Improving Software Quality

Eng. Asmaa Lafi

Importance of Software Quality

The way to manage software quality has become a critical component of project management at every step.

The cost-effectiveness and higher performance of your projects will be enabled by high-quality software.

Poor software quality increase issues and can become a time consuming and costly exercise.

Methodologies to Improve Software Quality

❑ Test at an early stage

- Testing is important when learning how to improve software quality and shouldn’t be neglected. Testing aims to catch defects early during the design phase so they don’t snowball and grow into bigger issues later.

- Testing early also invariably reduces the money spent on bug fixes. A defect that could cost your company $100 in the requirements phase can cost $10,000 or more in the product implementation stages.

Methodologies to Improve Software Quality (cont’d)

❑ Ensure Quality control

- In collaboration with developers, testers can monitor quality controls and raise awareness to guarantee that standards are consistently maintained. Quality controls begins at the beginning and continues throughout the delivery process.

❑ Implement quality Assurance

- QA should be presented at all times during the software development process. It is a type of team provided governance that installs trust in the overall software quality.

Methodologies to Improve Software Quality (cont'd)

❑ **Encourage innovations**

- Automation of testing where possible, to save time spent on rules is fantastic method for creativity. Moreover, innovations can lead to improvements in software quality that can change the way projects are delivered.

❑ **Outline your deliverables**

- A clear and comprehensive outline of what the project will provide can ensure that quality is prioritized right from the start. It also guarantees that money, resources and time are all adequately allocated to achieve high quality results.

Methodologies to Improve Software Quality (cont'd)

❑Communication is key

- For any relationship to be successful, whether it's personal or business, communication is key. To improve software quality it is important that all parties to the project have full information through fluid communication channels.
- show how software quality is measured at every step of the development process.
- all parties have the opportunity to provide feedback to the team to ensure that all expectations are being met.
- keep all stakeholders in the loop and not isolate team members from the vendors or end user of the software.

Methodologies to Improve Software Quality (cont'd)

❑ Have a risk register

- A risk register is a management tool to manage risks. A risk register is more synonymous with financial auditing, however it is still a vital element in software development.

- A risk register will provide everybody aligned on a software project a list of clearly identified risks and then assess them in regards to the importance of delivering the project. A risk register works well for software quality because its creation actively leads to risk mitigation.

Methodologies to Improve Software Quality (cont'd)

❑ Risk Factors

Risk factors are the uncertain conditions and influences that will affect the cost, duration and quality of the project negatively.

Some of the risks associated with systems development are as follows:

- Inadequate budget
- Impractical project schedule
- Allocation of appropriate resources
- Changing end user needs
- Non- compliance with industry regulations
- Lack of communication between project management team and system designers/developers
- Inaccurate data conversion
- Improper documentation of plan, design and maintenance

