Integration testing & System testing

Source: https://u-tor.com/topic/system-vs-integration

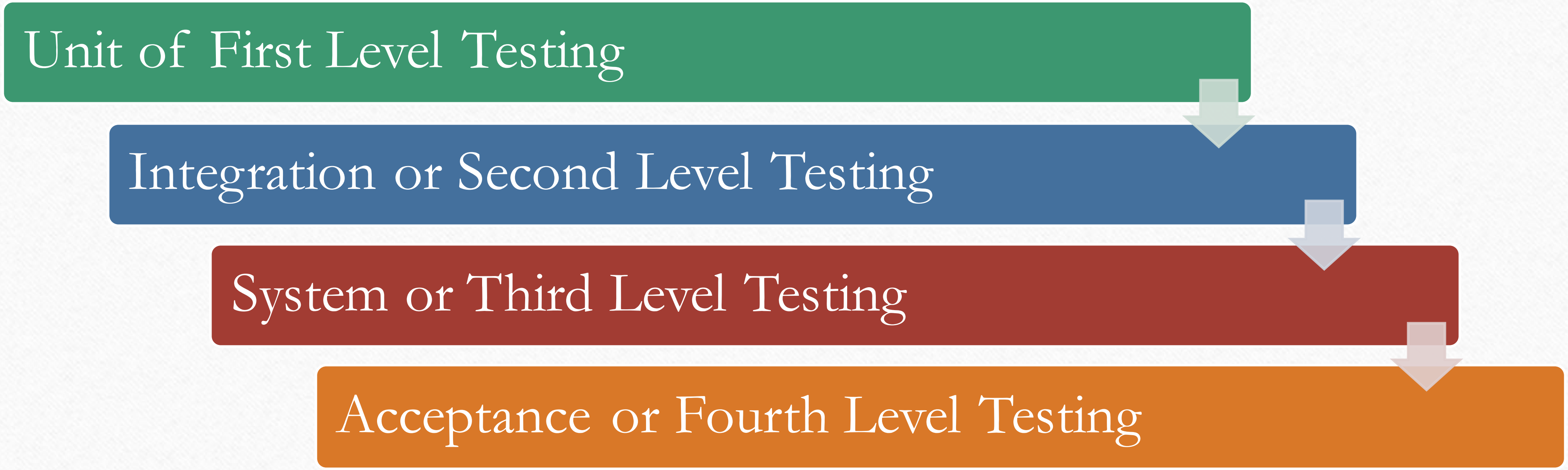
WHY Need testing?

HAVING MORE BUGS WHICH ARE COSTLIER TO FIX THAN IF THEY WERE FOUND EARLIER.

IN THE US , THE COST OF SOFTWARE BUGS IS PROJECTED TO BE ABOUT 2.8 TRILLION/YEAR

ACCORDING TO BETANEWS, 97% OF COMPANIES, INCLUDING IBM, MICROSOFT, CISCO, AT&T, ARE USING AGILE TECHNOLOGY FOR BOTH DEVELOPMENT AND TESTING.

