



# ĐẢM BẢO CHẤT LƯỢNG & KIỂM THỬ PHẦN MỀM

## CHƯƠNG 2 QUY TRÌNH KIỂM THỬ PHẦN MỀM

09/2022



- **Các vấn đề liên quan**
- **Software Testing Life Cycle**
- **Software Testing Process**
- **Test Plan**
- **Test case**
- **Requirements Traceability Matrix**

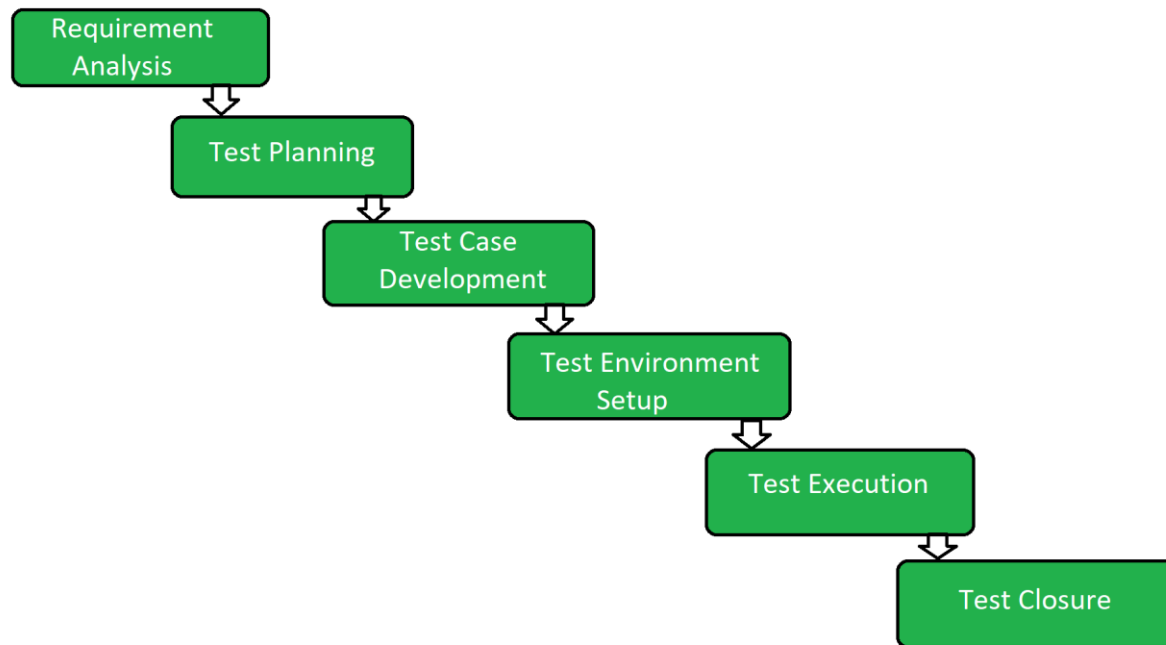
# CÁC VẤN ĐỀ LIÊN QUAN

---

- **Software Testing Life Cycle**
- **Software Development Models**
- **Vị trí của kiểm thử trong vòng đời phần mềm**
- **Tầm quan trọng của kiểm thử theo quy trình**

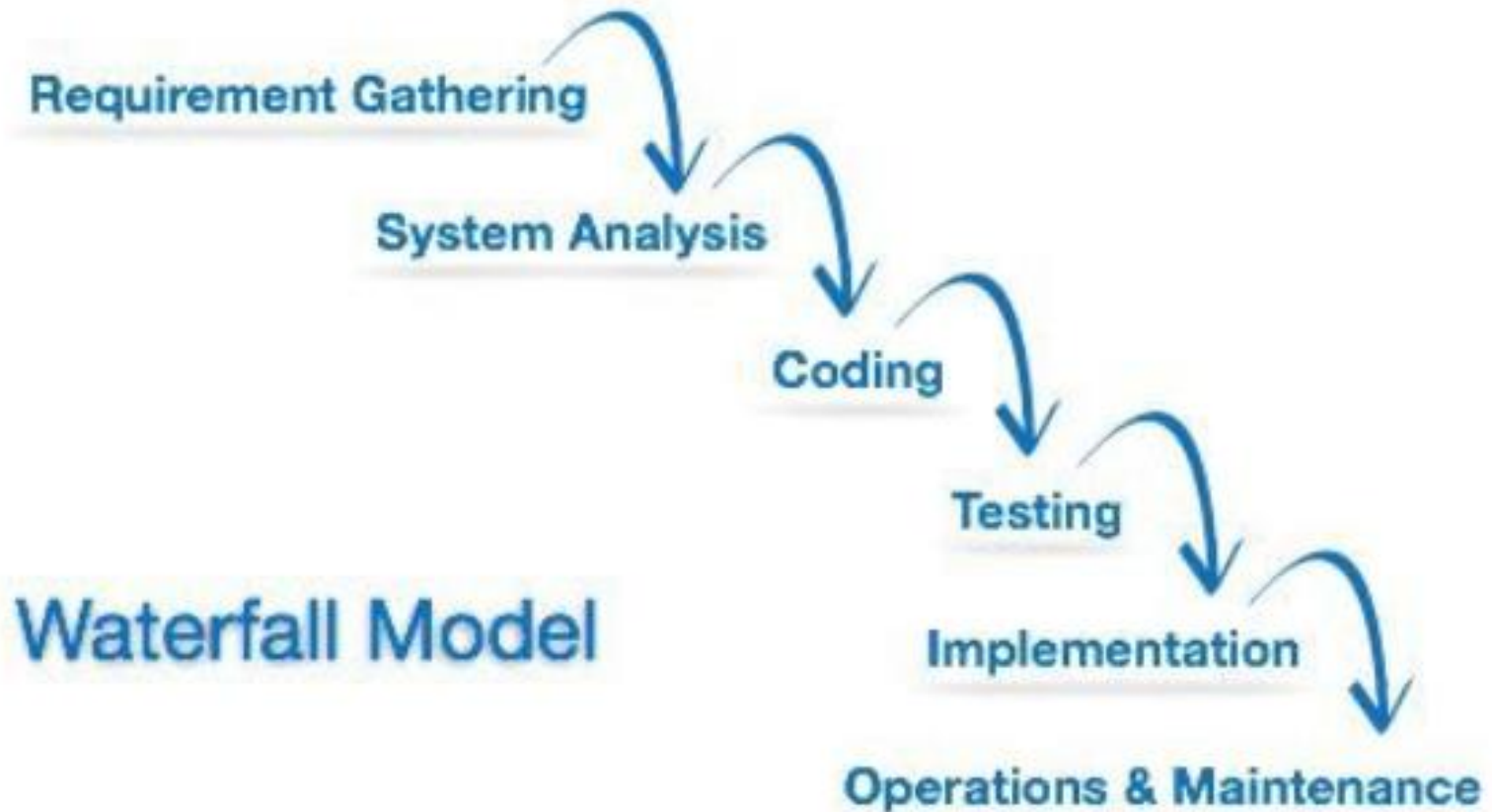
# Software Testing Life Cycle (STLC)

- **Software testing life cycle – STLC** is a sequence of specific activities conducted during the testing process to ensure software quality goals are met.

