The software is a humanistic approach that requires the involvement of human beings at every stage. Humans invariably create the software, test it, and in most cases, use and abuse it too. Thus, we require a lot of effort and understanding to be able to deal with the fact that we are fallible. In this article, we'll discuss the ***Psychology of Testing***, that is, various psychological factors that influence testing and success. We will focus on the following given points:

How is Human Psychology related to Testing, along with a real-life example?
Role & Characteristics of a Software Tester
How to improve communication and develop relationships between Testers and Developers?

Let’s learn the psychology of testing in detail.

Human Psychology and Testing

The mindset which we apply during ***Testing and Reviewing*** is different from the one that we use during ***Designing*** or ***Developing***. It means, while building the software, we work positively towards the software with an intent to meet customer requirements. However, when we test or review the product, we lookout for defects in the product.

Let's understand Psychology of Testing with the help of an example:

Suppose you are a chef in a five-star hotel. Your job is to cook the best meal you possibly can, with the available ingredients. You bring all your experience to the table and create a meal that is flawless from your point of view. Now imagine that there is an internal quality inspector hired by the hotel who will taste your meal, and give a report to the hotel management on the quality and taste of the meal.

You can correlate the chef's mindset with that of a Developer and the quality inspector's mindset with that of a Tester. The chef cooks the food with positive intent, whereas the quality inspector looks out for flaws/mistakes in the dish prepared by the chef.

A human psychology element called ***Confirmation bias*** refers to thinking that makes it difficult to accept information that disagrees with currently held beliefs. For example, *Developers believe that their code has no errors. So, it is difficult for them to take that their code is incorrect. Testing is often looked upon as bearer of bad news by Developers as it highlights* ***defects/failures*** *in the system. It's difficult to see the bigger picture that these defects eventually make the software better and more usable*.

Role & Characteristics of a Software Tester

Software Development and Software Testing go hand in hand, simultaneously. Both aim to meet pre-defined requirements and purposes. The work of developing software is constructive or creative. On the other hand, software testing usually falls in the category of destructive work or negative work. Because testing a software needs the mindset to break the application/software. Hence, it's considered a destructive process.

Additionally, software testing has established procedures and techniques that are designed to give a tester the best chance to find defects. To conclude, software testing is a destructive process to achieve a constructive purpose.

Software Testers also need to acquire the following skills in addition to technical skills:-

***Interpersonal skills**: Firstly, there should be effective communication between testers & developers about defects, failures, test results, the progress of tests and risks, etc. The way of conveying messages should be very concise, complete, and humble. Additionally, they should try to build friendly relations with developers so that they are comfortable to share feedback*.
***Sharp observation**: Secondly, the software testers must have an "**eye for detail**". The testers can quickly identify or detect many critical errors if they observe sharply. Moreover, they should examine the software for the parameters such as 'look & feel' of GUI, incorrect data representation, ease of use, etc*.
***Destructive creativity**: In addition to the above, the tester needs to develop destructive skills as well. In other words, the tester should not "**hesitate to do negative testing**". Negative testing is checking the system under unexpected conditions. For instance, if a login ID is specified to use only alphabets, then the tester should test it with numbers and special characters as well. A creatively oriented but destructive approach is necessary. It produces a more robust & reliable software*.
***Customer-oriented perspective**: The software testers should adopt a customer-oriented perspective while testing the software product. They should be able to place themselves in customer shoes and test the product as a mere end-user*.
***Cynical but friendly attitude**: Regardless of the nature of the project, the tester must be tenacious when questioning even the minor ambiguity until it is proven. Different situations may arise during the test. For instance, the detection of a large number of errors may cause a more significant delay in the shipment of the product. It can lead to a tight situation between testers and other development teams. The tester must balance this relationship. It should not happen at the expense of errors. Testers should convince and defend the intentions of "**attacking software issues but not software developers**"*.
***Organized, flexible, and patient at work**: Testers realize that they can also make mistakes. Therefore, they should be excellent organizers -they must-have checklists, facts, and figures to support their findings. Additionally, the tester should be flexible and open to new strategies. Sometimes, significant tests must be re-run that would otherwise change the fundamental functionality of the software. Therefore, the tester should have the patience to retest the software for as many new errors as may arise. Testers must be patient and stay prepared in projects where requirements change rapidly*.
***Objective and neutral attitude**: No one likes to hear and believe the bad news. Well, testers resemble messengers of bad news in a software project team. No matter how brilliant the testers are at their job; nobody wants to share the bad news. But, the tester always communicates the wrong part of the software, which the developers do not like. The tester must be able to deal with the situation in which he has to face the accusation of doing his job (i.e., detecting errors) too well. The tester's work should be appreciated, and the development team should welcome the*

errors. That is because every potential error encountered by the tester would mean a reduction of an error that the client might have encountered.

Regardless of the perceptions of testing being destructive, the role of a tester is to report honestly every known mistake found in the product with a specific objective and neutral attitude.

Defect reports need to be raised against the software, not against the individual who made the mistake.

How to improve communication and relationships between Testers and Developers?

As discussed above, testing highlights defects/errors. Therefore, the perception is that it is a destructive activity. It, in turn, makes it all the more important for testers to convey the defects/faults constructively. Apart from technical and domain skill-set, one of the primary skills of a tester/test manager is his ability to communicate effectively. This communication can be about defects, test results, test progress, and risks. They should also be able to build a positive relationship with their colleagues, specifically with their development counterparts.

Here are a few ways of effective communication:-

Communicate findings of the product in a neutral, fact-centric manner, without criticizing the person who created it. For example, write objectives and provide a logical, organized, and detailed defect report. The sections of this defect report may include: Complete description, build/Platform, steps to reproduce, actual results, and expected outcomes.
- *Do not boast - You're not perfect either!*
- *Do not blame - the errors are probably from the group instead of an individual*.
- *Be constructively critical and discuss the defect and steps to reproduce it*.

Discuss how to fix the defect so that the delivered system is better for the client.
- *Demonstrate the risk involved with the defect and clearly define the priority of the defects*.
- *Do not just see the pessimistic side, praise the efforts as well.*
- *Show discovered risks and the benefits of fixing the defects.*
- *Confirm that the developer has understood the defect clearly*

Collaborate instead of battling around. Remember everyone in the project has a common objective of creating better software.
- *Be collaborative, kind, and helpful to your colleagues*

Try to understand how the other person feels and why he/she reacts the way they do.
Confirm that the other person has understood what you said and vice versa.
Offer your work to be reviewed, too.

Try to understand how the other person may feel, and if they will react negatively to the information presented.

Conclusion

To achieve successful testing it is essential for software engineers to consider the psychology of testing. The tester must have a good working relationship with the developers. It, in turn, will not only help in creating a quality product but will also promote collaboration and learning opportunities.