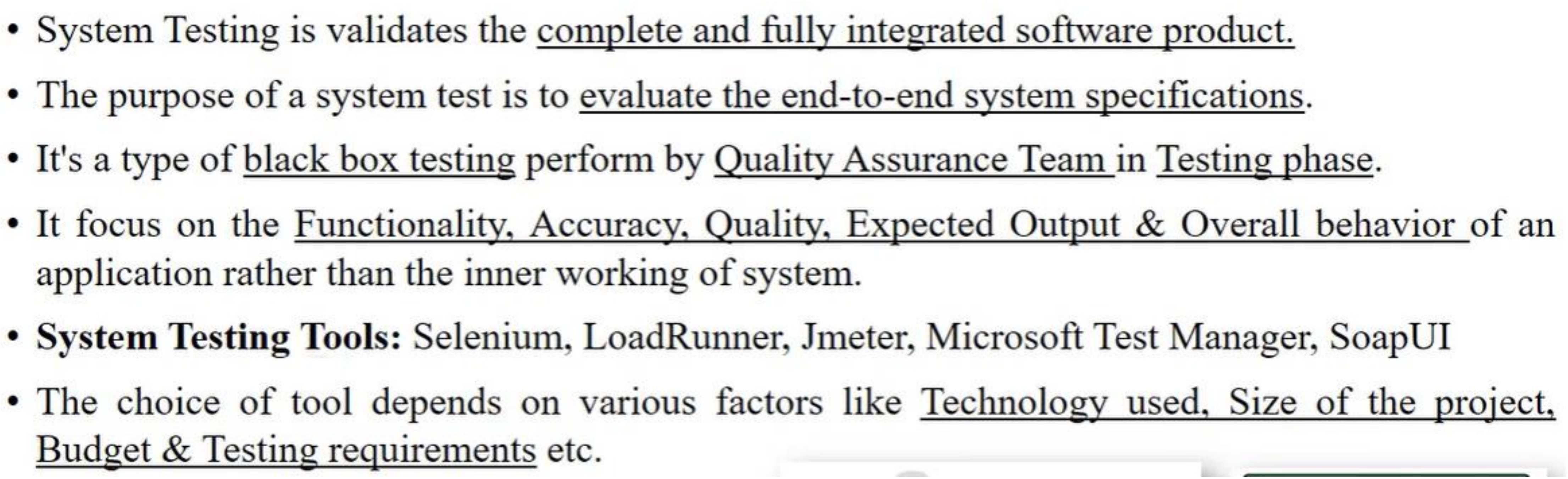
# Testing and Quality Assurance

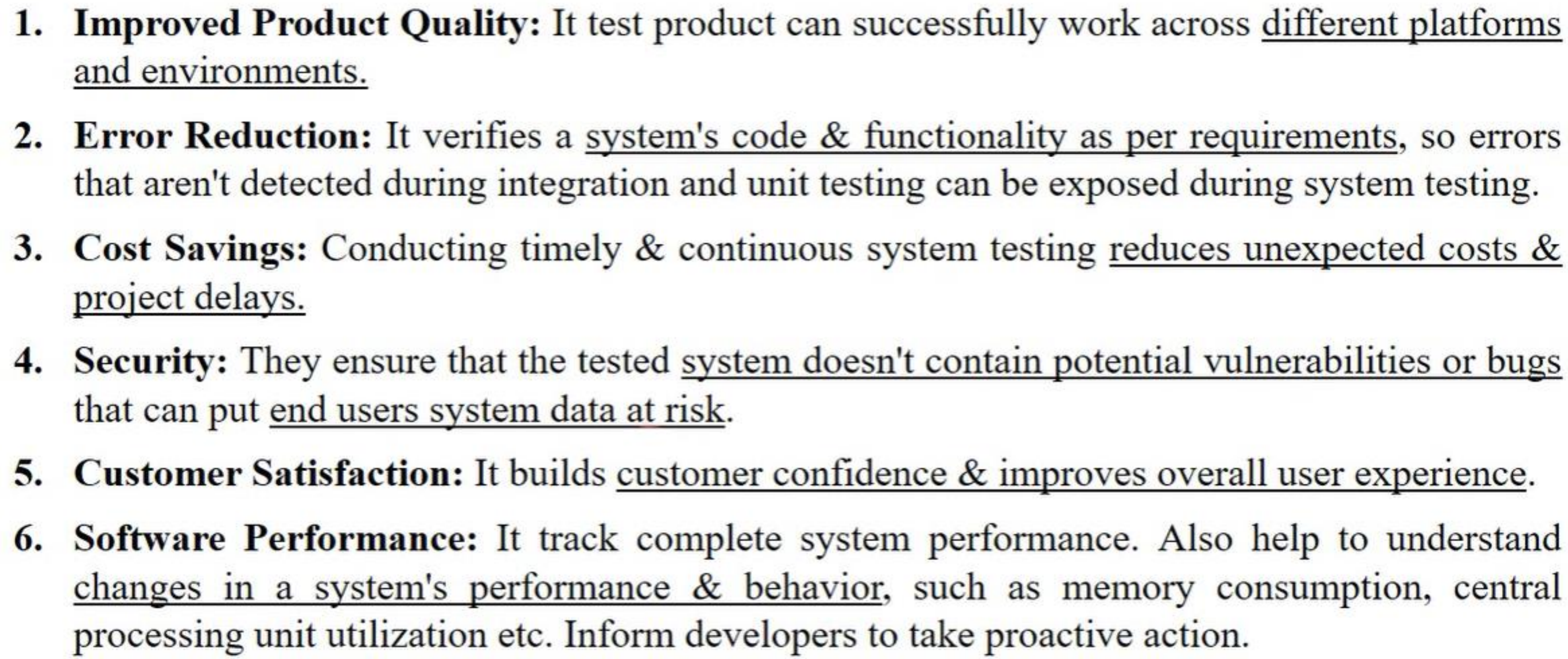
Testing ($2^{nd}$ Part)

# System Testing



- System Testing is validates the complete and fully integrated software product.
- The purpose of a system test is to evaluate the end-to-end system specifications.
- It's a type of black box testing perform by Quality Assurance Team in Testing phase.
- It focus on the Functionality, Accuracy, Quality, Expected Output & Overall behavior of an application rather than the inner working of system.
- **System Testing Tools:** Selenium, LoadRunner, Jmeter, Microsoft Test Manager, SoapUI
- The choice of tool depends on various factors like Technology used, Size of the project, Budget & Testing requirements etc.

# Importance of System testing



1. **Improved Product Quality:** It test product can successfully work across different platforms and environments.
2. **Error Reduction:** It verifies a system's code & functionality as per requirements, so errors that aren't detected during integration and unit testing can be exposed during system testing.
3. **Cost Savings:** Conducting timely & continuous system testing reduces unexpected costs & project delays.
4. **Security:** They ensure that the tested system doesn't contain potential vulnerabilities or bugs that can put end users system data at risk.
5. **Customer Satisfaction:** It builds customer confidence & improves overall user experience.
6. **Software Performance:** It track complete system performance. Also help to understand changes in a system's performance & behavior, such as memory consumption, central processing unit utilization etc. Inform developers to take proactive action.

# Types of Software Testing

### 1. Performance Testing:

- It measures Speed, Load time, Stability, Reliability & Response times of the system under various conditions.

### 2. Load Testing:

- It determine how system or software performs under a real-life extreme load.
- Such as Throughput, Number of users etc.

### 3. Usability Testing:

- It evaluate system is easy to use and functional for the end user.
- Such as User error rates, Task success rates, Time takes a user to complete a task & User satisfaction etc.

# Contd…

## 4. Regression Testing:

- It ensure that any changes done during the development process have not introduced a new defects or bugs.
- Also ensure that old defects or bugs will not exist on the addition of new software over the time.

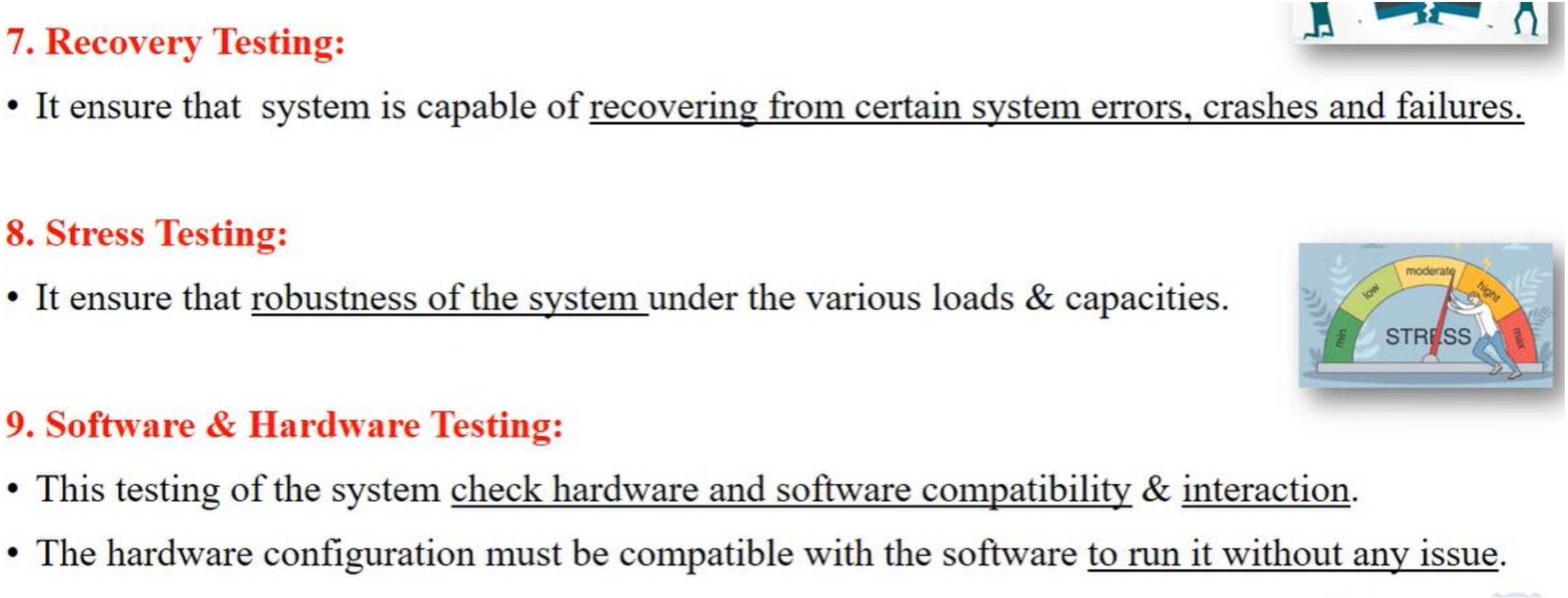
## 5. Migration Testing:

- It ensure that if system needs to be modified in new infrastructure so it should be modified without any issue.

## 6. Functional Testing:

- Tester find out any missing function in the system & make list of it.
- It can be added during functional testing and should improve quality of the system.

# Contd…



**7. Recovery Testing:**

- It ensure that  system is capable of recovering from certain system errors, crashes and failures.

