SAIGON UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

GROUP EXERCISE / LAB 4

ABC PROJECT

Intructor: Dr. Đỗ Như Tài

HO CHI MINH CITY

Team Introduction

	Full Name	Student Code
1	Trang Gia Huy	3122411068
2	Nguyễn Lê Quỳnh Hương	3122411078

REVISION HISTORY

Date	Version	Description	Author
05 Aug 10	1.0	Initial edition	Alexandre Stelmakh

Table of content

Test Plan

1. Introduction

1.1 Purpose

This Test Plan document for ABC project supports the following objectives:

- Identify existing project information and software components to be tested.
- List recommended Requirements for Test (high level).
- Recommend and describe testing strategies to be employed.
- Identify required resources and provide a test effort estimate.
- List the test project deliverable elements.

1.2 Background

The system developed is an instant messenger for Palm platform. The system should provide a capability to use ICQ instant messaging service. So, the protocol used by this service must be supported by the system.

The system developed is intended for Palm application market and positioned as standalone product.

1.3 Scope

This Test Plan describes the system tests that will be conducted within Build 1, Build 2, Build 3, Release candidates and Release versions.

This Test Plan applies to testing all ABC project requirements defined in the Software Requirements Specification [1].

1.4 Project Identification

The table below identifies the documentation and availability used for developing the Test Plan:

Document **(and version /date)**	**Created or Available**	**Received or Reviewed**	**Author or Resource**	**Notes**
Requirements Specification	■ Yes □ No	■ Yes □ No		
Functional Specification	■ Yes □ No	■ Yes □ No		
Use-Case Reports	■ Yes □ No	■ Yes □ No		
Project Plan	■ Yes □ No	■ Yes □ No		
Design Specifications	■ Yes □ No	■ Yes □ No		
Prototype	□ Yes ■ No	□ Yes ■ No		
User’s Manuals	□ Yes ■ No	□ Yes ■ No		
Business Model or Flow	□ Yes ■ No	□ Yes ■ No		
Data Model or Flow	□ Yes ■ No	□ Yes ■ No		
Business Functions and Rules	■ Yes □ No	■ Yes □ No		
Project or Business Risk Assessment	□ Yes ■ No	□ Yes ■ No		

1.5 References

The following table contains references to external documents.

No.	Document	Description
1	ABC project Software Requirements Specification, Version 1.0.3.3, Monday, 25/Aug/10	The **SRS** fully describes the external behavior of the application or subsystem identified. It also describes non- functional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the software requirements.
2	ABC project Software Development Plan, Version 1.0.5, 05/Sep/10	The purpose of Software Development Plan is to define tasks of ABC project (hereinafter referred to as IM or the project) to be developed. It focuses on the capabilities sought after by stakeholders, the target users in particular, and the reasons for this search.
3		

2. Feature to be tested

The listing below identifies those items − use cases, functional requirements, and non-functional requirements − that have been identified as targets for testing. This list represents what is to be tested.

2.1. Functionality

2.1.1 Download contact list from server

- Verify that the system provides the capability to download a contact list from the server. This operation should be performed automatically during connection to ICQ service.
- Verify that the system provides the capability to disable automatic download of the contact list.
- Verify that the system provides the capability to download the contact list upon the user request.

2.1.2 Upload contact list to server

- Verify that the system provides the capability to upload the contact list from the device to the server.
- Verify that the system correctly updates existing contact entries on the server.
- Verify that the system allows manual synchronization by user request.
- Verify that conflicts (e.g., different versions of the same contact) are handled properly.

2.1.3 Add new contact to contact list

- Verify that the system allows adding a new contact by ICQ number.
- Verify that the system allows adding a new contact by nickname or email (if supported).
- Verify that the system validates the entered data before saving.
- Verify that the new contact appears correctly in the contact list..

2.1.4 Delete contact from contact list

- Verify that the system provides the capability to delete a contact from the contact list.
- Verify that the deleted contact is removed from both local storage and server (after synchronization).
- Verify that a confirmation prompt is displayed before deletion..

2.1.5 Create new message

- Verify that the system provides the capability to create a new message for a selected contact.
- Verify that the system supports plain text messages.
- Verify that the system supports emoticons and supported character sets.
- Verify that the system prevents sending empty or invalid messages.

