LEVEL OF TESTING:

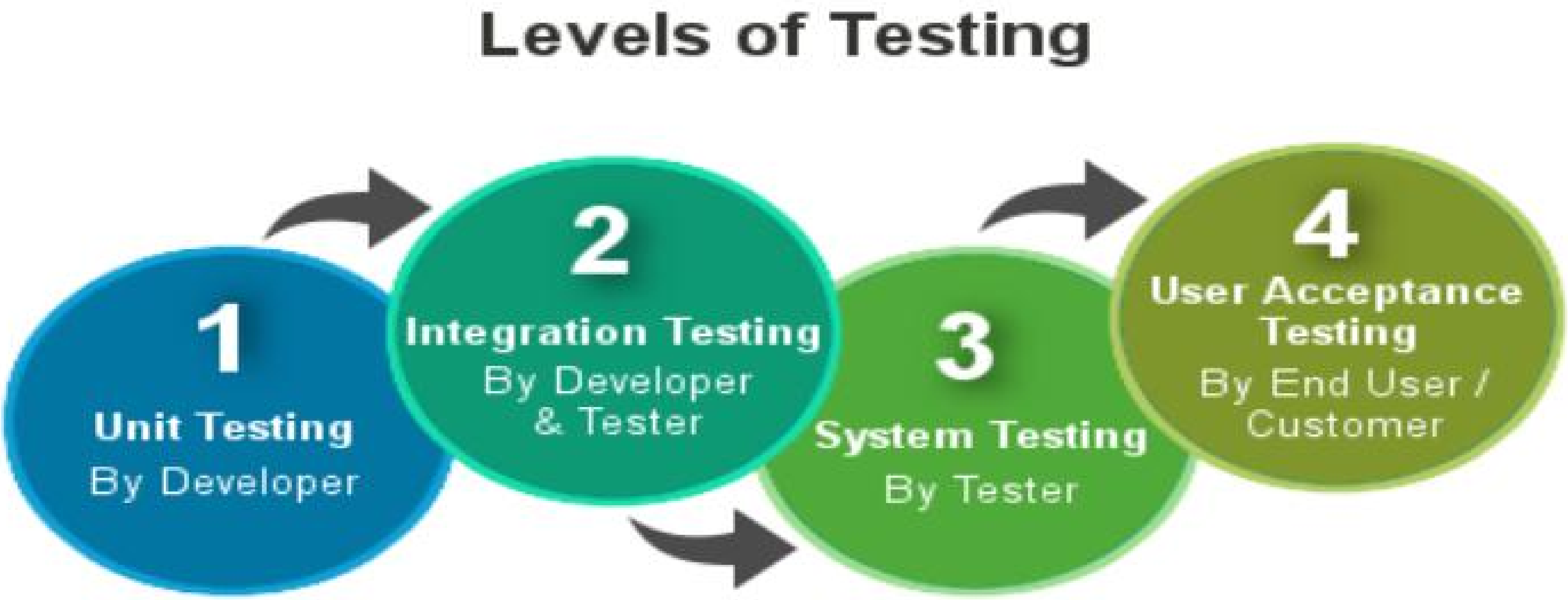
In order to detect an error, we will implement software testing; therefore, all the errors can be removed to find a product with more excellent quality.

DIFFERENT LEVEL OF TESTING:

The levels of software testing involve the different methodologies, which can be used while we are performing the software testing.

In Software Testing, we have four different levels of testing, which are as discussed below:

1. **Unit Testing**
2. **Integration Testing**
3. **System Testing**
4. **Acceptance Testing**



UNIT TESTING/COMPONENT TESTING:

- Unit testing is when every module of the application gets tested respectively.

- Unit testing is done by the developer himself. After he has written code for a feature, he will ensure it is working fine.
- Unit tests are the smallest testable component of the application.
- Nowadays we have Junit, Pytest, and TestNg frameworks for unit testing the application.

INTERGRATION TESTING:

- Integration testing is a testing technique where two or more independent components are tested together.
- Integration testing is done by the developer. Here test cases are written to ensure the data flowing between them is correct.
- For example, testing the signup form where UI validations are correct, data reaching API, and getting stored are all validated.
- Integration testing is done when the application is still developing to find bugs early on in the development process.

SYSTEM TESTING:

- System testing is done by the tester where the entire application is tested as a single unit.
- Hence, system testing test cases are also performance test cases, load testing, and stress testing test cases.
- System testing is done to find the errors which might have been overlooked during unit or integration testing.
- System testing evaluates both functional and non-functional test cases.

