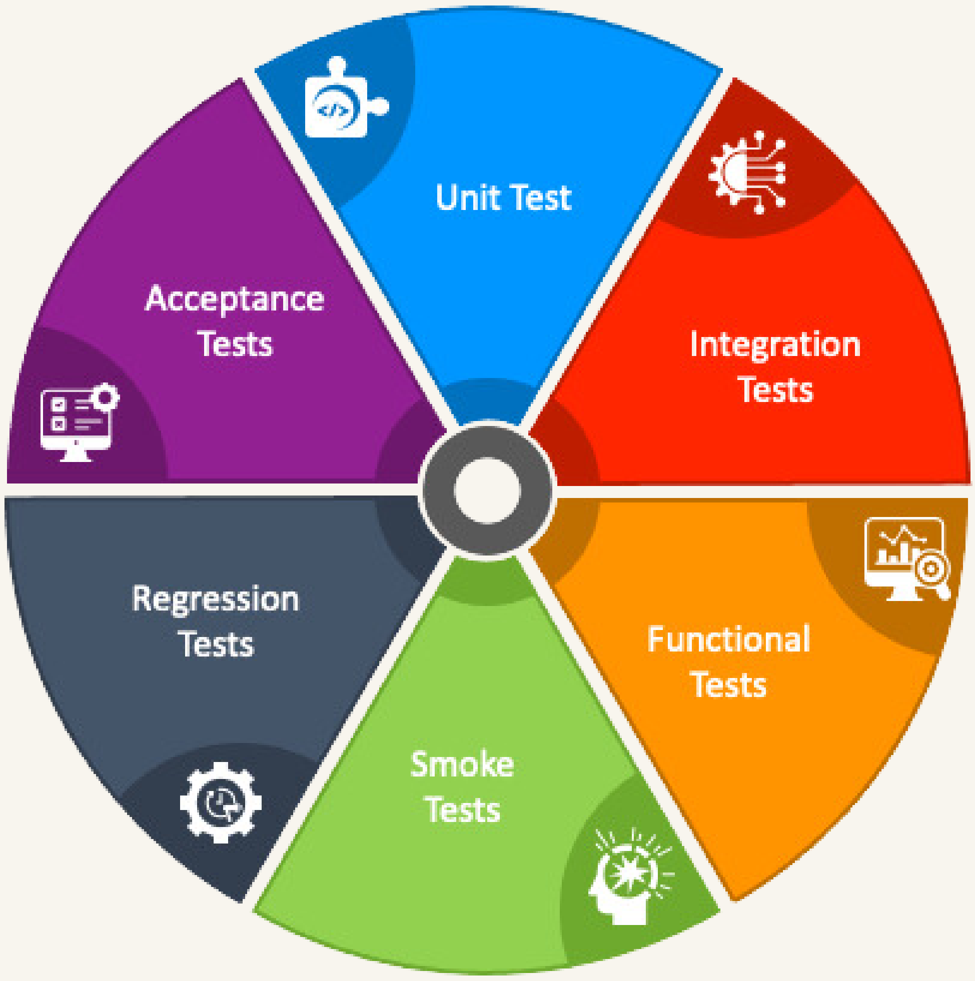
Software Testing

Types



Swipe →

Agenda

- ▼ Introduction to Software Testing
- ▼ Testing Objectives
- ▼ Types of Software Testing
- ▼ Functional Testing
- ▼ Non-Functional Testing

Introduction to Software Testing

- **Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that the software product is **DEFECT** free.

- It is a process that verifies and validates the software product.

- Importance of Software Testing:
  - Though testing itself costs money, companies can save millions per year in development and support if they have a good testing technique and QA processes in place. Early software testing uncovers problems before a product goes to market. The sooner development teams receive test feedback, the sooner they can address issues such as:
    - Architectural flaws
    - Poor design decisions
    - Invalid or incorrect functionality
    - Security vulnerabilities
    - Scalability issues
  - When development leaves ample room for testing, it improves software reliability and high-quality applications are delivered with few errors. A system that meets or even exceeds customer expectations leads to potentially more sales and greater market share.