2.1.6 Receive new message

- Verify that the system correctly receives new incoming messages in real-time.
- Verify that notifications are displayed to the user upon message receipt.
- Verify that messages are correctly displayed in the conversation window.
- Verify that messages from unknown contacts are handled according to user settings.

2.1.7 Message history

- Verify that the system stores sent and received messages in local message history.
- Verify that the user can view previous conversations with a selected contact.
- Verify that message history persists after restarting the application.
- Verify that the system supports search or navigation within the history (if applicable).

2.2 Usability

2.2.1 Sessions support

- Verify that the system allows multiple login sessions (if supported).
- Verify that the system restores user session after temporary disconnection

2.2.2 Fonts support

- Verify that the system supports multiple fonts and sizes for displaying messages.
- Verify that non-Latin character sets (e.g., Cyrillic, Asian languages) are displayed correctly.

2.2.3 Graphical emoticons

- Verify that the system provides graphical emoticons in messages.
- Verify that emoticons are correctly displayed on both sending and receiving ends.

2.2.4 Internet connection management

- Verify that the system automatically reconnects when internet connection is restored.
- Verify that the system provides error messages when connection is lost.
- Verify that the system minimizes data usage while maintaining stable connection.

2.2.5 Display support

- Verify that the system adjusts UI layout to different Palm device screen resolutions.
- Verify that messages are readable in both portrait and landscape modes (if supported).

2.3 Design Constraints

2.3.1 Maximum message length

- Verify that the system enforces the maximum message length defined in requirements.

- Verify that the system notifies the user when the maximum message length is exceeded.

2.3.2 Test environment

- Verify that the system operates correctly within the defined Palm OS version and hardware specifications.
- Verify that the system functions in both emulator and physical device environments.

2.3.3 Characters

- Verify that the system supports Unicode characters.
- Verify that unsupported characters are handled gracefully (e.g., replacement with “?” symbol).

2.4 Interfaces

2.4.1 User Interfaces

- Verify that the system provides a user-friendly interface for messaging functions.
- Verify that UI elements (buttons, menus, lists) are intuitive and consistent with Palm UI standards.
- Verify that error messages and prompts are clearly displayed to the user.
- Verify that navigation between screens (contacts, chat, settings) is smooth and logical.

3. Feature not to be tested

3.1 Setup a user account

- Verify that the system provides the capability to setup user account for ICQ service.

- Verify that the system provides the capability to setup several user accounts.
- Verify that the system provides the capability to set option “Remember password” for each user account individually.