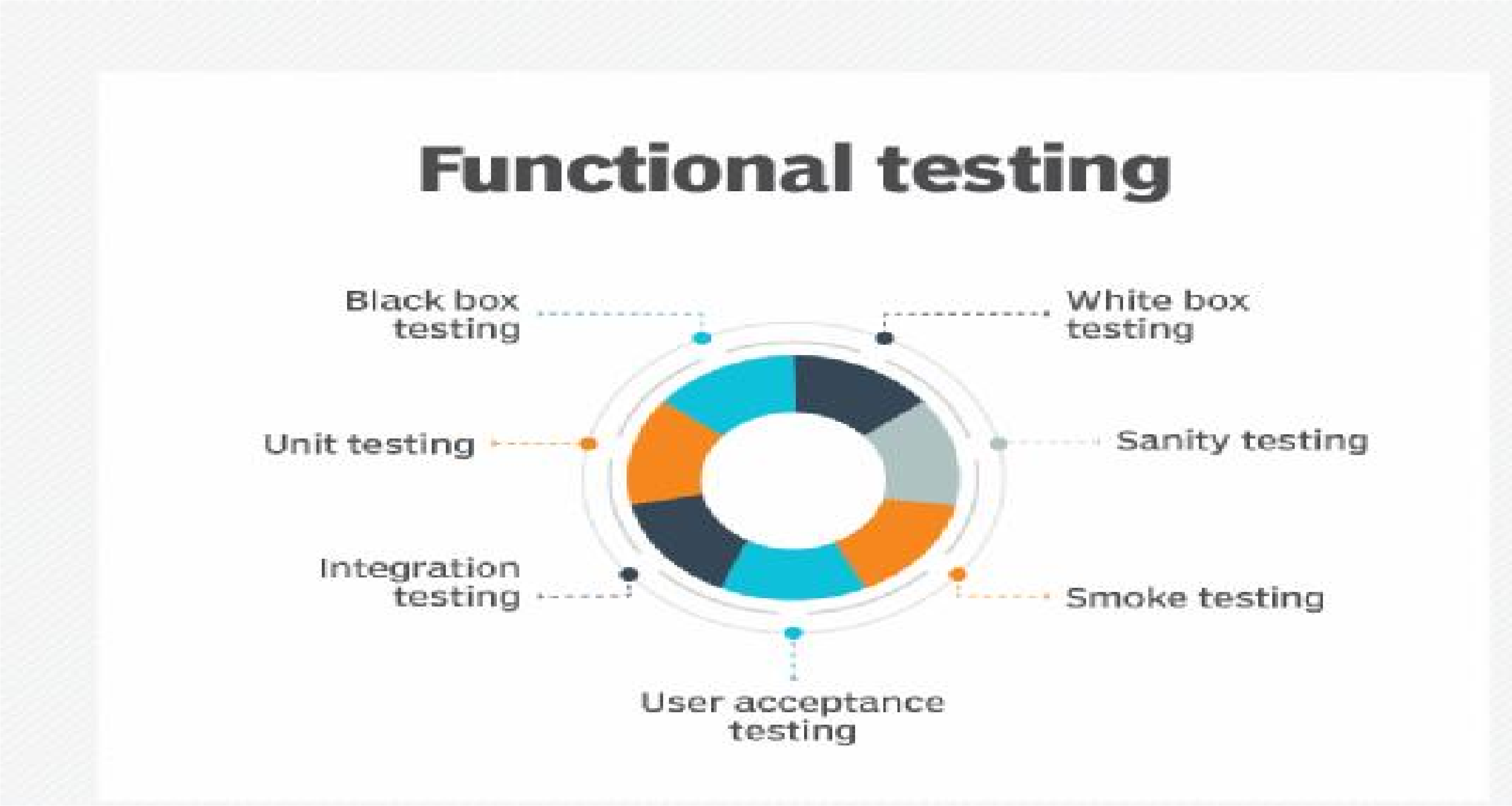
ACCEPTANCE TESTING:

- Acceptance testing is done by the client where he evaluates whether the product is made by the requirement he listed out.
- Acceptance testing is done at the UAT server where a well-tested product is deployed by the team for the client's reference so he can track ongoing changes in the project
- There is a defined acceptance criterion that is laid at the time of requirement listing so that the client can validate that the product is meeting the acceptance criteria.
- Once the client completes acceptance testing the product goes to production where users can use the final application.

FUNCTIONAL TESTING:

Functional testing is a type of testing which verifies that each **function** of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing, and it is not concerned about the source code of the application.

Every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results. This testing involves checking of User Interface, APIs, Database, security, client/ server applications and functionality of the Application Under Test. The testing can be done either manually or using automation.



TYPE OF TESTING IN FUNCTIONAL :