**8. Stress Testing:**

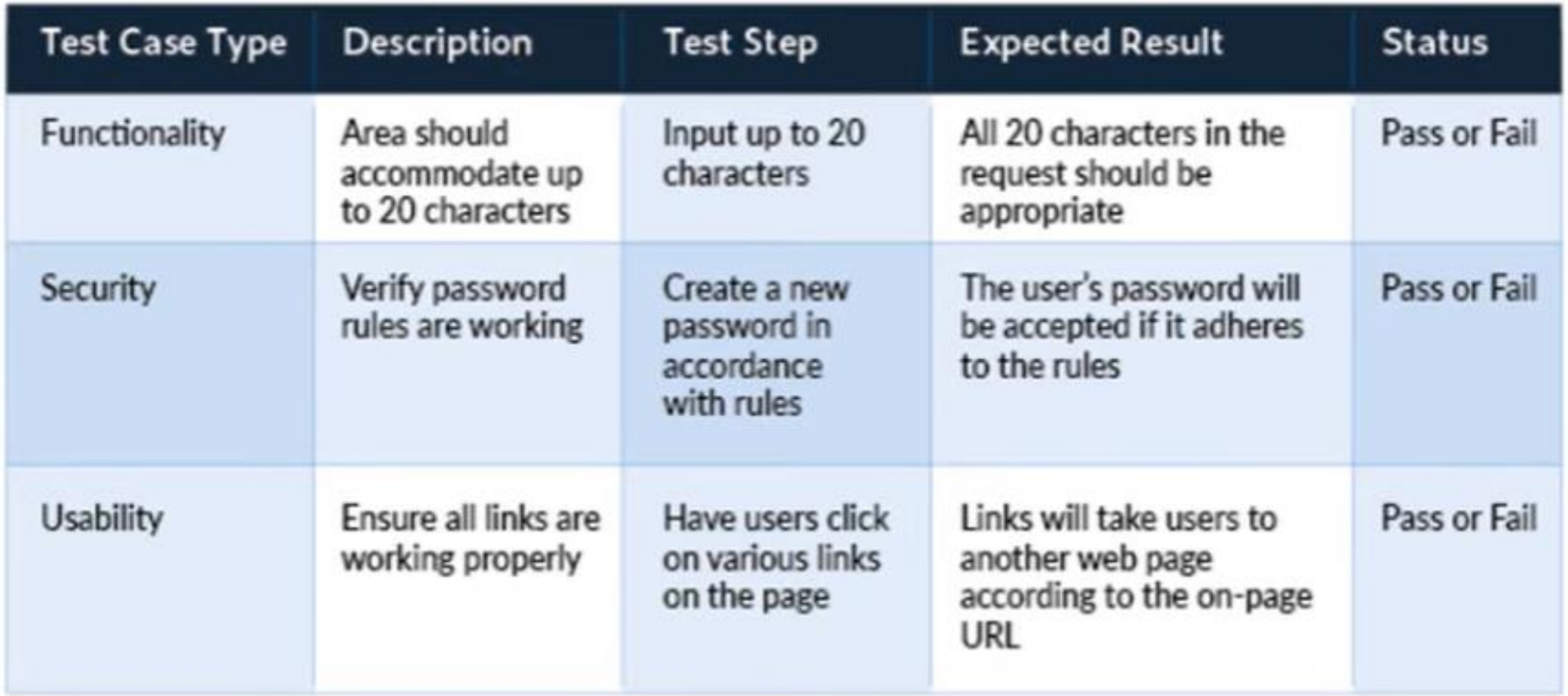
- It ensure that robustness of the system under the various loads & capacities.

min
low
moderate
hight
max
STRESS

**9. Software & Hardware Testing:**

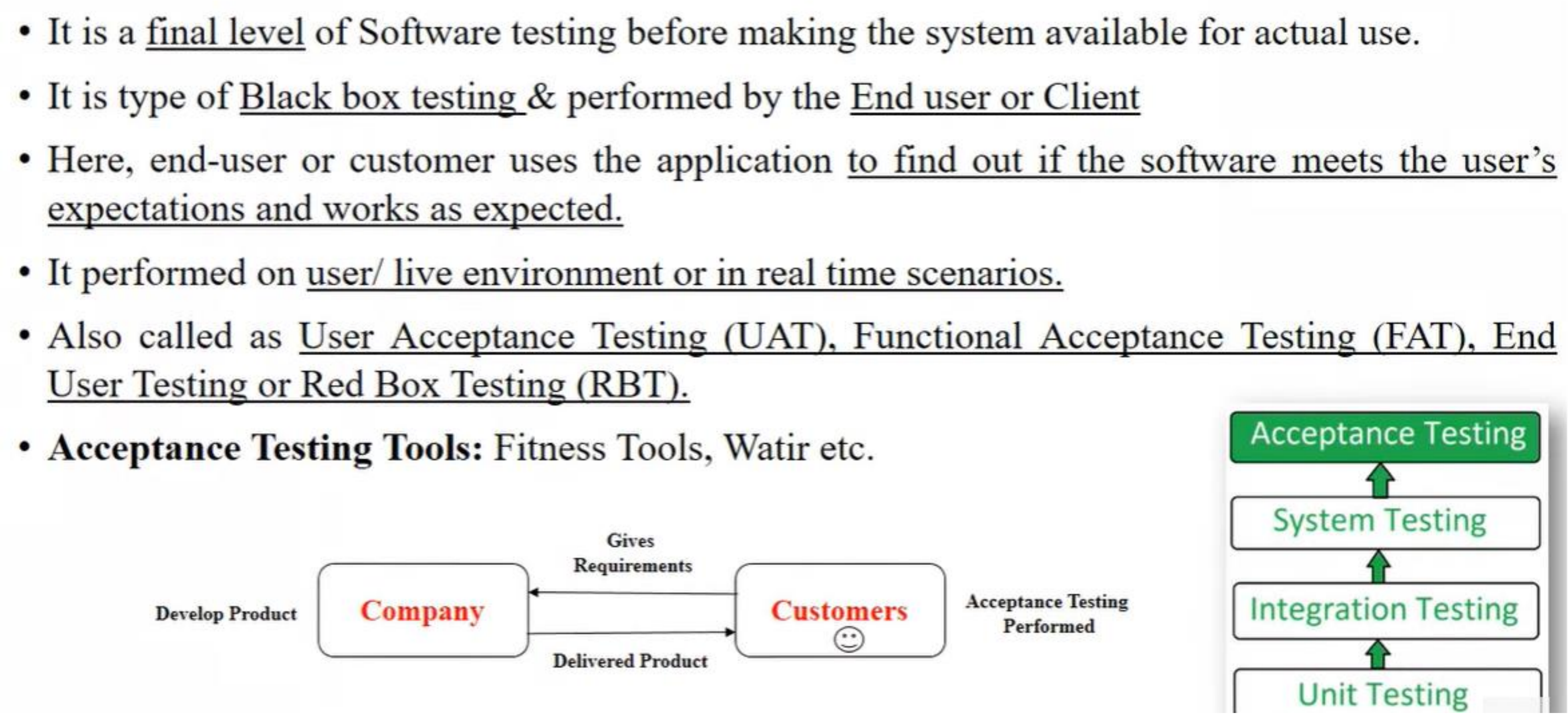
- This testing of the system check hardware and software compatibility & interaction.
- The hardware configuration must be compatible with the software to run it without any issue.

# System Testing - Example



| Test Case Type | Description | Test Step | Expected Result | Status |
|---|---|---|---|---|
| Functionality | Area should accommodate up to 20 characters | Input up to 20 characters | All 20 characters in the request should be appropriate | Pass or Fail |
| Security | Verify password rules are working | Create a new password in accordance with rules | The user's password will be accepted if it adheres to the rules | Pass or Fail |
| Usability | Ensure all links are working properly | Have users click on various links on the page | Links will take users to another web page according to the on-page URL | Pass or Fail |

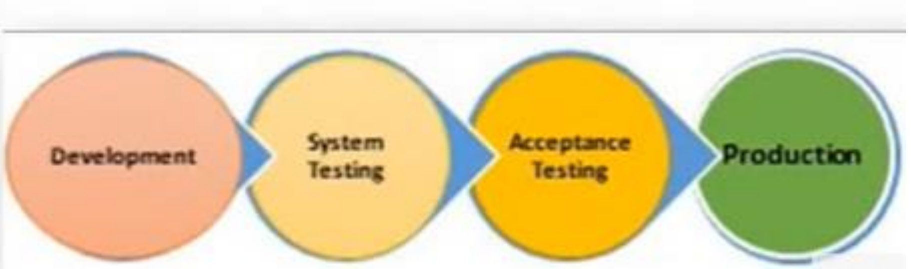
# Acceptance Testing



- It is a final level of Software testing before making the system available for actual use.
- It is type of Black box testing & performed by the End user or Client
- Here, end-user or customer uses the application to find out if the software meets the user's expectations and works as expected.
- It performed on user/ live environment or in real time scenarios.
- Also called as User Acceptance Testing (UAT), Functional Acceptance Testing (FAT), End User Testing or Red Box Testing (RBT).
- **Acceptance Testing Tools:** Fitness Tools, Watir etc.

# Importance of Acceptance Testing

1. Helps identify bugs that may have been left during the development process.
2. It helps to identify that the end product works according to client's or user's expectations or requirements.
3. It brings confidence and satisfaction to the clients as they are directly involved in the testing process.
4. Helps to deliver the end product without any bugs.
5. To satisfy all the functionalities that mentioned in SRS document.


