What is Testing in the Testing Phase?

Testing Phase is the stage of SDLC where the developed software is **verified and validated** to ensure it works as per the **business requirements (BRD/FRD)** and is **free from defects** before going live.

In this phase, the system is checked for:

- **Functionality**
- **Accuracy**
- **Performance**
- **Security**
- **User experience**

✅ Objectives of Testing Phase

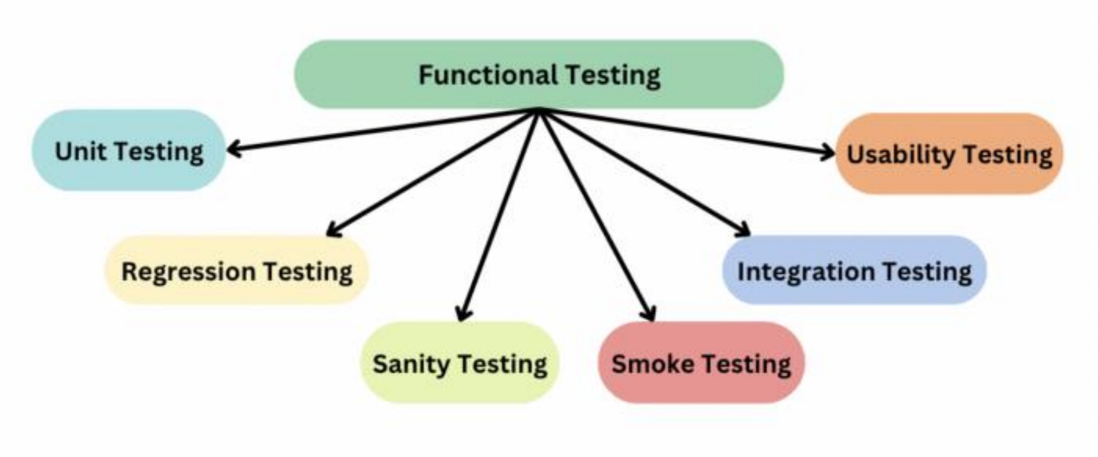
✔ Find defects & bugs
✔ Ensure requirements are implemented correctly
✔ Validate business scenarios
✔ Confirm system readiness for production
✔ Reduce production failures
✔ Improve software quality

✅ Types of Testing Done in the Testing Phase

1 Functional Testing

Checks whether the system works as per requirements.
Example: Login, payment, report generation.

Functional Testing – Detailed Explanation



◆ What is Functional Testing?

Functional Testing is a type of software testing that verifies whether each **function of the application works according to the business requirements (BRD/FRD/SRS)**.

It answers one main question:
☞ **“Is the software doing what the business expects it to do?”**

It focuses on:

- **Input → Process → Output**
- User actions and system responses
- Business rules and validations

◆ Objective of Functional Testing

✓ Validate business requirements
✓ Ensure correct functionality
✓ Identify defects in features
✓ Verify data accuracy
✓ Ensure smooth user experience
✓ Confirm integration between modules

What is Tested in Functional Testing?

Functional testing checks:

- Login & authentication
- Forms & data validation
- Transactions (Add, Update, Delete)
- Reports & dashboards
- Payment & refund
- Notifications (Email, SMS)
- Role-based access
- Error handling

Types of Functional Testing (In Detail)

1 Unit Testing

Done by developers. Tests **individual components or functions**.
Example: Testing only the "Login" function.

2 Integration Testing

Tests how **different modules work together**.
Example:
Order module → Payment Gateway → ERP update.

3 System Testing

Tests the **complete application as a whole**.
Example: Full e-commerce flow from login to delivery.

4 User Acceptance Testing (UAT)

Done by **business users/client** to confirm system meets business needs.
Example: Finance team verifying GST reports.

5 Regression Testing

Ensures **new changes do not break existing functionality**.
Example: After adding OTP login, checking old password login still works.

6 Smoke Testing

Quick basic testing to verify that the **build is stable**.

7 Sanity Testing

Quick check after a **small change or bug fix**.

Functional Testing Process (Step-by-Step)

1 **Requirement Analysis**
Tester/BA studies BRD, FRD, SRS

2 **Test Planning**
Define scope, tools, resources

3 **Test Case Design**
Write test cases based on business logic

4 **Test Environment Setup**

5 **Test Execution**
Execute test cases & record results

6 **Defect Logging**
Report bugs in defect management tool

7 **Re-testing & Regression Testing**

8 **Test Closure & Reporting**

Example of Functional Testing (Real-Time – Banking App)

Scenario: Fund Transfer

Step	Expected Result
Enter valid account	Account verified
Enter amount	Amount accepted
Click transfer	OTP generated
Enter OTP	Transaction successful
Check statement	Entry updated

If any step fails → **DefECT**

Example of Functional Testing (E-Commerce)

- User registration
- Product search
- Add to cart
- Apply coupon
- Make payment
- Order confirmation
- Invoice generation

Every step is functionally tested.