03

Types of Software Testing

▼ There are **two main types** of software testing:

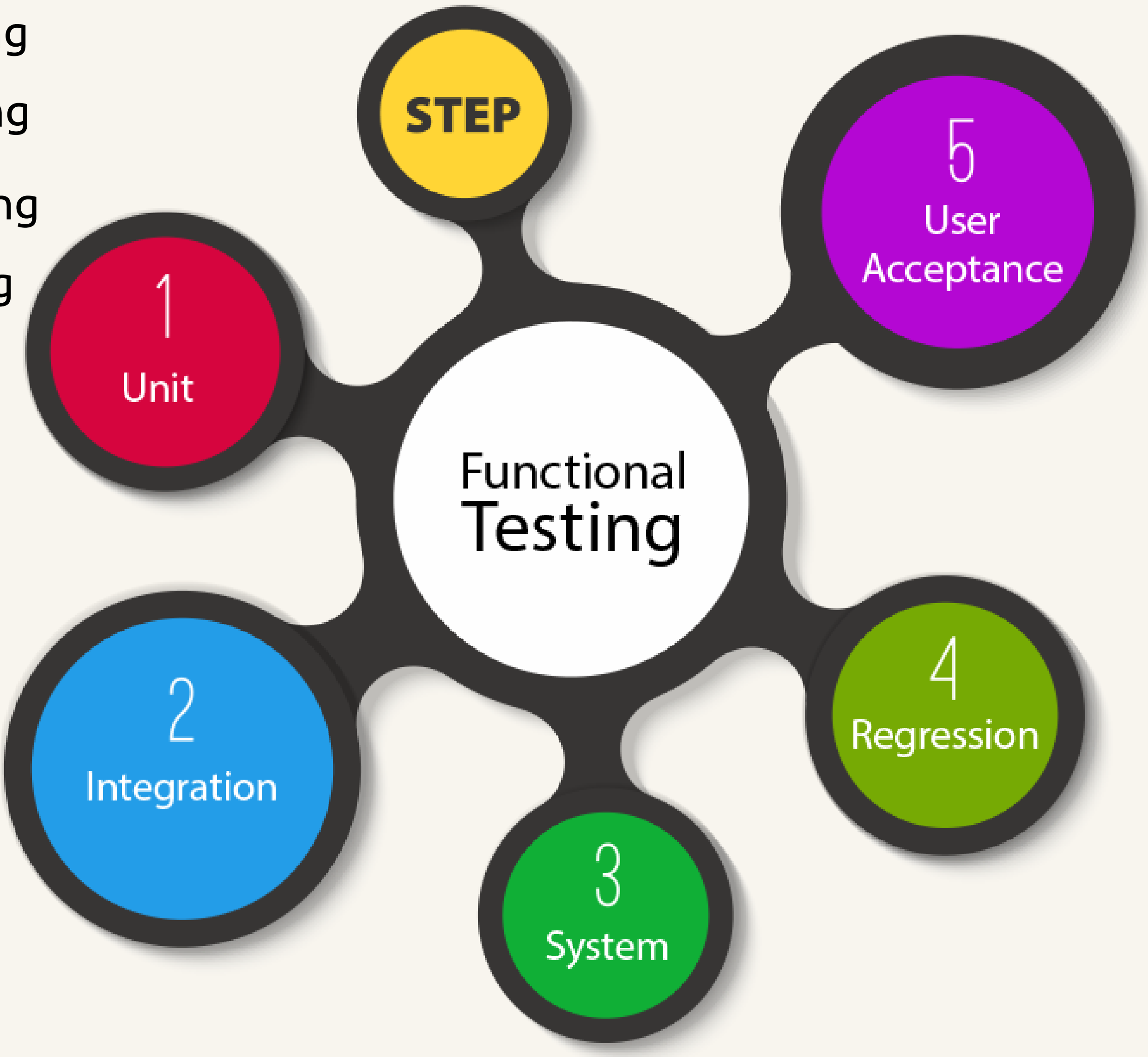
- Functional
- Non-functional Testing

▼ For each of these, we have many testing types that we will be discussing further in this presentation.

Functional Testing

- **Functional testing** is a stage in the software delivery lifecycle in which quality engineers verify whether the application under test's features behaves as per their requirements.