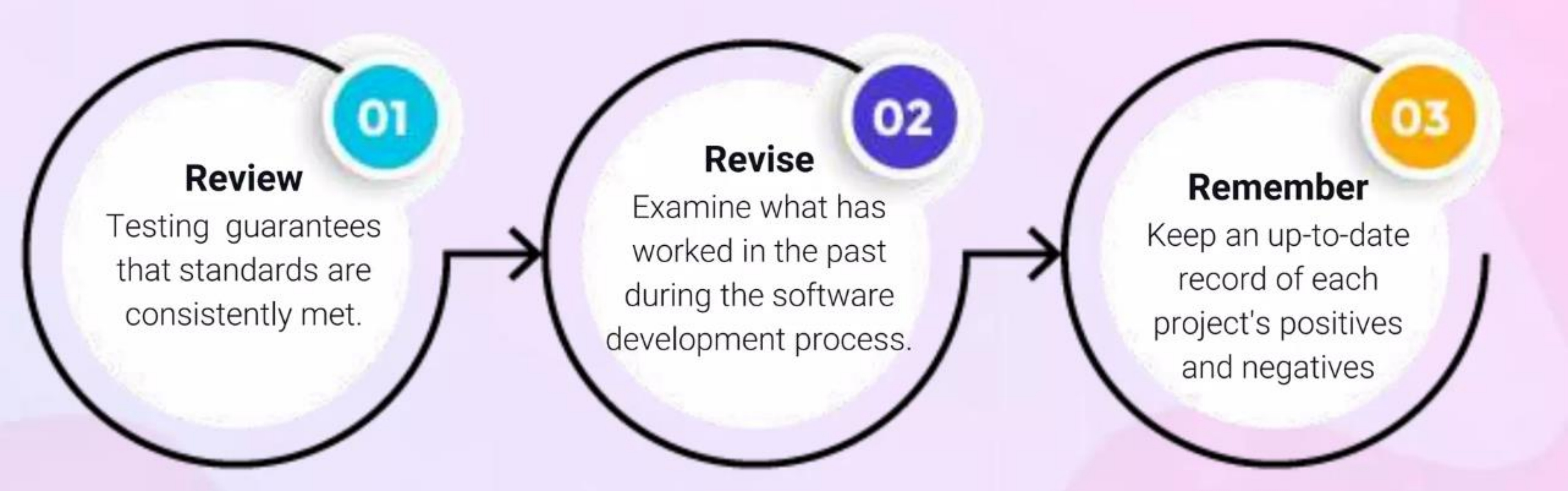
Methodologies to Improve Software Quality (cont'd)

❑ Long term strategy

- the advantages of long-term strategy for producing software quality:
    - Doing it right first means you don't have to spend time doing it over.
    - Placing as much importance on architecture and design as coding ensures the veracity of your project.
    - Creating coding standards with long-term vision eliminates unnecessary mistakes.
    - Effective peer review ensures errors are minimised even though it may seem time-consuming at that particular moment.
    - Testing often allows you to plan further ahead with certainty that errors and bugs have been fixed.

Methodologies to Improve Software Quality (cont’d)

❑ Three Rs for quality check



01

Review

Testing guarantees that standards are consistently met.

02

Revise

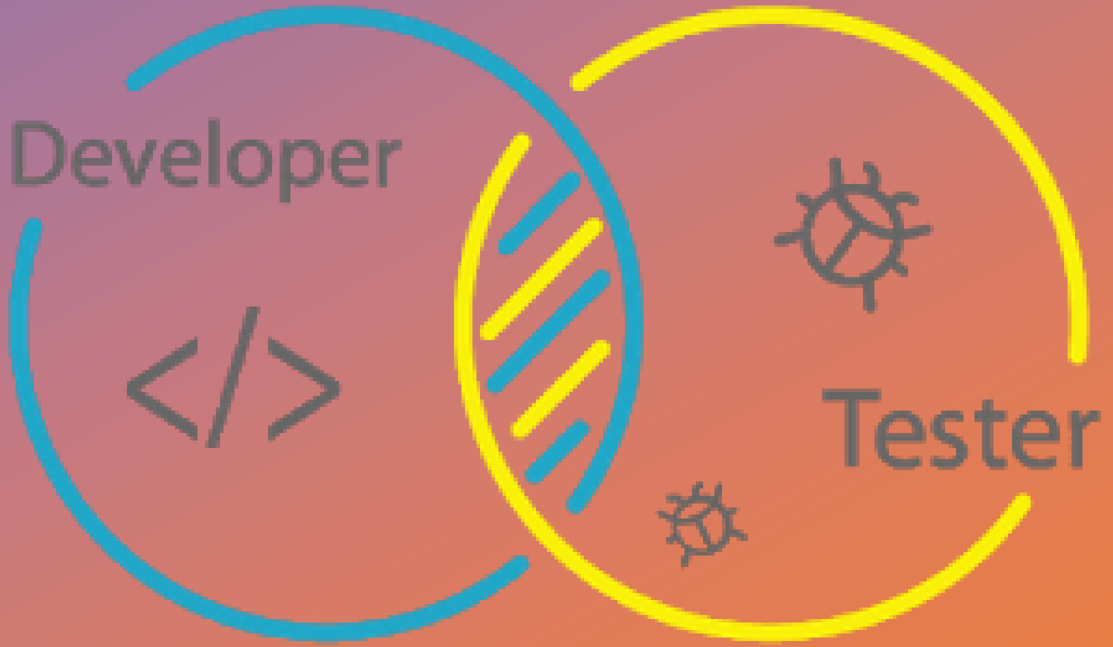
Examine what has worked in the past during the software development process.