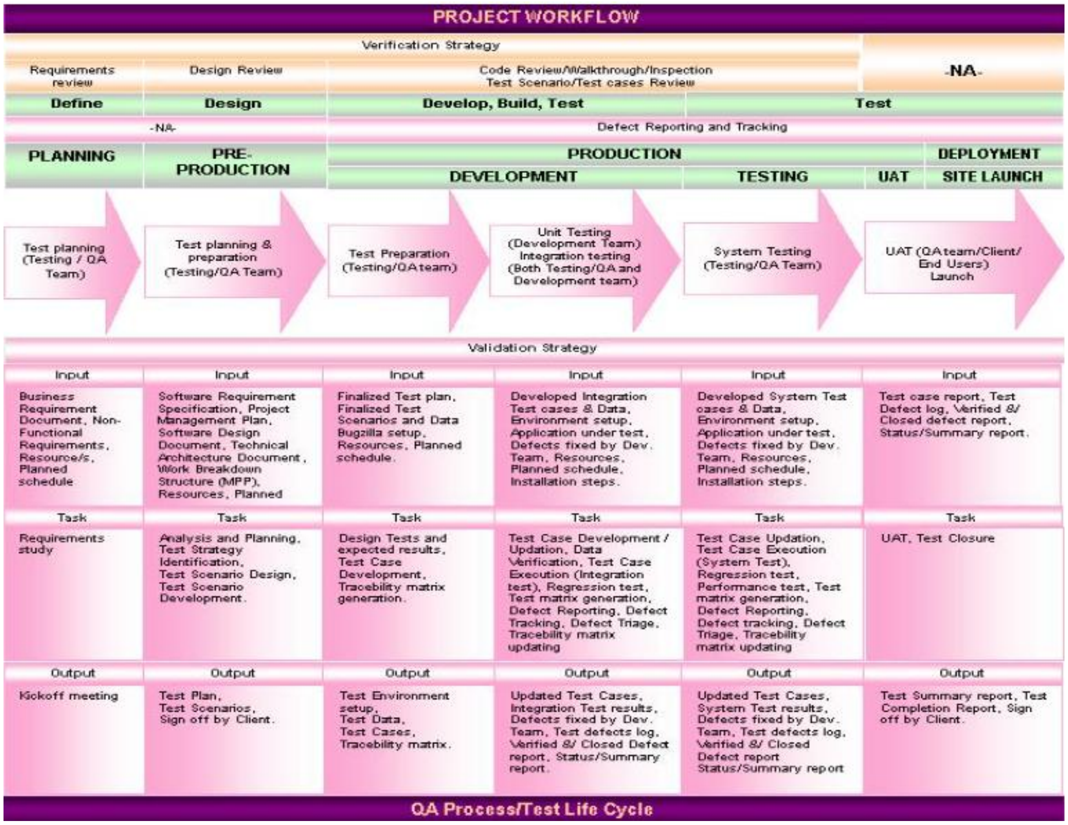
3.2 Remove user account

- Verify that the system provides the capability to remove an existing ICQ user account from the device.
- Verify that after removal, the account no longer appears in the login list.
- Verify that removing an account does not affect message history unless explicitly chosen by the user.
- Verify that the system provides a confirmation prompt before account removal.

4. Test Strategy

The Test Strategy presents the recommended approach to the testing of the ABC Development Project. The previous section on Test Requirements described what would be tested; this describes how it will be tested.

Test Process Workflow



PROJECT WORKFLOW

Verification Strategy					
Requirements review	Design Review	Code Review/Walkthrough/Inspection Test Scenario/Test cases Review			-NA-
Define	Design	Develop, Build, Test		Test	
-NA-		Defect Reporting and Tracking			
PLANNING	PRE-PRODUCTION	PRODUCTION			DEPLOYMENT
		DEVELOPMENT		TESTING	UAT SITE LAUNCH
Test planning (Testing / QA Team)	Test planning & preparation (Testing/QA Team)	Test Preparation (Testing/QA team)	Unit Testing (Development Team) Integration testing (Both Testing/QA and Development team)	System Testing (Testing/QA Team)	UAT (QA team/Client/ End Users) Launch
Validation Strategy					
Input	Input	Input	Input	Input	Input
Business Requirement Document, Non-Functional Requirements, Resource/s, Planned schedule	Software Requirement Specification, Project Management Plan, Software Design Document, Technical Architecture Document, Work Breakdown Structure (MPP), Resources, Planned	Finalized Test plan, Finalized Test Scenarios and Data Bugzilla setup, Resources, Planned schedule.	Developed Integration Test cases & Data, Environment setup, Application under test, Defects fixed by Dev. Team, Resources, Planned schedule, Installation steps.	Developed System Test cases & Data, Environment setup, Application under test, Defects fixed by Dev. Team, Resources, Planned schedule, Installation steps.	Test case report, Test Defect log, Verified &/ Closed defect report, Status/Summary report.
Task	Task	Task	Task	Task	Task
Requirements study	Analysis and Planning, Test Strategy Identification, Test Scenario Design, Test Scenario Development.	Design Tests and expected results, Test Case Development, Tracebility matrix generation.	Test Case Development / Updation, Data Verification, Test Case Execution (Integration test), Regression test, Test matrix generation, Defect Reporting, Defect Tracking, Defect Triage, Tracebility matrix updating	Test Case Updation, Test Case Execution (System Test), Regression test, Performance test, Test matrix generation, Defect Reporting, Defect tracking, Defect Triage, Tracebility matrix updating	UAT, Test Closure
Output	Output	Output	Output	Output	Output
Kickoff meeting	Test Plan, Test Scenarios, Sign off by Client.	Test Environment setup, Test Data, Test Cases, Tracebility matrix.	Updated Test Cases, Integration Test results, Defects fixed by Dev. Team, Test defects log, Verified &/ Closed Defect report, Status/Summary report.	Updated Test Cases, System Test results, Defects fixed by Dev. Team, Test defects log, Verified &/ Closed Defect report Status/Summary report	Test Summary report, Test Completion Report, Sign off by Client.

QA Process/Test Life Cycle

The above diagram explains the complete QA process/Test life Cycle in General. Following are the steps which would explain in detail , the Test Strategy to be followed ABCApplication.

