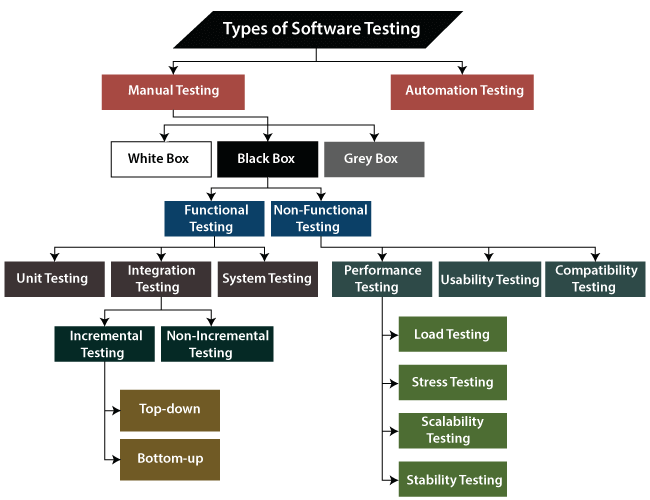
Manual or Automation testing always depend on how the methods are adopted to perform the task.



Using tools such as selenium, xunit(unit testing) etc.

In [computer programming](https://en.wikipedia.org/wiki/Computer_programming) and [software design](https://en.wikipedia.org/wiki/Software_design), **code refactoring** is the process of **restructuring existing**[**computer code**](https://en.wikipedia.org/wiki/Computer_code)—changing the [*factoring*](https://en.wikipedia.org/wiki/Decomposition_(computer_science))—**without changing its external behavior.** Refactoring is intended to improve the design, structure, and/or implementation of the [software](https://en.wikipedia.org/wiki/Software) (its [*non-functional*](https://en.wikipedia.org/wiki/Non-functional_requirement) attributes), while preserving its [functionality](https://en.wikipedia.org/wiki/Functional_requirement). Potential advantages of refactoring may include improved code [readability](https://en.wikipedia.org/wiki/Readability) and reduced [complexity](https://en.wikipedia.org/wiki/Cyclomatic_complexity)

# Automated Testing

Faster | Test your code frequently | Catch bugs before deploying | deploying with confidence.

# Types of Automated Testing

***Unit Testing-***Testing a unit of an application without its external dependencies.

Cheap to Write | Execute Fast

***Integration Test-***Testing a unit of an application with its external dependencies.

Take longer time to execute | more confidence about the code.

***End-To-End Test****-* Drives an application through its UI.

Give the greatest confidence | very slow | very brittle (small change leads to easily break)

# TDD (Test driven Development)