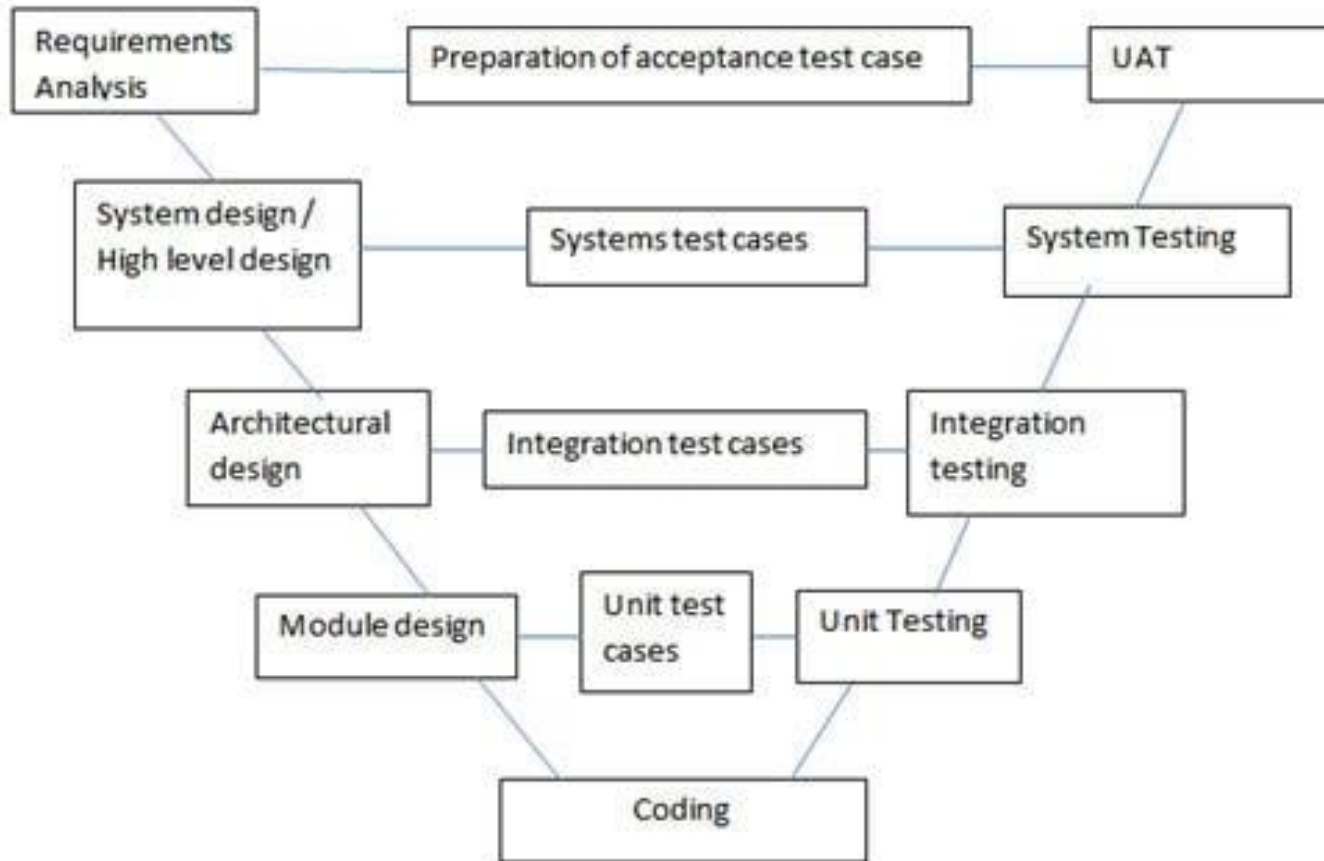


# Waterfall Model

---



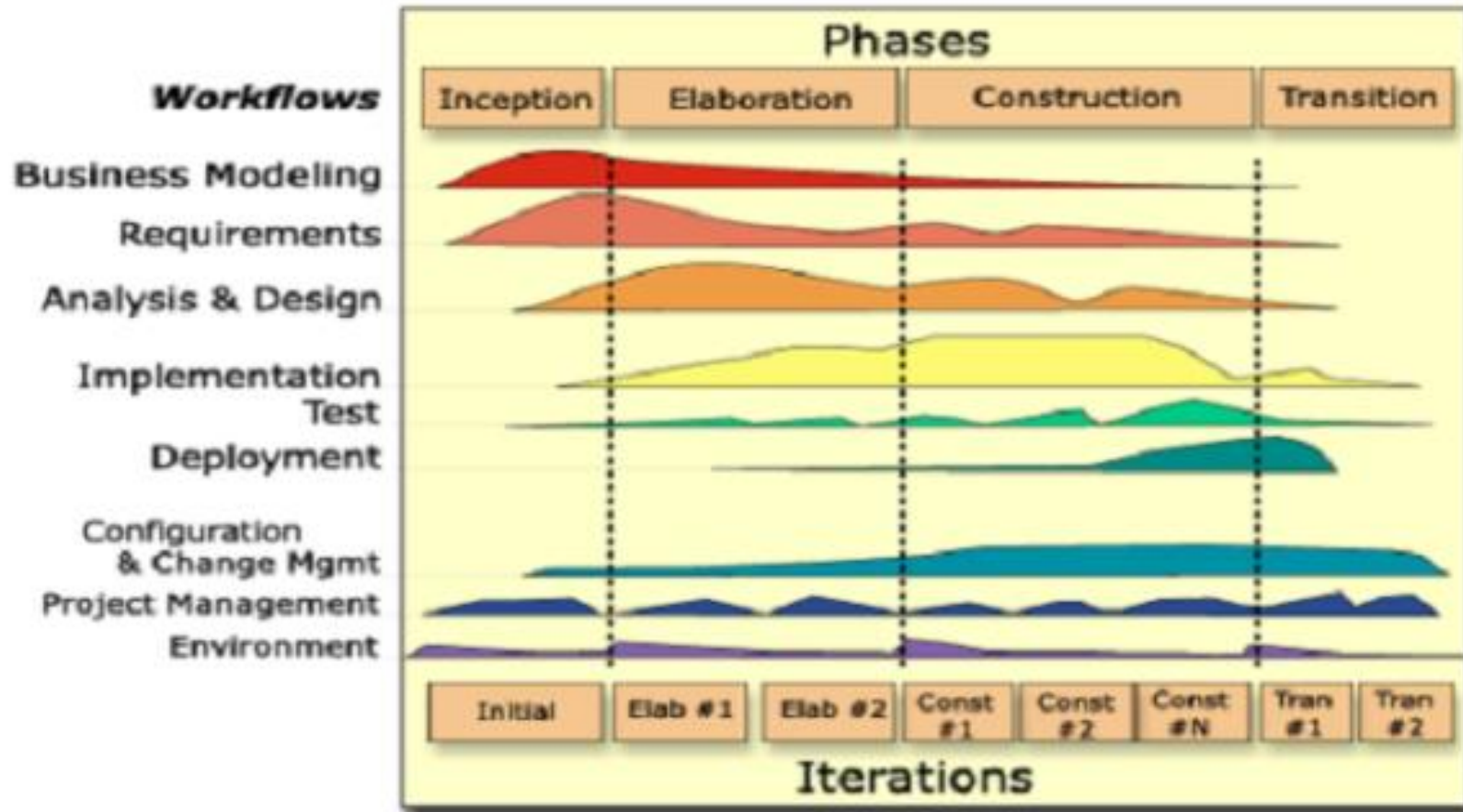
# V-Model



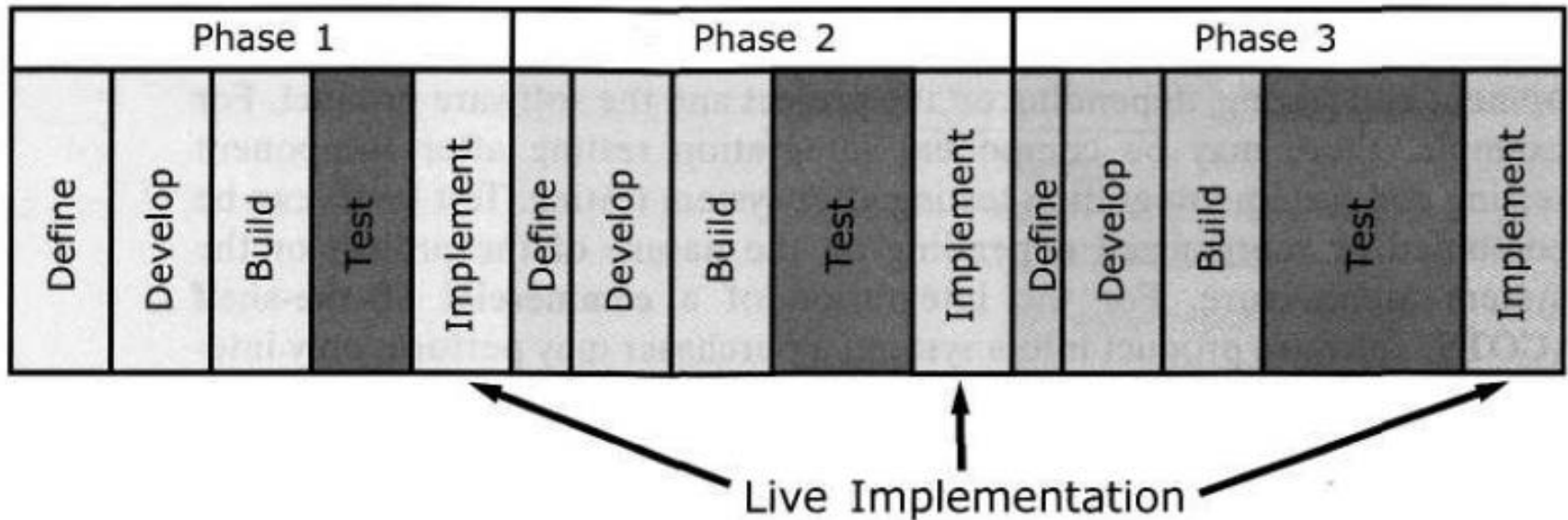
STLC - V Model - © [www.SoftwareTestingHelp.com](http://www.SoftwareTestingHelp.com)

- Các hoạt động kiểm thử có thể được tích hợp vào từng giai đoạn của vòng đời phát triển phần mềm.
- Hoạt động xác verification và validation được thực hiện cùng với hoạt động phát triển phần mềm.

# RUP (Rational Unified Process)



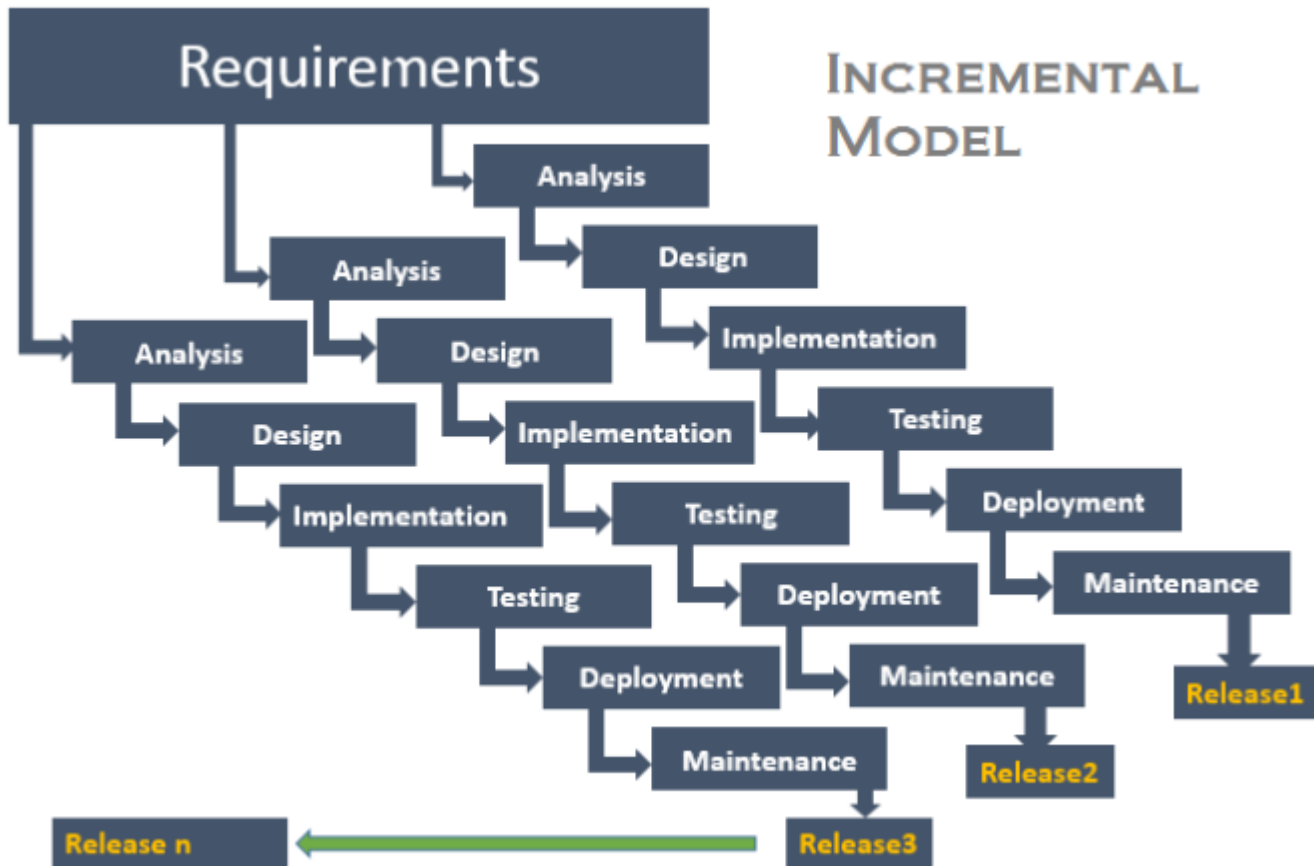
# Iterative life cycles



- Quá trình thực hiện được lặp đi lặp lại theo từng giai đoạn.
- Giai đoạn sau là cơ sở để hỗ trợ hoàn thiện chức năng được xây dựng từ các giai đoạn trước.
- Cuối mỗi lần lặp của mô hình - tạo ra một phiên bản mới của phần mềm.
- Kiểm thử: kiểm thử chức năng mới, kiểm thử hồi quy, kiểm thử tích hợp.