There are 4 level of testing



Unit of First Level Testing

Integration or Second Level Testing

System or Third Level Testing

Acceptance or Fourth Level Testing

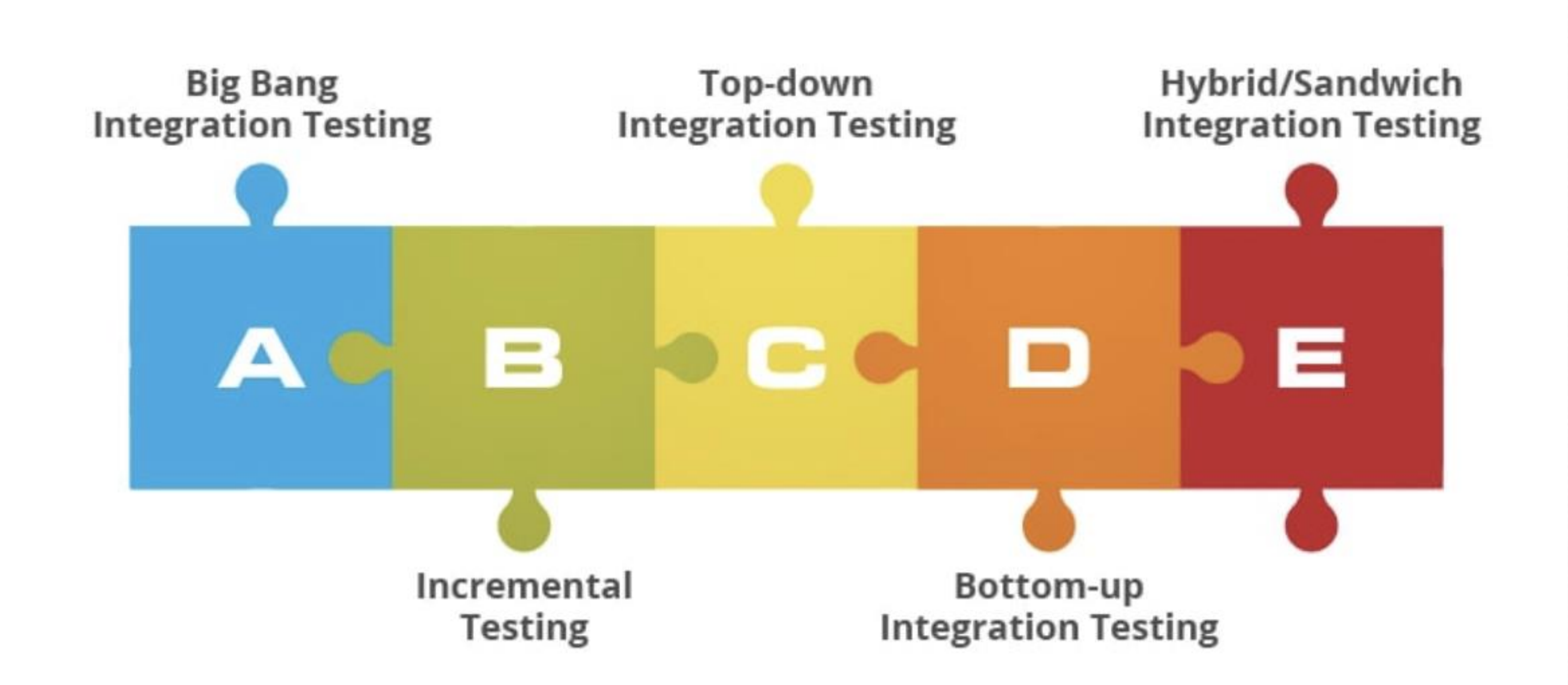
Integration Testing

- Combining all the components that make up the software and testing everything as a whole instead of individually as done during unit testing.
- About verifying the modules and checking their readiness and their collective, integral cooperation.
- Tests can be split into functional and non-functional types
- One can't tell with all certainty if the software is working to its full potential until an integration test is performed.

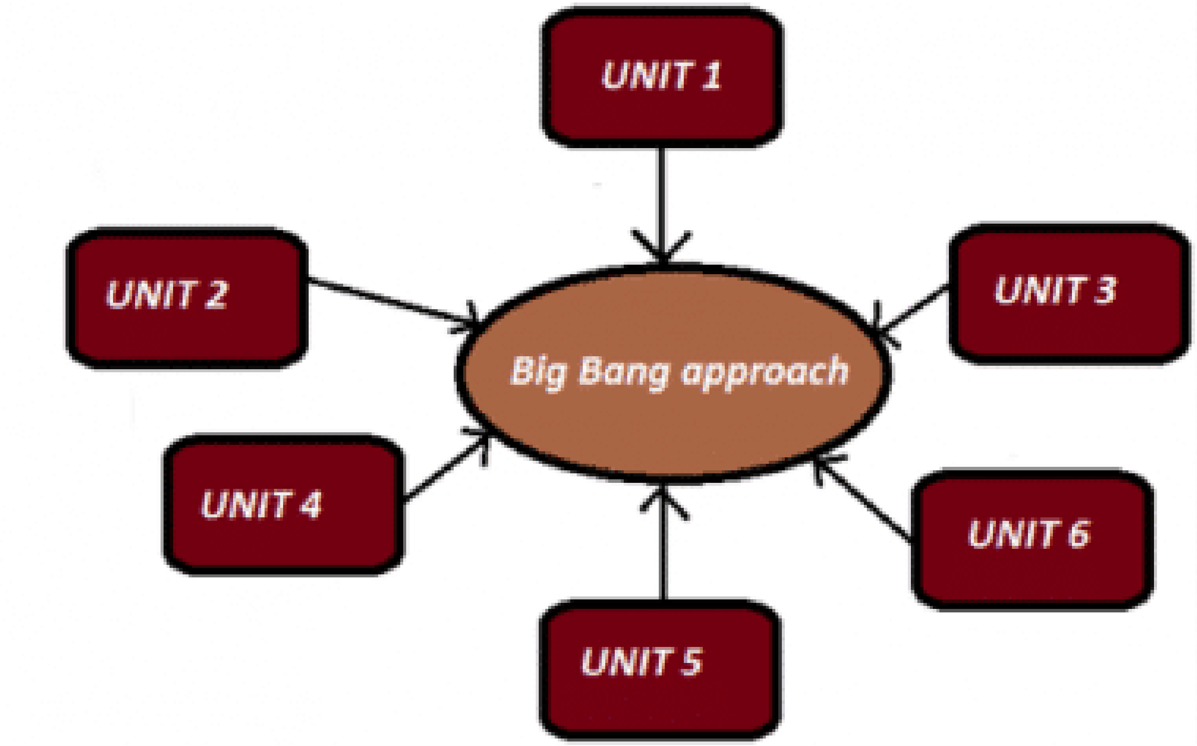
Why Intergration Testing?