03

Remember

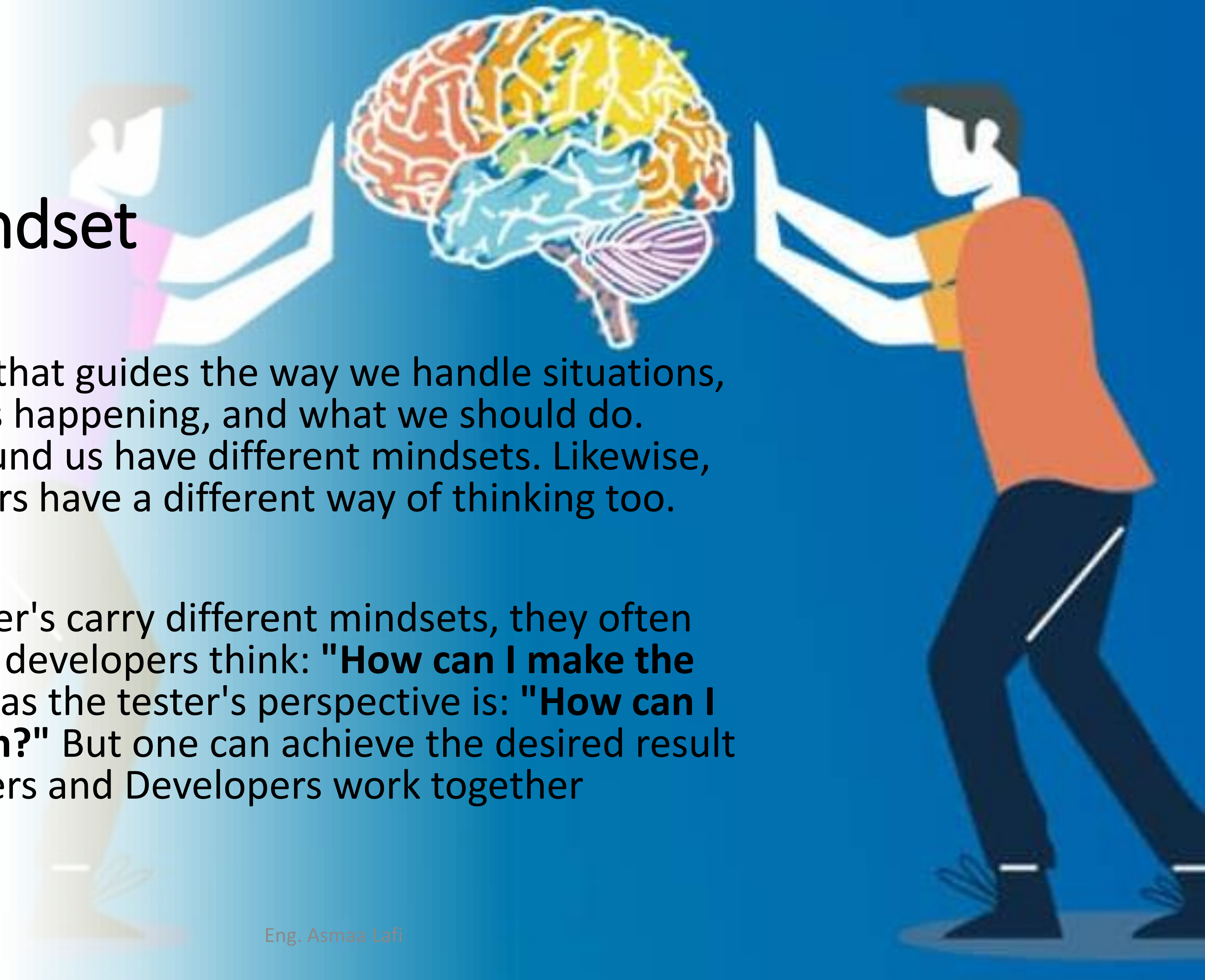
Keep an up-to-date record of each project's positives and negatives

Impact of psychology and mindset of tester and developer on The Software success



Eng. Asmaa Lafi

Tester and Developer mindset



- 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.

