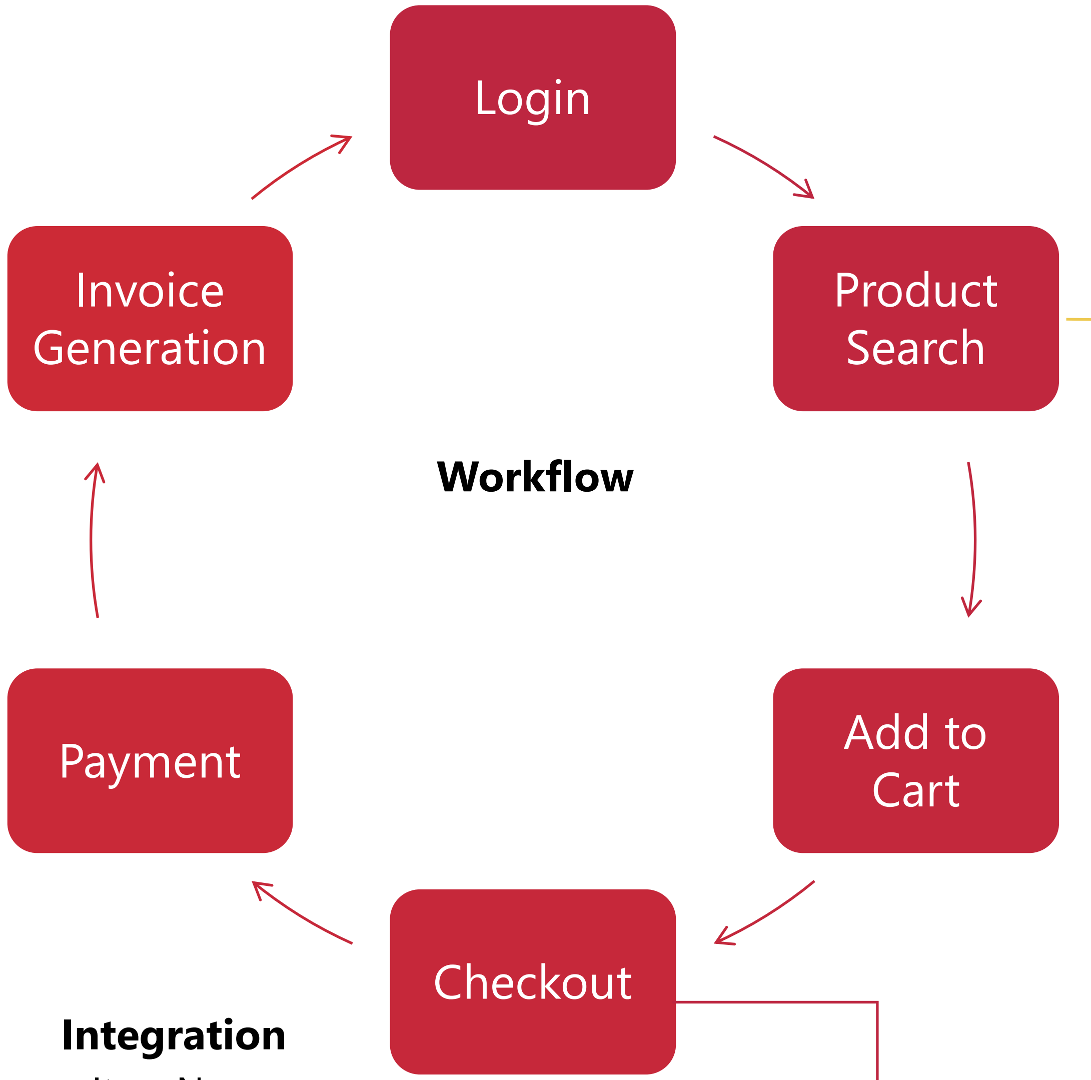
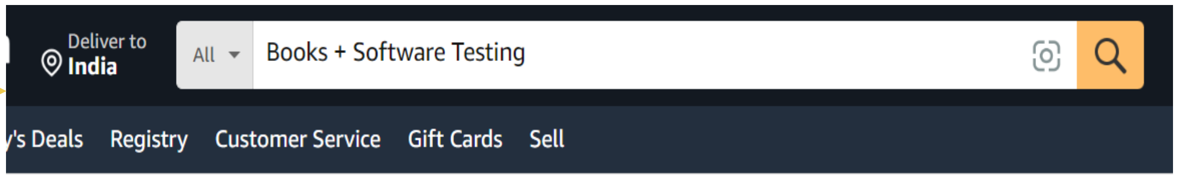
testhouse