Development

System Testing

Acceptance Testing

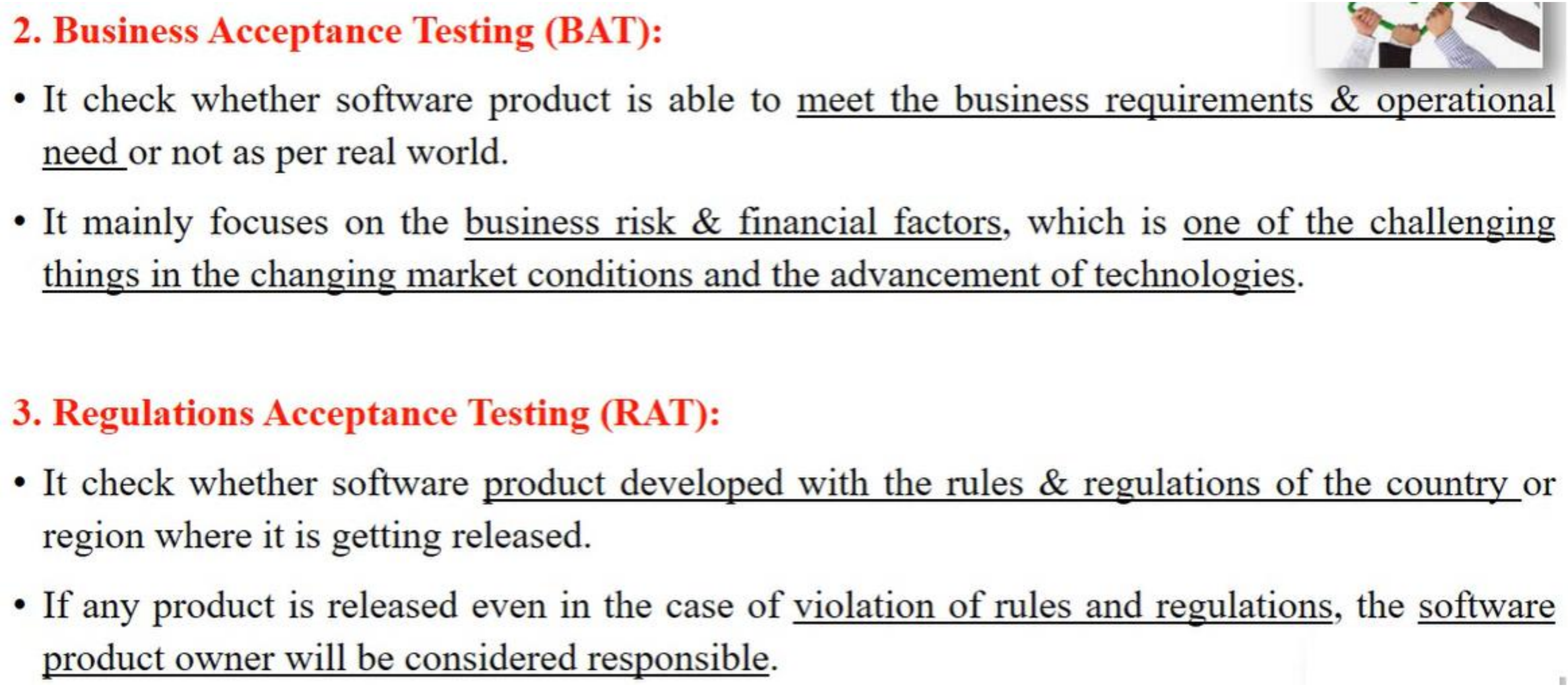
Production

# Types of Acceptance Testing

## 1. User Acceptance Testing (UAT):

- It performed from the end-user point of view.
- Client check whether the software product is working as per the requirements of the user correctly before moving it into the production environment.
- This testing is performed in a different production-like set-up environment.
- It does not focus on errors & bugs mainly focus on functionality required by the clients.
- This testing is also known as End- User- Testing.

# Contd…



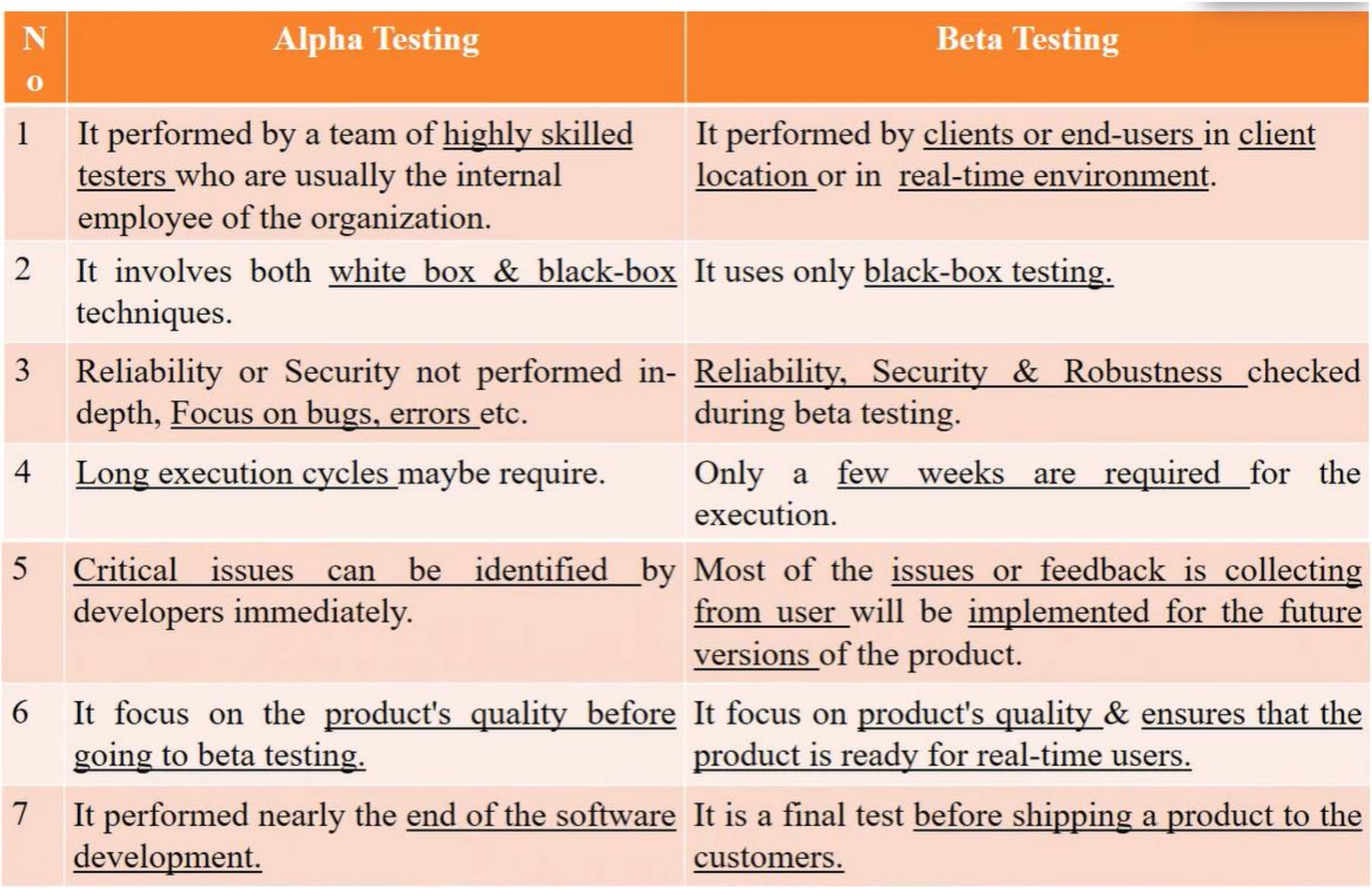
**2. Business Acceptance Testing (BAT):**

- It check whether software product is able to meet the business requirements & operational need or not as per real world.
- It mainly focuses on the business risk & financial factors, which is one of the challenging things in the changing market conditions and the advancement of technologies.

**3. Regulations Acceptance Testing (RAT):**

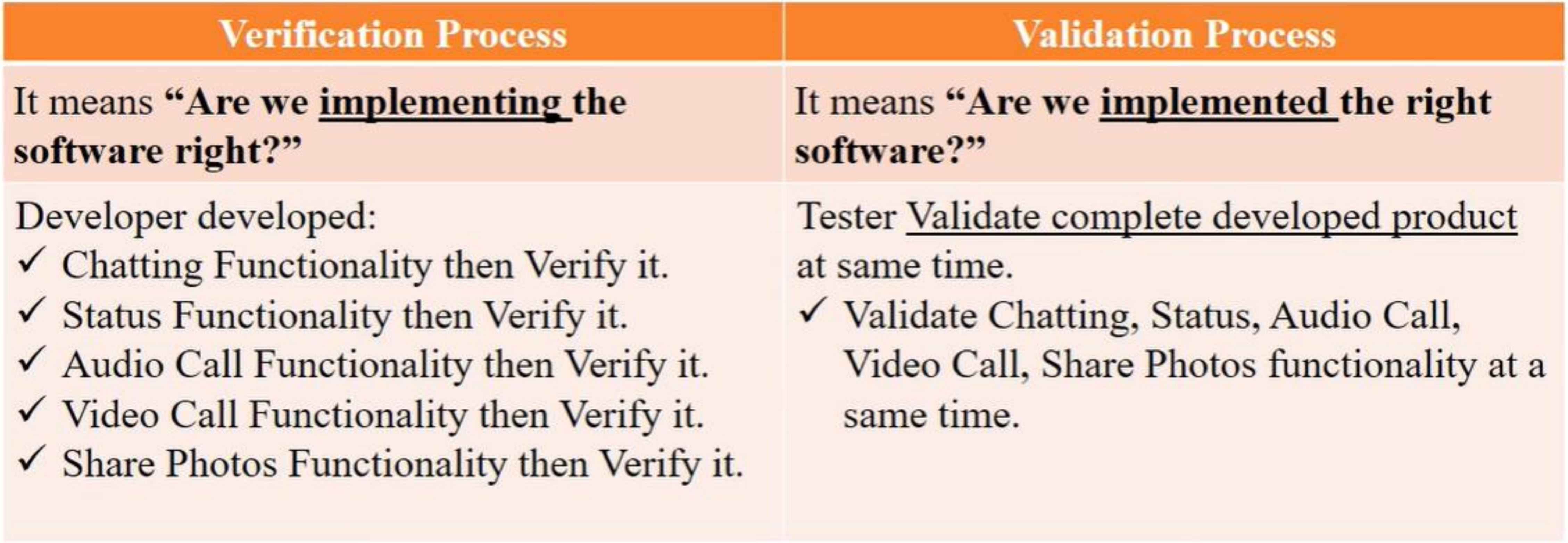
- It check whether software product developed with the rules & regulations of the country or region where it is getting released.
- If any product is released even in the case of violation of rules and regulations, the software product owner will be considered responsible.

Al



| No | Alpha Testing | Beta Testing |
|---|---|---|
| 1 | It performed by a team of highly skilled testers who are usually the internal employee of the organization. | It performed by clients or end-users in client location or in real-time environment. |
| 2 | It involves both white box & black-box techniques. | It uses only black-box testing. |
| 3 | Reliability or Security not performed in-depth, Focus on bugs, errors etc. | Reliability, Security & Robustness checked during beta testing. |
| 4 | Long execution cycles maybe require. | Only a few weeks are required for the execution. |
| 5 | Critical issues can be identified by developers immediately. | Most of the issues or feedback is collecting from user will be implemented for the future versions of the product. |
| 6 | It focus on the product's quality before going to beta testing. | It focus on product's quality & ensures that the product is ready for real-time users. |
| 7 | It performed nearly the end of the software development. | It is a final test before shipping a product to the customers. |

# Verification Vs Validation



| Verification Process | Validation Process |
|---|---|
| It means **“Are we implementing the software right?”** | It means **“Are we implemented the right software?”** |
| Developer developed:<br>✓ Chatting Functionality then Verify it.<br>✓ Status Functionality then Verify it.<br>✓ Audio Call Functionality then Verify it.<br>✓ Video Call Functionality then Verify it.<br>✓ Share Photos Functionality then Verify it. | Tester Validate complete developed product at same time.<br>✓ Validate Chatting, Status, Audio Call, Video Call, Share Photos functionality at a same time. |

| No | Verification | Validation |
|---|---|---|
| 1 | It means “Are we implementing the software right?” | It means “Are we implemented the right software?” |
| 2 | It comes before validation | It comes after verification |
| 3 | It is known as Static Testing. | It is known as Dynamic Testing. |
| 4 | It is executed by Quality assurance team or Developers. | It is executed by the Testing team. |
| 5 | It includes checking documents, design, code and program. | It includes testing and validating the actual product |
| 6 | Whether the software conforms to specification is checked | It checks whether the software meets the requirements and expectations of a customer |
| 7 | It does not involve executing the code | It always involves executing the code |

| No | Verification | Validation |
|---|---|---|
| 8 | It finds bugs early in the development cycle | It can find bugs that the verification process can not catch |
| 9 | It includes different methods like Inspections, Reviews & Walkthroughs. | It includes testing like Functional, System, Integration & User acceptance testing. |
| 10 | Developer can verify that the inputs follow the outputs or not. | Tester can validate that the user accepts the product or not. |

# Defect or Bug

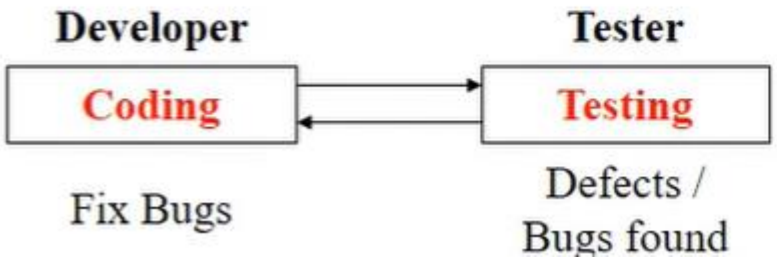
- A **Defect or Bug** is an errors in an application that is created during building or designing software and due to which software starts to show abnormal behaviors during its use.
- The variation between the actual results and expected results is known as **Defect**.