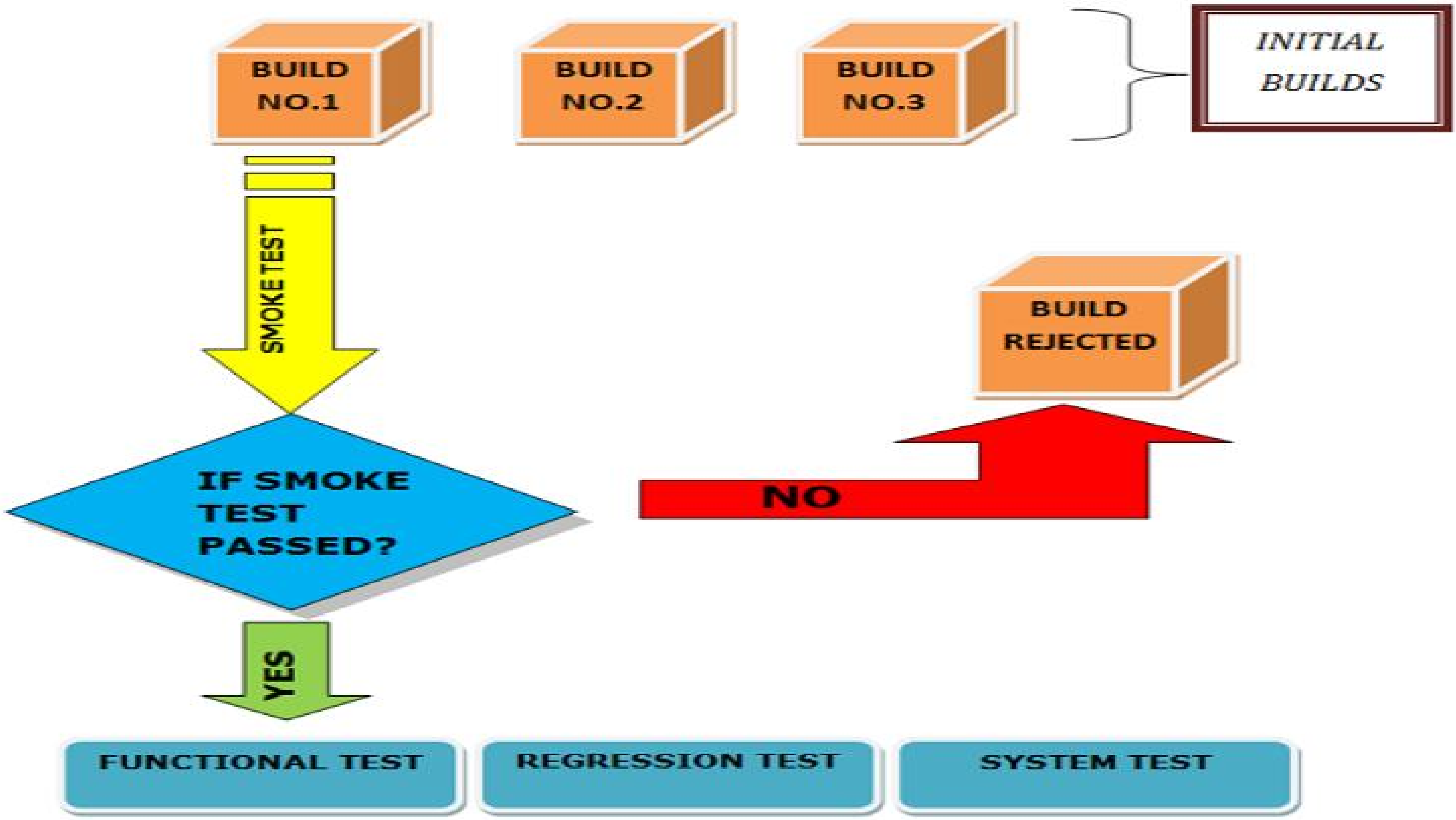
- Unit testing
- Smoke testing
- Sanity Testing
- User Acceptance
- Integration Testing
- Regression testing
- Localization
- Globalization
- Interoperability

SANITY TESTING:

Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

SMOKE TESTING:

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as “Build Verification Testing” or “Confidence Testing.”


BUILD NO.1
BUILD NO.2
BUILD NO.3
INITIAL BUILDS
SMOKE TEST
BUILD REJECTED
IF SMOKE TEST PASSED?
NO
YES
FUNCTIONAL TEST
REGRESSION TEST
SYSTEM TEST

USER ACCEPTANCE TESTING:

User acceptance testing (UAT), also called *application testing* or *end-user testing*, is a phase of software development in which the software is tested in the real world by its intended audience. UAT is often the last phase of the software testing process and is performed before the tested software is released to its intended market. The goal of UAT is to ensure software can handle real-world tasks and perform up to development specifications.

Types of UAT

Multiple types of software tests qualify as user acceptance testing. These tests include the following:

- **Beta testing.** The software is given to groups of end users who evaluate it for its intended purpose and provide feedback to developers for improvements.
- **Black box testing.** An end user tests specific software functions without seeing the internal code.