Training Session 4

System Testing





Field Level Validation

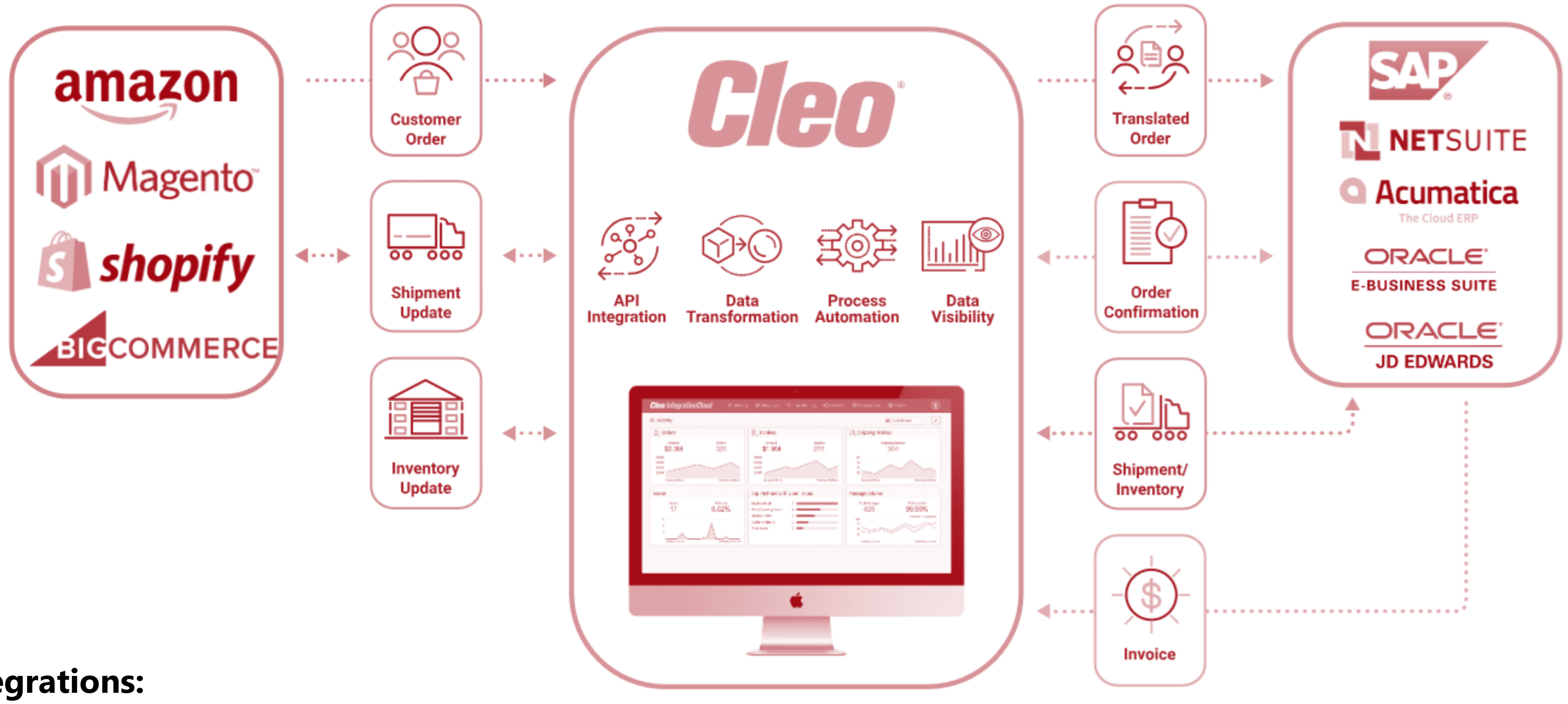
Integration

- Item Name
- Seller Name
- Type of Item
- Item Price

Business Rule

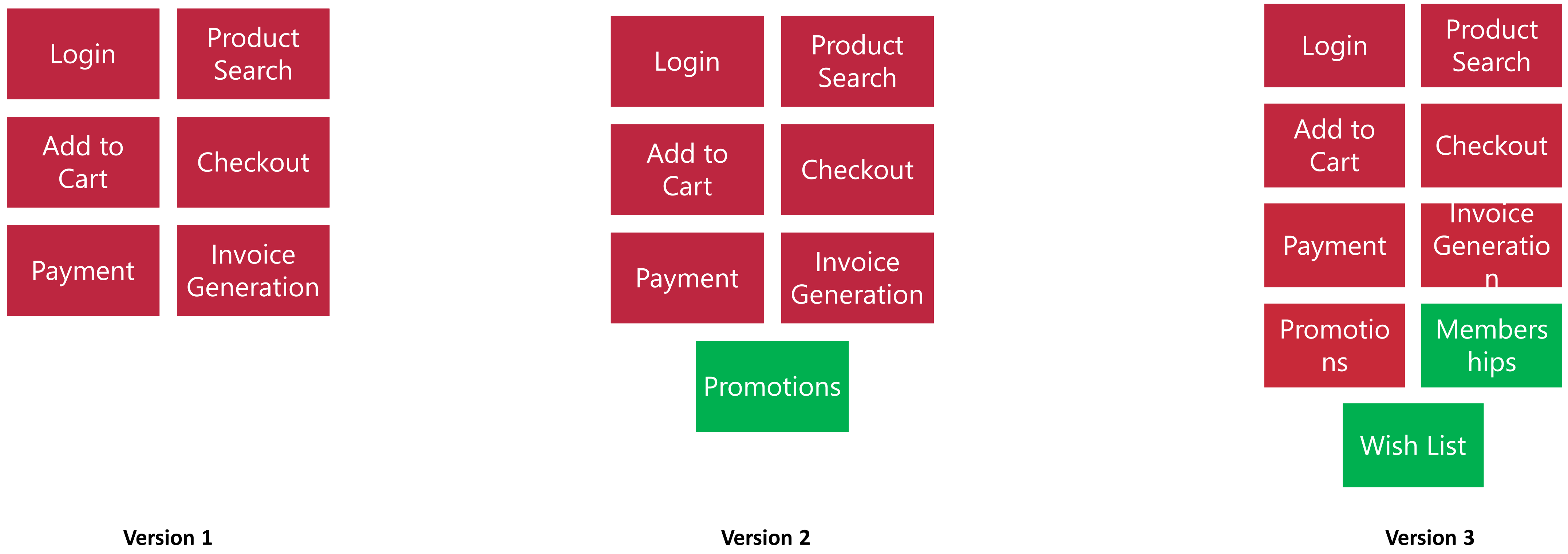
The e-commerce platform sets a minimum purchase amount of $50 for free shipping