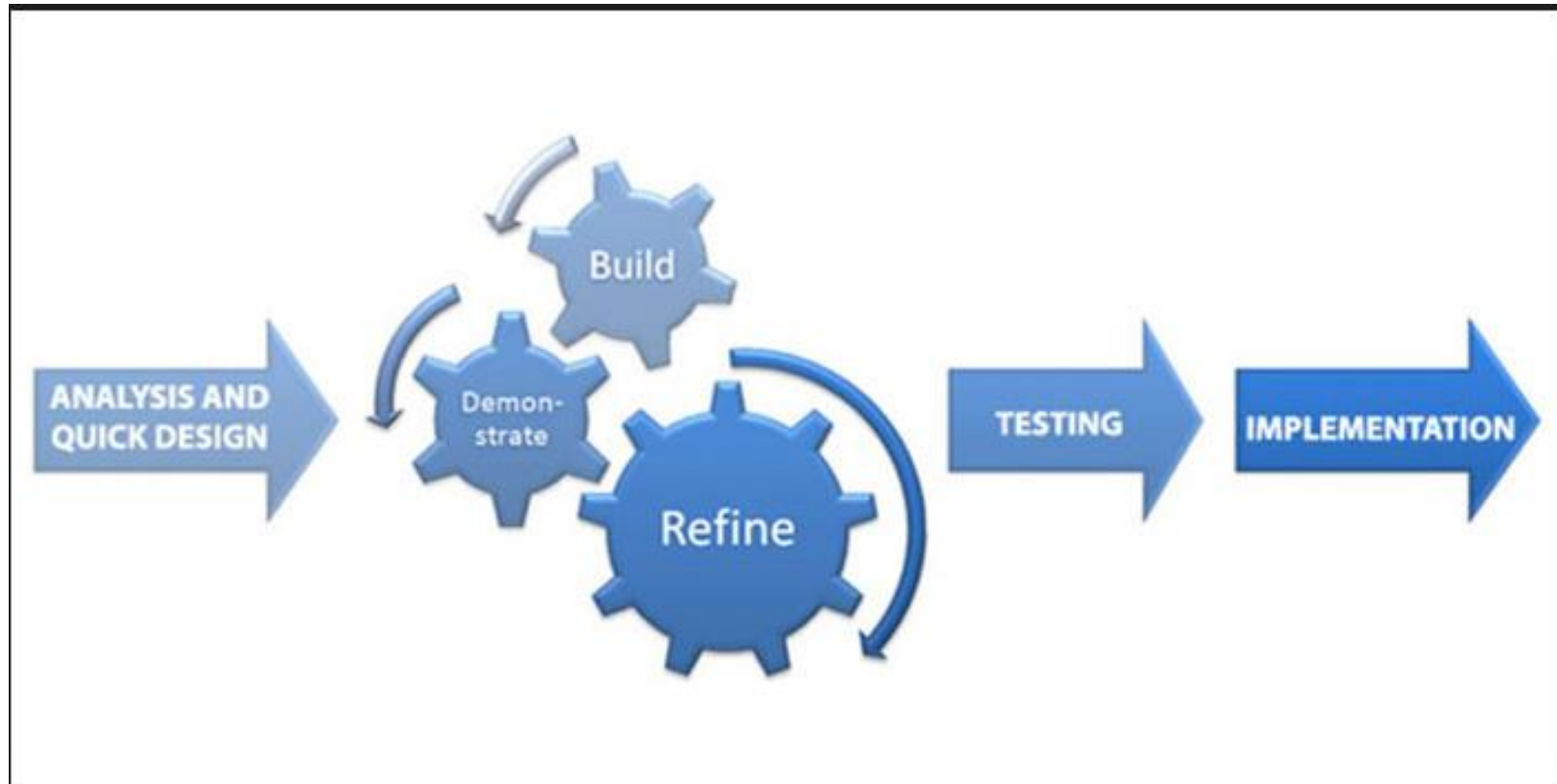


# Incremental model



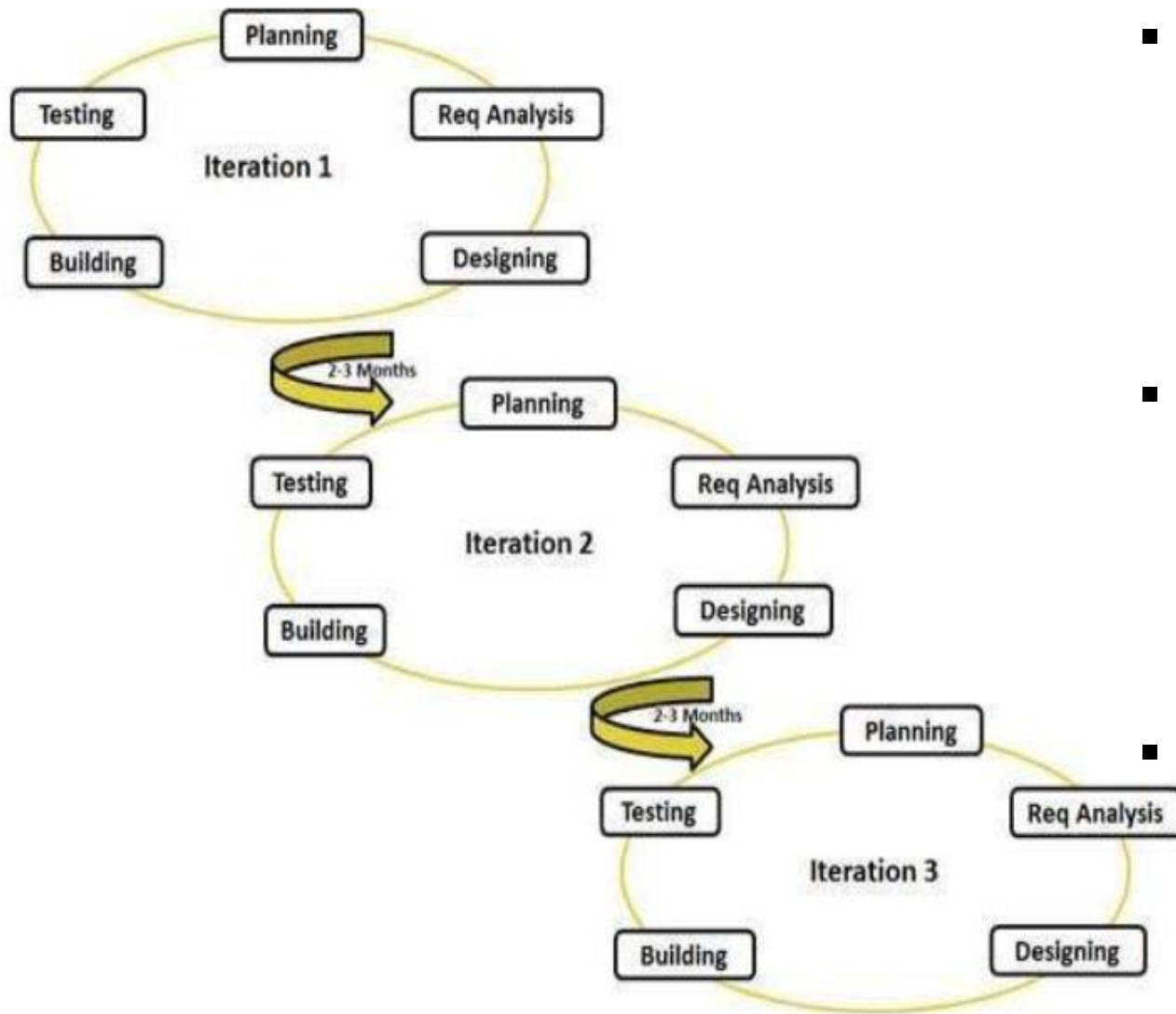
- Các yêu cầu được chia thành nhiều mô đun và mỗi mô đun được phát triển riêng biệt;
- Tích hợp các mô đun đã phát triển trở thành một hệ thống hoàn chỉnh.

# Rapid Application Development



- Phương pháp phát triển phần mềm sử dụng quy hoạch tối thiểu có lợi cho việc tạo mẫu nhanh.
- Các mô đun chức năng được phát triển song song và được tích hợp để tạo ra sản phẩm hoàn chỉnh → phân phối sản phẩm nhanh hơn

# Agile



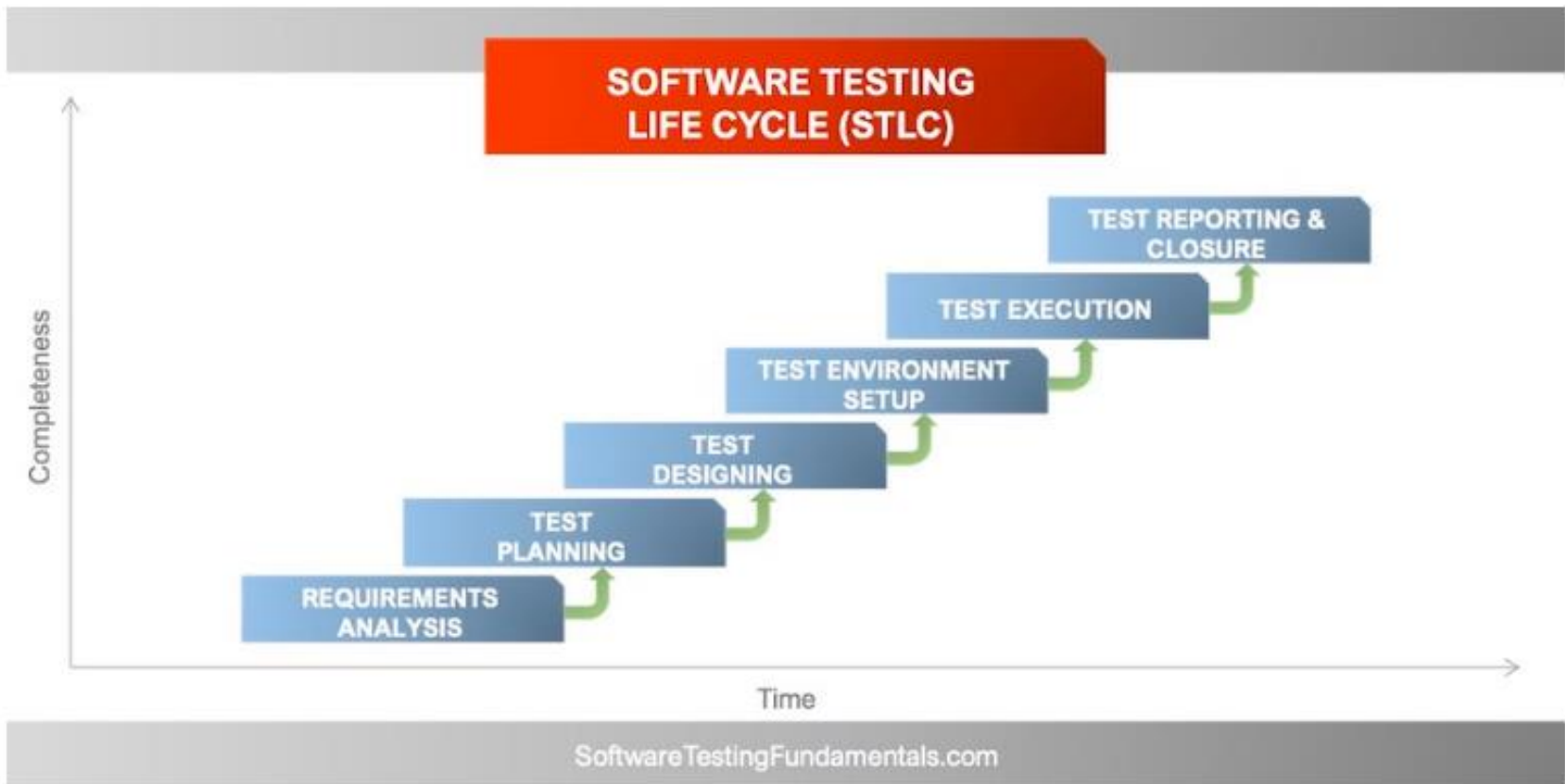
- Phát triển dựa vào mô hình lặp (Iterative life cycles) và gia tăng (Incremental model).
- Tập trung vào khả năng thích ứng của quy trình và sự hài lòng của khách hàng
- Phân phối nhanh sản phẩm phần mềm