- Examples of functional Testing:
  - Unit Testing
  - Integration Testing
  - Smoke Testing
  - Sanity Testing
  - Regression Testing
  - End-to-End Testing
  - Acceptance Testing
  - White Box Testing
  - Black Box Testing
  - Interface Testing



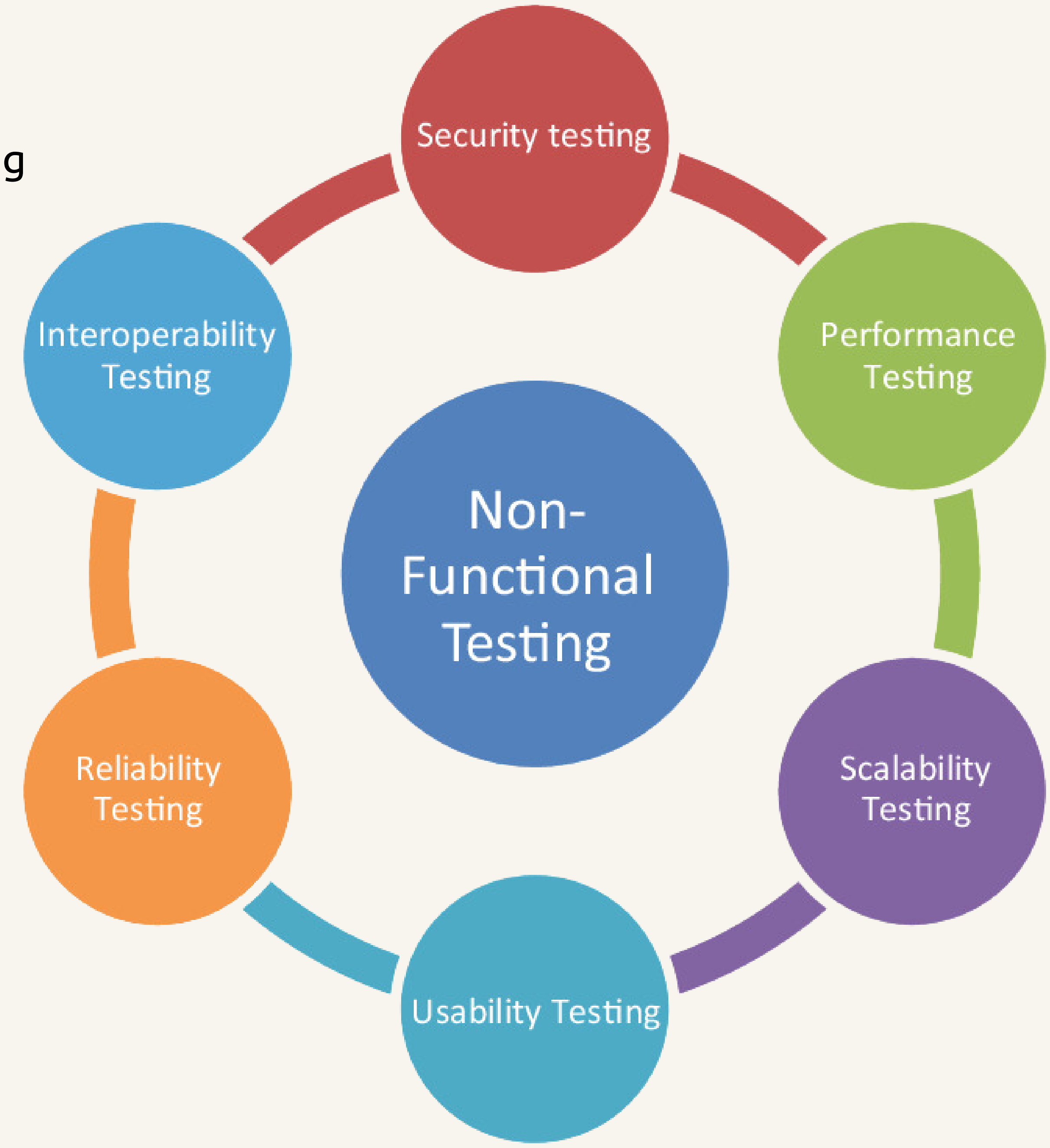
05

Non-Functional Testing

▼ When performing **non-functional testing**, testers evaluate other attributes of system behavior, such as the **system's performance, reliability, and stability.**

▼ Examples of non-functional Testing:

- Performance Testing
- Security Testing
- Usability Testing
- Installation Testing
- Reliability Testing



TechGlobal
SCHOOL

Unit Testing

- **Unit testing** is a type of software testing which is done on an **individual unit or component** to test its corrections.
- Typically, **Unit testing is done by the developer** at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure, or object.
- Developers often use test automation tools such as **NUnit, Xunit, JUnit, Chai, Jest** for the test execution.