System Integration Testing

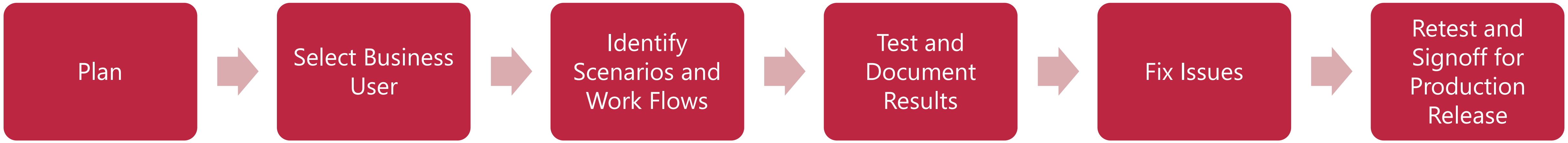


Types of Integrations:
- API
- Data Integration
- Middleware
- Workflow Automation Tools like IFTTT, Zapier
- Webhooks

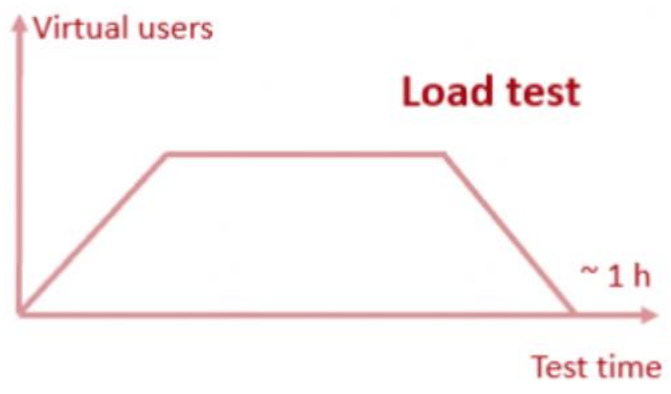
Regression Testing

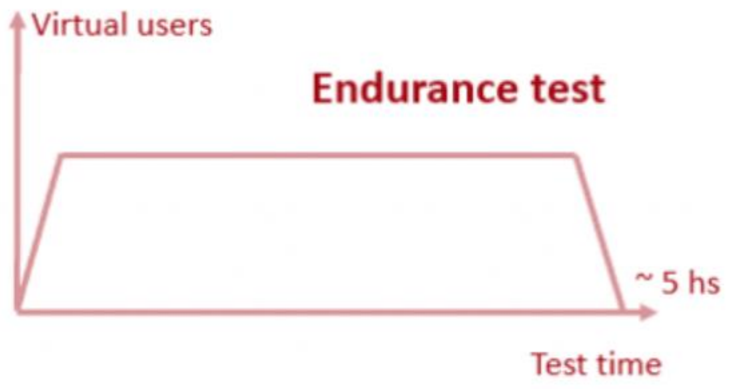


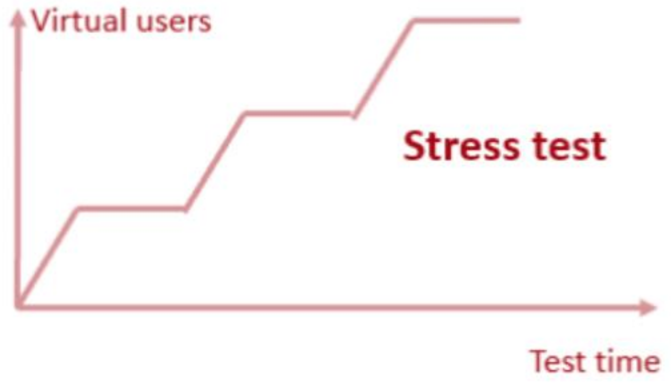
User Acceptance Testing


Plan
Select Business User
Identify Scenarios and Work Flows
Test and Document Results
Fix Issues
Retest and Signoff for Production Release

Performance Testing







User Experience Testing

Gather Feedback from Clickable Prototypes

Conduct usability tests to evaluate product functionality.

Usability Tests Conducted in Labs or Remote

Monitor user behavior with analytics and heatmaps.

Test across various devices for consistent user experience.

Ensure responsive design for mobile and desktop users.

