

Why we go for Manual Testing?

1. Human eye captures more defects.
2. Everything cannot be automated like Captcha, Image, Animation, Video, etc.
3. Report prepared by the manual testers will be always detailed because manual testers are strong in Domain knowledge.
while automation testers are strong in technical knowlege.

Software Testing:

1. Functional Testing
2. Non Functional Testing

1.Functional Testing:

Checking the functionalities of the application.

---> Manual Testing (Testing the functionalities by manual)

---> Automation Testing (Testing the functionalities by using tools/scripts)

---> Webservices Testing (Validating request and response) - API Testing

Automation Tools:

1. Selenium - Web Based Applications
2. Appium - Mobile Applications
3. QTP(Quick Test Professional)/UFT(Unified Functional Testing) - Desktop Related Applications [Ex: MS Office Applications, Web Browsers,etc]
4. Test Complete - Both Web Based and Desktop Based Applications

2.Non-Functional Testing:

Checking the non-functional aspects of the application.

Types of Non-Functional Testing:

1.Performance Testing:

*They add virtual number of users accessing the application simultaneously and at a

same time to check the performance of the application.

*Tools Used: i.)J-Meter	---> Open Source Tool
ii.)Load Runner	---> Paid Version

Load/Stress Testing:

Process of adding load and making the application stress is called Load/Stress Testing.

2.Usability Testing:

To check how good the application is user friendly.

3.Accessibility Testing:

*To check how good the application is accessible to different users.

[Note: This testing mainly focus on physically challenged peoples]

*Tools Used: JAWS and NVDA

Principles of Software Testing:

1. Testing shows the presence of defect not their absence.
2. Exhaustive testing is not possible (Testing all the combinations of valid or invalid data is impossible).
3. Early testing saves time and money.
4. Defects cluster together (Concentrating the module even it is small which has more defects and testing it thoroughly will raise of more defect).
It was found that 80% of defects comes from 20% of the modules.
5. Beware of pesticide paradox (Aware of which module the testing is required. Don't Spend time in testing the module which doesn't have any defects).
6. Testing is context dependent (Whatever the application we are going to test, we have to act based on the context as what the application related to).
7. Absence of error is a fallacy (Instead of telling bug free, make sure the environments are matching with the client requirement).