# Software Testing Life Cycle

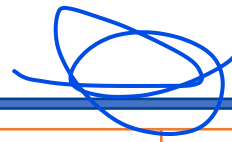
---

- **Requirement Analysis**
- **Test Planning**
- **Test Case development**
- **Test Environment setup**
- **Test Execution**
- **Test Cycle closure**

# STLC Phases



# STLC Phases



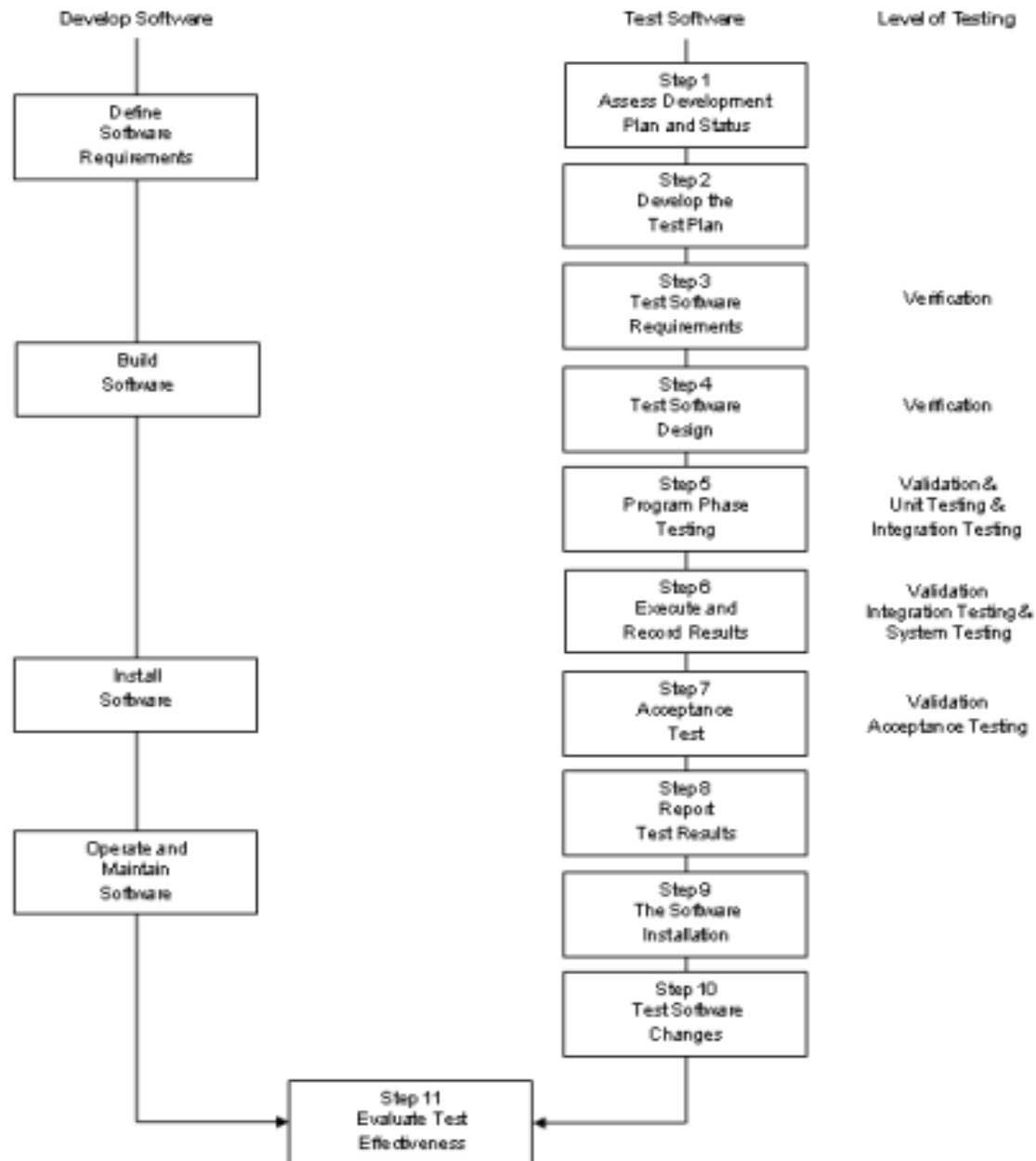
Giai đoạn	Đầu vào	Đầu ra
<b>Requirements Analysis</b>	<ul style="list-style-type: none"><li>- Tài liệu đặc tả yêu cầu (SRS)</li><li>- Tài liệu khách hàng</li><li>- Tài liệu thiết kế hệ thống</li></ul>	<ul style="list-style-type: none"><li>- Q&amp;A document</li><li>- Tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm</li></ul>
<b>Test Planning</b>	<ul style="list-style-type: none"><li>- Tài liệu đặc tả yêu cầu (updated)</li><li>- Tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm</li></ul>	<ul style="list-style-type: none"><li>- <b>Test Plan</b></li><li>- Test Estimation</li><li>- Test Schedule</li></ul>
<b>Test case development</b> <b>Test Designing</b>	<ul style="list-style-type: none"><li>- Tài liệu đặc tả yêu cầu (updated)</li><li>- Test plan</li><li>- Test estimation</li><li>- Test schedule</li></ul>	<ul style="list-style-type: none"><li>- Test design</li><li>- <b>Test case</b>/check list</li><li>- Test data</li><li>- Test automation script</li><li>- <b>Requirements Traceability Matrix</b></li></ul>
<b>Test Environment Setup</b>	<ul style="list-style-type: none"><li>- Test plan</li><li>- Smoke test case</li><li>- Test data</li></ul>	<ul style="list-style-type: none"><li>- Môi trường kiểm thử</li><li>- Kết quả của smoke test case</li></ul>
<b>Test Execution</b>	<ul style="list-style-type: none"><li>- Test plan</li><li>- Test design, test case, check list</li><li>- Test data</li><li>- Test automation script</li></ul>	<ul style="list-style-type: none"><li>- Test results</li><li>- Defect reports</li></ul>
<b>Test Reporting &amp; Closure</b>	<ul style="list-style-type: none"><li>- Tài liệu đặc tả yêu cầu</li><li>- Test plan</li><li>- Test results</li><li>- Defect reports</li><li>- Tài liệu Q&amp;A</li></ul>	<ul style="list-style-type: none"><li>- Test report</li><li>- Test results (final)</li></ul>

# Software Testing Process

---

- **CSTE: Certified Software Tester**
  - **QAI: Quality Assurance Institute**
- **ISTQB: International Software Testing Qualifications Board**

# Software Testing Process (CSTE-QAI)





# Software Testing Process (ISTQB)

---

A test process consists of the following main groups of activities:

- Test planning
- Test monitoring and control
- Test analysis
- Test design
- Test implementation
- Test execution
- Test completion

Each constituent activity consists of multiple individual tasks, which would vary from one project or release to another.

**Tài liệu đặc tả yêu cầu:** mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống.

- Tài liệu quan trọng cho đội phát triển (system analyst, business analyst, code) và kiểm thử (tester);
- Mô tả các chức năng và cấu trúc của hệ thống: Function Requirement và Non- Function Requirement ;
- Stakeholders đều hiểu được hệ thống theo cùng hướng;
- Đội phát triển xây dựng hệ thống một cách chính xác, đặc tả được các tính năng;
- Nhà kiểm thử hệ thống đọc hiểu từ đó xây dựng nên kịch bản kiểm thử chi tiết nhất;
- Bảo trì hệ thống và cải tiến những chức năng của hệ thống một cách nhanh chóng và dễ dàng;

# Software Requirement Specification

## **Thành phần chính SRS:**

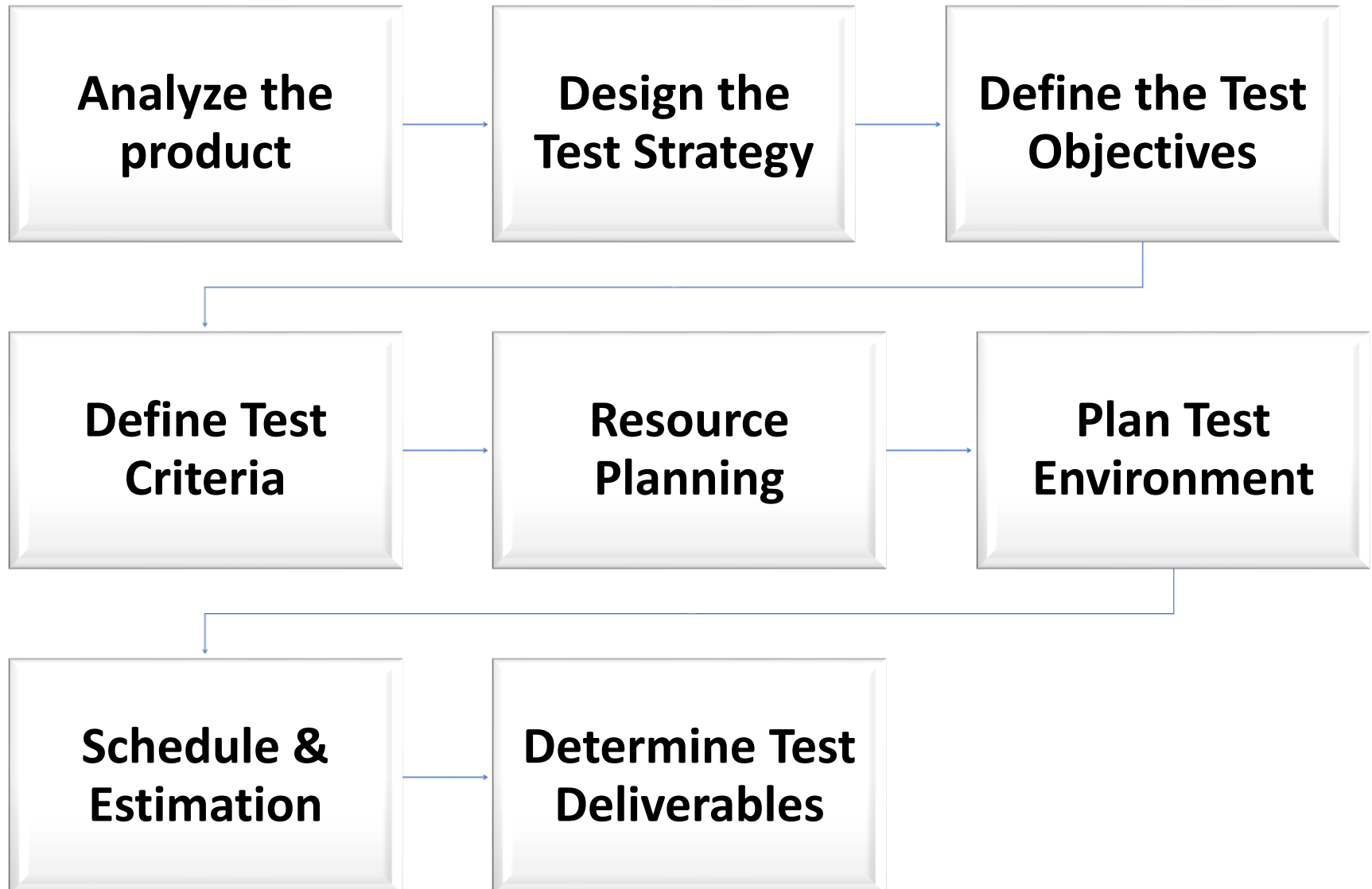
- Introduction: Phần giới thiệu
- High Level Requirement: Yêu cầu mức tổng thể
- Security Requirement: Yêu cầu về bảo mật
- Use Case Specification: Đặc tả Use Case
- Wireframe: Thiết kế màn hình
- Other Requirement: Các yêu cầu khác
- Integration: Yêu cầu tích hợp
- Appendices: Phụ lục

# Test Plan

**Kế hoạch kiểm thử:** tài liệu chi tiết phác thảo chiến lược kiểm thử, mục tiêu, tài nguyên cần thiết & schedule kiểm thử.