Step1. Test Organize/Review Project Documentation

Step2. Test plan

Step3. Test Design/Development

Step4. Unit Test Execution

Step5. Integration/System Test Execution

Step6. Defect Tracking and Management

Step7. Update Documents and Results

Step8. Test Reports

Step9. UAT and Closure

Test Organize/Review Project Documentation:

Documentation reviews provide a means for testing the accuracy and completeness of the planning, requirements and specifications. Throughout the project, periodic reviews will be held to assure the quality of project documentation. These reviews will:

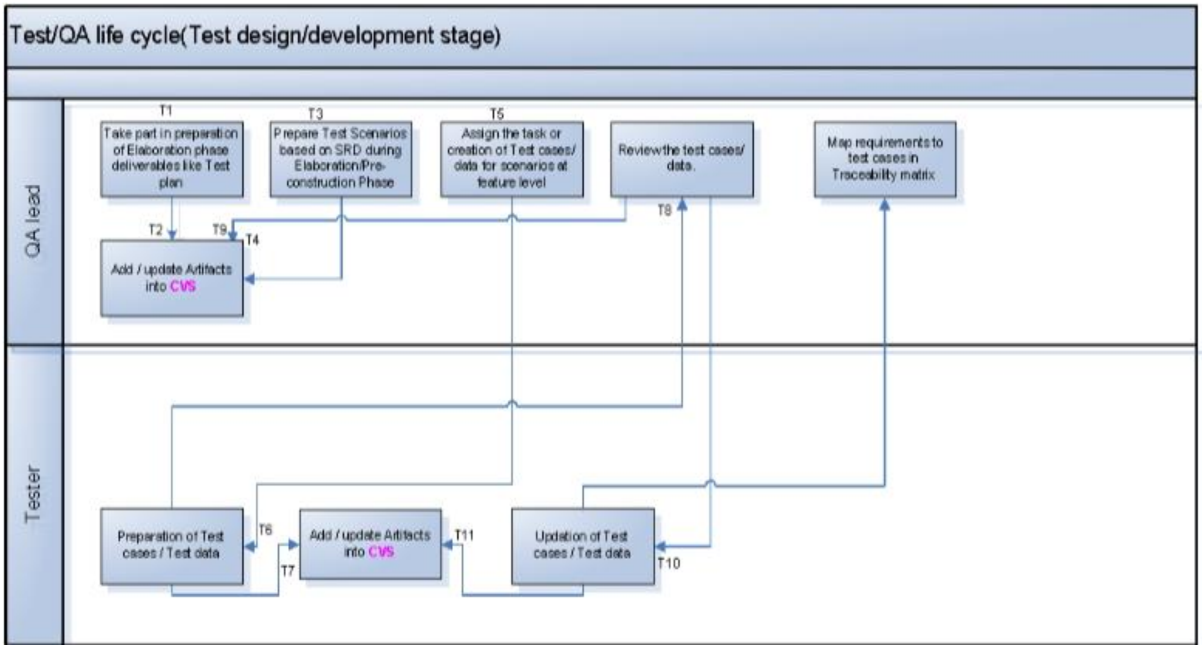
- Ensure project plans have adequate time allocated for testing activities and determine limitations.
- Ensure that the Business Requirements, Information Site Flow, Use Cases, Business Rules, and Technical Design documents clearly articulate the functionality of the ABC Project

Develop System Test Plan:

This step of the testing process involves creation of the System Test Plan (this document). This will serve as the guidepost for development of test cases and for integration of testing with other project activities.

- This plan describes at a high level the overall testing plan and strategy for the ABC Application.
- Professional Access will follow this plan to develop test scenarios/cases and scripts that will be used for system testing.
- Test scenarios will be described in separate document(s).
- Test Cases will be described in separate document(s)
- Professional Access will obtain test accounts and Ids for Interface testing (see Scope)

Test Design/Development:


Test/QA life cycle(Test design/development stage)
QA lead
Tester
T1
Take part in preparation of Elaboration phase deliverables like Test plan
T3
Prepare Test Scenarios based on SRD during Elaboration/Pre-construction Phase
T5
Assign the task or creation of Test cases/ data for scenarios at feature level
Reviewthe test cases/ data.
Map requirements to test cases in Traceabilty matrix
T2
T9
T4
T8
Add / update Artifacts into CVS
Preparation of Test cases / Test data
T6
T7
Add / update Artifacts into CVS
T11
Updation of Test cases / Test data
T10