◆ Sample Functional Test Case Format

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
TC_01	Login with valid credentials	Enter username & password	Login successful	Pass	✓

Role of Business Analyst in Functional Testing

- Map requirements to test cases
- Review test cases with QA
- Clarify business logic

- ◆ Validate UAT scenarios
- ◆ Approve UAT sign-off
- ◆ Ensure requirement traceability
- ◆ Support defect validation

Entry & Exit Criteria of Functional Testing

✓ Entry Criteria:

- Requirements finalized
- Development completed
- Test cases ready
- Test environment available

✓ Exit Criteria:

- All critical test cases passed
- High-priority defects fixed
- UAT approval received
- Test report approved

Common Defects Found in Functional Testing

- Incorrect calculations
- Validation errors
- Broken workflows
- Role access issues
- Data mismatch
- Failed transactions
- Incorrect report values

Functional Testing vs Non-Functional Testing

Functional Testing	Non-Functional Testing
What the system does	How the system performs
Business logic	Performance, security
Example: Login works	Example: Login under 2 sec

Interview-Perfect Definition (You Can Use This)

“Functional Testing is a type of testing that validates the application’s features and business workflows against the specified requirements to ensure the system behaves as expected.”

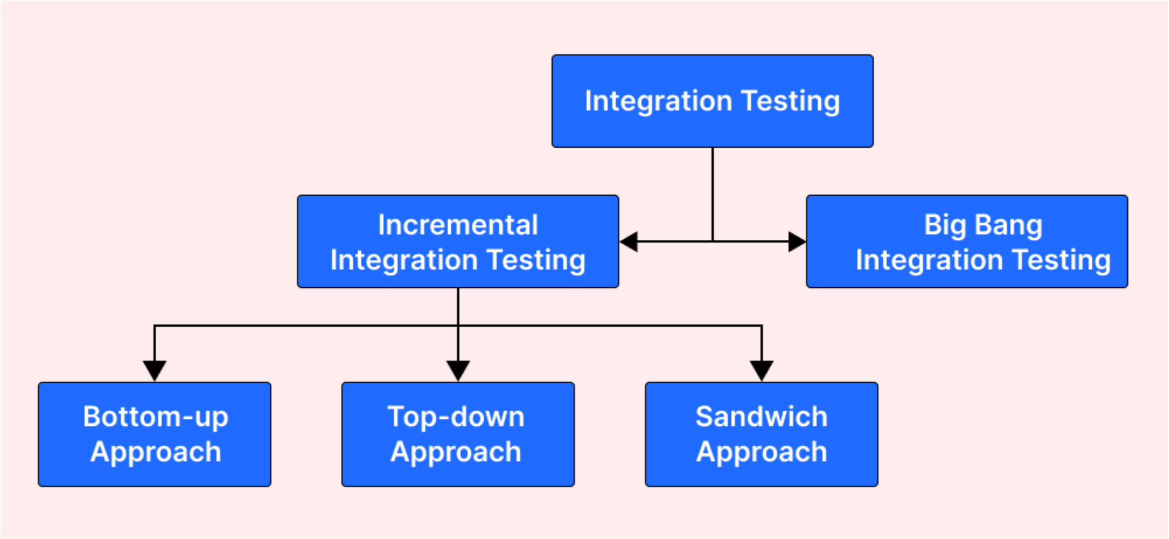
One-Line BA-Oriented Definition

“As a Business Analyst, Functional Testing ensures that all business processes are correctly implemented in the system as per BRD and FRD.”

2 Integration Testing

Checks data flow between multiple modules/systems.
Example: Order system ↔ Payment Gateway ↔ ERP.

Integration Testing



What is Integration Testing?

Integration Testing is a type of software testing where **individual modules or components are combined and tested together** to verify that they **interact with each other correctly** as per the business and functional requirements.

It ensures that:
☞ **Data flows correctly between integrated modules/systems**
☞ **Interfaces, APIs, and dependencies work as expected**

Objective of Integration Testing

✓ Verify communication between modules
✓ Validate data transfer accuracy
✓ Identify interface defects
✓ Ensure end-to-end business workflows work smoothly
✓ Detect issues early before system testing

What is Tested in Integration Testing?

- Data flow between modules
- API requests & responses
- Interface compatibility
- Error handling between systems
- Database interactions
- Third-party integrations (Payment gateway, SMS, Email, Bank, ERP, CRM)

Types of Integration Testing (In Detail)

1 Big Bang Integration Testing

All modules are integrated at once and tested together.

✅ Advantage: Easy to execute
❌ Disadvantage: Difficult to identify root cause of defect

2 Top-Down Integration Testing

Testing starts from **top-level modules (UI)** and moves downward.

Uses **stubs** for lower modules (dummy components).

✅ Advantage: Early testing of main features
❌ Lower-level defects found late

3 Bottom-Up Integration Testing

Testing starts from **lower-level modules (database/services)** and moves upward.

Uses **drivers** to simulate upper modules.

✅ Advantage: Core logic tested early
❌ UI validation done late

4 Hybrid (Sandwich) Integration Testing

Combination of **Top-Down + Bottom-Up** approach.