1. Codes were written by different developers
2. Ensure it is correctness
3. Ascertain how well the units work together
4. Pre-approved requirements can lose their relevance and change – that’s especially true for Agile projects.

How to test:


Big Bang Integration Testing
Top-down Integration Testing
Hybrid/Sandwich Integration Testing
A
B
C
D
E
Incremental Testing
Bottom-up Integration Testing

Big-Bang Testing



Incremental Testing

Fig2

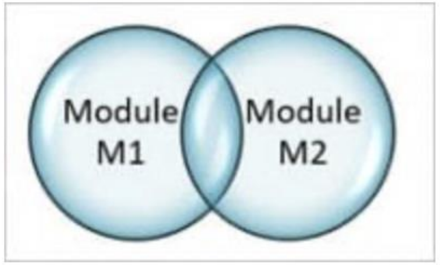


Fig3

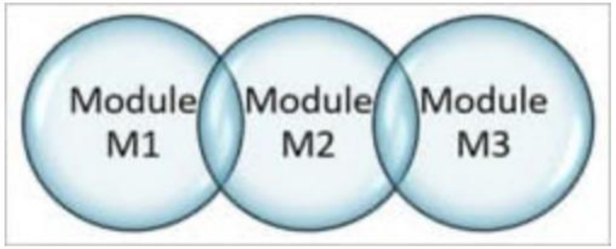
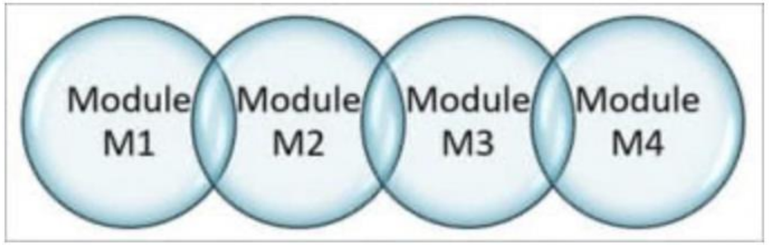
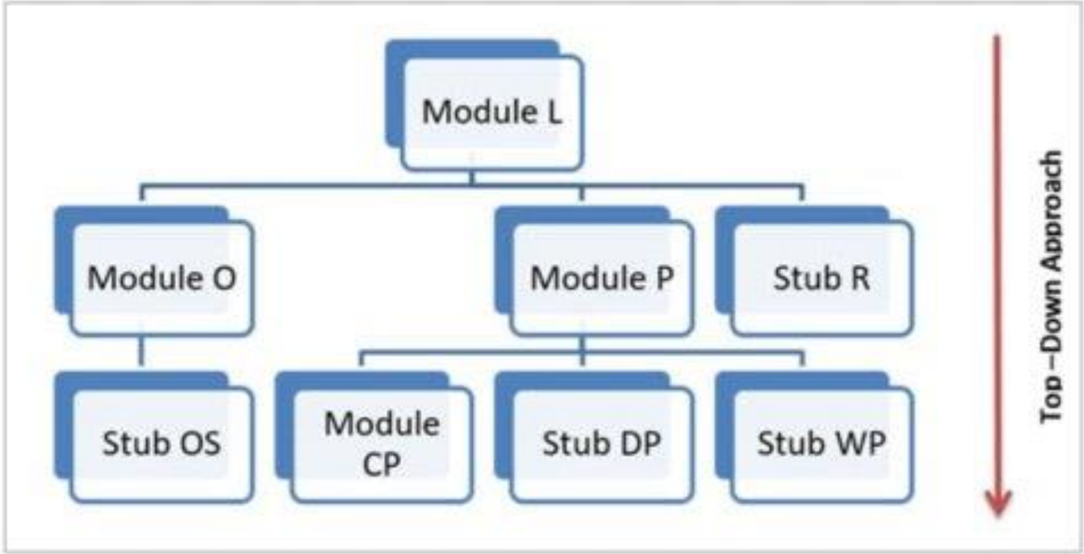