Brief explanation of Test Design/Development workflow with respect to the Process flow diagram displayed above:

Task ID	Description
T1, T2	Test Lead takes part in preparation of Elaboration phase deliverables like Test Plan and updates the Artifacts in CVS for further reference.
T3, T4	From Post-elaboration phase to Pre-construction phase, Test Scenarios are designed by the Test Lead for the modules/features available in SRD. Once the final draft version of SRD with all modules/features specifications is received, Test Scenarios are completed and stored in CVS.
T5	Test Lead assigns the task of test case/test script creation to test team members during the Construction phase.
T6, T7	For all Test Scenarios created earlier, Test Team members design Test Cases/Test Scripts during Construction Phase. All created Test Cases/Test Scripts are stored in CVS.
T8, T9	Test Lead reviews all created Test Cases/Test Scripts and updates the review comments in CVS.

T10	Test Team members check the review comments, update respective Test Cases, and store the updated versions in CVS.
T11	Test Lead maps the requirements to Test Cases in the Traceability Matrix, showing which Test Case covers which functionality/requirement.

Written test cases and scripts will be used to direct system testing efforts. Professional Access test team will write these in accordance with the System Test Plan.

- Tests will be developed to exercise the required functionality for the website, validate data integrity, and ensure that data is passed or received successfully from external interfaces. Test Cases will be written in a separate document appended to this plan.
- Each test case will document the steps or actions required to exercise a specified area of functionality. The test cases will be reviewed to verify that they properly validate the intended functionality. Actual testing will be performed by executing the steps of the test case. A pass/fail notation will be made for each step.
- Each test case will be executed manually and using automated testing tool(for Performance/Load testing) using the browser versions mentioned in Test environment section. A pass/fail notation will be recorded for each condition tested, noting the severity and reason for each instance of failure. Test scripts to perform Performance/Load testing will be executed automatically during the System testing phase.

Unit Test Execution:

Unit testing verifies each module, component, object, or program developed is functionally correct and conforms to requirements. A unit is defined as a single program function in terms of inputs, processes and outputs. A program unit is small enough that the developer who developed it can test it in great detail.

The developer that wrote the code is responsible for creating, updating and executing the unit tests after each successful build in the development environment. Separate document has been prepared drafting Unit test strategy.

Integration Testing:

The objective of these tests is to ensure that all the components of the system function properly together and that the application interfaces properly with external applications.

Entrance Criteria:

- All functions to be tested have successfully passed unit testing
- All Severity 1 and 2 defects are fixed and have successfully passed unit testing. (See Defect Management portion of this document for severity definitions)
- Software build is properly version controlled
- Build report has been completed and submitted with build
- All hardware and software configurations are in place and ready to test
- All test cases required for integration testing have been prepared
- All required integrated systems are available

Exit Criteria:

- All components delivered and tested function as detailed in the documents in the References portion of
- this document
- Test cases have been updated if and when functionality has changed
- Test results report is developed/updated
- All new defects have been logged into the issues tracking database

System Testing:

The test team will conduct a system test to verify that the software matches the defined requirements. Once the application has executed successfully under integration test, each test suite will be executed against the other supported configurations to ensure defects are not created because the system configuration has changed. A separate test environment must be established for all hardware, software, and browser configurations supported.

Entrance Criteria:

- All functions tested have successfully passed integration testing

- All severity 1 and 2 defects are fixed and have successfully passed regression testing
- Test cases have been updated if and when functionality has changed
- All test cases required for system testing have been prepared
- All hardware and software configurations are in place and ready to test
- All required integrated systems are available

Exit Criteria:

- All Severity 1 and 2 defects are fixed and have successfully passed regression testing
- The risks associated with not correcting any outstanding Severity 3and 4 defects have been identified and signed off by the Project Manager, Technical Lead, QA Lead
- All components delivered and tested function as detailed in the documents in the References portion of this document
- Regression tests have been performed and executed successfully
- Test results report is developed/updated
- All new defects have been logged into the issues tracking database

4.1 Testing Types

4.1.1 Function and Usability Testing

Test Objective:	To ensure proper target-of-test functionality, including navigation, data entry, processing, and retrieval.
Technique:	Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following: • The expected results occur when valid data are used. • The appropriate error or alert messages are displayed when invalid data are used.
Completion Criteria:	• All planned tests have been executed. • All identified defects have been fixed.