- 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.

Testers and developers involvement at different phases of SDLC

- ❑ **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.

Testers and developers involvement at different phases of SDLC

❑ **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.

Testers and developers involvement at different phases of SDLC

❑ **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.

Testers and developers involvement at different phases of SDLC

❑ **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.

Testers and developers involvement at different phases of SDLC

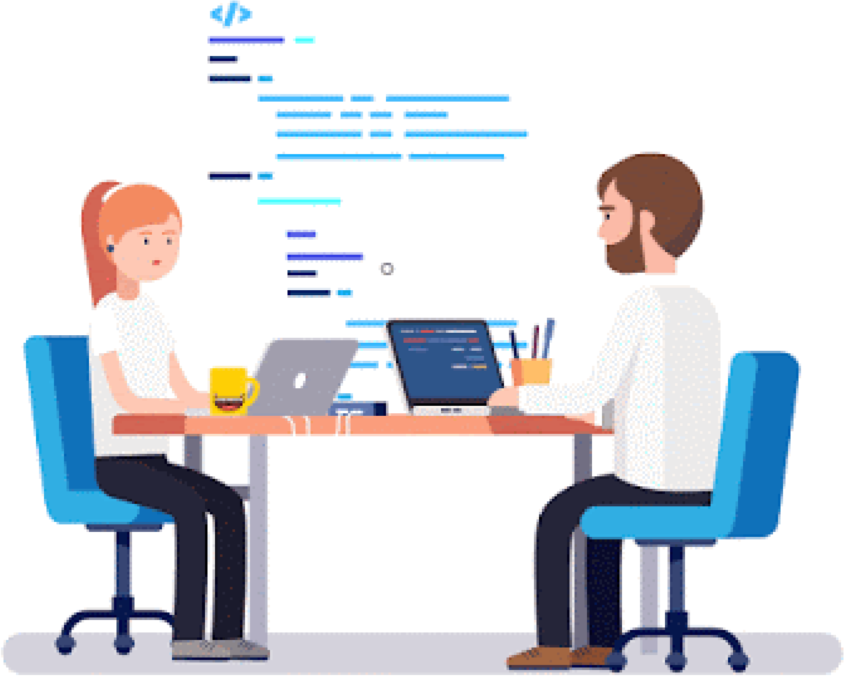
❑ **Maintenance phase:** The maintenance phase starts once the system deployment to production happens, and customers begin using the product. This phase includes the post-deployment supports and fixes.

- **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.

Comparison of Tester's and Developer's Mindsets

- The testing and reviewing of the applications are different from the analysing and developing of it.
- If we are building or developing applications we are working positively to solve the problems during the development process and to make the product according to the user specification.
- Testing or reviewing a product we are looking for the defects or failures in the product.
- Thus building the software requires a different mindset from testing 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.

- 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 self-testing.

- 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 feedback on defects between tester and developer

- We all make mistakes and we sometimes get annoyed and upset or depressed when someone points them out. So, when as testers we run a test which is a good test from our viewpoint because we found the defects and failures in the software.

- Because testing can be seen as destructive activity we need to take care while reporting our defects and failures to the programmers as objectively and politely as possible.

Conclusion

- ✓ 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.

References

- https://medium.com/@madhavikau/psychology-of-testing-in-software-testing-a59bdeb90ae0
- https://www.toolsqa.com/software-testing/istqb/testers-and-developers-mindsets/
- http://tryqa.com/what-is-the-psychology-of-testing/
- https://www.browserstack.com/guide/how-to-improve-software-quality
- https://testpoint.com.au/11-ways-to-improve-software-quality/
- https://eujournal.org/index.php/esj/article/viewFile/2624/2485