➤**Defect / Bug Life Cycle:**

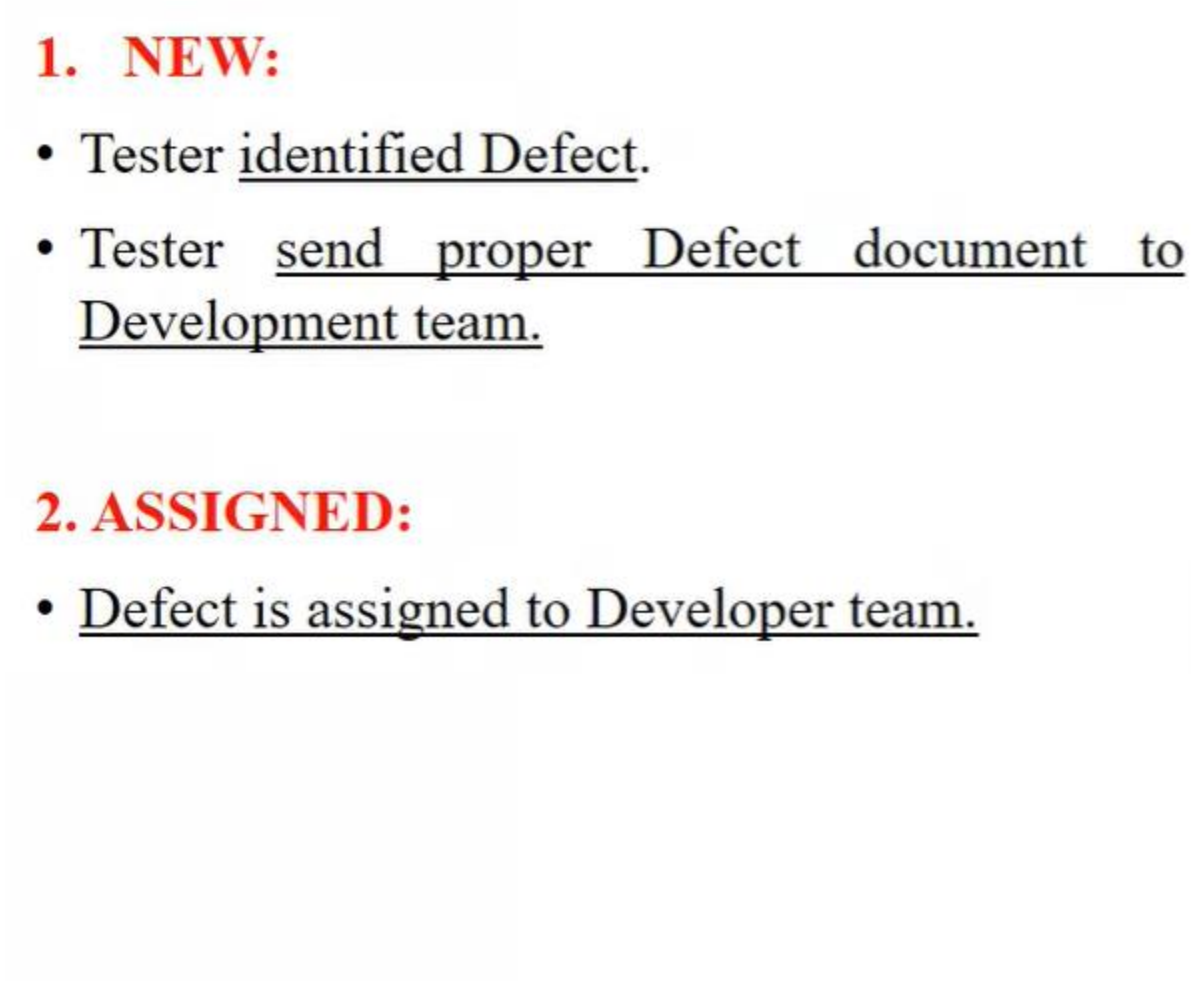
- In software testing is the specific set of states that defect or bug goes through in its entire life.
- The purpose of Defect life cycle is to easily coordinate and communicate current status of defect make the defect fixing process systematic and efficient.

➤**Performed by:** Developer & Tester

➤**Tools:** JIRA, Trac, Redmine etc.



# Defect Life Cycle

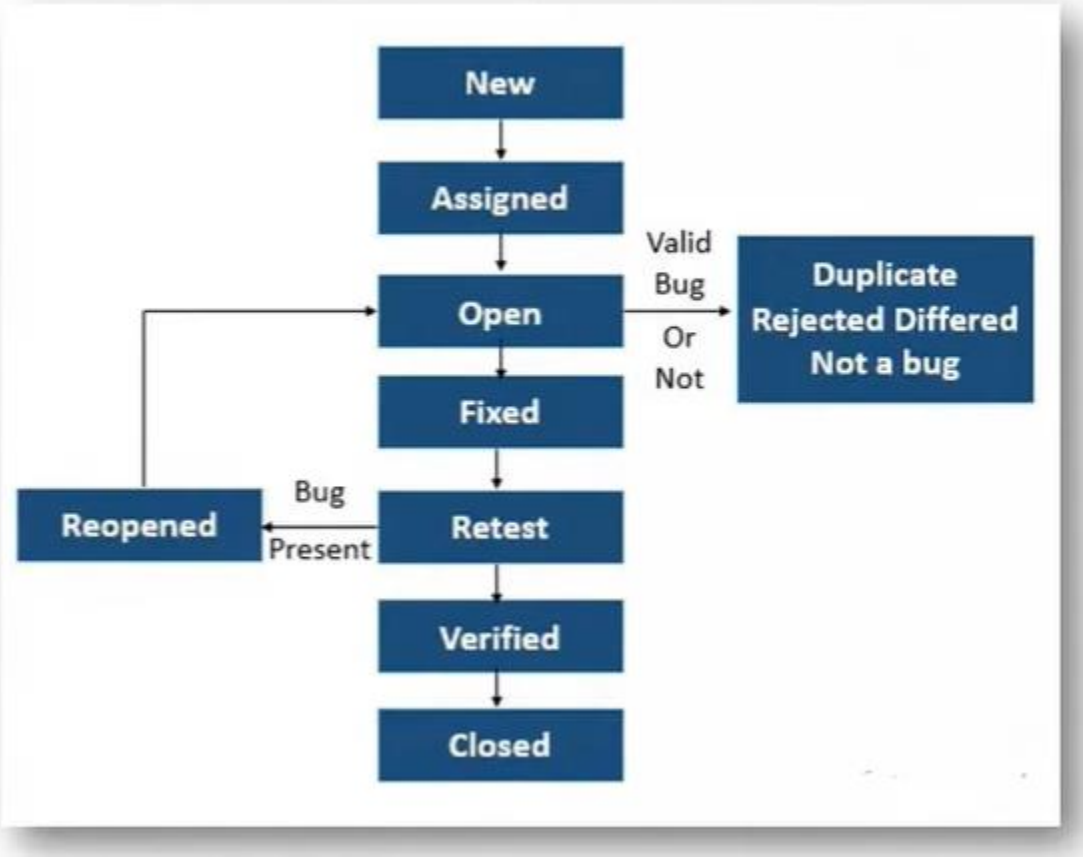


## 1. NEW:

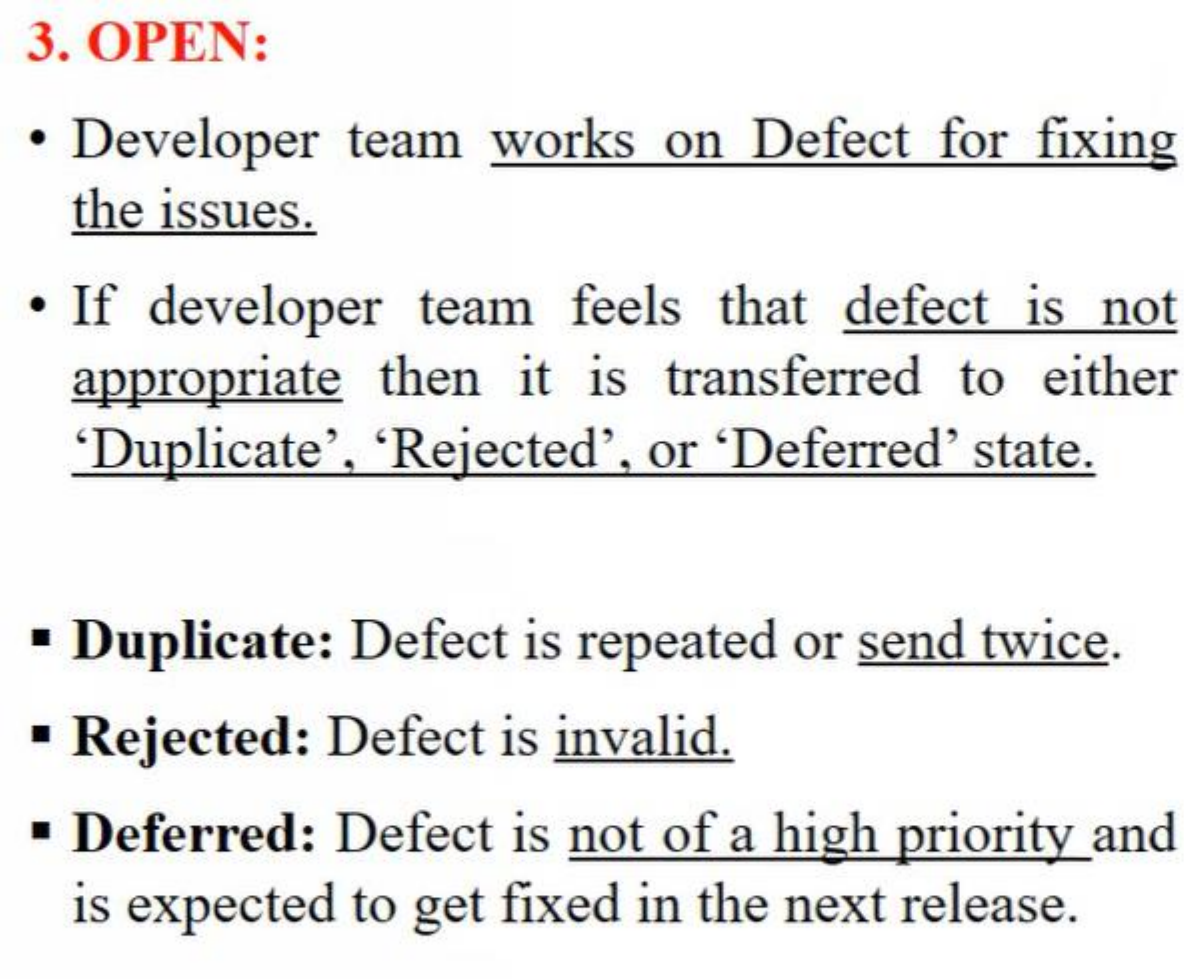
- Tester identified Defect.
- Tester send proper Defect document to Development team.