- Kế hoạch chi tiết để tiến hành các hoạt động kiểm thử phần mềm như một quy trình xác định;
- Được giám sát và kiểm soát từng bước bởi Test Manager;
- Xác định effort cần thiết để xác nhận chất lượng của ứng dụng đang kiểm thử;
- Giúp cho nhà phát triển, quản lý doanh nghiệp, khách hàng hiểu chi tiết về kiểm thử;
- Giống như một cuốn sách quy tắc, cần phải được tuân theo;
- Test Estimation, Test Scope, chiến lược test được ghi lại;
- Nhóm quản lý có thể xem xét và sử dụng lại cho các dự án khác.

# How to write a Test Plan



# Test Plan: 1 - Analyze the product

## Phân tích sản phẩm:

- Who will use the website?
- What is it used for?
- How will it work?
- What are software/ hardware the product uses?



Interview client,  
designer and  
developer

Review product  
and project  
documentation



perform product  
walkthrough



**Analyze** the banking website: <http://demo.guru99.com/V4>

# Test Plan: 1 - Analyze the product



Demo Site

Selenium - Insurance Project Agile Project Bank Project - Security Project Telecom Project Payment Gateway Project New Tours

SEO -

## Guru99 Bank

UserID

Password

## Steps To Generate Access

1. Visit - [here](#)
2. Enter your email id
3. Login credentials is allocated to you and mailed at your id
4. Login credentials are only valid for 20 days! So Hurry Up and quickly complete your tasks

# Test Plan: 2. Design the Test Strategy

## Xây dựng chiến lược kiểm thử:

- **Test Strategy:** Chiến lược kiểm thử
  - Bước quan trọng trong việc lập Test Plan
  - Tài liệu high-level
  - Thường được phát triển bởi Test Manager
  - Định nghĩa mục tiêu kiểm thử và các phương tiện để đạt được chúng
  - Xác định effort và chi phí kiểm thử





# Test Plan: 2.1 Define Scope of Testing

## Định nghĩa phạm vi của kiểm thử

- **In scope:** Các thành phần của hệ thống sẽ được kiểm thử (phần cứng, phần mềm, phần mềm trung gian, v.v.)
- **Out of scope:** Các thành phần của hệ thống sẽ không được kiểm thử cũng cần được xác định rõ ràng.
- Xác định scope của dự án kiểm thử: quan trọng đối với tất cả các bên liên quan:
  - Cung cấp cho mọi người một sự chắc chắn và thông tin chính xác về kiểm thử mà các bạn đang làm;
  - Tất cả các thành viên dự án sẽ có một sự hiểu biết rõ ràng về những gì được kiểm thử và những gì không.

# Test Plan: 2.1 Define Scope of Testing

---

## **How do you determine scope your project?**

- Precise customer requirement
- Project Budget
- Product Specification
- Skills & talent of your test team

# Test Plan: 2.1 Define Scope of Testing

## Project Guru99 Bank

- **In scope:** functions , external interface of website
- **Out of scope:** stress, performance or logical database

## Problem Scenario

- The customer wants you to test his API
- The project budget does not permit to do so
- In such a case what will you do?

# Test Plan: 2.1 Define Scope of Testing

---

## Project Guru99 Bank

- **In scope items:** functional testing, API testing.
- **Out of scope:** database testing, hardware & any other external interfaces.

# Test Plan: 2.2 Identify Testing Type

**Testing Type:** is a standard test procedure that gives an expected test outcome.

## Unit Test

- Test the **smallest** piece of **verifiable** software in the application

## API Testing

- **Test** the **API's** created for the application

## Integration Test

- Individual software modules are **combined** and tested as a group

## System Test

- Conducted on a **complete, integrated** system to evaluate the system's compliance with its specified requirements

## Install/uninstall Testing

- Focuses on what **customers** will **need** to do to **install** /**uninstall** and set up/remove the new software successfully

## Agile Testing

- Testing the system using Agile methodology

# Test Plan: 2.3 Document Risk & Issues

---

- Risk is future's **uncertain event** with a probability of **occurrence** and a **potential** for loss.
- When the risk actually happens, it becomes the '**issue**'.

# Test Plan: 2.4 Create Test Logistics

---

## Who will test?

- Ability to **understand** customers point of view
- Strong **desire** for quality
- **Attention** to detail
- Good **cooperation**

# Test Plan: 2.4 Create Test Logistics

## When will the test occur?

Test activities must be matched with associated development activities.

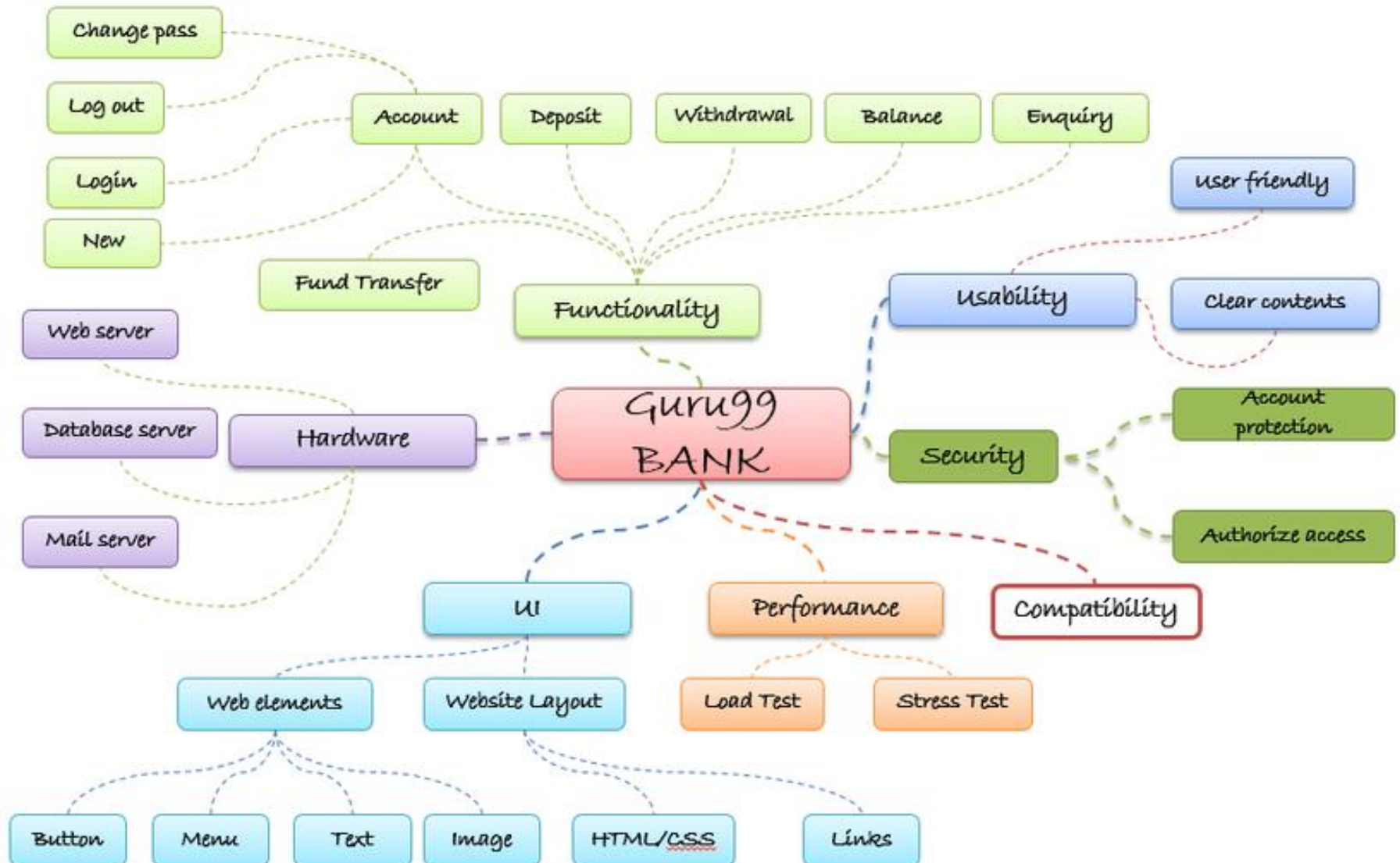




# Test Plan: 3. Define the Test Objectives

- **Test Objective:** is the overall goal and achievement of the test execution.
- The objective of the testing is finding as many software defects as possible.
- Ensure that the software under test is **bug free** before release.
- 2 following steps to define the test objectives:
  - List all the software features
  - Define the **target** or the **goal** of the test based on above features.