Integration Testing

- **Integration testing** is a type of software testing where **two or more modules** of an application are logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system.

- This type of testing is done on integrating modules of a system or between systems. For example, a user is buying a flight ticket from any airline website. Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

Regression Testing

08

- Any new change or feature added to the software can wreck its existing functionalities. **Regression testing** is performed every time alterations are made to check for the software's stability and functionalities. Due to its work-intensive nature, regression testing is often automated.

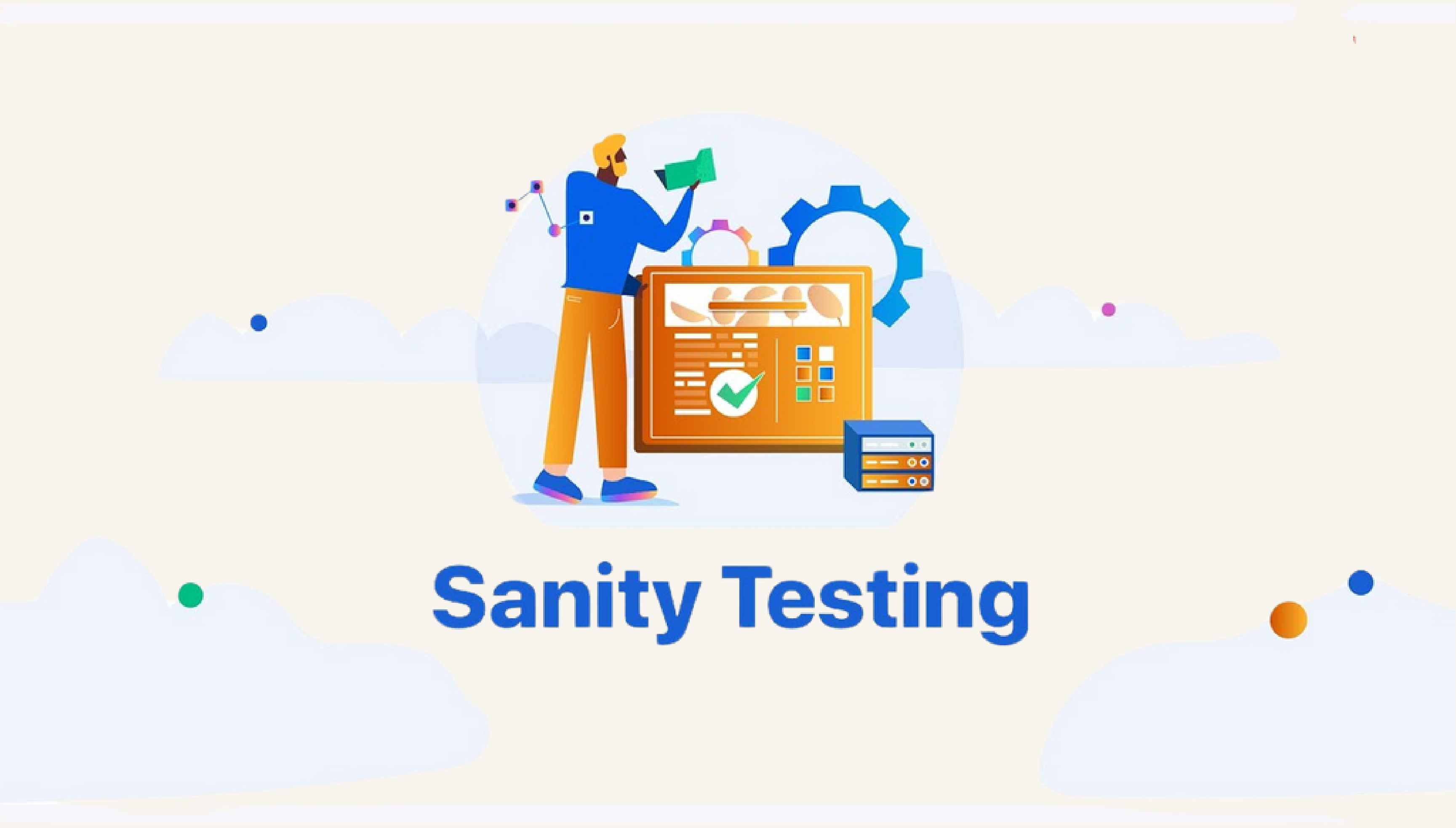
- Example: A food delivery app added a function to help users add multiple promotions on top of each other. A regression test needs to be done to make sure the checkout and payment process is not affected.