✅ Most commonly used in large real-time projects
✅ Balanced risk and coverage

Integration Testing Process (Step-by-Step)

1 Study integration points & APIs
2 Identify dependent modules
3 Prepare integration test scenarios
4 Design integration test cases
5 Set up test environment & data
6 Execute test cases
7 Log defects in tool (JIRA, ALM)
8 Re-test after fix
9 Perform regression testing
10 Prepare integration test report

Real-Time Examples of Integration Testing

✅ Example 1: Banking System

- Customer → Internet Banking → Core Banking → SMS Gateway
- Tests:
    - Payment debit from account
    - Balance update
    - SMS alert generated

✅ Example 2: E-Commerce Website

Modules:

- User Login
- Product Catalog
- Cart
- Payment Gateway
- Order Management
- Inventory
- Email System

Integration checks:

- Order placed → Stock reduced → Invoice generated → Email sent

✓ Example 3: ERP System

- Sales Module → Finance Module
- Purchase Module → Inventory Module
- Payroll → Accounting

Integration ensures automatic data reflection across departments.

◆ Sample Integration Test Case Format

Test Case ID	Module 1	Module 2	Test Scenario	Expected Result
IT_01	Order	Payment	Successful payment	Order confirmed & invoice generated

Role of Business Analyst in Integration Testing

❖ Identify integration points during requirement phase
❖ Define business workflows
❖ Validate data mapping between systems
❖ Review integration test cases
❖ Support testers with business logic
❖ Validate UAT integration issues
❖ Ensure traceability between BRD and integration tests

Entry & Exit Criteria of Integration Testing

✓ Entry Criteria:

- Individual modules unit tested
- Interfaces & APIs ready
- Integration test cases prepared
- Test environment available

✓ Exit Criteria:

- All critical integration defects resolved
- Smooth data flow verified
- No high-priority open defects
- Integration test report approved

Common Defects Found in Integration Testing

- Incorrect API response
- Data mismatch between modules
- Incorrect status updates
- Failed third-party connections
- Duplicate or missing records
- Timeout and interface errors

Integration Testing vs System Testing

Integration Testing	System Testing
Tests interaction between modules	Tests entire system
Focus on interfaces	Focus on overall application
Done after Unit Testing	Done after Integration Testing

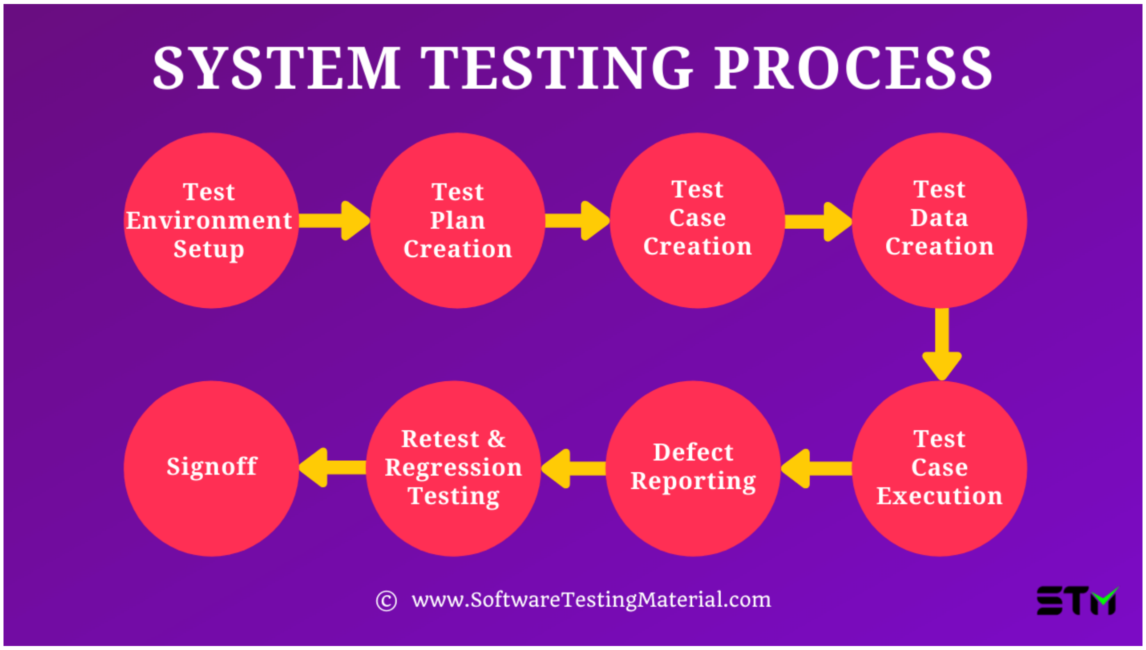
✓ Interview-Perfect Definition

"Integration Testing is performed to verify proper communication and data flow between interconnected modules and systems to ensure the application works as a combined unit."

✓ One-Line BA-Oriented Definition

"As a Business Analyst, Integration Testing ensures that data moves correctly between different business modules as per defined workflows."

3 System Testing



What is System Testing?

System Testing is a level of software testing in which the **complete and fully integrated application is tested as a whole** to verify that it meets the **functional and non-functional requirements** specified in the **SRS / FRD / BRD**.

It answers the question:
☞ **“Does the entire system work correctly as per business expectations?”**

System Testing is performed **after Integration Testing and before UAT**.

Objective of System Testing

✓ Validate end-to-end business flows
✓ Ensure the system works as a complete unit
✓ Verify all functional requirements
✓ Validate non-functional requirements (performance, security, usability)
✓ Identify defects before UAT
✓ Ensure system is production-ready

What is Covered in System Testing?