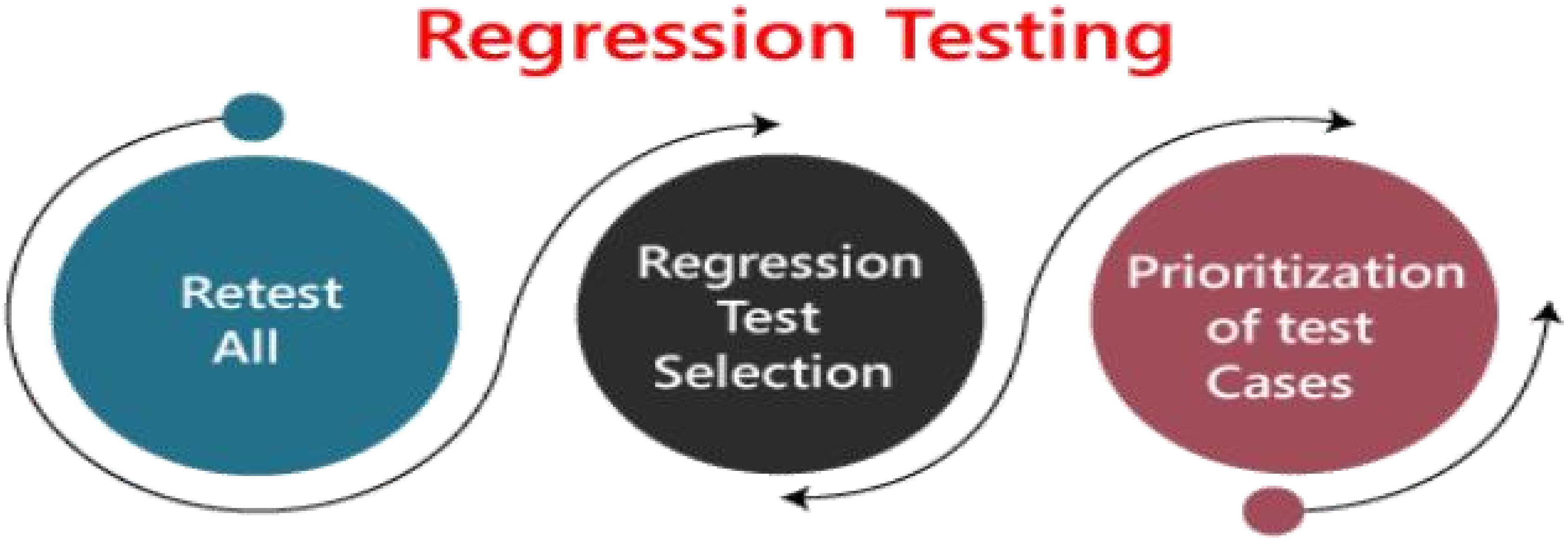
REGRESSION TESTING:

Regression testing is a black box testing techniques. It is used to authenticate a code change in the software does not impact the existing functionality of the product. Regression testing is making sure that the product works fine with new functionality, bug fixes, or any change in the existing feature.

Regression testing is a type of software testing. Test cases are re-executed to check the previous functionality of the application is working fine, and the new changes have not produced any bugs.

Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged.

Regression tests are also known as the Verification Method. Test cases are often automated. Test cases are required to execute many times and running the same test case again and again manually, is time-consuming and tedious too.



LOCALIZATION TESTING:

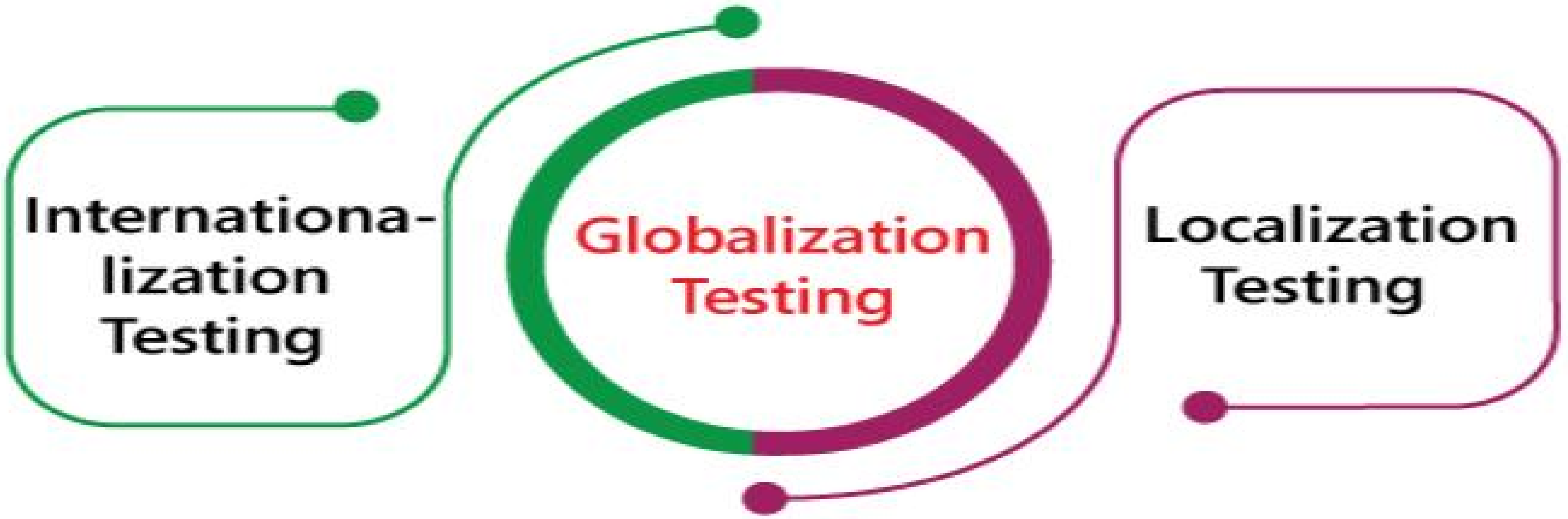
Localization Testing is a software testing technique in which the behavior of a software is tested for a specific region, locale or culture. The purpose of doing localization testing for a software is to test appropriate linguistic and cultural aspects for a particular locale. It is the process of customizing the software as per the targeted language and country. The major area affected by localization testing includes **content and UI.**

It is a process of testing a globalized application whose UI, default language, currency, date, time format, and documentation are designed as per the targeted country or region. It ensures that the application is capable enough for using in that particular country.