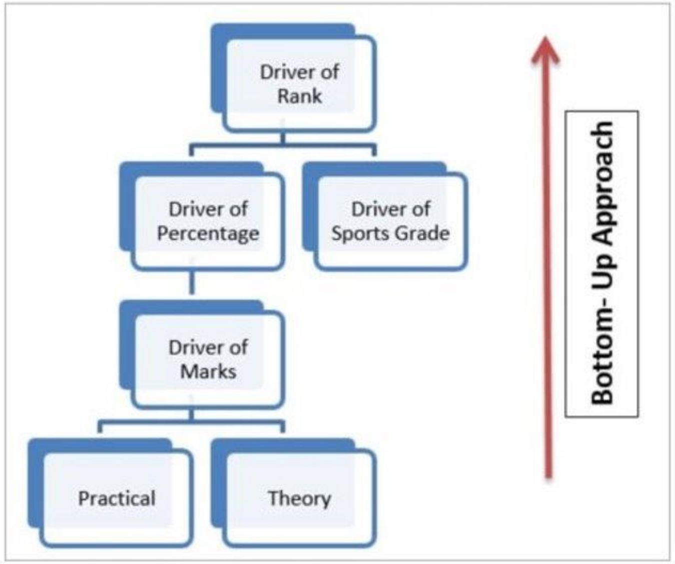
Fig4



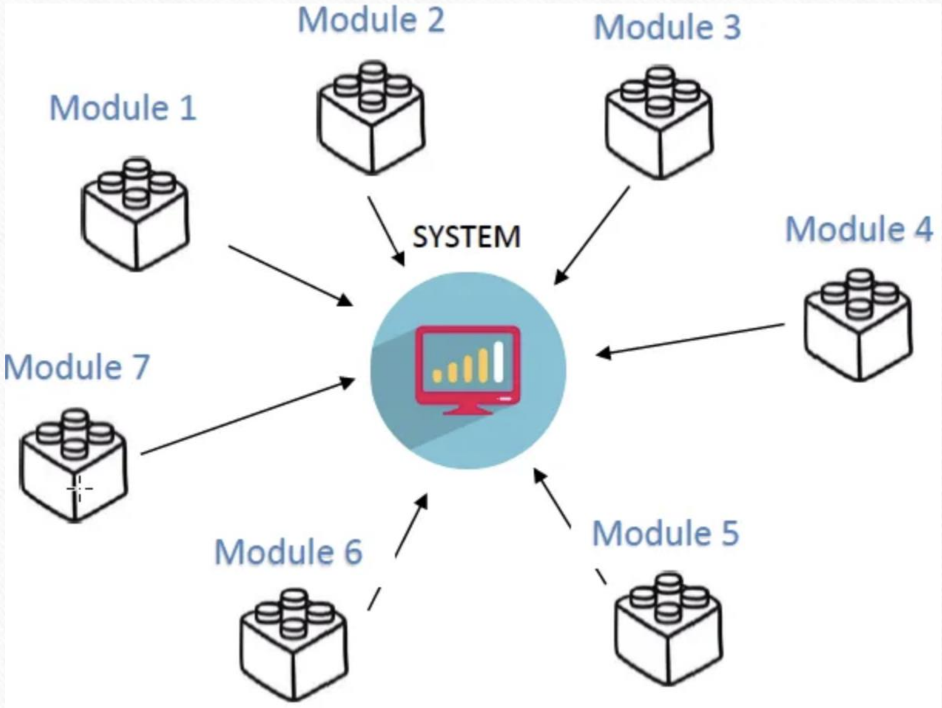
Top to Down

Button-Up





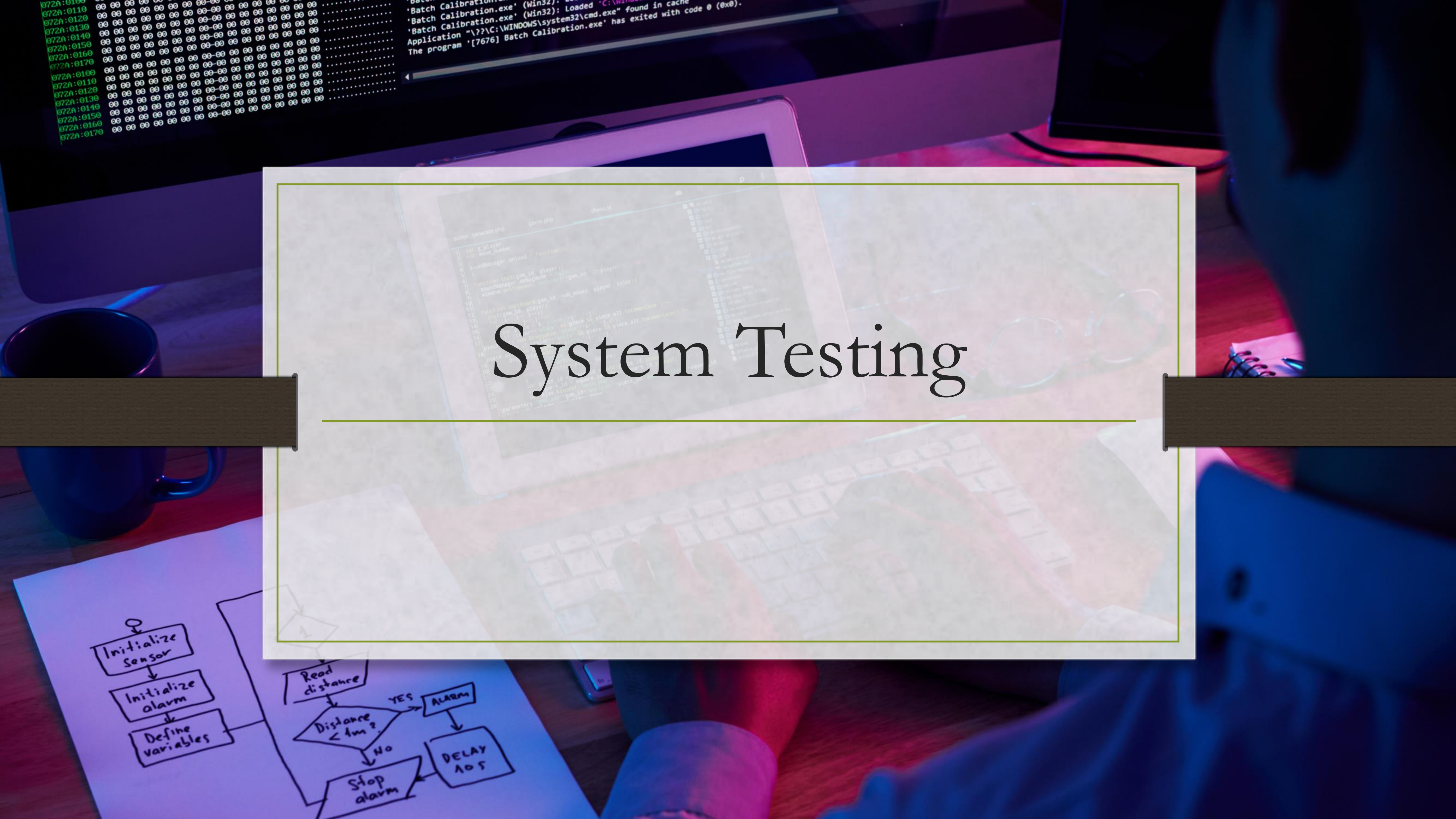
A testing team interacts with an app and its units via the user interface – by clicking on buttons and links, scrolling, swiping, etc. They don’t need to know how code works or consider the backend part of the components.



Who perform this test?