✓ Functional Coverage:

- Login & user management
- Role-based access control
- CRUD operations (Create, Read, Update, Delete)
- Reports & dashboards
- Payment & refund
- Notifications
- Error messages & validations

✓ Non-Functional Coverage:

- Performance
- Load & stress
- Security
- Usability
- Compatibility
- Recovery & reliability

Types of System Testing (In Detail)

1 Functional System Testing

Validates all features as per requirements.

2 Performance Testing

System behavior under normal & heavy load.

3 Security Testing

Checks for vulnerabilities, authentication & authorization.

4 Usability Testing

Ease of use & user experience.

5 Compatibility Testing

Across browsers, devices, OS.

6 Recovery Testing

System behavior after crash, power failure, or network failure.

7 Installation & Configuration Testing

Checks setup and deployment accuracy.

System Testing Process (Step-by-Step)

1 Requirement & SRS Review
2 Test Planning
3 Test Scenario & Test Case Preparation
4 Test Environment Setup
5 Test Data Preparation
6 Test Execution
7 Defect Logging in JIRA/ALM
8 Defect Re-testing
9 Regression Testing
10 Test Closure & System Test Report

Real-Time Examples of System Testing

✅ Example 1: Banking Application

- Login
- Check balance
- Transfer funds
- Generate statement
- Logout
  ✓ All tested end-to-end

✅ Example 2: E-Commerce Website

- User registration
- Product browse
- Add to cart
- Checkout
- Payment
- Order confirmation
- Invoice generation
- Email notification

✅ Example 3: ERP System

- Sales → Billing → Inventory → Finance
- HR → Payroll → Accounting
  All modules tested as **one system**.

◆ Sample System Test Case Format

Test Case ID	Test Scenario	Test Steps	Expected Result	Status
ST_01	Complete Order Flow	Login → Add product → Pay	Order successful	Pass

◆ Role of Business Analyst in System Testing

◈ Ensure all business requirements are covered
◈ Validate end-to-end workflows
◈ Review system test cases
◈ Support QA for business clarifications
◈ Validate defect impact on business
◈ Ensure traceability (RTM)
◈ Approve system testing completion

Entry & Exit Criteria of System Testing

✔ Entry Criteria:

- Integration Testing completed
- System test cases approved
- Test environment ready
- Test data prepared

✔ Exit Criteria:

- All critical defects fixed
- Test execution completed
- System test report prepared
- Ready for UAT

Common Defects Found in System Testing

- End-to-end process failure
- Missing validations
- Incorrect business rules

- Performance degradation
- Security loopholes
- Data inconsistency
- Incorrect report generation

◆ System Testing vs Other Testing

System Testing	Integration Testing	UAT
Tests full application	Tests module interaction	Done by business users
Technical + business	Technical focus	Business focus
Before UAT	Before System Testing	Final approval

✅ Interview-Perfect Definition

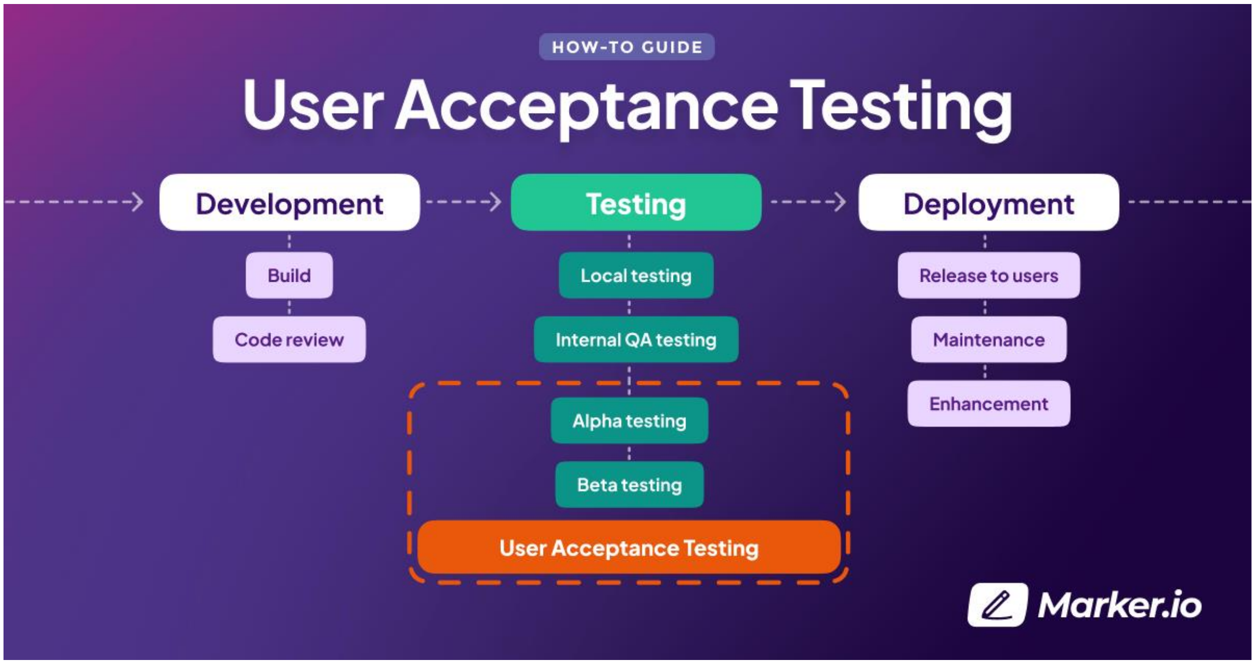
"System Testing is a testing level where the complete integrated application is validated end-to-end against functional and non-functional requirements to ensure it is ready for UAT."