GLOBALIZATION TESTING:

Globalization testing is another type of software testing which is used to test the software that is developed for multiple languages, is called **globalization testing,** and improving the application or software for various languages is known as **globalization**.

This testing ensures that the application will support multiple languages and multiple features because, in current scenarios, we can see the enhancement in several technologies as the applications are planned in such a way that it is used globally.



Interoperability Testing:

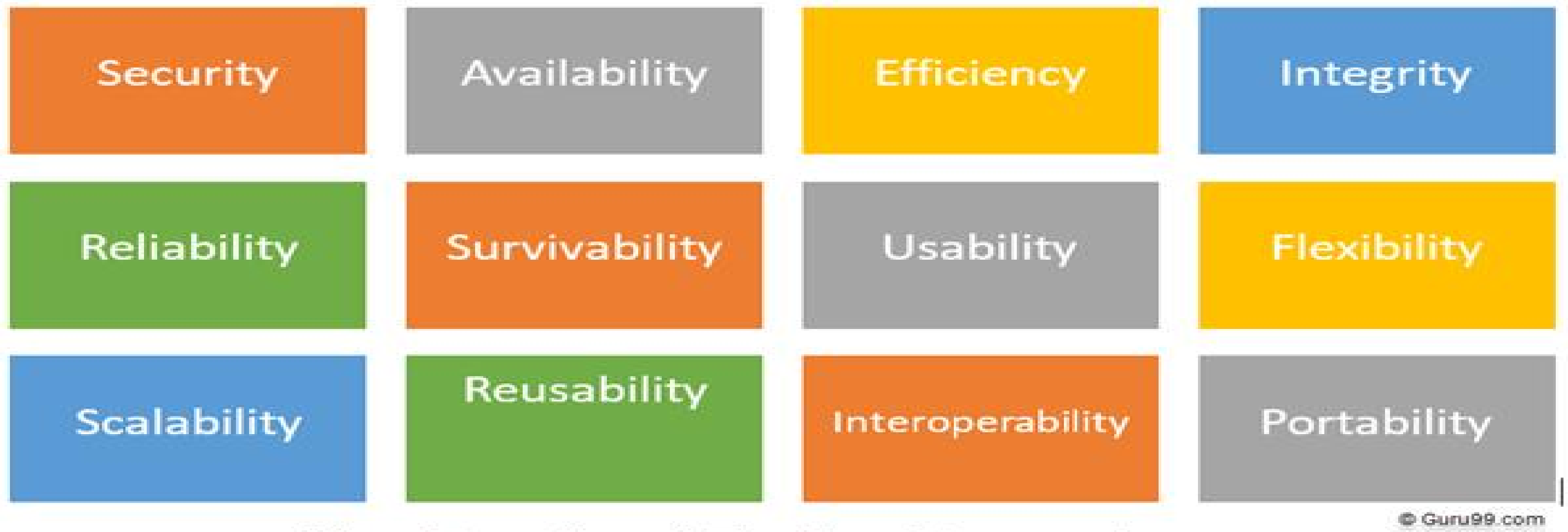
Interoperability Testing is a software testing type, that checks whether the software can interact with other software components and systems. The purpose of Interoperability tests is to ensure that the software product is able to communicate with other components or devices without any compatibility issues.

NON FUNCTIONALITY TESTING:

Non-functional testing is a type of testing to check non-functional aspects (performance, usability, reliability, etc.) of a software application. It is explicitly designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

A good example of non-functional test would be to check how many people can simultaneously login into a software.

Non-functional testing is equally important as functional testing and affects client satisfaction.



Security	Availability	Efficiency	Integrity
Reliability	Survivability	Usability	Flexibility
Scalability	Reusability	Interoperability	Portability

© Guru99.com

Non Functional Testing Parameters

NON FUNCTIONAL TESTING TYPE:

- Performance Testing
- Volume Testing
- Scalability
- Usability Testing
- Load Testing
- Stress Testing
- Compliance Testing
- Portability Testing
- Disaster Recover Testing

PERFORMANCE TESTING:

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. The process incorporates “performance” indicators such as:

- Browser, page, and network response times
- Server request processing times
- Acceptable concurrent user volumes
- Processor memory consumption; number and type of errors that might be encountered with app

Performance testing gathers all the tests that verify an application's speed, robustness, reliability, and correct sizing. It examines several indicators such as a browser, page and network response times, server query processing time, number of acceptable concurrent users architected, CPU memory consumption, and number/type of errors which may be encountered when using an application.