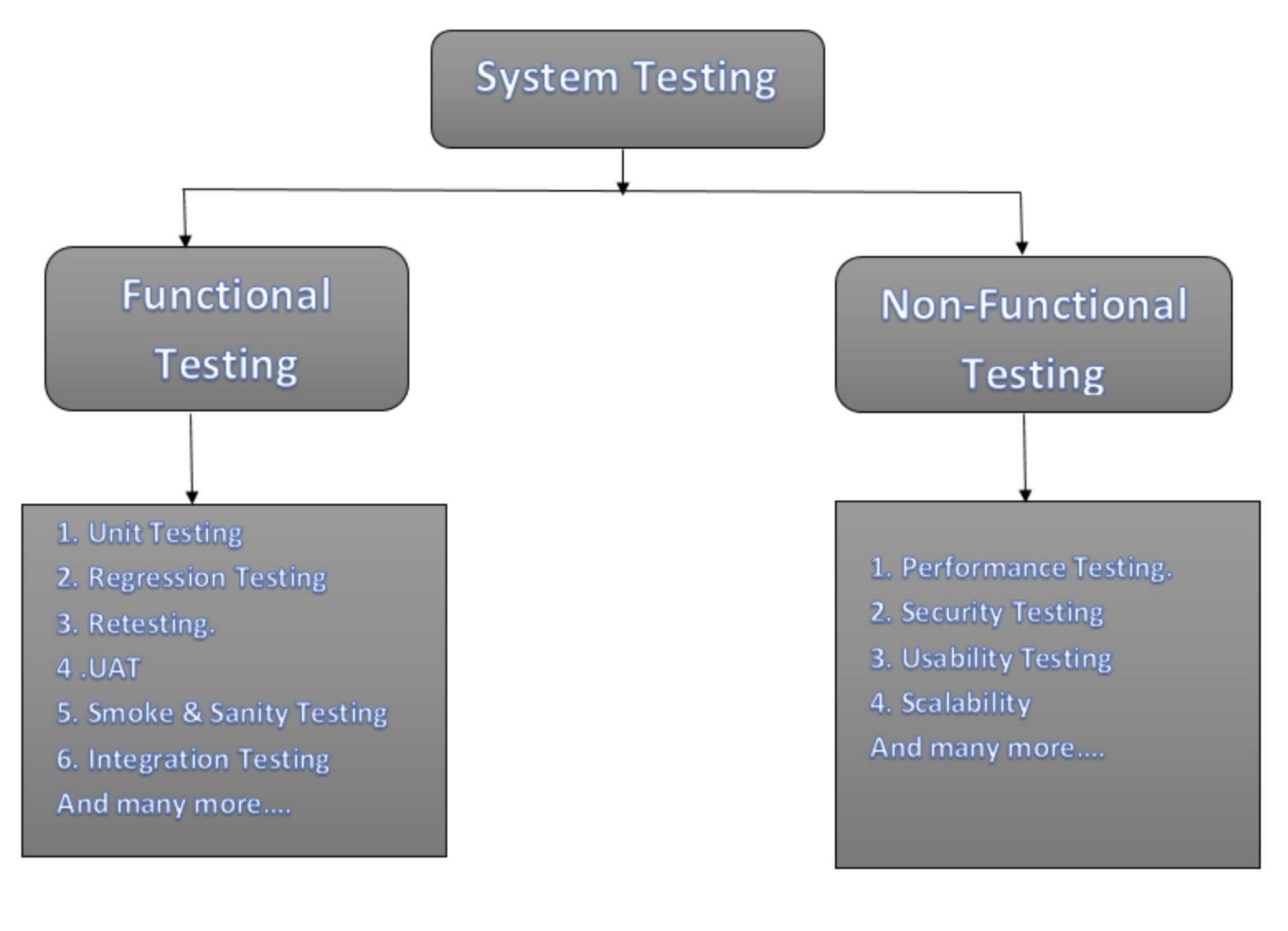
- Tester(s)

System Testing



Why System Testing?

- Verifying the required operations of the software and its compatibility with operating systems.
- Test both the technicalities and the business logic of the software
- Run functional tests to check what the various functions of a system do, and non-functional tests to check how those functions work.
- It is very important to complete a full test cycle and ST is the stage where it is done.



System Testing

Functional Testing

1. Unit Testing
2. Regression Testing
3. Retesting.
4 .UAT
5. Smoke & Sanity Testing
6. Integration Testing
And many more....