✅ One-Line BA-Oriented Definition

"As a Business Analyst, System Testing ensures that all business workflows work correctly across the entire application."

End-to-end testing of the complete application.

4 User Acceptance Testing (UAT)



Done by **business users/clients** to confirm the system meets business needs.

What is UAT?

User Acceptance Testing (UAT) is the **final phase of testing** where **actual business users** verify whether the application meets their **business needs and expectations** before it is released into production.

It answers the key question:
☞ **“Does the system solve the real business problem?”**

UAT is performed **after System Testing** and **before Go-Live/Production**.

◆ Objective of UAT

✓ Validate business requirements
✓ Confirm real-world business scenarios
✓ Ensure system usability
✓ Identify gaps missed in system testing
✓ Get formal business approval
✓ Reduce production risk
✓ Ensure end-user satisfaction

Who Performs UAT?

- Business Users
- Clients / Customers
- Product Owners
- SMEs (Subject Matter Experts)
- Operations Team

Testers & Developers usually **do not perform UAT**; they only **support** it.

What is Tested in UAT?

- End-to-end business workflows
- Real life process scenarios
- Data accuracy and reports
- Role-based access
- Legal & compliance rules
- User-friendliness
- Input validations from a business perspective

Types of UAT

1 Alpha Testing

Performed at **developer site** with internal users.

2 Beta Testing

Performed by **real users in a live environment**.

3 Operational Acceptance Testing (OAT)

Validates operational readiness such as:

- Backup & recovery
- Monitoring
- Security
- Maintenance

UAT Process (Step-by-Step)

1 Requirement Review
2 UAT Planning
3 UAT Scenario & Test Case Preparation

4 UAT Environment Setup
5 UAT Test Data Preparation
6 UAT Execution by users
7 Defect Reporting
8 Defect Fix by Dev Team
9 Re-testing
10 Business Sign-Off

UAT Entry & Exit Criteria

✓ Entry Criteria:

- System testing completed
- No critical defects open
- UAT environment ready
- UAT test cases approved
- Business users identified

✓ Exit Criteria:

- All UAT scenarios executed
- All critical UAT defects fixed
- Final business sign-off received
- Go-Live approval granted

Real-Time UAT Examples

✓ Example 1: Banking Application

Business users validate:

- Account opening
- Fund transfer
- Interest calculation
- Statement generation

✓ Example 2: E-Commerce Application

Business validates:

- Order processing
- Discounts
- Taxes
- Refund flow

- Invoice & email

✓ Example 3: ERP System

Finance validates:

- Billing
- GST calculation
- Ledger posting
  HR validates:
- Payroll
- Attendance
- Salary slips

Sample UAT Test Case Format

UAT ID	Business Scenario	Steps	Expected Result	Actual Result	Status
UAT_01	Create Sales Order	Login → Create order → Save	Order created	Pass	✓

Role of Business Analyst in UAT

◆ Convert business requirements into UAT scenarios
◆ Identify UAT users
◆ Conduct UAT walkthrough sessions
◆ Support users during execution
◆ Validate UAT defects
◆ Coordinate between business & IT
◆ Track UAT progress
◆ Get UAT sign-off

Common UAT Defects

- Business rule mismatch
- Incorrect reports
- Missing validations
- Workflow gaps
- Usability issues
- Data mismatch

Difference Between System Testing & UAT

System Testing	UAT
Done by QA	Done by business users
Technical focus	Business focus
Checks system functionality	Checks business usability
Before UAT	Final testing phase

✓ Interview-Perfect Definition

“User Acceptance Testing is the final testing phase where business users validate the application against real business scenarios to decide whether the system is ready for production.”