Security Testing

testhouse

SAST

Review the source code of an application for security vulnerabilities to secure code

Secret Scanning

Scan codebase for detecting and preventing any secrets like passwords, api keys, tokens in code repos

Software Composition Analysis

Find and fix the vulnerabilities found in open-source software libraries

DAST

Identify vulnerabilities present in the web/ mobile applications through simulated attacks

Operational Acceptance Testing

Verify Operational Readiness of an Application for Production Move

Focuses on Recovery, Backup and Maintainence Process

Testing is done in an Pre-Production Environment

Verifies System Documentation and Training Adequacy

Verify Effectiveness of system monitoring and logging capabilities

Pick a Unique Topic and Submit a Presentation of 2 Slides covering What , How and Why?

testhouse

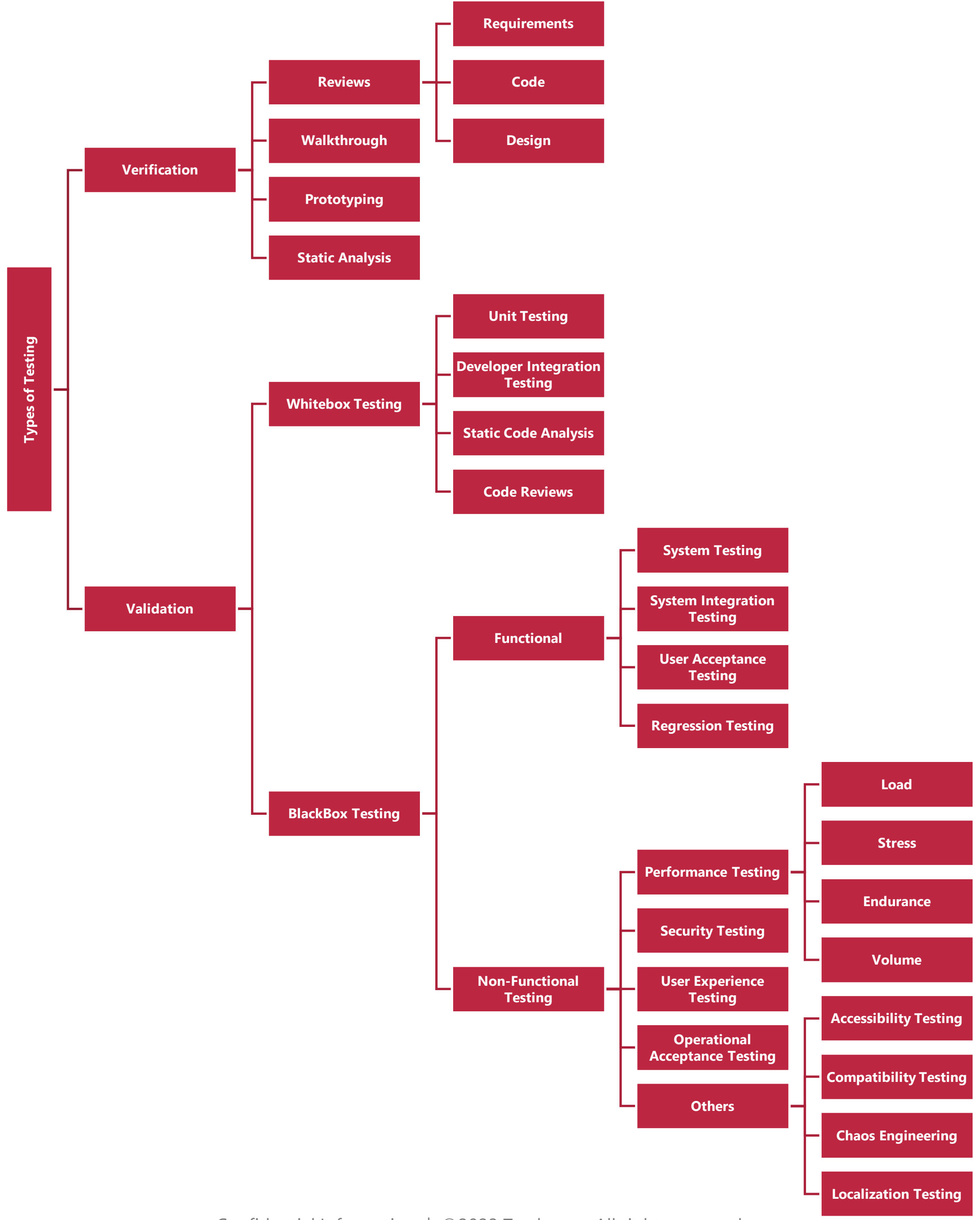
- Accessibility Testing
- Localization Testing
- Volume Testing
- Chaos Engineering
- Comptability Testing
- Compliance Testing
- Concurrency Testing
- Smoke Testing
- Sanity Testing
- Installation Testing
- Card Sorting
- Crowd Sourced Testing
- A/B Testing
- Beta Testing
- Exploratory Testing
- Gray Box Testing
- Database Testing

testhouse

A Quality Engineering Solutions Company

contact@testhouse.net
www.testhouse.net

United Kingdom (HQ)	**United States**	**United States**	**Australia**	**Middle East**	**India**
Level 18, 40 Bank Street Canary Wharf London E14 5NR, UK	260, Madison Avenue 8th Floor New York 10016	10100 Santa Monica Boulevard #300 Los Angeles California 90067	Level 35, Tower One International Towers, 100 Barangaroo Avenue Sydney, NSW 2000, Australia	Module No. 3, Office No. 214 Block-B Business Village Al-Maktoum Street, Deira, Dubai	2nd Floor, Nila Building Technopark Campus Trivandrum 695581
+44 20 8555 5577	**+1 917 793 9266**	**+1 224 323 6188**	**+61 452 043 774**	**+971 459 16013**	**+91 471 270 0117**


