**LEVELS OF TESTING**

When it comes to the different levels of software testing, there are four that take place. Each level checks for something different, and each of them must be conducted thoroughly—nothing should be rushed or skipped.

* Unit Testing
* Intergration Testing
* System Testing
* Acceptance Testing

**Unit Testing:**

Unit testing is a software testing process for testing specific units, components, or software elements. This is the most basic type of testing, and the goal for this level of testing is to validate that each unit of code performs how it should and is free of bugs, errors, and glitches.

Typically, this level is done during the development (or coding) phase by software developers who isolate a section of code and verify that it’s correct. It’s considered a Whitebox testing method, meaning the code can be viewed but usually not altered, and should always take place early in development to save money, time, and effort in the long run.

With unit testing, the earlier, the better! In many cases, this testing is completed by developers before they hand the code over for further testing.

**Integration Testing:**

Next is integration testing, which is when different software components and modules are combined and tested as a group to make sure everything is ready for the next level. Since a standard software project will likely consist of various modules, coded by multiple programmers, the goal is to test to expose potential defects or bugs between the various modules. Sometimes, this phase is referred to as I & T (integration and testing), thread testing, or string testing.

This method focuses on checking the data flow from one module to the next and is performed by testers. It’s another crucial level of testing because it verifies everything is working as one singular unit.

**System Testing:**

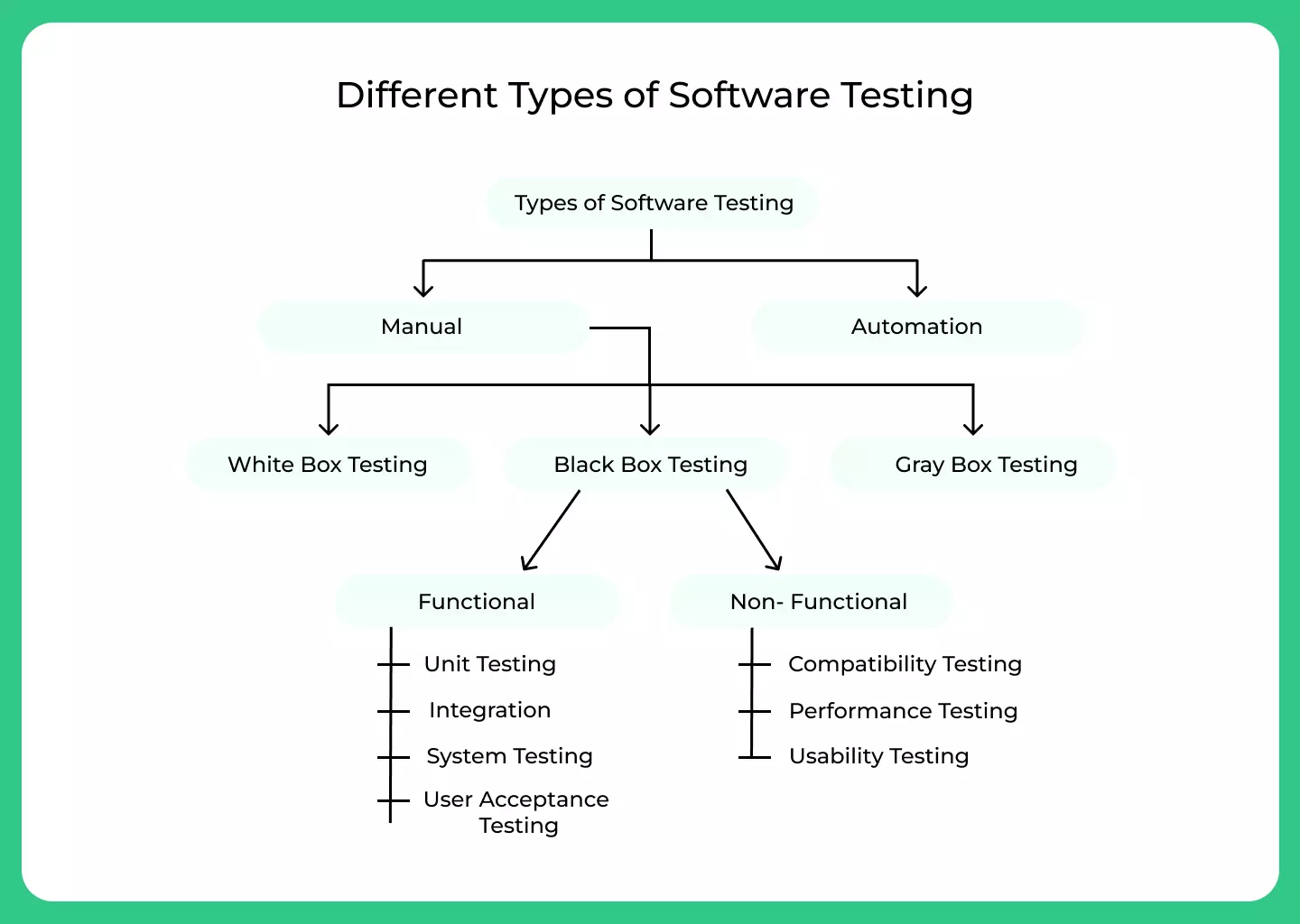
The next level of software testing is system testing. This checks for a system’s compliance in accordance with the necessary given requirements. System testing inspects components like performance, load, reliability, and security with the goal of evaluating the end-to-end system specifications.

Typically, this method is done by a professional testing agent on the completed software product before it can be introduced to the market with real users. This step is important because the project is so close to being complete, so it should be tested in an environment similar to what the user will experience once it’s finished.

**Acceptance Testing:**

Finally, there’s acceptance testing. What makes this level different from the rest is that it’s conducted by the user or customer since the goal is to find out if the requirements have been met on delivery of the final product. Since acceptance testing is the final phase, it needs to validate the end-to-end business flow and check for things like cosmetic errors, spelling mistakes, and usability.

**Diagram:**

****