TechGlobal SCHOOL

Sanity Testing

- Similar to regression testing, sanity testing is conducted for a new build with **minor bug fixes, or new code added.** If rejected in the sanity testing phase, the build will not proceed to further testing. While regression testing checks the entire system after alterations, **sanity testing** targets specific areas that are affected by the new code or bug fixes only.

- Example: On an e-commerce webpage, users cannot add a particular product to their cart even when the stock is available. After the issue was fixed, sanity testing is performed to ensure that the "add to cart" function is indeed working.



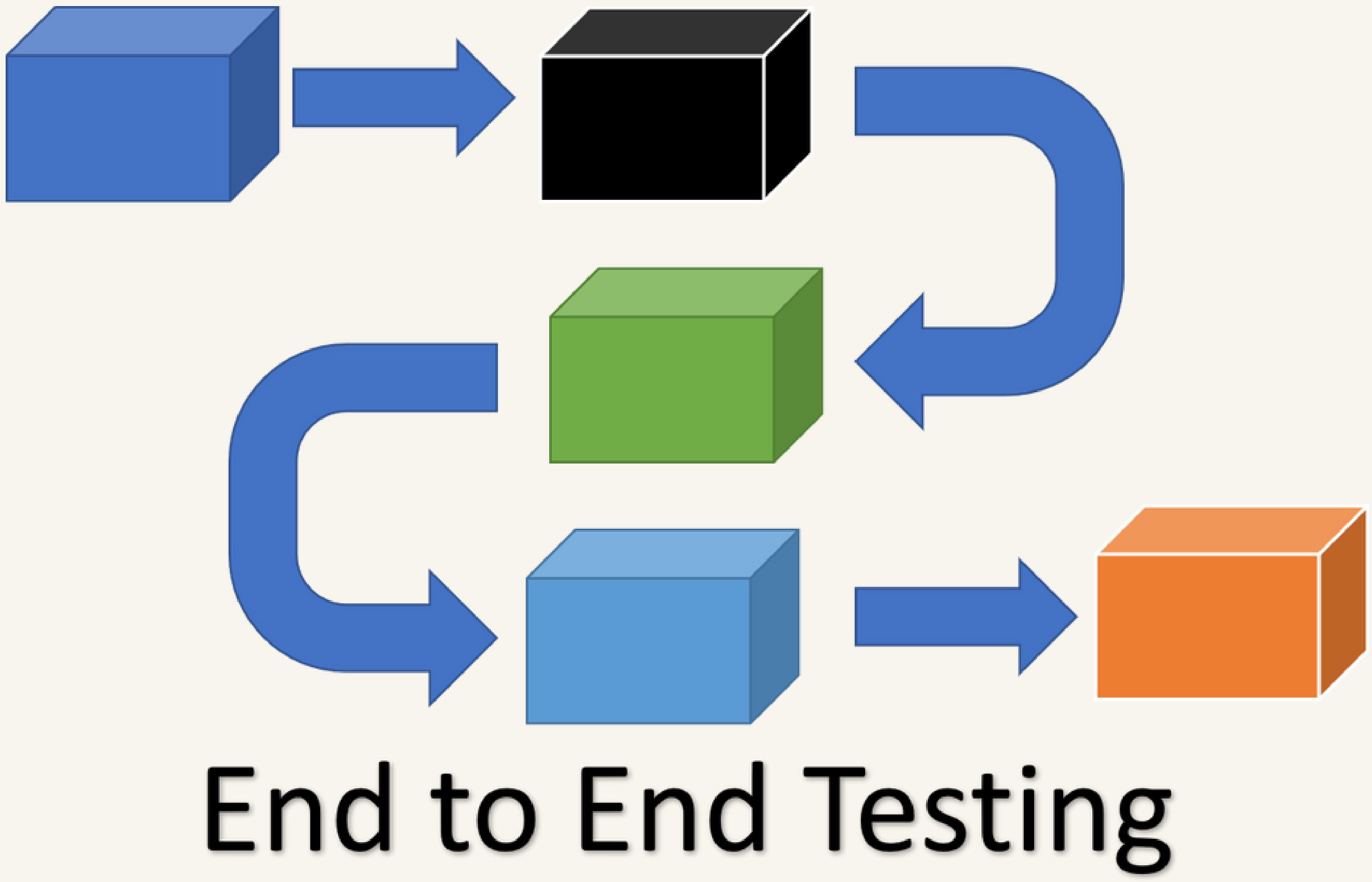
Smoke Testing

- When a new build is completed, it is handed to the QAs for smoke testing. In this phase, only **the most critical and core functionalities** are tested to ensure that they yield the intended results. As an early-stage acceptance test, **smoke testing** adds a verification layer to determine whether or not the new build can proceed to the next stage or needs re-work.
- Example: A utility company built an app with the function to report outages in customers' homes. This function reports the address and other relevant information as well as notifies the homeowner when a dispatcher is on the way to help. Smoke testing will validate this feature on a fundamental level to assure that when an outage is reported, the correct information is sent so a dispatcher can be there on time.

End to End Testing

- It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a **database,** using **network communications,** or interacting with other **hardware, applications, or systems** if appropriate.

- For example, a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users' accounts, updating user address information, receiving order confirmation emails and policy documents.



Security Testing

▼ It is a type of testing performed by a special team. Any hacking method can penetrate the system.

▼ **Security Testing** is done to check how the software, application, or website is secure from **internal and/or external threats.** This testing includes how much software is secure from **malicious programs, viruses** and how secure & strong the authorization and authentication processes are.