Types of Testing
Verification
Reviews
Requirements
Code
Design
Walkthrough
Prototyping
Static Analysis
Validation
Whitebox Testing
Unit Testing
Developer Integration Testing
Static Code Analysis
Code Reviews
BlackBox Testing
Functional
System Testing
System Integration Testing
User Acceptance Testing
Regression Testing
Non-Functional Testing
Performance Testing
Load
Stress
Endurance
Volume
Security Testing
User Experience Testing
Operational Acceptance Testing
Others
Accessibility Testing
Compatibility Testing
Chaos Engineering
Localization Testing

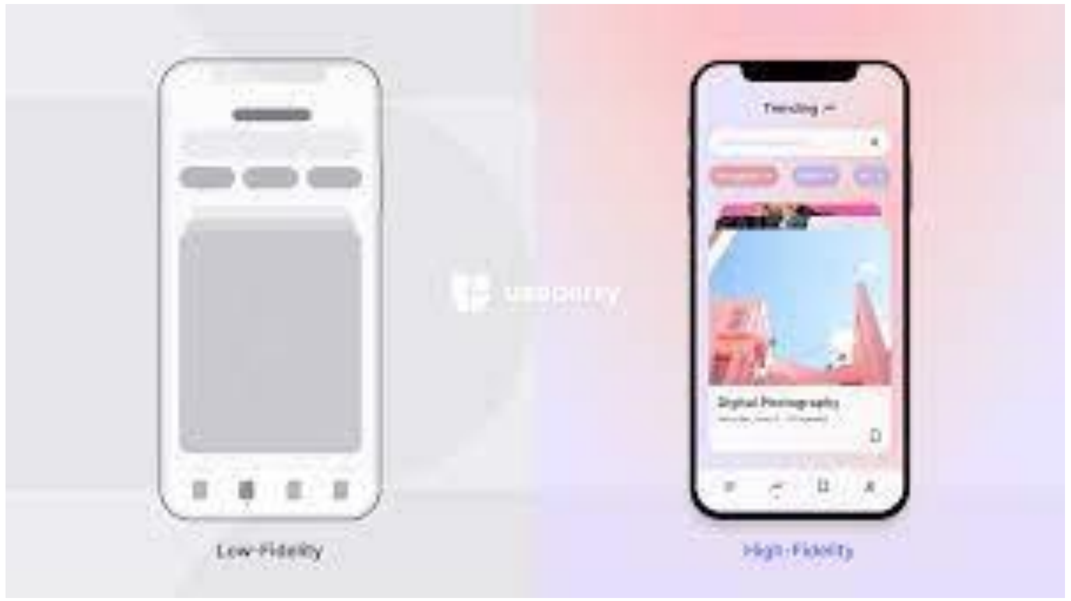
Reviews and Walkthroughs

Aspect	Reviews	Walkthroughs
Focus	Evaluate Overall Quality	Gain Better Understanding
Participants	Management, technical staff, and quality assurance members.	Author, colleagues, cross-functional team members, and sometimes customers.
Formality	Highly Formal	Less Formal
Purpose	Ensure Final Product Meet Standards or Specifications	Gain Insights, Improve Understanding, Find Defects

Prototypes

- Visualizes concepts, enables early feedback.
- Identifies design flaws and usability issues early.
- Enhances user involvement and stakeholder communication.
- Facilitates iterative development and refinement.
- Reduces development time and costs long-term.

Protype Low Fidelity and High Fidelity



Black Box and White Box Testing

testhouse

Aspect	White Box	Black Box
Knowledge of Code	Requires internal code knowledge.	No internal code knowledge.
Focus	Tests internal structures, logic.	Tests external functionalities.
Test Basis	Based on code paths.	Based on requirements, specs.
Tester's Skill	Needs programming knowledge.	No programming knowledge needed.
Test Level	Common in unit testing.	Used in system testing.

Code Static Analysis

- Analyzes code without executing the program.
- Detects potential errors and code inefficiencies early.
- Improves code quality and standards compliance.
- Identifies security vulnerabilities and risky constructs.
- Facilitates code reviews and maintenance processes.

Unit Testing

Validates individual software components' functionality.