## 2. ASSIGNED:

- Defect is assigned to Developer team.



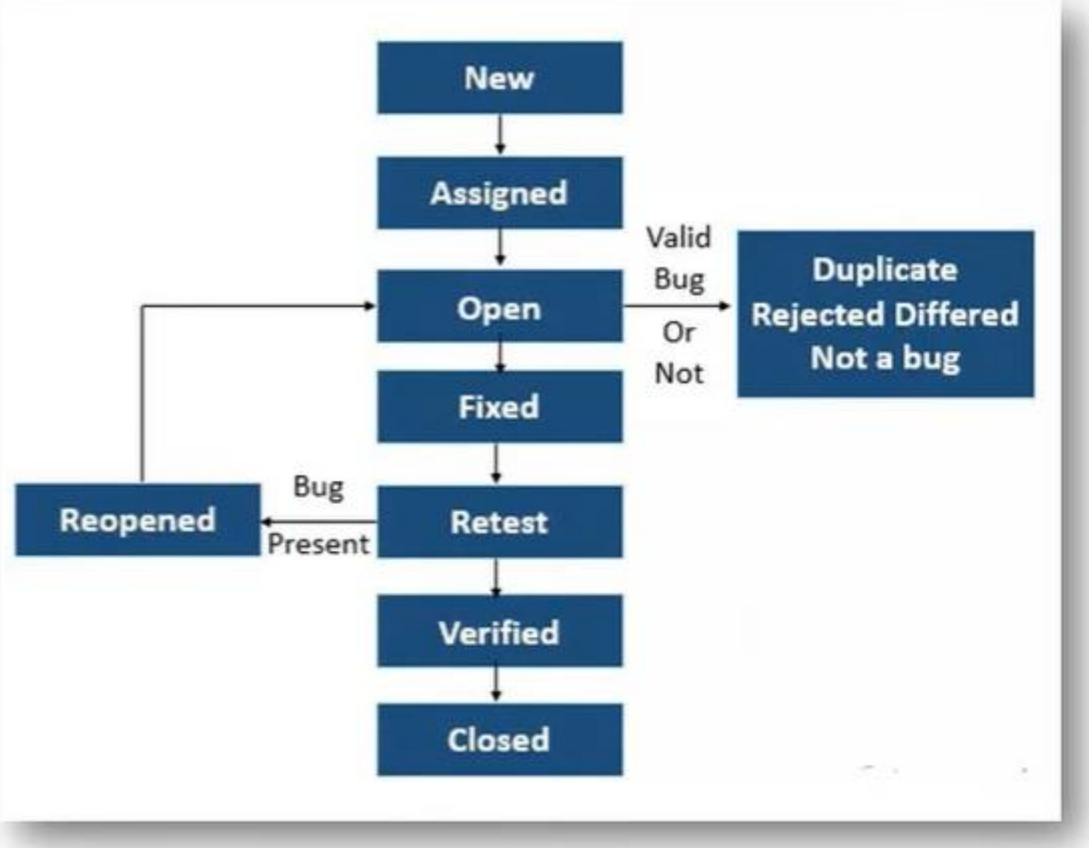
# Contd…



**3. OPEN:**

- Developer team works on Defect for fixing the issues.
- If developer team feels that defect is not appropriate then it is transferred to either 'Duplicate', 'Rejected', or 'Deferred' state.

- **Duplicate:** Defect is repeated or send twice.
- **Rejected:** Defect is invalid.
- **Deferred:** Defect is not of a high priority and is expected to get fixed in the next release.



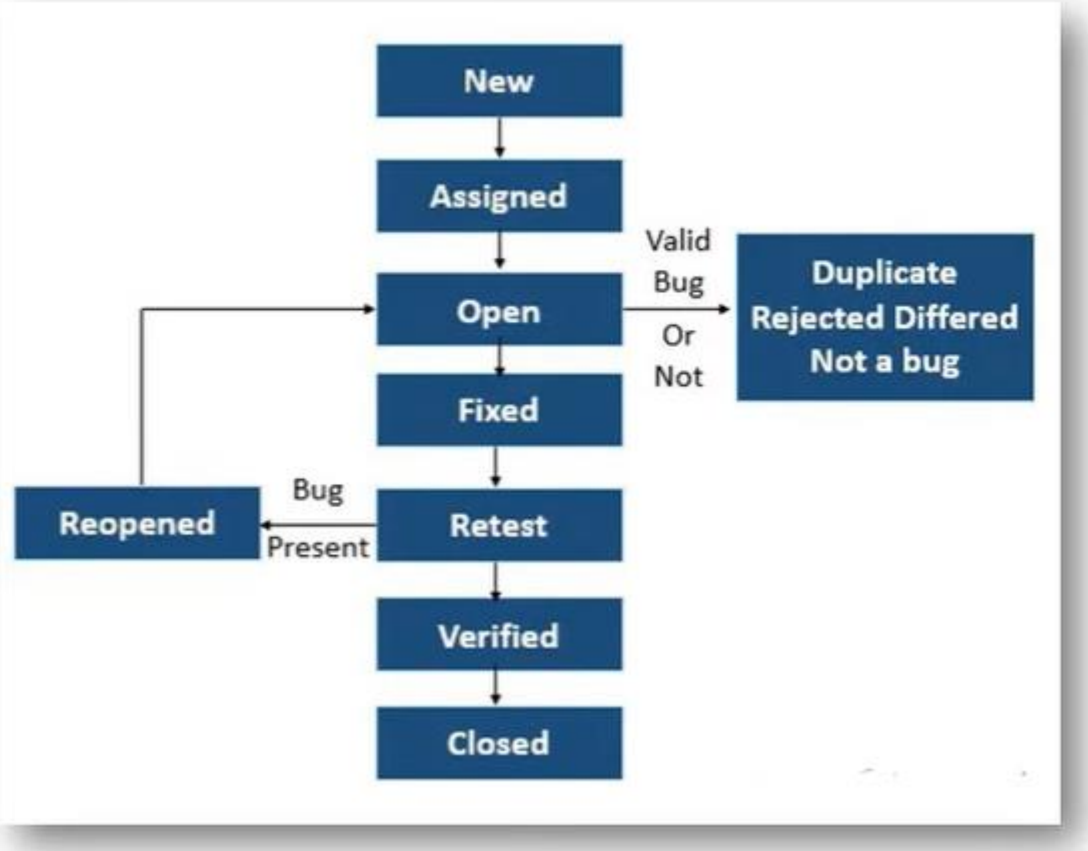
# Contd…

## 4. FIXED:

- Developer takes necessary actions on coding to fix the Defect.
- The defect will remove from the application.

## 5. RETEST:

- Testing team to check whether the bug has fixed by the developer or not.



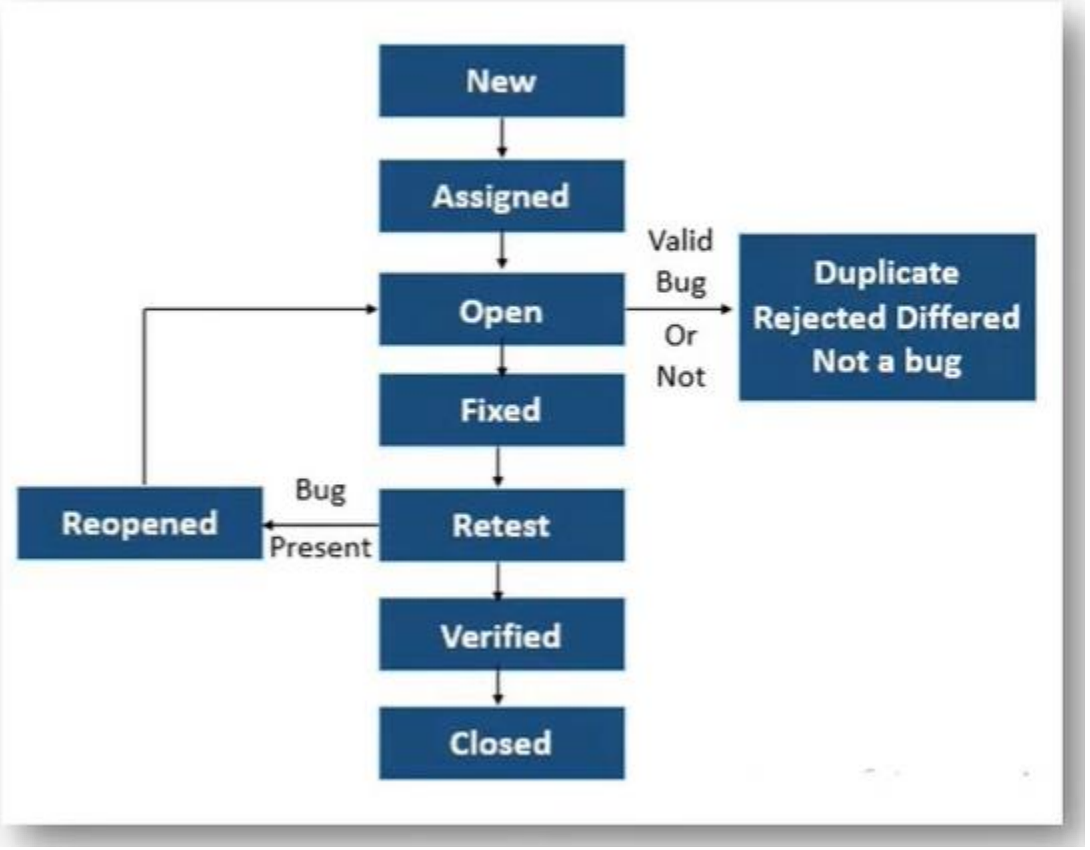
# Contd…

## 6. REOPENED:

- If Defect still exists even after Developer team has fixed the bug.
- The lifecycle restarts.

