QA Interview Quick Notes – Manual Testing Differences

Verification vs Validation

Verification	Validation
Ensures we are building the product right	Ensures we are building the right product
Static testing – no code execution	Dynamic testing – code executed
Focus on reviews and documentation	Focus on actual testing
Performed by QA or development team	Performed by testers or end users

Verification → Process check (no execution)
Validation → Product check (with execution)

Retesting vs Regression Testing

Retesting	Regression Testing
Done to check whether a specific defect is fixed	Done to ensure existing features are not broken after changes
Performed on failed test cases	Performed on passed test cases
Planned and specific testing	Generic and broad testing
Manual testing is preferred	Often automated for efficiency

Retesting → Verify defect fix
Regression → Verify nothing else broke

https://www.linkedin.com/in/vishnupriyaravichandran

Priority vs Severity

Priority	Severity
Defines how soon a defect should be fixed	Defines how serious the defect is
Set from a business/project point of view	Set from a technical/impact point of view
Can change frequently during the project	Usually constant unless impact changes
Example: Spelling mistake on homepage – High Priority, Low Severity	Login crash – Low Priority, High Severity

Priority → Urgency to fix

Severity → Impact on system

Smoke Testing vs Sanity Testing

Smoke Testing	Sanity Testing
Done to check basic stability of the build	Done to check specific functionality after changes
Broad and shallow testing	Narrow and deep testing
Performed on new build	Performed on regression or bug fix build
Usually scripted	Usually unscripted

Smoke → "Is build stable?"

Sanity → "Is fix working fine?"

Functional vs Non-Functional Testing

Functional Testing	Non-Functional Testing
Validates what the system does	Validates how well the system performs
Based on business requirements	Based on performance/usability needs
Includes system, integration, UAT	Includes load, stress, usability tests
Focuses on features	Focuses on quality attributes

Functional → Works or not
Non-functional → Works well or not

Positive Testing vs Negative Testing

Positive Testing	Negative Testing
Uses valid inputs to check expected behavior	Uses invalid inputs to check error handling
Confirms software works as intended	Confirms software handles failure properly
Example: Valid username/password	Example: Blank or invalid password

Positive → System should accept
Negative → System should reject

Static Testing vs Dynamic Testing

Static Testing	Dynamic Testing
Verification activity – no execution	Validation activity – requires execution
Detects defects early in SDLC	Detects defects during/after execution
Includes reviews, walkthroughs	Includes functional, regression tests
Focus on process	Focus on product

Static → Check without running code
Dynamic → Check by running code

System Testing vs Integration Testing

System Testing	Integration Testing
Tests the entire system as a whole	Tests the interaction between modules
Done after integration testing	Done after unit testing
Validates end-to-end functionality	Validates data flow and interfaces
Performed by QA team	Performed by developers or QA

Integration → Check connections
System → Check complete product

Alpha Testing vs Beta Testing

Alpha Testing	Beta Testing
Conducted internally by QA team	Conducted externally by real users
Done before release	Done after release candidate build
Controlled environment	Real user environment
Helps find internal bugs	Helps get user feedback

Alpha → Internal testing
Beta → External testing

White Box Testing vs Black Box Testing

White Box Testing	Black Box Testing
Based on code logic	Based on requirements
Done by developers	Done by testers
Requires programming knowledge	No programming required
Focus on internal paths	Focus on input-output behavior

https://www.linkedin.com/in/vishnupriyaravichandran

White box → Inside the code
Black box → Outside the code

QA vs QC

QA (Quality Assurance)	QC (Quality Control)
Process-oriented	Product-oriented
Prevents defects	Finds defects
Focuses on process improvement	Focuses on product validation
Done throughout SDLC	Done after development

QA → Prevent defects
QC → Detect defects

Test Case vs Test Scenario

Test Case	Test Scenario
Step-by-step procedure to test a function	High-level idea to test a feature
Detailed: input, action, expected result	Broad: user journey or flow
Example: Verify login with valid credentials	Example: Verify login functionality

Scenario → What to test
Case → How to test

https://www.linkedin.com/in/vishnupriyaravichandran

Manual Testing vs Automation Testing

Manual Testing	Automation Testing
Performed manually by testers	Performed using scripts/tools
Time-consuming but flexible	Fast and repeatable
Best for exploratory/ad-hoc testing	Best for regression/repetitive testing
No initial cost	Requires tool setup cost

Manual → Human effort
Automation → Tool-driven execution

Adhoc Testing vs Exploratory Testing

Adhoc Testing	Exploratory Testing
Done without planning or documentation	Done with learning + test design simultaneously
Based on tester's intuition	Based on tester's skill and domain knowledge
Unstructured	Semi-structured

Adhoc → Random testing
Exploratory → Smart unscripted testing

Load Testing vs Stress Testing

Load Testing	Stress Testing
Checks system under expected load	Checks system under extreme load

https://www.linkedin.com/in/vishnupriyaravichandran

Goal: Find performance limit	Goal: Find breaking point
Measures response time	Measures stability under pressure

Load → Normal pressure

Stress → Beyond limit

SDLC vs STLC

SDLC (Software Development Life Cycle)	STLC (Software Testing Life Cycle)
Covers overall development process	Covers testing process only
Includes requirement → maintenance	Includes test planning → test closure
Focus on building product	Focus on validating product

SDLC → Build it

STLC → Test it

https://www.linkedin.com/in/vishnupriyaravichandran

Use Case vs Test Case

Use Case	Test Case
Describes end-user interaction	Describes steps to verify functionality
Written by BA or stakeholder	Written by tester
Covers business flow	Covers detailed conditions and data

Use Case → User perspective

Test Case → Tester perspective

Bug vs Defect vs Error vs Failure

Term	Meaning
Error	Mistake by developer while coding or designing
Defect	Error found during testing
Bug	Common term for defect found in testing
Failure	When defect appears during software execution

Error → By dev

Defect/Bug → Found by tester

Failure → Seen by user

Test Plan vs Test Strategy

Test Plan	Test Strategy
Document for a specific project	Document for overall organization
Defines scope, schedule, resources	Defines approach, methodology, objectives
Prepared by Test Lead/Manager	Prepared by Project Manager

Plan → Project level
Strategy → Organization level

Error, Defect, Failure vs Root Cause, Effect

Error/Defect/Failure	Root Cause/Effect
Technical issues during development/testing	Why issue occurred and what impact it caused
Focus on what happened	Focus on why it happened

Error = Symptom
Root Cause = Reason