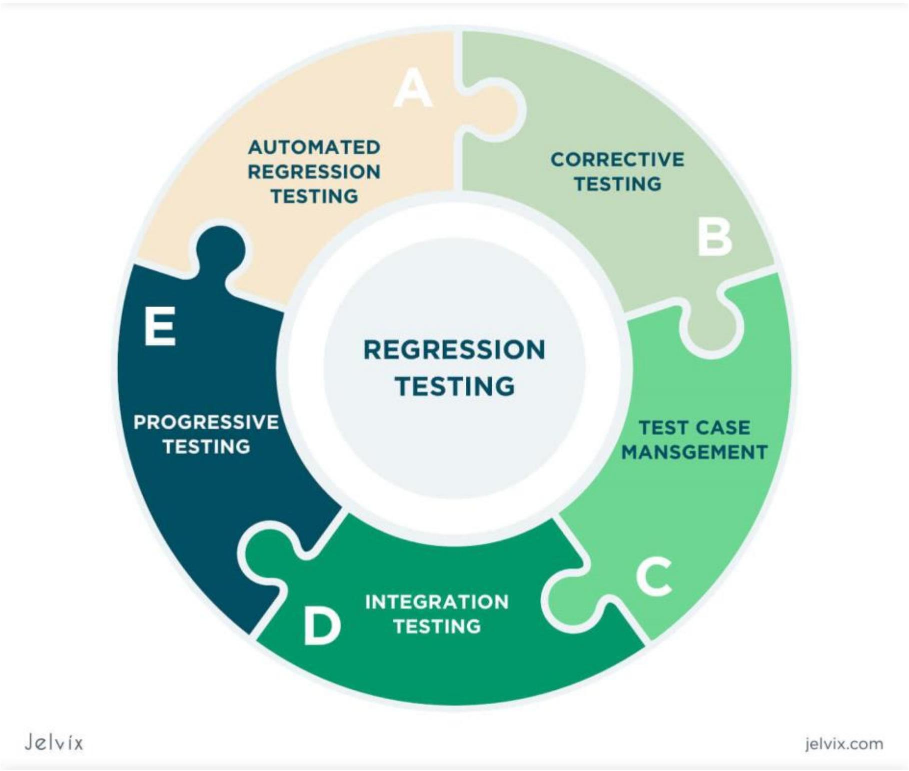
✓ One-Line BA-Oriented Definition

“As a Business Analyst, UAT ensures that the developed solution truly fulfills business objectives before production deployment.”

✓ Key Documents Used in UAT

- BRD
- FRD
- SRS
- UAT Test Cases
- UAT Sign-off Document
- Defect Tracker

5 Regression Testing



Ensures new changes do not break existing functionality.

Regression Testing

What is Regression Testing?

Regression Testing is a type of software testing performed to **verify that recent changes (bug fixes, enhancements, or new features) have not broken or negatively impacted the existing functionality** of the application.

In simple words:
☞ **“It ensures old features still work after new changes.”**

Regression Testing is done **after bug fixes, code changes, patches, or configuration updates**.

Objective of Regression Testing

✓ Ensure system stability after changes
✓ Detect unintended side effects
✓ Maintain software quality
✓ Protect existing working functionality

✓ Reduce production failures
✓ Build confidence in every release

When is Regression Testing Performed?

- After defect fix
- After change in requirement
- After enhancement or new feature addition
- After backend or database change
- Before every release or deployment
- After integration of new modules

What is Covered in Regression Testing?

- Core business functionalities
- Critical workflows
- Frequently used features
- Integration points
- Reports & calculations
- User roles & permissions

Types of Regression Testing

1 Corrective Regression Testing

When **no change in existing functionality**, only defect is fixed.

2 Selective Regression Testing

Only **impacted modules** are tested.

3 Progressive Regression Testing

When **new features are added**.

4 Full Regression Testing

Entire application is tested end-to-end.

5 Partial Regression Testing

Tests only **related modules** after changes.

Regression Testing Process (Step-by-Step)

1 Analyze the change/defect fix
2 Identify impacted areas
3 Select regression test cases
4 Prepare test data
5 Execute regression test cases
6 Log defects
7 Re-test after fix
8 Generate regression report

Real-Time Examples of Regression Testing

✓ Example 1: Banking Application

Change: OTP login added
Regression Check:

- Old password login
- Fund transfer
- Balance check
- Statement download

✓ Example 2: E-Commerce Website

Change: New discount feature added
Regression Check:

- Add to cart
- Checkout
- Payment
- Order confirmation
- Invoice generation

Example 3: ERP System

Change: GST calculation updated
Regression Check:

- Billing
- Ledger posting
- Reports
- Tax summary

Manual vs Automated Regression Testing

Manual	Automated
Human execution	Tool-based execution
Time consuming	Fast execution
Suitable for small apps	Ideal for large apps
Low cost initially	High initial investment
Error-prone	More reliable

◆ Role of Business Analyst in Regression Testing

◆ Identify business-critical impacted areas
◆ Ensure requirement traceability
◆ Validate business workflow stability
◆ Review regression scope
◆ Support UAT regression
◆ Approve release readiness from business side

Entry & Exit Criteria of Regression Testing

✓ Entry Criteria:

- Code changes completed
- Unit & integration testing done
- Regression test cases identified
- Test environment ready

✓ Exit Criteria:

- All regression test cases executed
- No high-priority regression defects open
- Regression report approved
- Ready for release

Common Defects Found During Regression Testing

- Broken workflows
- Calculation mismatches
- Failed integrations
- Role-based access issues
- Report inaccuracies
- Performance degradation

Regression Testing vs Re-Testing

Regression Testing	Re-Testing
Tests unaffected areas	Tests only the fixed defect
Checks overall system stability	Checks specific fix
Broad scope	Narrow scope

✓ Interview-Perfect Definition

“Regression Testing is performed to ensure that recent changes in the application do not adversely affect the existing working functionalities.”