## 7. VERIFIED:

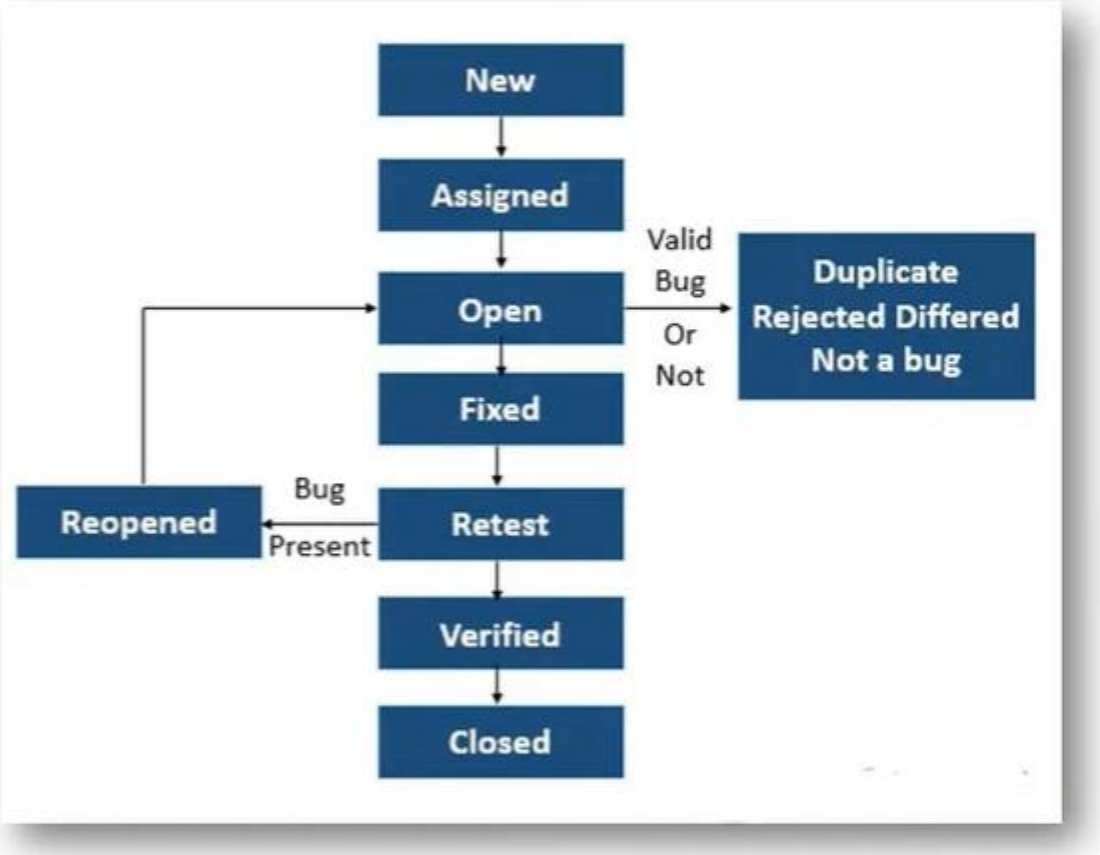
- The tester re-tests Defect after it got fixed by Developer team.
- If tester does not find any kind of Defect then status assigned is verified.



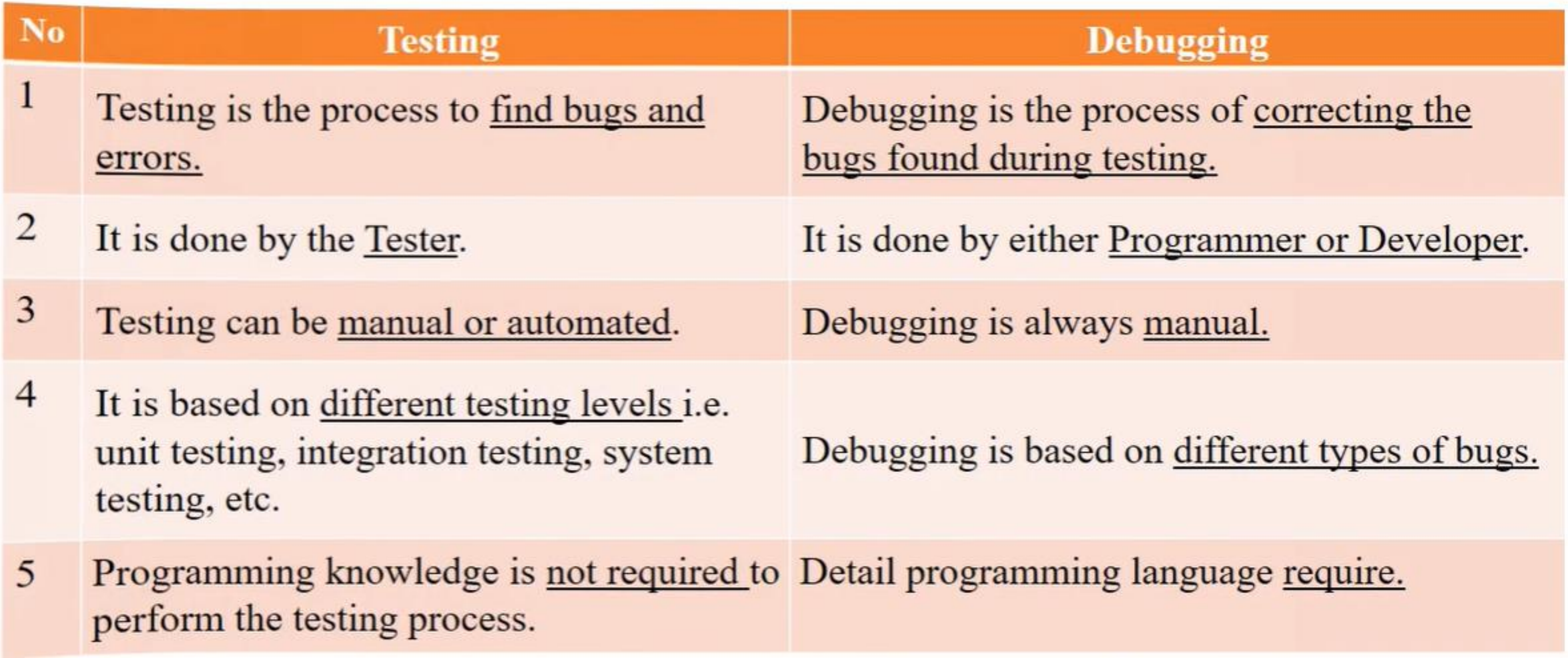
# Contd…

## 8. CLOSED:

- Once the Defect has been verified as fixed.
- The Testing team closes the issue.

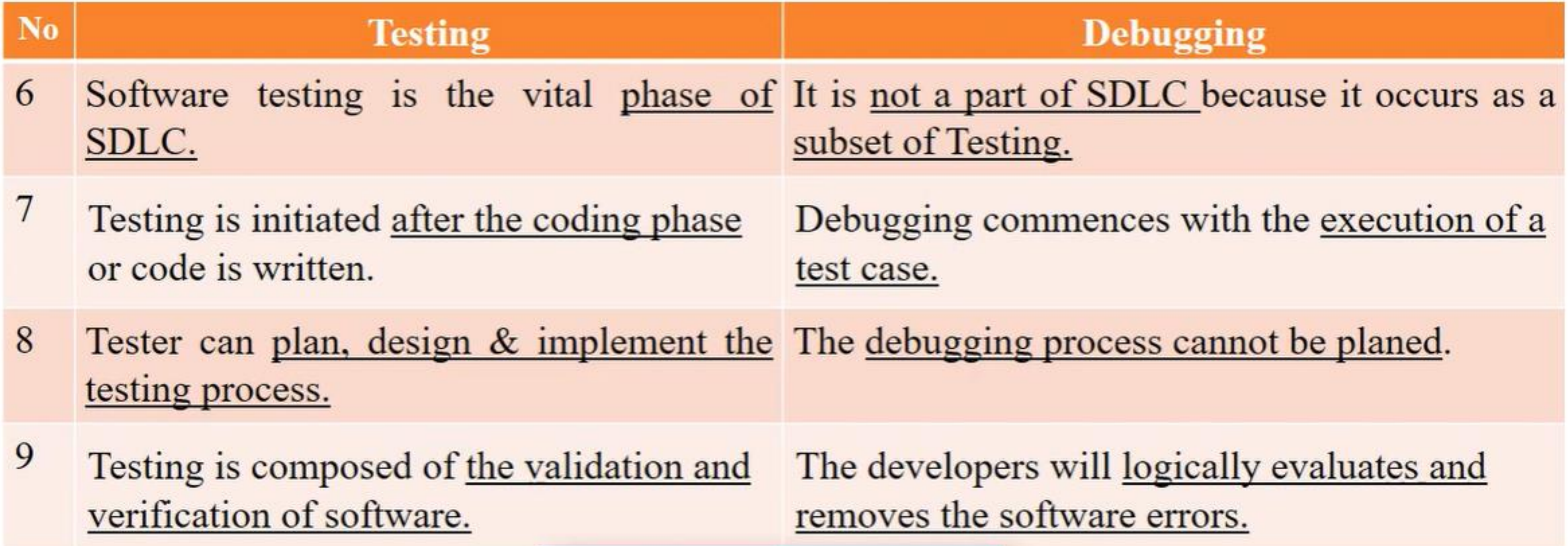


# Testing VS Debugging



| No | Testing | Debugging |
|---|---|---|
| 1 | Testing is the process to find bugs and errors. | Debugging is the process of correcting the bugs found during testing. |
| 2 | It is done by the Tester. | It is done by either Programmer or Developer. |
| 3 | Testing can be manual or automated. | Debugging is always manual. |
| 4 | It is based on different testing levels i.e. unit testing, integration testing, system testing, etc. | Debugging is based on different types of bugs. |
| 5 | Programming knowledge is not required to perform the testing process. | Detail programming language require. |

# Contd…



| No | Testing | Debugging |
|---|---|---|
| 6 | Software testing is the vital phase of SDLC. | It is not a part of SDLC because it occurs as a subset of Testing. |
| 7 | Testing is initiated after the coding phase or code is written. | Debugging commences with the execution of a test case. |
| 8 | Tester can plan, design & implement the testing process. | The debugging process cannot be planed. |
| 9 | Testing is composed of the validation and verification of software. | The developers will logically evaluates and removes the software errors. |

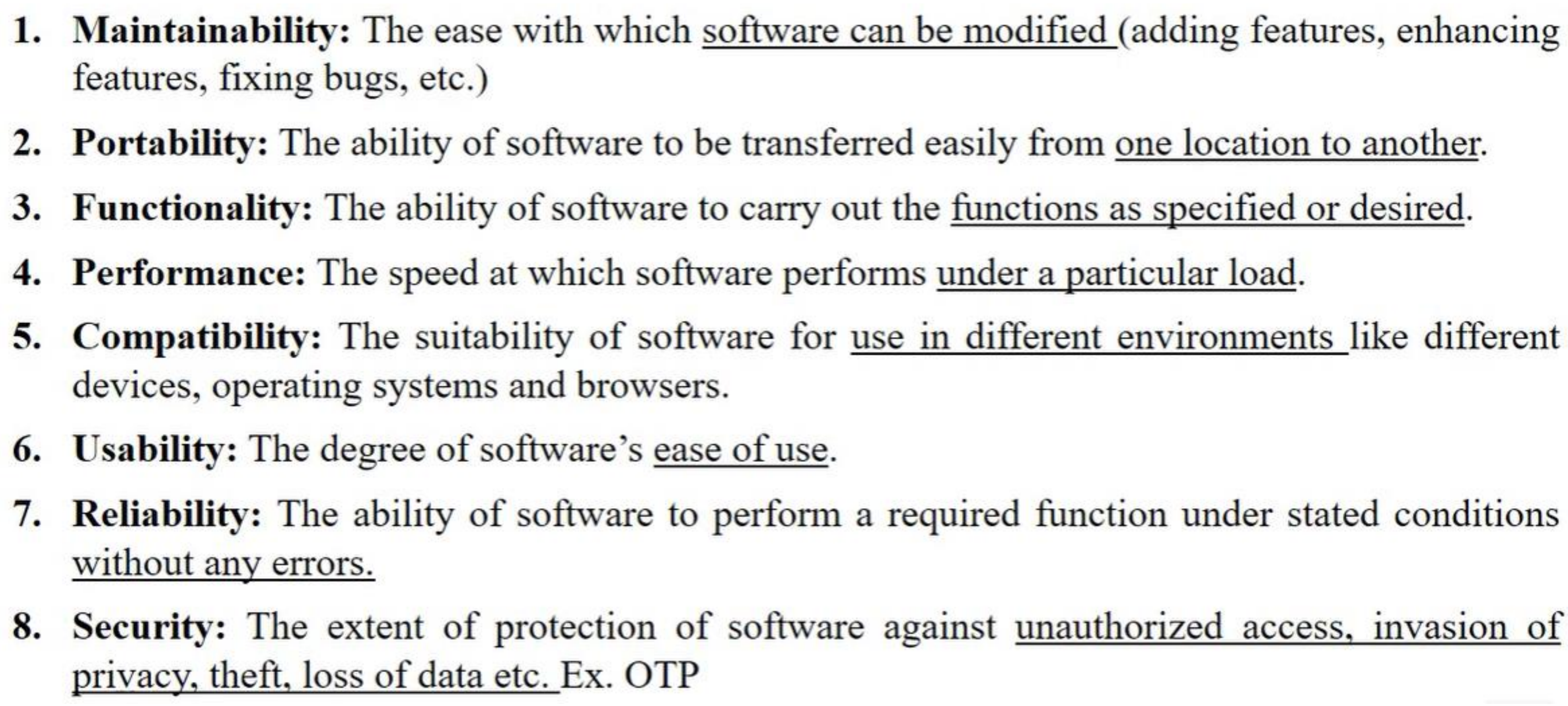
# Software Quality

- The quality of software can be defined as the ability of the software to function as per user requirement.
- When it comes to software products it must satisfy all the functionalities written down in the SRS document.