VOLUME TESTING:

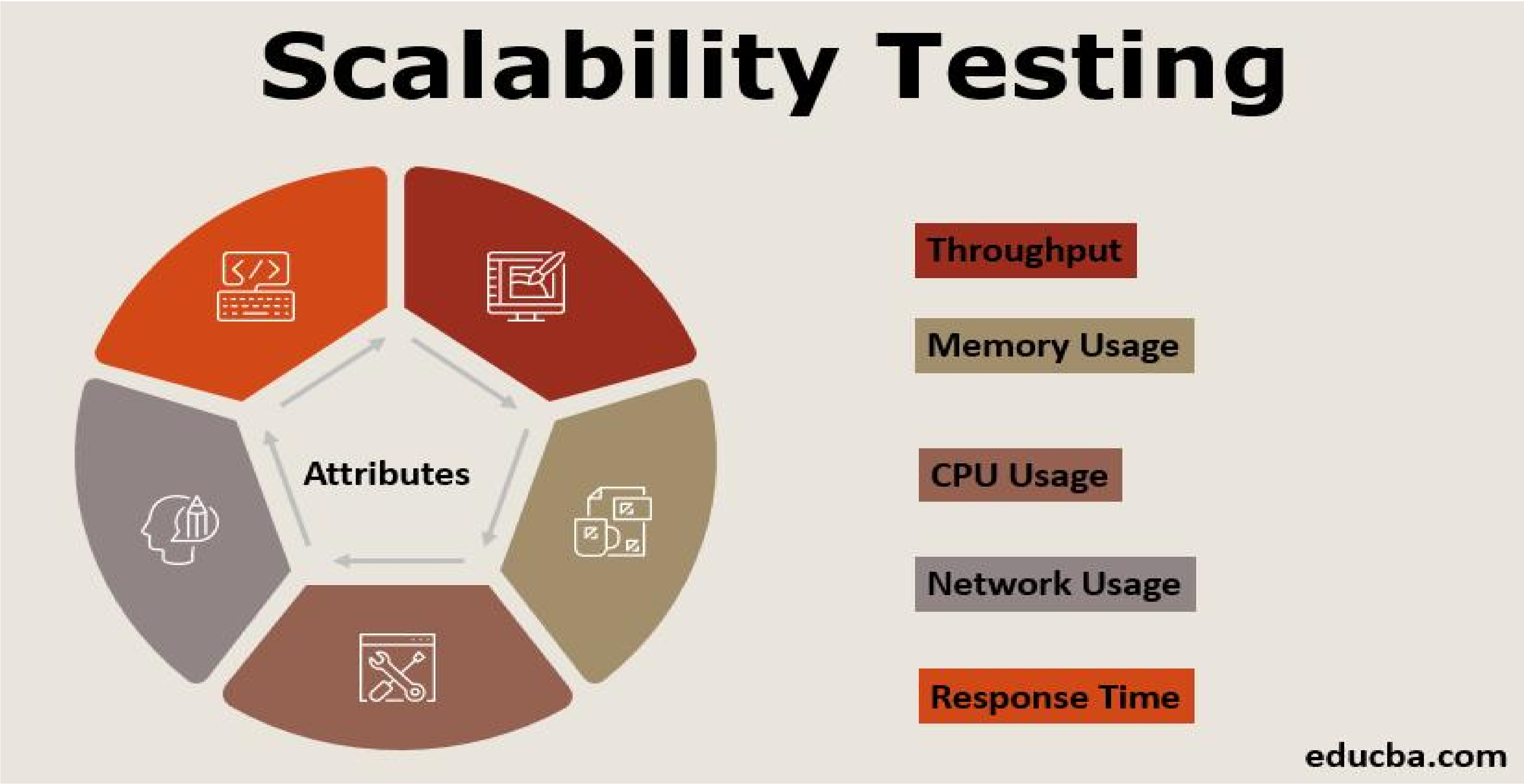
Volume Testing is a type of Software Testing, where the software is subjected to a huge volume of data. It is also referred to as **flood testing.** Volume testing is done to analyze the system performance by increasing the volume of data in the database.
With the help of Volume testing, the impact on response time and system behavior can be studied when exposed to a high volume of data.

For example, testing the music site behavior when there are millions of user to download the song.

Scalability Testing:

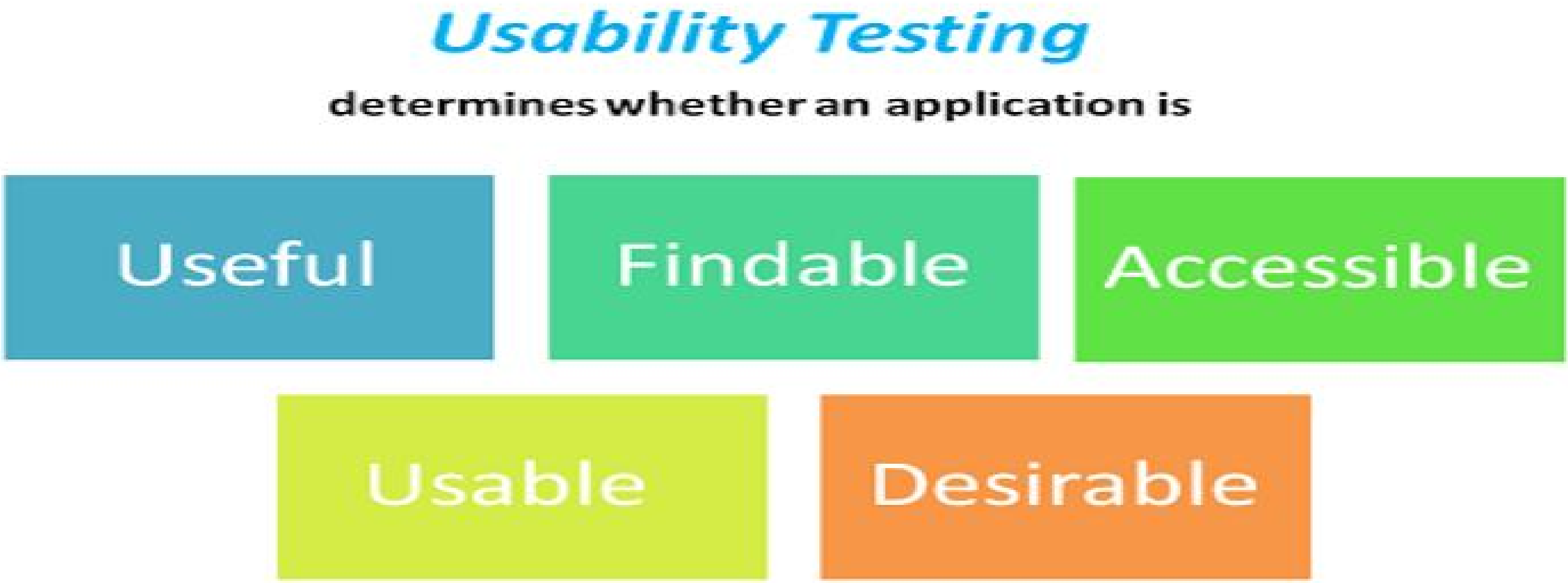
Scalability Testing is a non functional testing method that measures performance of a system or network when the number of user requests are scaled up or down. The purpose of Scalability testing is to ensure that the system can handle projected increase in user traffic, data volume, transaction counts frequency, etc. It tests system ability to meet the growing needs.

It is also referred to as performance testing, as such, it is focused on the behavior of the application when deployed to a larger system or tested under excess load. In Software Engineering, Scalability Testing is to measure at what point the application stops scaling and identify the reason behind it.



USABILITY TESTING:

Usability Testing also known as User Experience (UX) Testing, is a testing method for measuring how easy and user-friendly a software application is. A small set of target end-users, use software application to expose usability defects. Usability testing mainly focuses on user’s ease of using application, flexibility of application to handle controls and ability of application to meet its objectives.
This testing is recommended during the initial design phase of SDLC, which gives more visibility on the expectations of the users.



Load testing

The load testing is the most important essential part of performance testing which is used to check the performance of an application by applying some load like the numbers of users.

Stress testing

The stress testing is testing, which checks the behavior of an application by applying load greater than the desired load. Since it is non-functional testing, so we use this testing when the application is functionally stable.