Non-Functional Testing

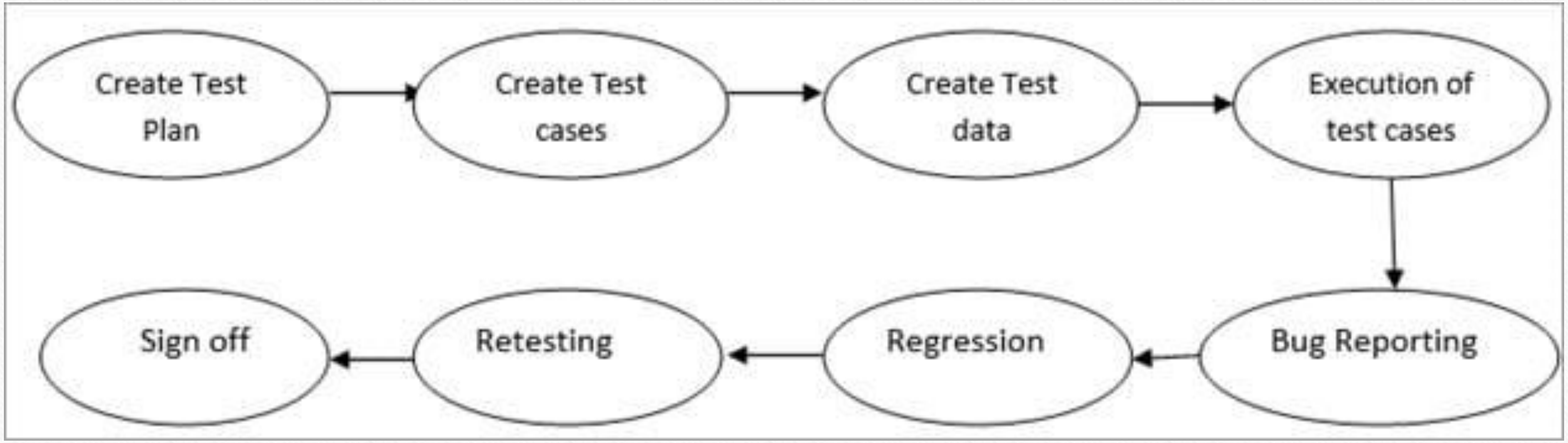
1. Performance Testing.
2. Security Testing
3. Usability Testing
4. Scalability
And many more....

Approach

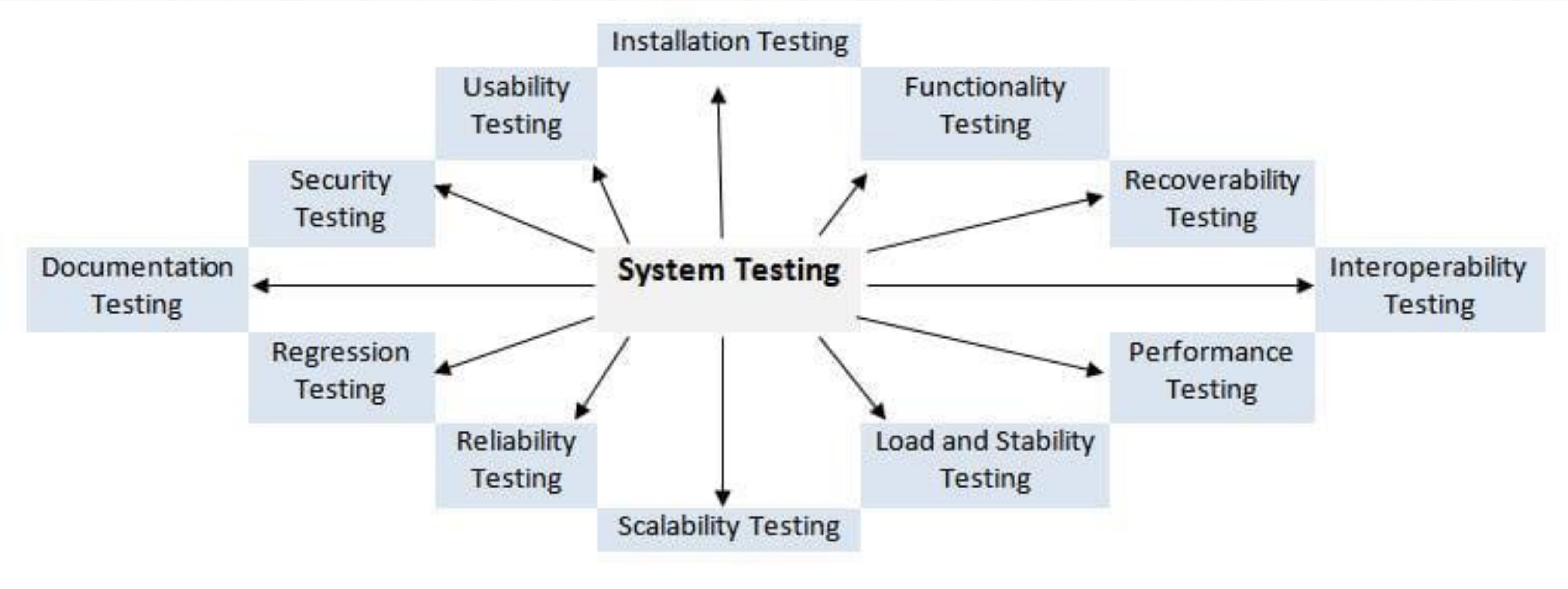
Clack-box test technique.

- Technique does not require internal knowledge of the code whereas the white box technique requires internal knowledge of the code.