➢**Software Quality include:**

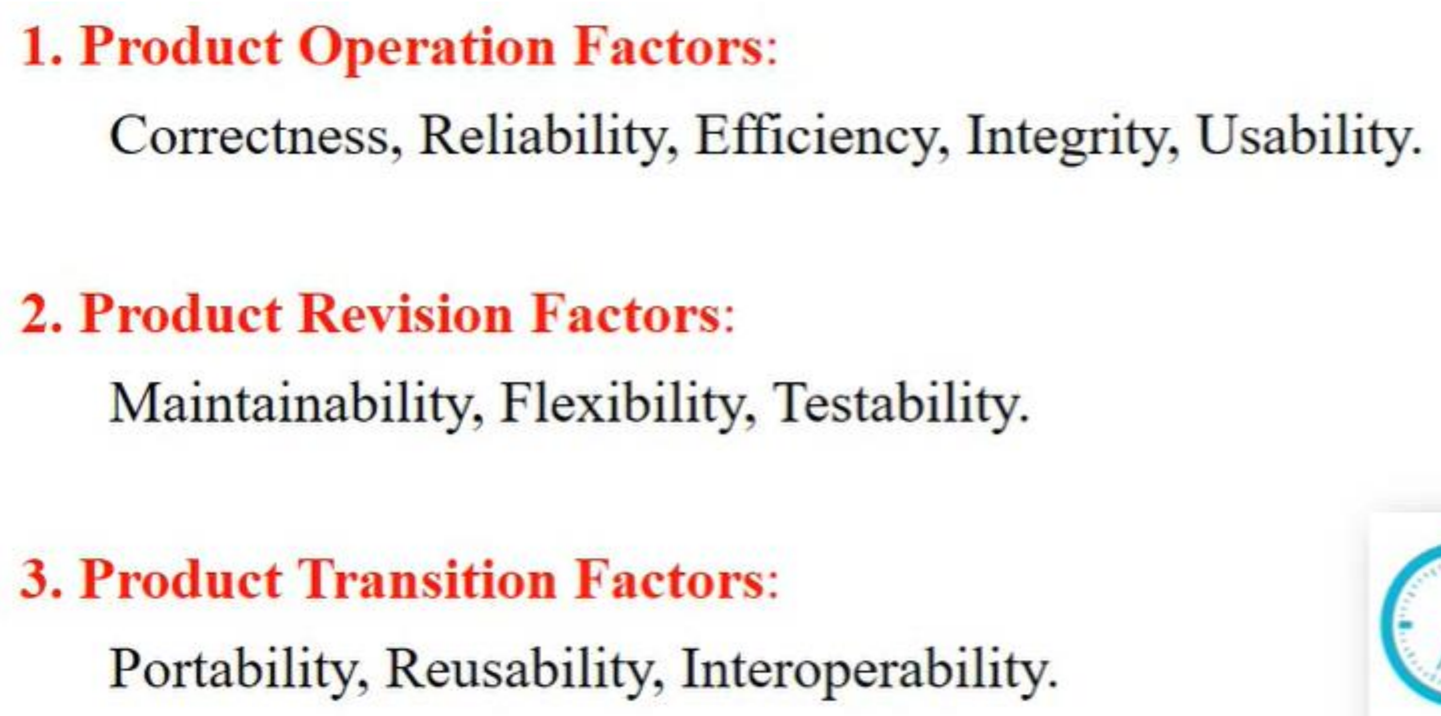
1. **Good Design:** Good visualization design to attract users.
2. **Durability:** The software work without any issue for a long period of time.
3. **Consistency:** Software perform consistently different platform and other devices.
4. **Maintainability:** Capture and fix bugs quickly. New features are added easily.
5. **Value for money:** Customer & companies who make this app should feel that the money spent on this app has not to waste.

# Software quality dimensions/parameters



1. **Maintainability:** The ease with which software can be modified (adding features, enhancing features, fixing bugs, etc.)
2. **Portability:** The ability of software to be transferred easily from one location to another.
3. **Functionality:** The ability of software to carry out the functions as specified or desired.
4. **Performance:** The speed at which software performs under a particular load.
5. **Compatibility:** The suitability of software for use in different environments like different devices, operating systems and browsers.
6. **Usability:** The degree of software's ease of use.
7. **Reliability:** The ability of software to perform a required function under stated conditions without any errors.
8. **Security:** The extent of protection of software against unauthorized access, invasion of privacy, theft, loss of data etc. Ex. OTP

# Factors affect on software quality



**1. Product Operation Factors:**

Correctness, Reliability, Efficiency, Integrity, Usability.

**2. Product Revision Factors:**

Maintainability, Flexibility, Testability.

**3. Product Transition Factors:**

Portability, Reusability, Interoperability.

# Software quality metrics

❑SQM ensures that the software product is of highest quality and standard.

### 1. Customer Problem Metrics:

- Measuring the problems encountered by the customers while using the product.
- PUM =Total problems reported by a customer + Total number of license months

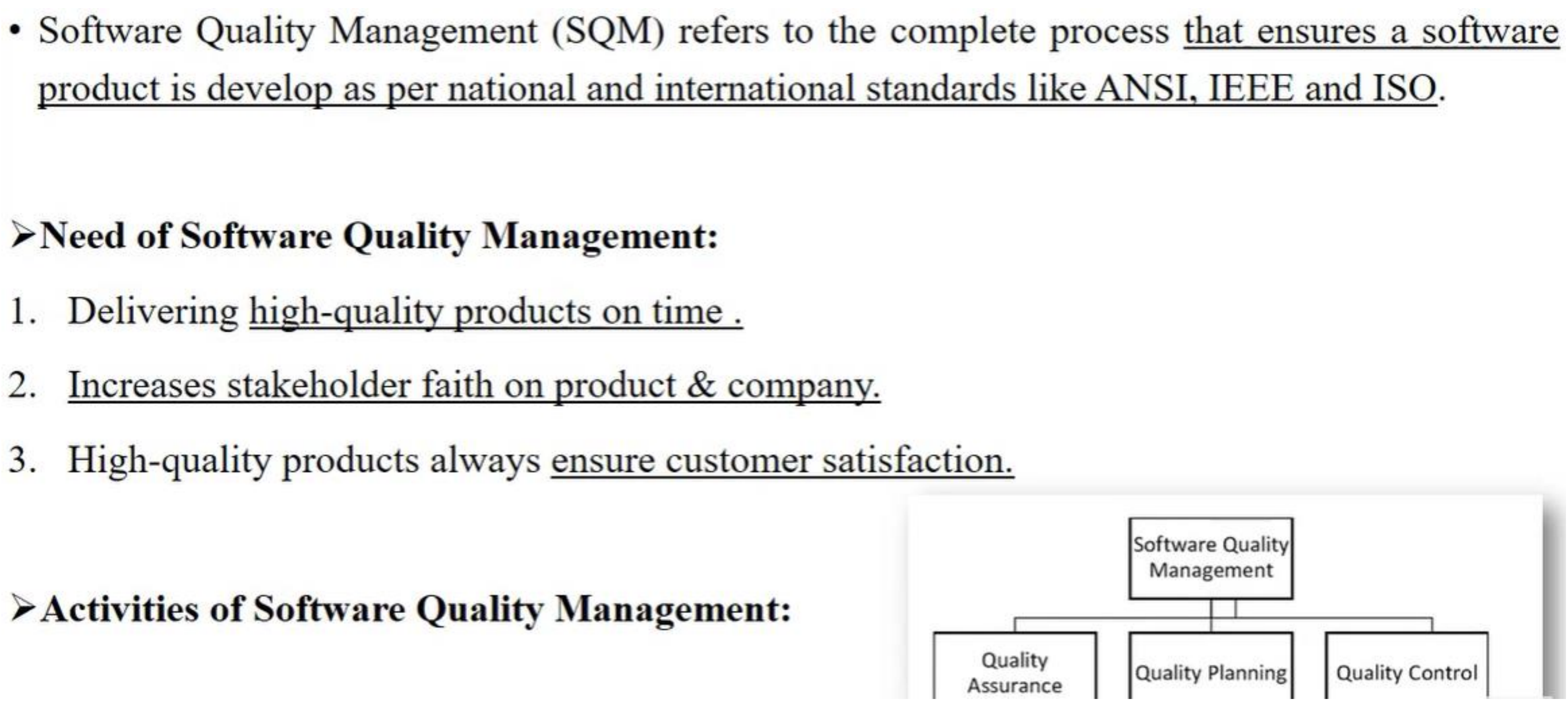
### 2. Customer Satisfaction Metrics:

- It deals with overall quality of product & how much a customer is satisfied with that product.
- It is measured by Very Satisfied, Satisfied, Neutral, Dissatisfied, Very Dissatisfied.

### 3. Software Maintenance Metrics:

- After completion of Development & Testing product release in market.
- During this interval, How many defect arrived at customer environment?