# Test Plan: 3. Define the Test Objectives



# Test Plan: 4. Define Test Criteria

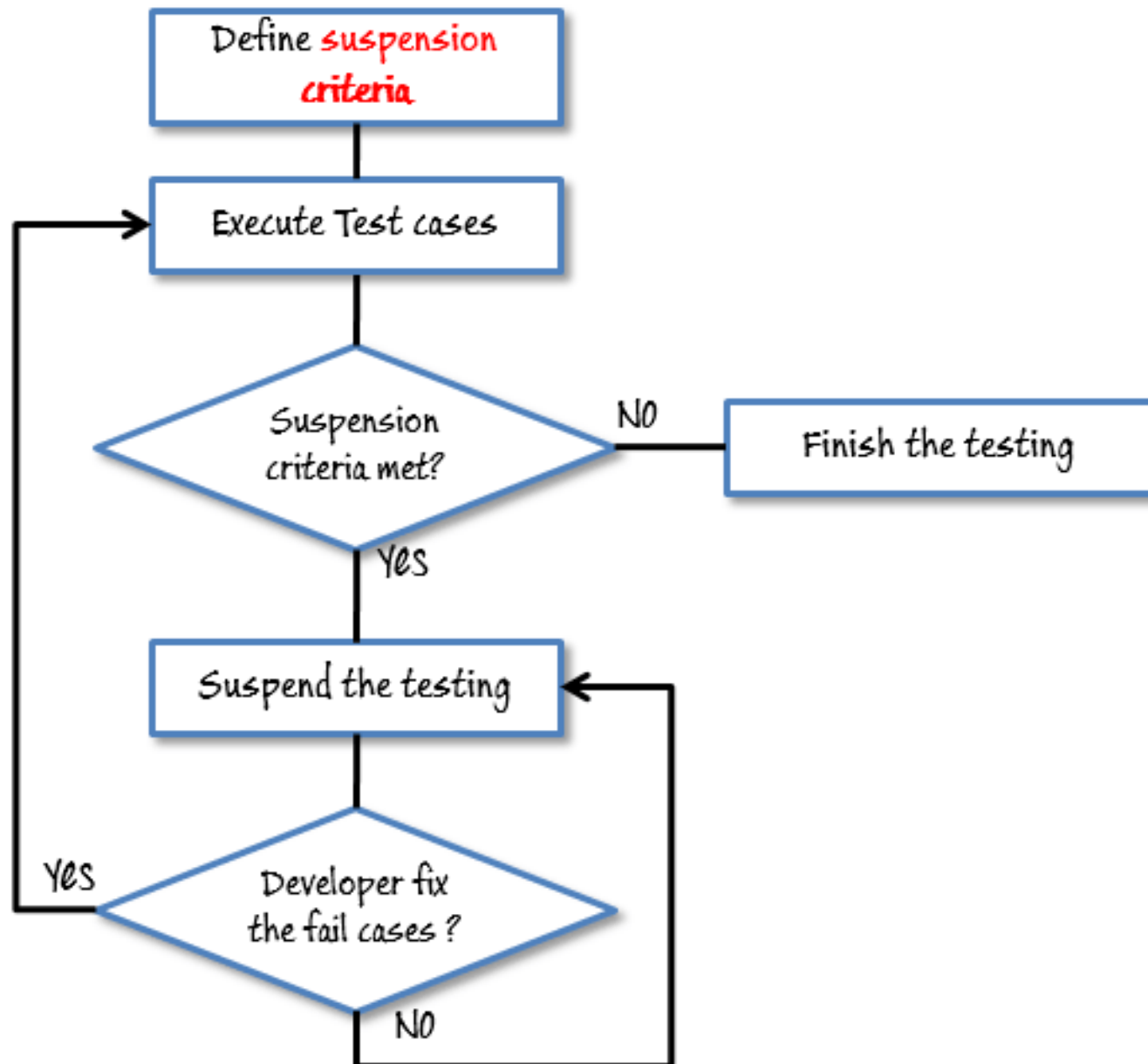
---

- **Test Criteria** is a standard or rule on which a test procedure or test judgment can be based.
- There're 2 types of test criteria as following:
  - **Suspension Criteria**
  - **Exit Criteria**

# Test Plan: 4. Define Test Criteria

- **Suspension Criteria: tiêu chí đình chỉ kiểm thử**
  - Specify the critical suspension criteria for a test.
  - If the suspension criteria are met during testing, the active test cycle will be **suspended** until the criteria are **resolved**.
  - Test Plan Example: If your team members report that there are **40%** of test cases failed, you should **suspend** testing until the development team fixes all the failed cases.

# Example: Define Suspension Criteria



# Test Plan: 4. Define Test Criteria

- **Exit Criteria: tiêu chí kết thúc kiểm thử**
  - It specifies the criteria that denote a **successful** completion of a test phase.
  - The exit criteria are the targeted results of the test and are necessary before proceeding to the next phase of development.
  - Example: **95%** of all critical test cases must pass.

# Test Plan: 4. Define Test Criteria

- Some methods of defining exit criteria are by specifying a targeted **run rate** and **pass rate**
- **Run rate:** ratio between **number test cases executed/total test cases** of test specification.
  - For example, the test specification has total 120 TCs, the tester only executed 100 TCs → the run rate is  $100/120 = 0.83$  (83%)
- **Pass rate:** ratio between **numbers test cases passed / test cases executed**.
  - For example, in above 100 TCs executed, there're 80 TCs that passed, so the pass rate is  $80/100 = 0.8$  (80%)

# Test Plan: 4. Define Test Criteria

- This data can be retrieved in Test Metric documents.
  - **Run** rate is mandatory to be **100%** unless a clear reason is given.
  - **Pass** rate is dependent on project scope, but **achieving high pass rate** is a goal.





# Test Plan: 5. Resource Planning

- **Resource Planning:** is a **detailed summary** of all types of resources required to complete project task.
- Resource could be human, equipment and materials needed to complete a project.
- The resource planning is important factor of the test planning because helps in **determining** the **number** of resources (employee, equipment...) to be used for the project..
- Manager can make the correct schedule & estimation for the project.

# Resource Planning – Human Resource

No.	Member	Tasks
1.	<b>Test Manager</b>	Manage the whole project Define project directions Acquire appropriate resources
2.	<b>Tester</b>	Identifying and describing appropriate test techniques/ tools/ automation architecture Verify and assess the Test Approach Execute the tests, Log results, Report the defects.
3.	<b>Developer in Test</b>	Implement the test cases, test program, test suite etc.
4.	<b>Test Administrator</b>	Builds up and ensures Test Environment and assets are managed and maintained Support Tester to use the test environment for test execution
5.	<b>SQA members</b>	Take in charge of quality assurance Check to confirm whether the testing process is meeting specified requirements

# Resource Planning – System Resource

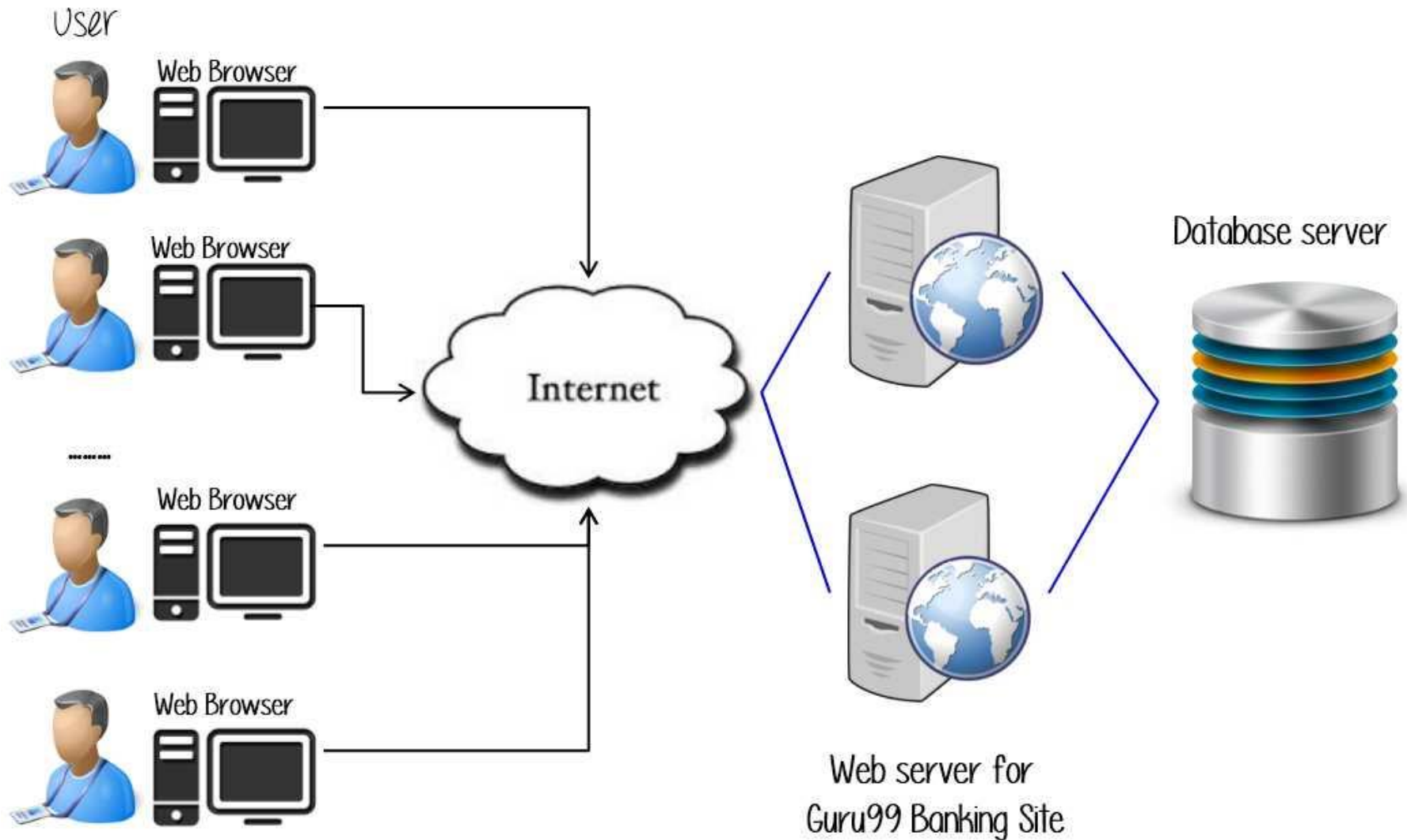
No.	Resources	Descriptions
1.	<b>Server</b>	<p>Install the web application under test</p> <p>This includes a separate web server, database server, and application server if applicable</p>
2.	<b>Test tool</b>	<p>The testing tool is to automate the testing, simulate the user operation, generate the test results</p> <p>There are tons of test tools you can use for this project such as Selenium, QTP...etc.</p>
3.	<b>Network</b>	<p>You need a Network include LAN and Internet to simulate the real business and user environment</p>
4.	<b>Computer</b>	<p>The PC which users often use to connect the web server</p>

# Test Plan: 6. Plan Test Environment

---

- A testing environment is a setup of software and hardware on which the testing team is going to execute test cases.
- The test environment consists of **real business** and **user** environment, as well as physical environments, such as server, front end running environment.