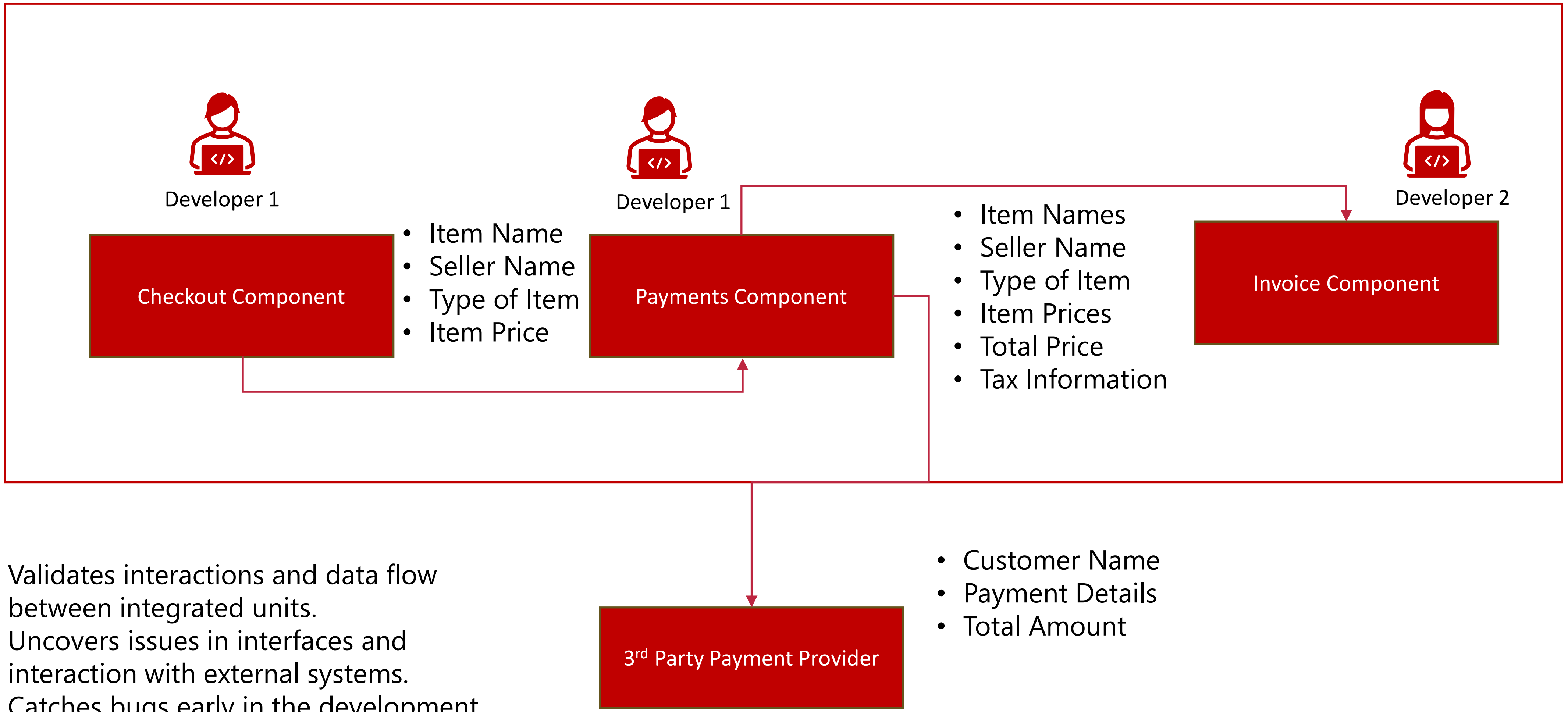
Utilize frameworks like JUnit, NUnit, or PyTest.

Mocking used to simulate external dependencies.

Carried out by the Developer

Enhances code quality and maintainability.

Developer Integration Testing



- Validates interactions and data flow between integrated units.
- Uncovers issues in interfaces and interaction with external systems.
- Catches bugs early in the development cycle, reducing costs.
- Checks if combined units meet specified requirements and functionalities.

testhouse

A Quality Engineering Solutions Company

contact@testhouse.net
www.testhouse.net

United Kingdom (HQ)	United States	United States	Australia	Middle East	India
Level 18, 40 Bank Street Canary Wharf London E14 5NR, UK	260, Madison Avenue 8th Floor New York 10016	10100 Santa Monica Boulevard #300 Los Angeles California 90067	Level 35, Tower One International Towers, 100 Barangaroo Avenue Sydney, NSW 2000, Australia	Module No. 3, Office No. 214 Block-B Business Village Al-Maktoum Street, Deira, Dubai	2nd Floor, Nila Building Technopark Campus Trivandrum 695581
+44 20 8555 5577	**+1 917 793 9266**	**+1 224 323 6188**	**+61 452 043 774**	**+971 459 16013**	**+91 471 270 0117**

What are the different Characteristics of An Application?

- Functionality
- Scalability
- Reliability
- Usability
- Performance
- Maintainability
- Security
- Portability
- Quality of Service
- Compatability

What is the Impact of Software Failures

- Financial Loss
- Loss of Employee Productivity
- Data and Security Breach
- Customer Dissatisfaction
- Safety Hazard
- Compliance Lawsuits and Fines
- Reputation Damage

Examples of Software Failures

Boeing 737 MAX Crashes (2018-2019)

- **Business Impact**: Grounding of the entire 737 MAX fleet, loss of trust in Boeing, significant financial losses, and legal consequences.
- **Reason for Defect**: A software flaw in the Maneuvering Characteristics Augmentation System (MCAS), which was designed to prevent the plane from stalling. The system received erroneous sensor data, causing it to forcefully push down the nose of the plane, leading to two fatal crashes.

Knight Capital Group Trading Disaster (2012)

- **Business Impact**: A loss of $440 million in about 45 minutes, severely damaging the company's financial position and leading to its acquisition.
- **Reason for Defect**: A software glitch in the automated high-frequency trading system. An old, unused piece of software was accidentally reactivated, which interacted wrongly with the new system, causing rapid, unintended stock trades.

HealthCare.gov Website Failure (2013)

- **Business Impact**: Damaged public perception of the Affordable Care Act, technical and customer service costs, and political fallout.
- **Reason for Defect**: The website, intended to provide health insurance under the Affordable Care Act, was plagued with issues like slow performance, error messages, and crashes. The problems were due to inadequate testing, high user traffic, and complex system integrations.

testhouse

What is a Software Defect?

- A software defect, also known as a bug, is a problem in a computer program that makes it work incorrectly or unexpectedly.

