analyzing the requirements document, the testing team can ask their set of queries. The testers can also find requirement defects. It saves time and money if detected and fixed at this stage.*

2. System design: *In this phase, the architecture, interfaces, modules, and data for a system are defined to meet the specified requirements.*

Developer's role: The transformation of the requirements identified in the requirements analysis phase into a system design document happens here. This document accurately describes the system design. Additionally, it works as an input for the development of the system in the next phase. Based on these detailed specifications, developers write the code for the software.

Tester's role: The testers, from their understanding and creative thinking, analyze all the possible scenarios for all the new features, integrations, etc. Preparation of test scenarios & test data happens to ensure smooth testing of the application. For this, they create a test strategy, integration test plan, test cases, checklists, and test data.

3. Coding phase: The coding phase is also called the "**implementation**" or "**development**" phase. It involves the development of the actual product. The developer writes the code and then tests it continuously and incrementally to ensure that the different components work together. It is the most time-consuming phase of the SDLC process.

Developer's role: Once the system design phase ends, the next stage is coding. In this phase, developers begin to build the entire system by writing code using the chosen programming language. In the coding phase, the task division into units or modules takes place and thereafter, it is assigned to different developers.

Tester's role: In current agile methodology, progressive automation and functional testing happen. Whereby, a tester automates and tests the application after coding. At this stage, the tester would test each component and also carry out component integration testing. Therefore, the tester needs to work closely with developers to make this phase successful.

4. System Testing: Once the software is complete and deployed in the test environment, the testing team begins to test the functionality of the entire system. It is to ensure that the entire application works according to customer requirements.

Developer's role: The testing team may find some defects and communicate them to the developers. The development team corrects the error and sends it back to the testing team for the retest. This process continues until the software is error-free, stable, and functioning & in accordance with the business requirements.

Tester's role: In this phase, the tester executes the end to end test cases and verifies every aspect of the system. Apart from the desired functionality, they also check the system from a user's perspective. So the testers use their creative thinking and explore each possible scenario. System integration testing, which involves integration with third-party systems, also occurs at this stage.

5. Maintenance phase: The maintenance phase starts once the system deployment to production happens, and customers begin using the product. This phase includes the post-deployment supports and fixes.

Developer's role: In this phase, the developer performs the following three activities:

Bug fixes: Developer fixes the bugs that are reported by the customer.

Update: They update the application to the latest versions of the software.

Enhancements: Developers also add some new features to existing software based on customer feedback.

Tester's role: When the developer finishes his job, the tester retests the application. The tester ensures that the system is working correctly after the code change or enhanced functionality. Tester is also responsible for doing regression testing to ensure existing functionality does not break by the latest changes.

Thus, we see that developers and testers interact a lot during each phase of the software development life cycle. Therefore, they must work together as a single team to ensure the project is a success.

Comparison of Tester's and Developer's Mindsets

Creating software products is a complex effort that requires the cooperation of people with different skills, knowledge, and thinking. **Developers and testers** have to work together to create and deliver quality products. These people can have different points of view, perceptions, and knowledge. To understand the difference in their approaches and opinions let us consider the following points-

Comparison of the tester and developer approach: *The testing and reviewing of the applications are different from their analysis and development. A developer, while creating or developing applications, is working positively to make the product according to the user's specifications. He continuously solves the problems during the development process. However, during testing or reviewing a product, testers lookout for defects or faults in the product. Therefore, building software requires a different mindset to test the software.*

The developer plays the role of a tester: *Even though, Testers and Developers are separate roles; it does not mean that their roles are not reversible. In other words, the tester can be the developer or the developer can be the tester. Developers always test the component that they built before giving it to anyone. This process is known as unit-testing. However, we all know that it is difficult to find our own mistakes. So, the developer sends the applications to test specialists or professional testers which allows independent testing of the system. This degree of independence avoids the author's bias and is often more effective in finding defects and failures.*

Clear and courteous communication and comments about defects between the tester and the developer: *Responsibility of tester is to test the software against specified requirements and report the defects and failures. But the developer who builds the application can react defensively and take this reported defect as a personal criticism. Therefore, a tester oughts to be very careful when acting or reporting bugs/flaws to the developer.*

To sum up, appreciating differences is essential for productive teams. But different approaches help to find solutions and lead to the delivery of a product that works in the best way. The testers and the developers together form a capable team. It is their responsibility to guarantee the best product. And, it is possible only if both work hand in hand with proper understanding and positive feedback.