# The test environment of the banking website



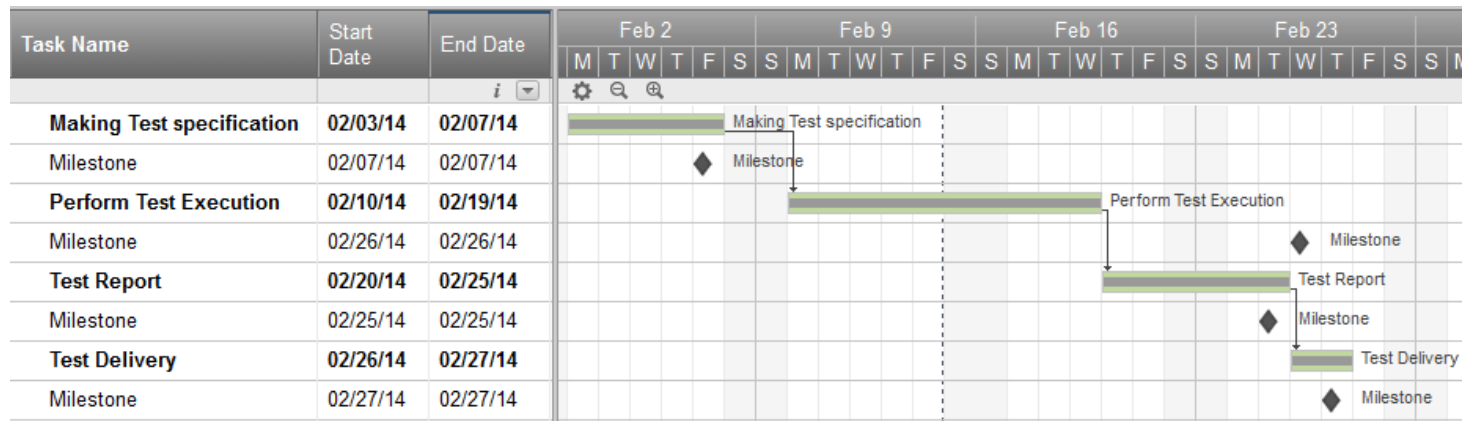
# Test Plan: 7. Schedule & Estimation

In the Test Estimation phase, suppose you break out the whole project into small tasks and add the estimation for each task as below:

Task	Members	Estimate effort
Create the test specification	Test Designer	170 man-hour
Perform Test Execution	Tester, Test Administrator	80 man-hour
Test Report	Tester	10 man-hour
Test Delivery		20 man-hour
Total		280 man-hour

# Test Plan: 7. Schedule & Estimation

- To create the project schedule, the Test Manager needs several types of input:
  - Employee and project deadline
  - Project estimation
  - Project Risk



# Test Plan: 8. Determine Test Deliverables

- Test Deliverables is a list of all the documents, tools and other components that has to be developed and maintained in support of the testing effort.

## **Before** testing

- Test plans document.
- Test cases documents
- Test Design specifications

## **During** testing

- Test Scripts
- Simulators.
- Test Data
- Test Traceability Matrix
- Error logs and execution logs.

## **After** the testing

- Test Results/reports
- Defect Report
- Installation/ Test procedures guidelines
- Release notes



# Test case

---

- **Test case là gì?**
- **Các phương pháp thiết kế Test case**
- **Cấu trúc Test case**
- **Ví dụ**
- **Lưu ý**

# What is a Test case?

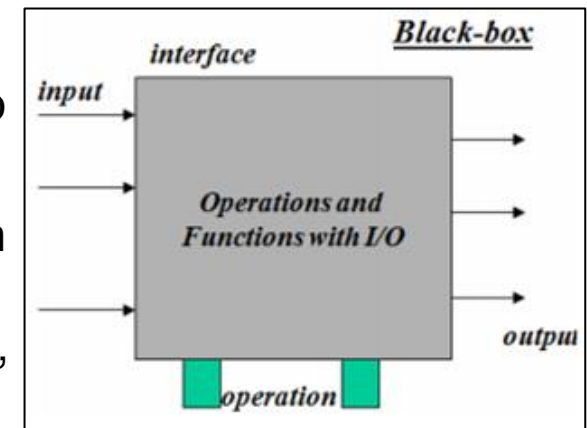
- Test case là đơn vị nhỏ nhất của kế hoạch kiểm thử.
- Mô tả các hành động và thông số cần thiết để đạt được cũng như xác minh rõ hành vi mong muốn của một chức năng cụ thể hoặc một phần nào đó của phần mềm cần được kiểm tra.



# Kỹ thuật thiết kế Test case

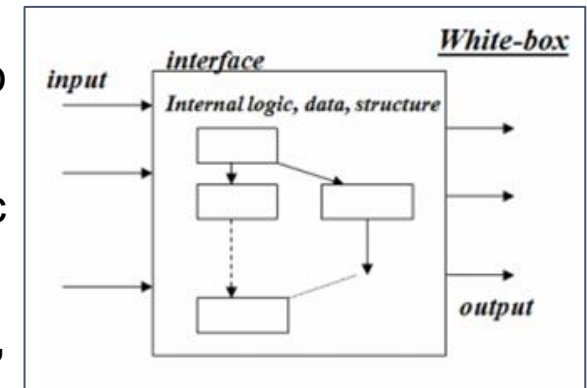
## ■ Specification-based

- Còn gọi là kỹ thuật hộp đen (black box): theo góc nhìn người sử dụng.
- Không cần kiến thức về chi tiết thiết kế và hiện thực bên trong.
- Kiểm thử dựa trên các yêu cầu và đặc tả sử dụng các thành phần phần mềm.



## ■ Structure-based

- Còn gọi là kỹ thuật hộp trắng (white box) theo góc nhìn người lập trình.
- Cần kiến thức về chi tiết thiết kế và hiện thực bên trong.
- Kiểm thử dựa vào phủ các lệnh, các nhánh, các điều kiện con,...



## ■ Experience-based

- Phụ thuộc vào hiểu biết và năng lực của tester.
- Kiến thức, kinh nghiệm của tester là cơ sở để thiết kế test case.

# Specification-based

---

- **Equivalence partitioning**
- **Boundary value analysis**
- **Decision table testing**
- **State transition testing**
- **Use case testing**

# Structure-based

---

- **Statement testing**
- **Condition testing**
- **Multiple condition testing**
- **Path testing**

# Experience-based

---

- **Exploratory testing**
- **Error guessing**

# Types of test cases

---

- **Positive**
- **Negative**
- **Boundary value**
- **Integration**
- **Testing localisation**

# A structure of a Test case

ID	Test Case Field	Description
1	Test case ID	Nên có một ID duy nhất
2	Test Priority	Các loại priority: Low, Medium, High
3	Test Designed by	Tên của người viết test cases (tester)
4	Date of test designed	Ngày mà các thử nghiệm được tạo ra
5	Test Executed by	Người thực thi việc kiểm thử (tester)
6	Date of the Test Execution	Ngày thực thi việc kiểm thử
7	Name or Test Title	<p>Tiêu đề: cung cấp sự mô tả ngắn gọn về trường hợp thử nghiệm, ví dụ "Nhập lại mật khẩu".</p> <p>Tiêu đề: chìa khóa giúp người kiểm thử nhanh chóng tìm ra các trường hợp thử nghiệm đúng.</p>



# A structure of a Test case

ID	Test Case Field	Description
8	Description/ Summary of Test	Mô tả chi tiết cho test cases Thiết lập các danh mục để tổ chức các test cases thành các nhóm hợp lý.
9	Pre-condition	Yêu cầu cần được hoàn thành trước khi thực thi test case
10	Test Steps	Các bước kiểm thử đưa ra cho tester một danh sách được đánh số các bước thực hiện trong hệ thống → test case dễ hiểu hơn.
11	Test Data	Có thể nhập dữ liệu kiểm thử trực tiếp vào các trường dữ liệu kiểm thử, hoặc chỉ ra một tập tin riêng biệt chứa dữ liệu kiểm thử cho 1 hoặc nhiều test cases.

# A structure of a Test case

ID	Test Case Field	Description
12	Expected Results	Kết quả mong muốn: lỗi/thông báo xuất hiện trên màn hình. Chi tiết mức độ tối ưu thay đổi tùy theo tình hình.
13	Post-Condition	Tình trạng của hệ thống sau khi chạy trường hợp thử nghiệm?
14	Status (Fail/Pass)	Fail: (actual result) <>(expected result). Pass: (actual result) = (expected result).
15	Notes/Comments/ Questions	Nếu có một vài điều kiện đặc biệt cần ghi chú liên quan đến những trường hợp phía trên
16	Requirements	Danh sách các yêu cầu cho một chu kỳ kiểm tra cụ thể.
17	Attachments/ References	Các file và tài liệu được gắn vào trường hợp kiểm thử Ví dụ: file ảnh chụp màn hình và các tài liệu hỗ trợ khác
18	Automation? (Yes/No)	Điền vào "Yes" khi các trường hợp thử nghiệm được chạy tự động

## Example 1: Test the Login Functionality in Banking

Poject Name: Banking System	Test Case
Test Case ID: BU_001	Test Designed by: <Name>
Test Priority (Low/Medium/High): Med	Test Designed date: <Date>
Module Name: Bank login screen	Test Executed by: <Name>
Test Title: Test the Login Functionality in Banking	Test Execution date: <Date>
Description: Verify login with valid username and password	

**Pre-conditions: User has valid username and password**

**Dependencies:**

## Example 1: Test the Login Functionality in Banking

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page		User should be able to login	User is able to login	Pass	
2	Provide valid username	<a href="#">User= example@gmail.com</a>	Credential can be entered	As Expected	Pass	
3	Provide valid password	Password: 1234	Credential can be entered	As Expected	Pass	
4	Click on Login button		User logged	User logged successfully	Pass	

## Example 2: Test the Change password Functionality

Test Case ID	Test Case Description	Test Steps		Test Data	Actual Result	Status
		Steps to Perform	Step Expected Result			
TC01	Change password with wrong old password	Click on button "đổi mật khẩu"	A screen is showed to change password		Error message	Pass
		Input a wrong old password	Error message			
TC02	Change password which "mật khẩu" field does not match with "nhập lại mật khẩu" field	Click on button "đổi mật khẩu"	A screen is showed to change password		Error message	Pass
		Input two passwords does not match together	Error message			
TC03	Change password with valid data	Click on button "đổi mật khẩu"	A screen is showed to change password		Successful modification	Pass
		Input valid data	Successful modification			

## Example 3: Test the Account Register Functionality

- Click on the link "Đăng ký" on the top right corner
- Type the field "Họ và tên"
- Type the field "Điện thoại" (length 10, consists of digits only)
- Type the field "Username"
- Type a password ( $\geq 6$  characters) into the field "Mật khẩu" or the field "Nhập lại mật khẩu".
- Type an email into the field "Email" and the field "Nhập lại email".
- Type the field "Nhập mã xác nhận"
- Fill out all the mandatory fields on the form.
- Click on the button "Đồng ý"
- .....