# Software quality management



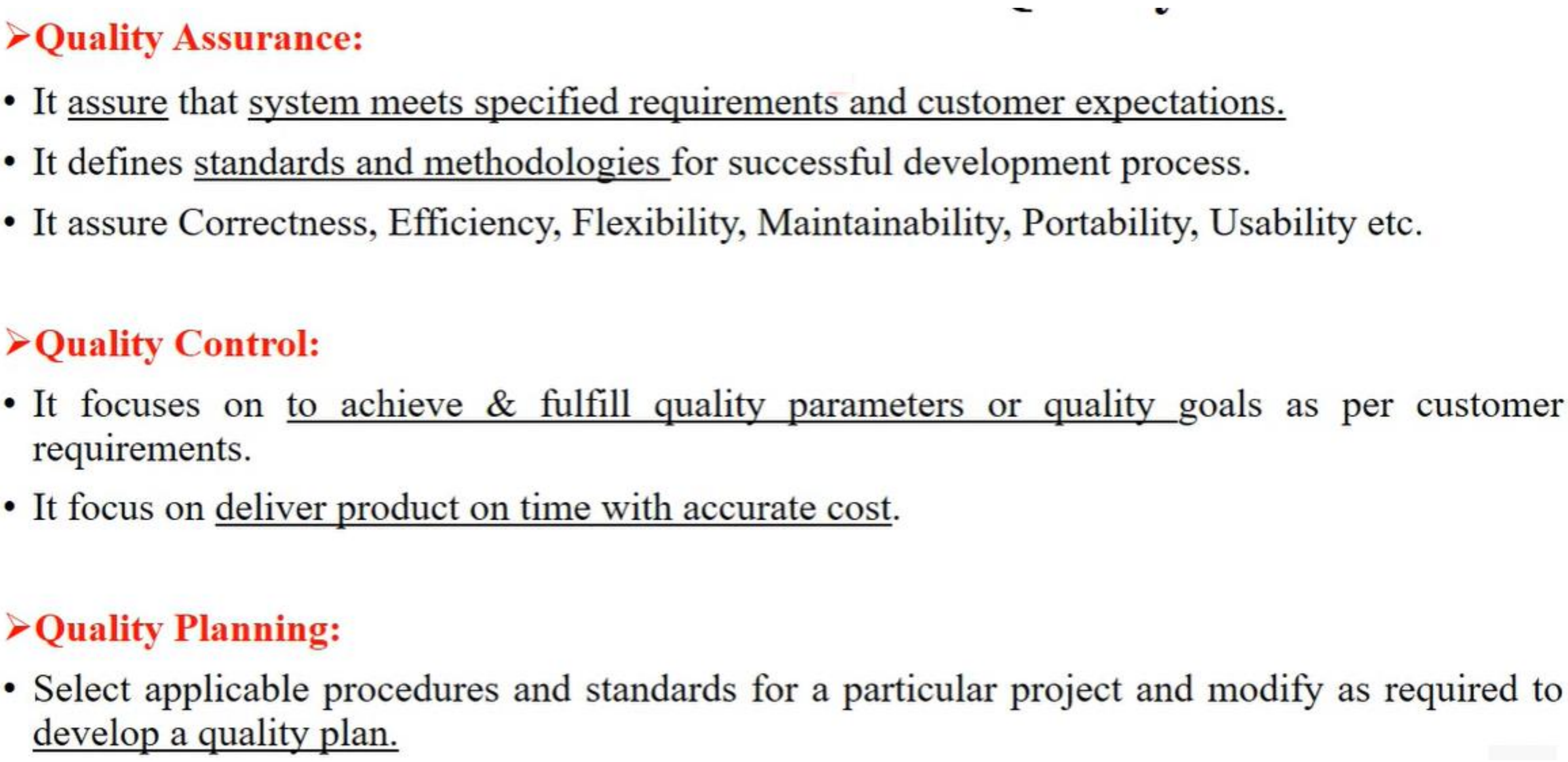
- Software Quality Management (SQM) refers to the complete process that ensures a software product is develop as per national and international standards like ANSI, IEEE and ISO.

➢**Need of Software Quality Management:**

1. Delivering high-quality products on time .
2. Increases stakeholder faith on product & company.
3. High-quality products always ensure customer satisfaction.

➢**Activities of Software Quality Management:**

# How to achieve software quality?



### ➢Quality Assurance:

- It assure that system meets specified requirements and customer expectations.
- It defines standards and methodologies for successful development process.
- It assure Correctness, Efficiency, Flexibility, Maintainability, Portability, Usability etc.

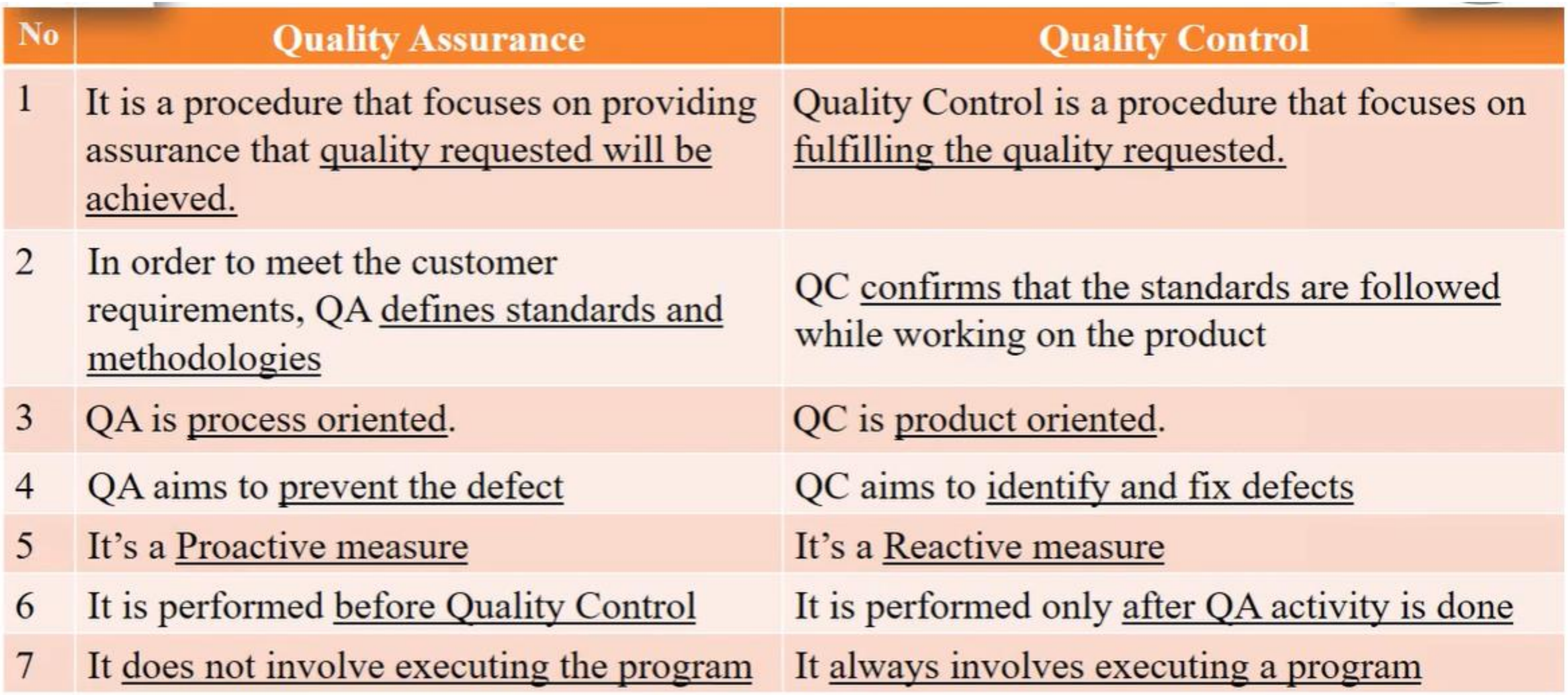
### ➢Quality Control:

- It focuses on to achieve & fulfill quality parameters or quality goals as per customer requirements.
- It focus on deliver product on time with accurate cost.

### ➢Quality Planning:

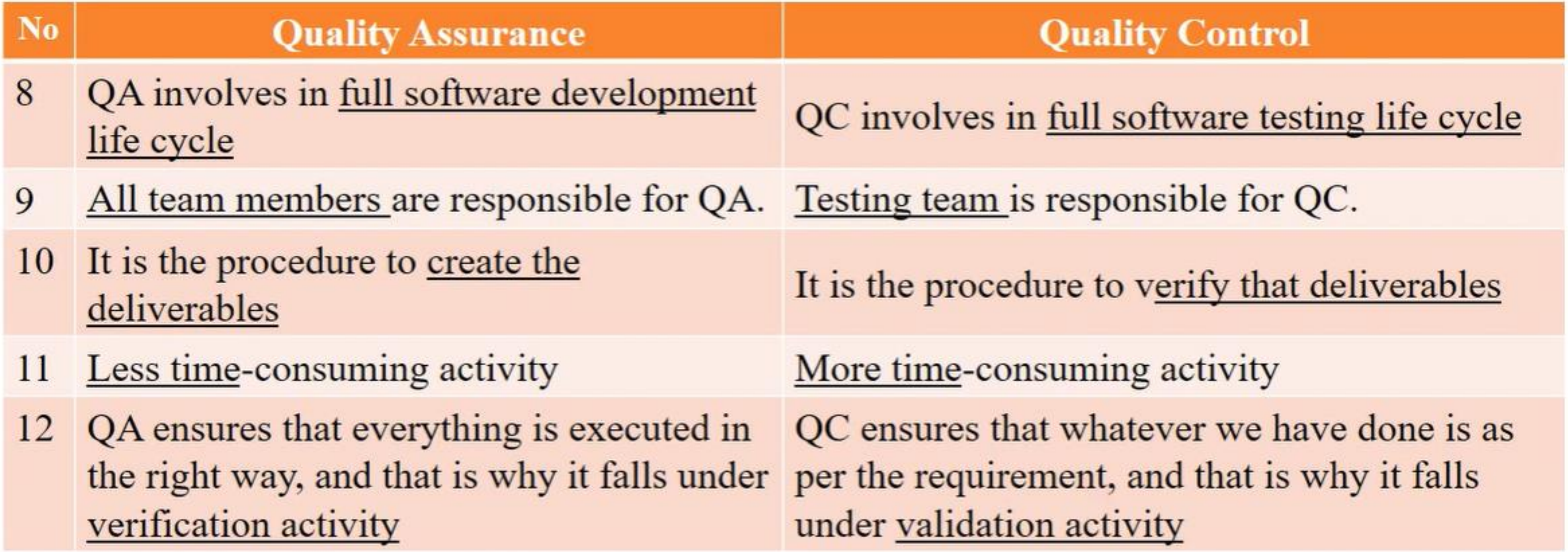
- Select applicable procedures and standards for a particular project and modify as required to develop a quality plan.

# Quality Assurance VS Quality Control



| No | Quality Assurance | Quality Control |
|---|---|---|
| 1 | It is a procedure that focuses on providing assurance that quality requested will be achieved. | Quality Control is a procedure that focuses on fulfilling the quality requested. |
| 2 | In order to meet the customer requirements, QA defines standards and methodologies | QC confirms that the standards are followed while working on the product |
| 3 | QA is process oriented. | QC is product oriented. |
| 4 | QA aims to prevent the defect | QC aims to identify and fix defects |
| 5 | It’s a Proactive measure | It’s a Reactive measure |
| 6 | It is performed before Quality Control | It is performed only after QA activity is done |
| 7 | It does not involve executing the program | It always involves executing a program |

# Contd…



| No | Quality Assurance | Quality Control |
|---|---|---|
| 8 | QA involves in full software development life cycle | QC involves in full software testing life cycle |
| 9 | All team members are responsible for QA. | Testing team is responsible for QC. |
| 10 | It is the procedure to create the deliverables | It is the procedure to verify that deliverables |
| 11 | Less time-consuming activity | More time-consuming activity |
| 12 | QA ensures that everything is executed in the right way, and that is why it falls under verification activity | QC ensures that whatever we have done is as per the requirement, and that is why it falls under validation activity |

Thank YOU!!!