▼ It also checks how software behaves for any **hacker's attack & malicious programs** and how software is maintained for data security after such a hacker attack.

Performance Testing

- **Performance testing** is testing of an application's **stability and response time** by applying load.

- The word **stability** means the ability of the application to withstand in the presence of load. **Response time** is how quickly an application is available to users. Performance testing is done with the help of tools. **Loader.IO, JMeter, LoadRunner,** etc. are good tools available in the market.

Types of Performance Testing

▼ Load testing

- Load testing is testing of an application's stability and response time by applying load, which is equal to or less than the designed number of users for an application.
- For example, your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

▼ Stress Testing

- Stress testing is testing an application's stability and response time by applying load, which is more than the designed number of users for an application.
- For example, your application handles 1000 users at a time with a response time of 4 seconds, then stress testing can be done by applying a load of more than 1000 users. Test the application with 1100,1200,1300 users and notice the response time. The goal is to verify the stability of an application under stress.

▼ Endurance Testing

- Endurance testing is testing an application's stability and response time by applying load continuously for a longer period to verify that the application is working fine.
- For example, car companies soak testing to verify that users can drive cars continuously for hours without any problem.

Usability Testing

- **Usability testing** is testing an application from the **user's perspective** to check the look and feel and user-friendliness.

- For example, there is a mobile app for stock trading, and a tester is performing usability testing. Testers can check the scenario like if the mobile app is easy to operate with one hand or not, scroll bar should be vertical, background color of the app should be black and price of and stock is displayed in red or green color.

- The main idea of usability testing of this kind of app is that as soon as the user opens the app, the user should get a glance at the market.