Special Considerations:	Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test.

4.1.2 User Interface Testing

Test Objective:	Ensure that the system provides a user-friendly, intuitive, and consistent interface that allows users to easily interact with the core application functions. The UI should correctly reflect the business requirements and functionalities, including smooth navigation between different screens.
Technique:	Perform testing by directly observing and interacting with the user interface to verify that: • UI components (buttons, menus, lists, icons) are displayed in the correct positions, are easy to understand, and consistent with the defined design standards. • Error messages, warnings, and prompts are displayed clearly and are easily recognizable. • Navigation between screens (e.g., Contacts, Chat, Settings) is smooth, logical, and consistent. • The interface supports proper user experience (UX) without causing confusion during use.
Completion Criteria:	• All planned UI test cases have been executed. • Each screen has been successfully verified against the test version or within

	acceptable tolerance.
Special Considerations:	• Testing must be performed across multiple screen resolutions and different browsers. • Not all object properties may be accessible for testing.

4.2 Tools

The following tools are employed in this project:

	Tool	**Vendor/In-house**	**Version**
Bug Tracking	Mantis	Open Source	Latest
Project Management	MS project	Microsoft	2019/365
Automation Testing for web UI	Selenium	Open Source	15.x
Compatibility Testing	BrowserStack	BrowserStack(Cloud)	Latest

5. Resources

5.1 Roles

This table shows the staff required for the project.

Human Resources		
Employee	Minimum Resources Recommended (number of full-time roles allocated)	Specific Responsibilities/Job Description
Test Developer		Identifies, sets priorities for, and implements test cases. Responsibilities: • generates test plan • evaluates effectiveness of test effort
Tester		Executes tests. Responsibilities: • develops test case specifications • develops test cases • executes tests • logs results • monitors bug fixing • requests for documentation to be changes
QA Lead		• defines overall test strategy and scope. • prepares and maintains the master

		test plan. • allocates resources and assigns responsibilities. • tracks progress and ensures milestones are met. • communicates test status and risks to stakeholders. • approves final test deliverables and sign-off.
Environment Manager		• sets up and maintains the test environments (hardware, OS, simulators, emulators). • installs necessary software, tools, and patches. • ensures environment consistency across builds. • supports testers with environment issues.

6. Project Mileston

Test activities and milestones are very much dependent upon the development iterations. The Construction Phase is split into 3 builds, release candidate and release version. Each build contains a full test cycle of test planning,design, development, execution, and evaluation.

The following table shows Test Milestones. effort, start date, and end date.

Milestone Task	Effort	Start Date	End Date
Build 1 Plan Test Design Test Execute Test Evaluate Test	123.7	07/Sep/2010	07/Sep/2010
Build 2 Plan Test Design Test Execute Test Evaluate Test	124.0	20/Sep/2010	20/Sep/2010
Build 3 Plan Test Design Test Execute Test Evaluate Test	124.6	09/Oct/2010	09/Oct/2010

7. Deliverables

7.1 Test Model

Test Model defines all test cases, and references, test procedures and test scripts which are associated with each test case.

7.2 Test Logs

Microsoft Word is used to record and report test results.

7.3 Bug Reports

Mantis bug tracker is used for logging and tracking individual malfunctions.

7.4 Test Scenarios

High-level descriptions of what needs to be tested, grouped by functional area.

They help identify coverage before creating detailed test cases.

7.5 Test Case Execution Report

Summarizes the status of executed test cases, including number of passed, failed, blocked, or deferred cases.

7.8 Test Summary

A final consolidated report of the test cycle, summarizing coverage, defect density, quality risks, and overall readiness for release.

Appendix A Project Tasks

Test-related tasks are given below:

- Test Planning
  - identify requirements for testing
  - assess risks
  - develop test strategy
  - identify test resources
  - create schedule
  - generate Test Plan
- Test Design
  - prepare workload analysis
  - identify and describe test cases
- Test Execution
  - execute test procedures
  - verify results
  - investigate unexpected results
  - log bugs
- Evaluate Test
  - evaluate test-case coverage
  - evaluate code coverage
  - analyze defects
  - determine whether Test Completion Criteria and Success Criteria have been achieved