How to perform:



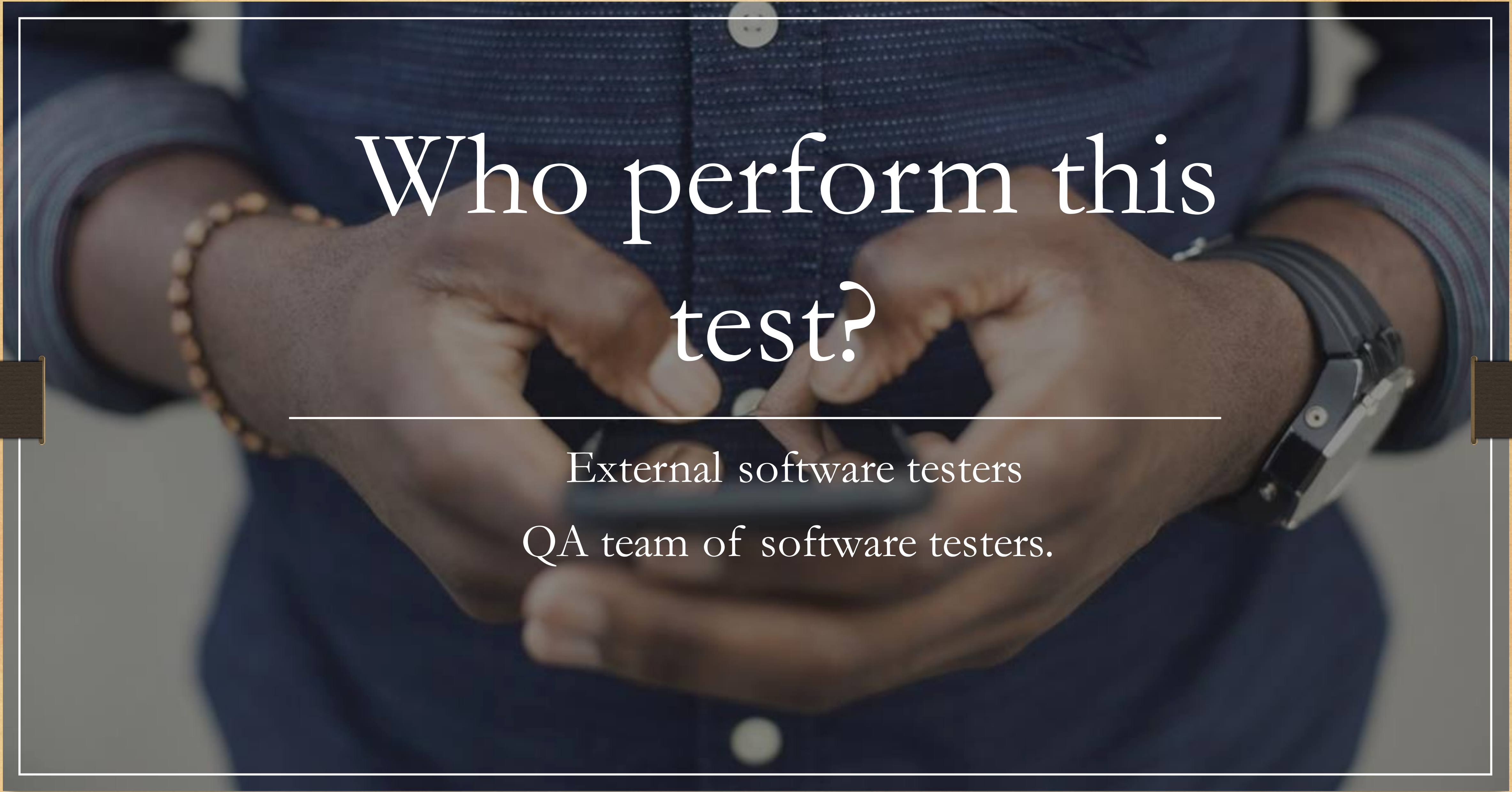
Type of System Testing



Example:

- Login feature responds when the user enters a password. But in non-functional testing, we check how long it takes the user to log in after password entry.

Who perform this test?



External software testers

QA team of software testers.

Integration vs System

	System Tests	Integration Tests
Intention	To guarantee that the total build fulfills the business specifications.	To guarantee that joined units can act together without problems.
Type	Nonfunctional and functional type of test. It falls in the acceptance testing class.	Functional type of test. It's not in the acceptance testing class.
Technique	Black box testing	White and black box testing or gray box testing
Level	Three (3)	Two (2)
Value	Helps to identify system errors.	Helps to identify interface errors.
Teams involved	Developers and Testers	QA