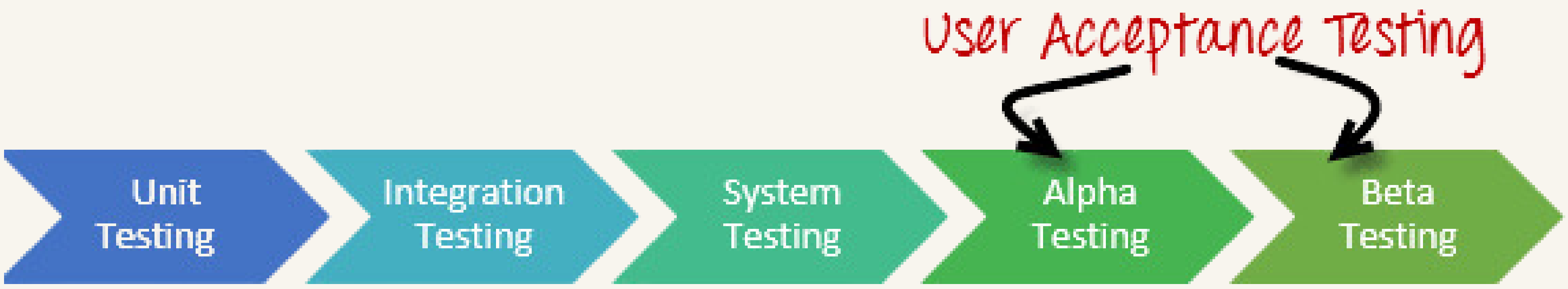
USABILITY TESTING

UAT (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.**
- **User acceptance testing validates the testing done at the end of the development cycle.** It is typically completed after unit testing, quality assurance, system testing and integration testing. The software may undergo other testing phases and be completely functional but might still not meet its requirements if it is not well received by its intended users.
- **Who performs UAT?**
  - End users normally perform user acceptance testing. They are the most effective group to test software in this form because they know exactly how the software will be used on a daily basis and what changes need to be made to be suitable for this day-to-day use.
  - Internal functional experts also play a role in UAT, as they help shape UAT cycles and test management, as well as interpret the results.

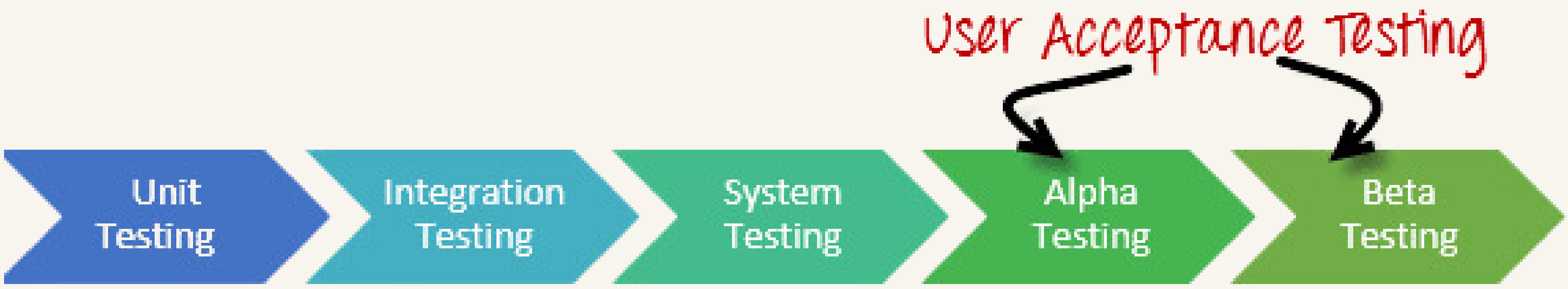
Alpha Testing

- **Alpha Testing is a type of acceptance testing;** performed to identify all possible issues and bugs before releasing the final product to the end users. Alpha testing is carried out by the testers who are **internal employees of the organization.** The main goal is to identify the tasks that a typical user might perform and test them.
- To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing. The main focus of alpha testing is to simulate real users by using a black box and white box techniques.



Beta Testing

- **Beta Testing is performed by "real users" of the software application in "real environment"** and it can be considered as a form of external User Acceptance Testing. It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to test products in customer's environment.
- **Beta version of the software is released to a limited number of end-users** of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.



A/B Testing

- **A/B testing** is an experimental method in which **two versions** of anything are contrasted to see which is **"better"** or more effective.

- This is often done in marketing when two different types of content—whether it be emailing copy, a display ad, a call-to-action (CTA) on a web page, or any other marketing asset—are being compared. This is usually done before launching any product in the market so that the company can get better results.

- This also helps in comparing the performance of two or more variants of emails and then selecting the best among them based on the result given by the audience. So, now without waiting any time let's move forward and take a look that what is A/B testing:

Ad Hoc-Monkey Testing

- **Ad Hoc Testing** is a kind of testing where testers who know the software well test it **without a strict plan.** It's also called **Random Testing or Monkey Testing.**
- Testers might use some existing test cases or choose them **randomly** to test the software.
- The term **"Monkey Testing"** comes from the idea that testers are essentially "monkeying around" with the software, mimicking a playful and **exploratory** approach to uncover hidden problems.