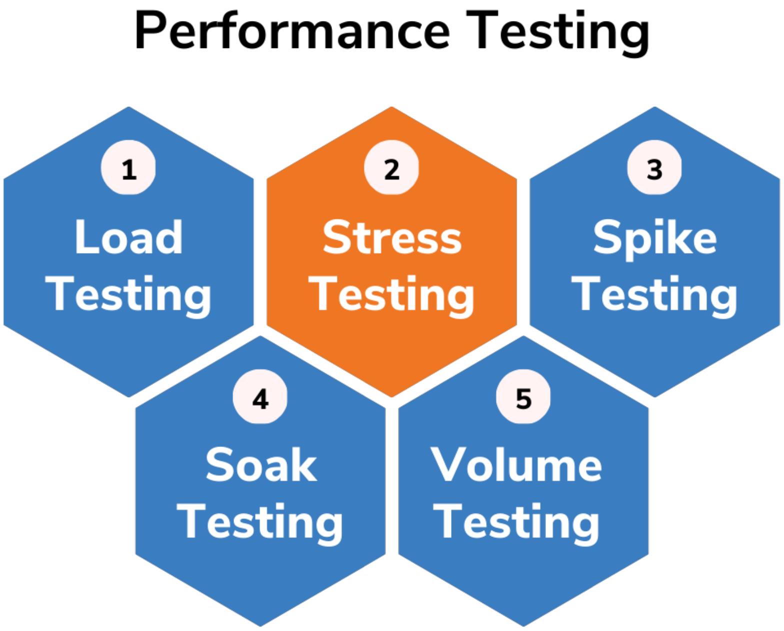
✓ One-Line BA-Oriented Definition

“As a Business Analyst, Regression Testing ensures that business-critical processes remain stable after system changes.”

✓ Key Documents Used in Regression Testing

- BRD / FRD
- Test Cases
- Regression Test Suite
- RTM (Requirement Traceability Matrix)
- Defect Report
- Release Notes

6 Performance Testing



Checks speed, load, and stability.

Performance Testing

◆ What is Performance Testing?

Performance Testing is a type of **non-functional testing** that checks how a software application performs under different workloads.
It measures:

- **Speed (Response Time)**
- **Scalability**
- **Stability**
- **Reliability**

In simple words:
☞ **“It ensures the system works fast, smoothly, and without crashing under load.”**

◆ Objectives of Performance Testing

✓ Check system response time
✓ Identify performance bottlenecks
✓ Validate system stability under heavy load
✓ Ensure scalability
✓ Verify resource usage (CPU, memory)
✓ Prevent performance failures in production

Key Performance Parameters (Metrics)

Metric	Meaning
Response Time	Time taken to respond to a request
Throughput	Number of transactions per second
Load	Number of concurrent users
Latency	Delay in data transmission
Error Rate	% of failed requests
CPU Utilization	Processor usage
Memory Usage	RAM consumption

Types of Performance Testing (In Detail)

1 Load Testing

Checks system behavior under **expected user load**.

✅ Example:
10,000 users logging in at the same time.

2 Stress Testing

Checks system behavior under **extreme or unexpected load**.

✅ Example:
Sudden traffic spike during a sale.

3 Spike Testing

Tests **sudden rise and fall** in traffic.

✅ Example:
Flash sale from 5,000 to 50,000 users instantly.

4 Endurance (Soak) Testing

Tests system for **long duration** under normal load.

✅ Example:
Website running continuously for 24–72 hours.

5 Volume Testing

Tests system with a **huge amount of data**.

✅ Example:
1 crore customer records in database.

6 Scalability Testing

Checks system ability to **scale up or down** based on load.

✅ Example:
Add servers and see if performance improves.

◆ Performance Testing Process (Step-by-Step)

1 Identify critical business processes
2 Define performance requirements (SLA)
3 Select performance testing tools
4 Prepare test environment
5 Create performance test scenarios & scripts
6 Execute test
7 Monitor system behavior
8 Analyze results
9 Tune system (code, DB, server)
10 Re-test & final reporting

Real-Time Examples of Performance Testing

✅ Example 1: Banking Application

- 1 lakh users checking balance simultaneously
- 10,000 concurrent fund transfers
- Measure response time & failure %

✅ Example 2: E-Commerce Website

- Big Billion Day / Amazon Sale
- Millions of users add to cart & pay
- System must not crash

✅ Example 3: ERP System

- Salary processing for 50,000 employees at once
- Data must process within defined time

Sample Performance Test Scenario

Scenario	Load	Expected Result
Login	5,000 users	Response < 3 sec
Payment	2,000 users	Success rate > 99%
Report Download	1,000 users	< 5 sec

Role of Business Analyst in Performance Testing

🔹 Define business-specific performance requirements
🔹 Identify critical transactions
🔹 Set SLA (Service Level Agreements)
🔹 Validate business-impact of performance issues
🔹 Prioritize performance defects
🔹 Support UAT performance validation
🔹 Sign-off from business perspective

◆ Entry & Exit Criteria of Performance Testing

✓ Entry Criteria:

- Application stable build available
- Test environment ready
- Performance scenarios identified
- Test data prepared

✓ Exit Criteria:

- Performance goals achieved
- No critical performance defects
- Test report approved
- System ready for production

Common Performance Issues Found

- Slow page loading
- Server crash
- Memory leakage
- Database locking
- Network latency
- High error rates during peak load

Performance Testing vs Load Testing

Performance Testing	Load Testing
Overall performance check	Focus on expected load
Broad scope	Narrow scope
Includes stress, spike, endurance	Only load condition

✓ Interview-Perfect Definition

“Performance Testing is a non-functional testing technique used to evaluate the speed, scalability, and stability of an application under varying workloads.”

✓ One-Line BA-Oriented Definition

"As a Business Analyst, Performance Testing ensures that the system meets business SLAs and can handle real-world user load without failure."