Incorrect Requirement

Programming Error

Testing Error

Data

Integration

Design Flaw

Poor Configuration

H/W or S/W Compatibility

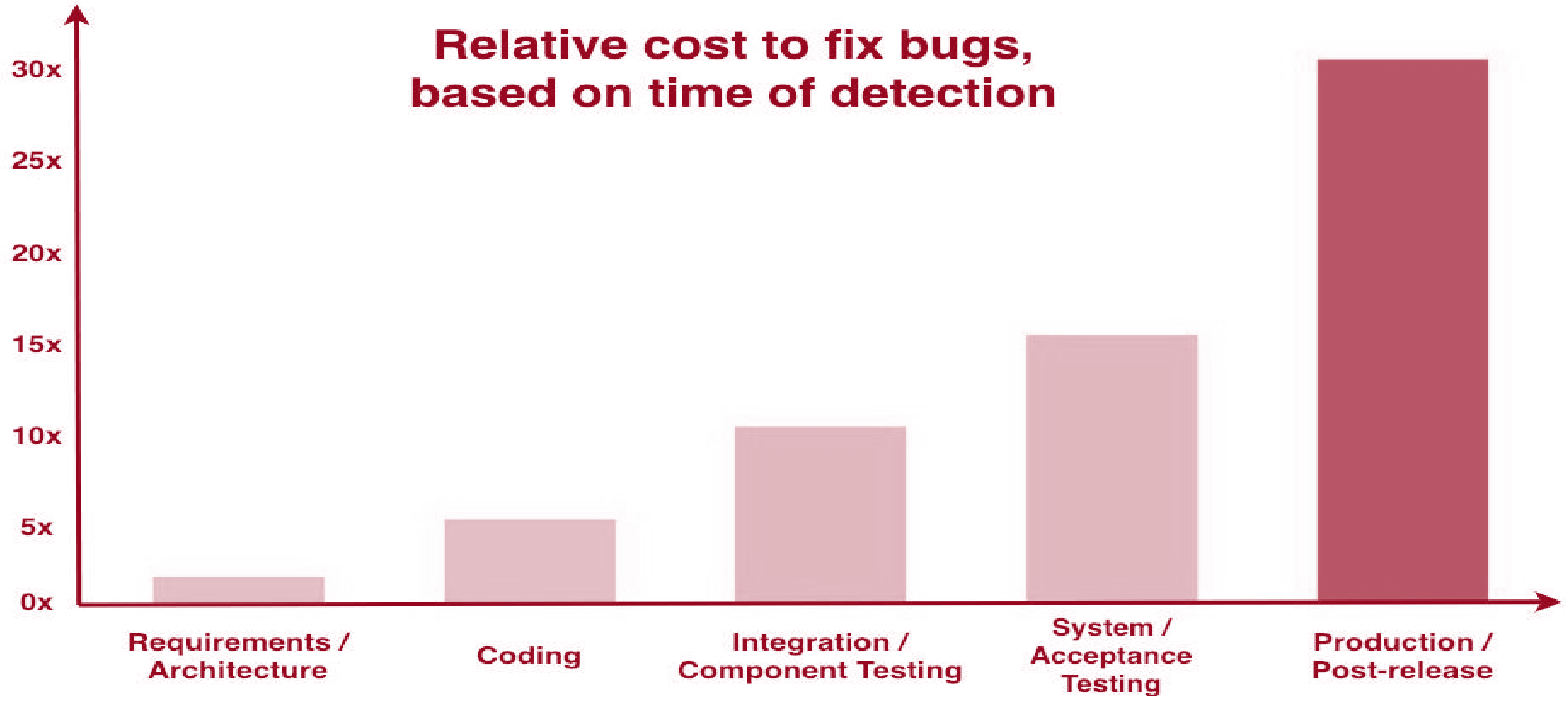
Poor UI Design

Poor Exception Handling

Why Test Software?

- Ensures quality and functionality
- Guarantees reliability and security
- Assesses performance under various conditions
- Enhances user satisfaction and experience
- Reduce Cost by Finding Defects Early
- Ensures legal and regulatory compliance
- Simplifies future maintenance and updates

Cost of Fixing Defects



Types of Applications

- **Cloud Based Application**
  - IAAS
  - PAAS
  - SAAS
- **Legacy Application**
  - Desktop
  - AS 400
  - Mainfram es
- **Digital Application**
  - Websites
  - Mobile Apps
  - Chatbots
- **Enterprise Applications**
  - CRM
  - SCM
  - ERP
  - PLM
  - HRMS
  - CMS
- **AI**
- **IOT Devices**
- **Embedded**
- **Gaming**
- **Virtual Reality**