## Example 3: Test the Account Register Functionality

Test Case ID	Test Case Description	Test Steps		Test Case Actual Result	Status
		Steps to Perform	Step Expected Result		
TC_001	+ Pre: None + Test description: The field "Họ và tên" is empty. + Post: An error message is displayed	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	An error message is displayed to inform user that the username already exists	Pass
		Leave the field "Họ và tên" empty			
		Click the button "Đồng ý"	An error message is displayed		
TC_002	+ Pre: None + Test description: The field "Điện thoại" have length <> 10 or consists non-numeric characters. + Post: An error message is displayed	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	An error message is displayed to inform user that the "Điện thoại" have length 10 and consists of digits only	Pass
		Type a field "Điện thoại" have length <> 10 or enter any non-numeric characters			
		Click the button "Đồng ý"	An error message is displayed		

## Example 3: Test the Account Register Functionality

Test Case ID	Test Case Description	Test Steps		Test Case Actual Result	Status
		Steps to Perform	Step Expected Result		
TC_003	+ Pre: None + Test description: Username already exists + Post: An error message is displayed	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	An error message is displayed to inform user that the username already exists	Pass
		Type an username that already exists into the field "Tên đăng nhập"			
		Click the button "Đồng ý"	An error message is displayed		
TC_004	+ Pre: None + Test description: Password is less than 6 characters + Post: An error message is displayed	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	An error message is displayed to inform user that the password is less than 6 characters	Pass
		Type a password less than 6 characters into the field "Mật khẩu" or the field "Nhập lại mật khẩu"			
		Click the button "Đồng ý"	An error message is displayed		



## Example 3: Test the Account Register Functionality

Test Case ID	Test Case Description	Test Steps		Test Case Actual Result	Status
		Steps to Perform	Step Expected Result		
TC_005	+ Pre: None + Test description: Password and confirm password don't match + Post: An error message is displayed	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	An error message is displayed to inform user that the password and the confirm password don't match	Pass
		Type a password into the field "Mật khẩu" and a different password into the field "Nhập lại mật khẩu"			
		Click the button "Đồng ý"	An error message is displayed		
TC_006	+ Pre: None + Test description: Email and confirm email don't match + Post: An error message is displayed	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	An error message is displayed to inform user that the email and the confirm email don't match	Pass
		Type an email into the field "Email" and a different email into the field "Nhập lại email"			
		Click the button "Đồng ý"	An error message is displayed		

## Example 3: Test the Account Register Functionality

Test Case ID	Test Case Description	Test Steps		Test Case Actual Result	Status
		Steps to Perform	Step Expected Result		
TC_007	+ Pre: None + Test description: The characters don't match the image verification + Post: An error message is displayed	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	An error message is displayed to inform user that the characters entered don't match the image verification	Pass
		Type characters different from the image verification into the field "Nhập mã xác nhận"			
		Click the button "Đồng ý"	An error message is displayed		
TC_008	+ Pre: None + Test description: Clean all the fields + Post: All fields are empty	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	All fields are empty	Pass
		Fill out some of the fields on the form			
		Click ont the button "Nhập lại"	All the fields are empty		

## Example 3: Test the Account Register Functionality

Test Case ID	Test Case Description	Test Steps		Test Case Actual Result	Status
		Steps to Perform	Step Expected Result		
TC_Signin_009	+ Pre: None + Test description: The form is filled out correctly + Post: A new user is created and stored in the databse	Click on the link "Đăng ký" on the top right corner	A register account form is displayed	A successful mesage is displayed to	Pass
		Fill out all the mandatory fields on the form		inform user that his/her account has been created	
		Click on tthe button "Đồng ý"	A successful mesage is displayed	successfully and the user should check email to activate the account	

# Tips on How to Write Effective Test Cases

---

- Test Case Naming Conventions
- Test Case Description
- Assumptions and Preconditions
- Test Data Input
- Cover all Verification Points in Test Design Steps
- Attach the Relevant Artefacts
- Expected Result
- Divide Special Functional Test Cases into Sets
- Legible & Easily Understandable
- Test Case Review
- Reusable
- Maintenance & Updates
- Post Conditions

# Traceability Matrix

---

## **Ma trận truy xuất nguồn gốc**

- A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship
- It is used to track the requirements and to check the current project requirements are met;

# Requirements Traceability Matrix

---

- **Requirement Traceability Matrix (RTM)** is a document that maps and traces user requirement with test cases;
- It captures all requirements proposed by the client and requirement traceability in a single document, delivered at the conclusion of the Software development life cycle;
- The main purpose of Requirement Traceability Matrix is to validate that all requirements are checked via test cases such that no functionality is unchecked during Software testing.

# Requirements Traceability Matrix

## Which Parameters to include in RTM?

- Requirement ID
- Requirement Type and Description
- Test Cases with Status

Req No	Req Desc	Testcase ID	Status
123	Login to the application	TC01,TC02,TC03	TC01-Pass TC02-Pass
345	Ticket Creation	TC04,TC05,TC06, TC07,TC08,TC09 TC010	TC04-Pass TC05-Pass TC06-Pass TC06-Fail TC07-No Run
456	Search Ticket	TC011,TC012, TC013,TC014	TC011-Pass TC012-Fail TC013-Pass TC014-No Run

# Requirements Traceability Matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
2																
3		Sno	Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution			Defects?	Defect ID	Defect Status	Req Coverage Status
4										Test Env	UAT Env	Prod Env				
5		1	Req01	Login to the Application	TC01	Login with Invalid Username and valid password	Completed	XYZ	No	Passed	No Run	No Run	None	None	N/A	Partial
6		2			TC02	Login with Valid Username and invalid password	Completed	YZA	No	Passed	No Run	No Run	None	None	N/A	Partial
7		3			TC03	Login with valid credentials	Completed	XYZ	Yes	Passed	Passed	No Run	None	None	N/A	Partial
8																

■ A requirement traceability matrix can:

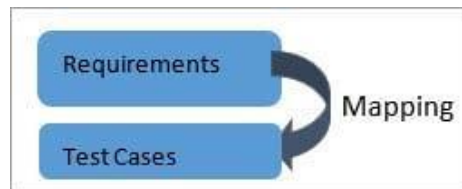
- Show the requirement coverage in the number of test cases;
- Design status as well as execution status for the specific test case;
- If there is any User Acceptance test to be done by the users, then UAT status can also be captured in the same matrix;
- The related defects and the current state can also be mentioned in the same matrix.



# Requirements Traceability Matrix

## Types of Traceability Test Matrix

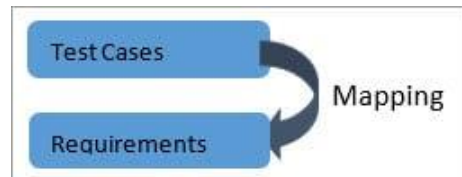
- **Forward traceability** - Truy xuất nguồn gốc xuôi



(Building the right product)

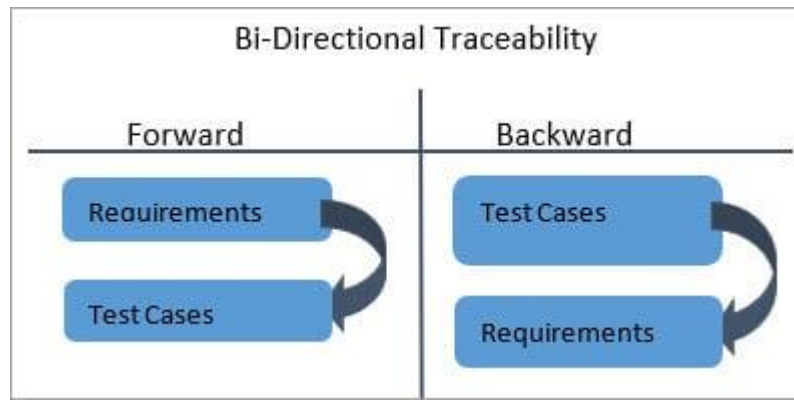
Requirement IDs >> Test Scenario IDs >> Test cases IDs >> Defect IDs.

- **Backward or reverse traceability** - Truy xuất nguồn gốc ngược



(Building the Product right)

- **Bi-directional traceability** - Truy xuất nguồn gốc hai chiều.



# Create RTM: Guru99 banking project

## BRD: Business Requirement Document

- The customer should be able to login to Guru99 banking website with the correct password and user
- Manager should be able to login to the website through customer login page..

BR#	Module Name	Applicable Roles	Description
B1	Login and Logout	Manager Customer	<b>Customer:</b> A customer can login using the login page <b>Manager:</b> A manager can login using the login page of customer. Post Login homepage will show different links based on role
B2	Enquiry	Customer	<b>Customer:</b> A customer can have multiple bank accounts. He can view balance of his accounts only <b>Manager:</b> A manager can view balance of all the customers who come under his supervision
B3	Fund Transfer	Manager Customer	<b>Customer:</b> A customer can have transfer funds from his "own" account to any destination account. <b>Manager:</b> A manager can transfer funds from any

Business Requirement # for  
Guru99 banking project

# Create RTM: Guru99 banking project

## TRD: Technical Requirement Document

### Login

**T92** User-ID must not be blank

**T93** Password must not be blank

**T94** If userid and password are valid. Login

Here is our TRD  
(Technical  
Requirement  
Document)

**Note:** QA teams do not document the BRD and TRD. Also, some companies use **Function Requirement Documents (FRD)** which are similar to Technical Requirement Document but the process of creating Traceability Matrix remains the same.

# Create RTM in Testing

**Step 1:** "Verify Login, when correct ID and Password is entered, it should log in successfully."

TestCase #	Test Case	Test Steps	Test Data	Expected Result
1	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

When correct password and id entered, it should login successfully

**Step 2:** Identify the Technical Requirement that this test case is verifying. For our test case, the technical requirement is T94 is being verified.

**T94** If userid and password are valid. Login

T94 is our technical requirement that verifies successful login

# Create RTM in Testing

**Step 3:** Note this Technical Requirement (T94) in the Test Case.

TestCase #	TR #	Note the Technical Requirement in the test case		Test Steps	Test Data	Expected
1	T94	Verify Login		1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

**Step 4:** Identify the Business Requirement for which this TR (Technical Requirement-T94) is defined.

BR#	Module Name	Applicable Roles	Description
B1	Login and Logout	Manager Customer	<b>Customer:</b> A customer can login using the login page <b>Manager:</b> A manager can login using the login page of customer. Post Login homepage will show different links based on role

# Create RTM in Testing

## Step 5: Note the BR (Business Requirement) in Test Case

BR#	Module Name	Applicable Roles	Description
B1	Login and Logout	Manager Customer	<b>Customer:</b> A customer can login using the login page <b>Manager:</b> A manager can login using the login page of customer. Post Login homepage will show different links based on role
B2	Enquiry	Customer	<b>Customer:</b> A customer can have multiple bank accounts. He can view balance of his accounts only <b>Manager:</b> A manager can view balance of all the customers who come under his supervision
B3	Fund Transfer	Manager Customer	<b>Customer:</b> A customer can have transfer funds from his "own" account to any destination account. <b>Manager:</b> A manager can transfer funds from any

Business Requirement # for Guru99 banking project

TestCase #	TR #	Test Steps	Test Data	Expected
1	T94	Verify Login 1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

Note the Technical Requirement in the test case

TestCase #	BR #	TR #	Test Case	Test Steps	Test Data	Expected
1	B1	T94	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

# Create RTM in Testing

**Step 6:** Do above for all Test Cases. Later Extract the First 3 Columns from your Test Suite. RTM in testing is Ready!

Business Requirement #	Technical Requirement #	Test Case ID
B1	T94	1
B2	T95	3
B3	T96	3
B4	T97	4

Requirement Traceability Matrix

# Advantage of RTM

---

- It confirms 100% test coverage
- It highlights any requirements missing or document inconsistencies
- It shows the overall defects or execution status with a focus on business requirements
- It helps in analyzing or estimating the impact on the QA team's work with respect to revisiting or re-working on the test cases.



# TỔNG KẾT CHƯƠNG 2

---

- **Software Testing Life Cycle**
- **Software Testing Process**
- **Test Plan**
- **Test case**
- **Requirements Traceability Matrix**

---

# HỎI & ĐÁP