✅ Tools Used for Performance Testing

- JMeter
- LoadRunner
- NeoLoad
- Gatling
- BlazeMeter

7 Security Testing

Checks data protection and system vulnerabilities.

What is Security Testing?

Security Testing is a type of **non-functional testing** performed to **identify vulnerabilities, threats, and risks** in a software application and to ensure that:

- **Data is protected**
- **Access is controlled**
- **System is safe from cyber attacks**
- **Confidential information is not leaked**

In simple words:
☞ **"Security Testing checks whether the system is safe from hackers and unauthorized access."**

◆ Objectives of Security Testing

✔ Prevent unauthorized access
✔ Protect sensitive data (PII, financial data)
✔ Identify system vulnerabilities
✔ Ensure data integrity
✔ Ensure compliance with security standards
✔ Avoid financial & reputational loss
✔ Build customer trust

What is Tested in Security Testing?

- User authentication
- Authorization & access control
- Data encryption
- Session management
- Input validation
- APIs and web services
- File uploads/downloads
- Database security
- Network security

Key Security Principles (CIA Triad)

Principle	Meaning
Confidentiality	Data accessed only by authorized users
Integrity	Data is accurate and not altered
Availability	System is accessible when required

Types of Security Testing (In Detail)

1 Vulnerability Scanning

Automated scan to find **known security flaws**.

2 Penetration Testing (Ethical Hacking)

Simulated **real-world hacker attacks** to find system weaknesses.

3 Security Scanning

Checks **network, ports, and configurations**.

4 Risk Assessment

Identifies security risks and their **business impact**.

5 Ethical Hacking

Certified security experts try to **break into the system legally**.

6 Compliance Testing

Checks if system follows **standards and regulations** such as:

- ISO 27001
- GDPR
- PCI-DSS
- HIPAA

7 Access Control Testing

Validates **role-based access control (RBAC)**.

Common Security Attacks Tested

- SQL Injection
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- Brute force attack
- Session hijacking
- Malware injection
- Man-in-the-Middle attack
- Unauthorized API access

Security Testing Process (Step-by-Step)

1 Understand application architecture
2 Identify sensitive data & entry points
3 Identify possible threats
4 Select security testing tools
5 Perform vulnerability scanning
6 Conduct penetration testing
7 Log security defects
8 Fix vulnerabilities
9 Re-test
10 Prepare security test report

Real-Time Examples of Security Testing

✓ Example 1: Banking Application

- Attempt login with invalid credentials
- Try multiple wrong OTPs
- Test encrypted data transmission
- Check session timeout after inactivity

✓ Example 2: E-Commerce Website

- Attempt fake payment manipulation
- Upload malware using file upload feature
- Access admin panel as normal user

✅ Example 3: ERP System

- HR data accessed by finance user
- Salary data exposure
- Unauthorized report downloads

Sample Security Test Case

Test Case ID	Scenario	Steps	Expected Result
SEC_01	Unauthorized Access	Try access admin URL as user	Access denied

Role of Business Analyst in Security Testing

- Identify sensitive business data
- Define security requirements in BRD/FRD
- Identify regulatory & compliance needs
- Review security test scenarios
- Validate business impact of vulnerabilities
- Support UAT security validation
- Ensure security sign-off before Go-Live

◆ Entry & Exit Criteria of Security Testing

✅ Entry Criteria:

- Application build ready
- Security requirements finalized
- Test environment available
- Security test cases prepared

✅ Exit Criteria:

- All critical vulnerabilities fixed
- No high-risk open issues
- Security test report approved
- System certified for production

Common Defects Found in Security Testing

- Weak password policy
- Missing session timeout
- Improper role access
- Data exposed in URL
- Unencrypted data transmission
- Vulnerable APIs
- Open ports & firewall misconfiguration

◆ Security Testing vs Performance Testing

Security Testing	Performance Testing
Protects from attacks	Ensures speed & load
Focused on data safety	Focused on system speed
Prevents breaches	Prevents crashes

✓ Interview-Perfect Definition

“Security Testing is a non-functional testing process used to identify vulnerabilities in a system and ensure that data and resources are protected from unauthorized access.”

✓ One-Line BA-Oriented Definition

“As a Business Analyst, Security Testing ensures that critical business and customer data is protected as per compliance and regulatory standards.”

✓ Tools Used for Security Testing

- Burp Suite
- OWASP ZAP
- Nessus
- Acunetix
- Metasploit
- Wireshark

✓ Key Documents Used in Security Testing

- BRD / FRD
- Security Requirement Specification

- Vulnerability Report
- Penetration Test Report
- Compliance Report

✓ Role of Business Analyst in the Testing Phase

◆ Review Test Cases against BRD/FRD
◆ Support QA team with business clarifications
◆ Validate UAT scenarios
◆ Coordinate with business users for UAT
◆ Verify defect impact on business
◆ Sign-off on UAT completion

✓ Entry & Exit Criteria of Testing Phase

◆ Entry Criteria:

- Development completed
- Test cases prepared
- Test environment ready

◆ Exit Criteria:

- All critical defects fixed
- Test cases executed
- UAT approval received
- Test summary report prepared

✓ Simple Interview Answer

“The Testing Phase is the stage in SDLC where the developed application is verified against business and functional requirements to identify defects and ensure the system is ready for deployment.”