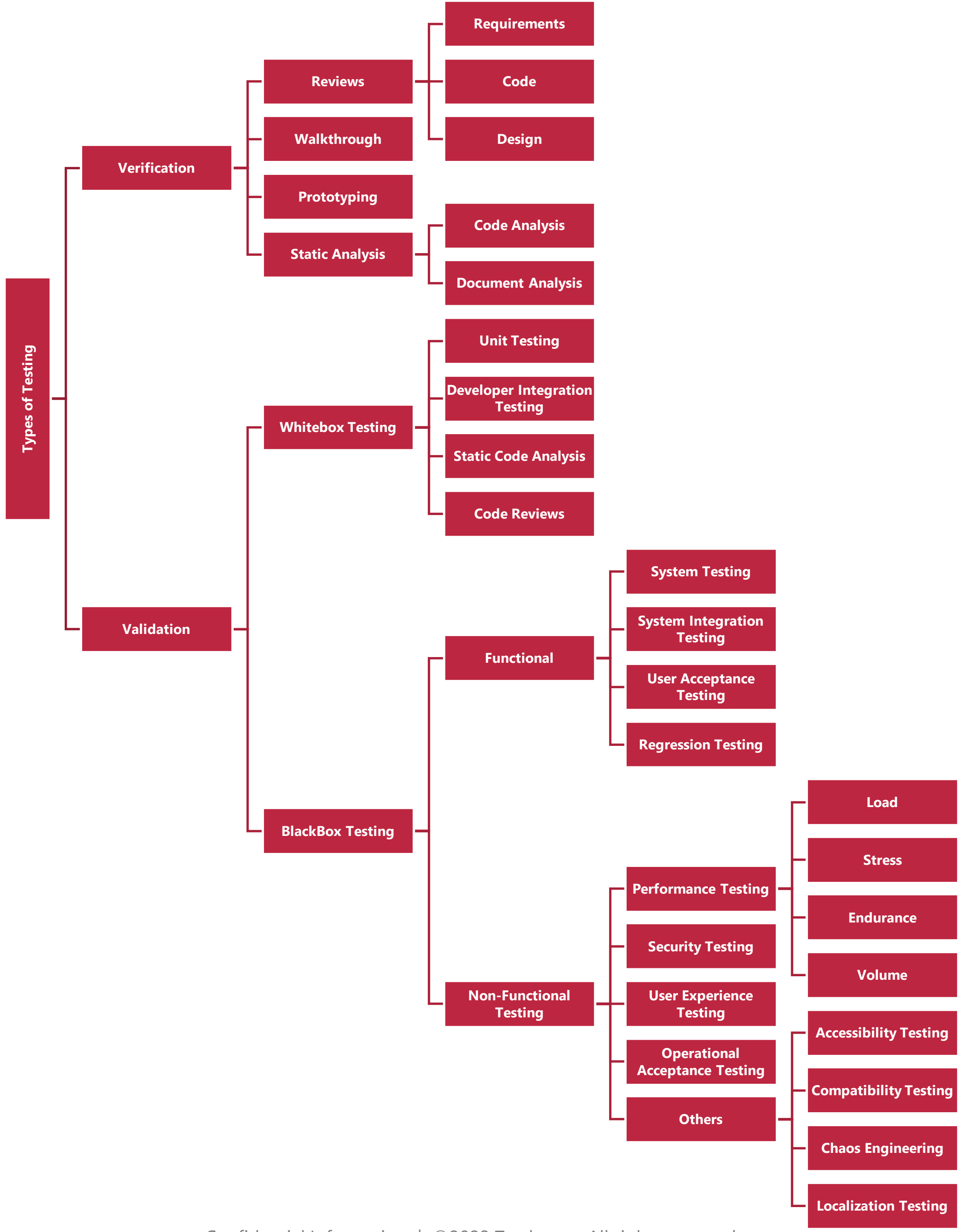
Software Testing Principles

testhouse

- Presence of Defects : Testing can indicate defects, not their absence.
- Exhaustive Testing Impossibility : Complete testing is unfeasible; prioritize based on risk.
- Early Testing : Begin testing early in the software development lifecycle.
- Defect Clustering : Defects tend to cluster; focus on high risk areas.
- Pesticide Paradox : Regularly revise tests to uncover new bugs.
- Context Dependent Testing : Tailor testing strategies to the specific software and context.
- Absence of Errors Fallacy : Software without defects may still not meet user needs.

Verification & Validation

Focus Area	Verification	Validation
Objective	The process of evaluating work-products of a development phase to ensure the software at that stage meets the specified requirements.	The process of evaluating the final product to ensure it meets the business and user requirements
Activities		Actual testing of the functionality and behavior of the software.
Outcome	Are we building the product, right?	Are we building the right product?


Types of Testing
Verification
Reviews
Requirements
Code
Design
Walkthrough
Prototyping
Static Analysis
Code Analysis
Document Analysis
Validation
Whitebox Testing
Unit Testing
Developer Integration Testing
Static Code Analysis
Code Reviews
BlackBox Testing
Functional
System Testing
System Integration Testing
User Acceptance Testing
Regression Testing
Non-Functional Testing
Performance Testing
Load
Stress
Endurance
Volume
Security Testing
User Experience Testing
Operational Acceptance Testing
Others
Accessibility Testing
Compatibility Testing
Chaos Engineering
Localization Testing

A Quality Engineering Solutions Company

contact@testhouse.net
www.testhouse.net

United Kingdom (HQ)
Level 18, 40 Bank Street
Canary Wharf
London E14 5NR, UK
+44 20 8555 5577

United States
260, Madison Avenue
8th Floor
New York 10016
+1 917 793 9266

United States
10100 Santa Monica Boulevard
#300 Los Angeles
California 90067
+1 224 323 6188

Australia
Level 35, Tower One International
Towers, 100 Barangaroo Avenue
Sydney, NSW 2000, Australia
+61 452 043 774

Middle East
Module No. 3, Office No. 214
Block-B Business Village
Al-Maktoum Street, Deira, Dubai
+971 459 16013

India
2nd Floor, Nila Building
Technopark Campus
Trivandrum 695581
+91 471 270 0117