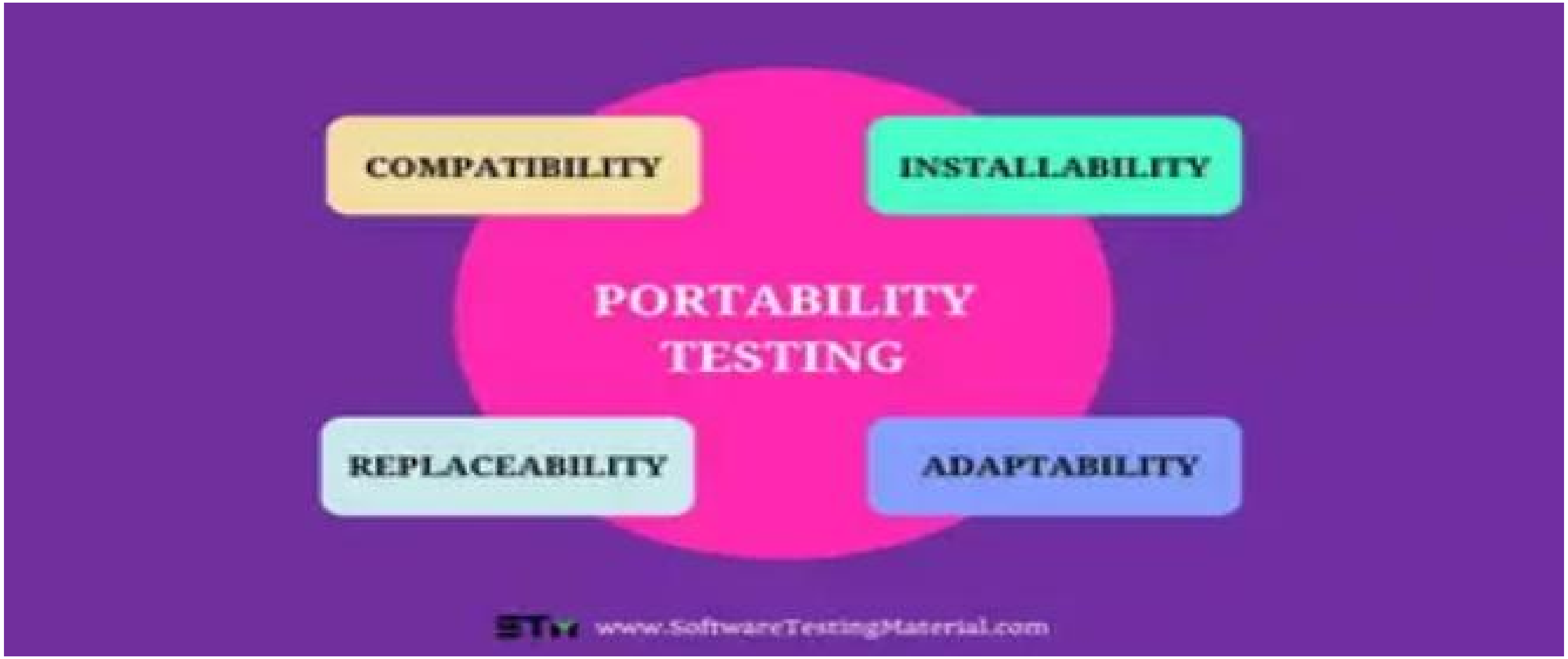
Compilance or Conformance Testing

Conformance Testing is a software testing technique used to certify that the software system complies with the standards and regulations as defined by IEEE, W3C or ETSI. The purpose of conformance testing is to determine how a system under test confirms to meet the individual requirements of a particular standard. Conformance Testing is also called Compliance Testing.

PORTABILITY TESTING:

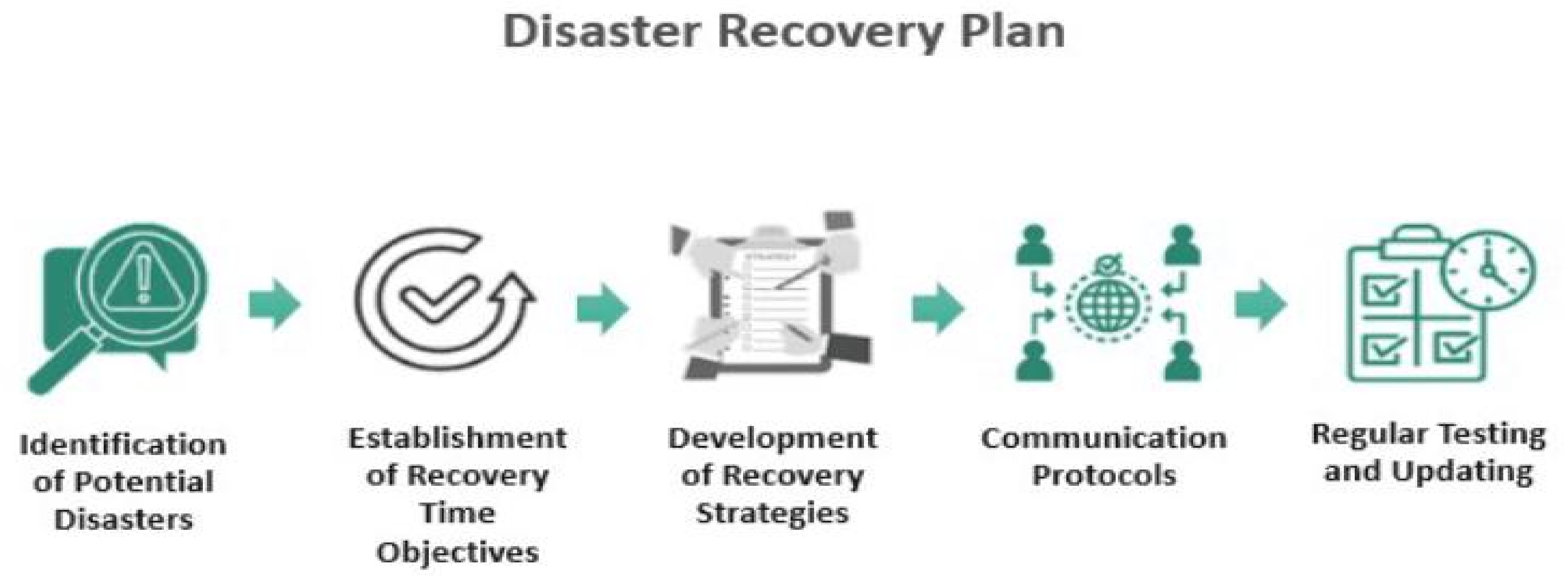
Portability testing is a process of testing with ease with which the software or product can be moved from one environment to another. It is measured in terms of maximum amount of effort required to transfer from one system to another system.

The portability testing is performed regularly throughout the software development life cycle in an iterative and incremental manner.



DISASTER RECOVERY TESTING:

Recovery Testing is software testing technique which verifies software's ability to recover from failures like software/hardware crashes, network failures etc. The purpose of Recovery Testing is to determine whether software operations can be continued after disaster or integrity loss. Recovery testing involves reverting back software to the point where integrity was known and reprocessing transactions to the failure point.


Disaster Recovery Plan
Identification of Potential Disasters
Establishment of Recovery Time Objectives
Development of Recovery Strategies
Communication Protocols
Regular Testing and Updating