

Chapter 5 – Evaluation

5.1 – Introduction

The test plan, test results, and user evaluation are all described in the Evaluation chapter. To check and verify the system's performance, a wide range of data should be confirmed. During this step, the system's feature is assessed. The program is put through its paces in accordance with the test strategy.

5.2 - Description of testing approach

“A test approach is the test strategy implementation of a project, defines how testing would be carried out. Test approach has two techniques:

Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

Reactive - An approach in which the testing is not started until after design and coding are completed.

There are many strategies that a project can adopt depending on the context and some of them are:

- Dynamic and heuristic approaches
- Consultative approaches
- Model-based approach that uses statistical information about failure rates.
- Approaches based on risk-based testing where the entire development takes place based on the risk
- Methodical approach, which is based on failures.
- Standard-compliant approach specified by industry-specific standards.” [6]

5.3 - Test plan

“A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project. It describes the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the

test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.” [7]

5.3.1 - Verification and Validation

In software engineering, the phrases verification and validation refer to two separate forms of investigation. The most common definitions are:

Validation: Are we building the right system?

Verification: Are we building the system, right?

In other words, “**Validation** is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product. **Verification** will help to determine whether the software is of high quality, but it will not ensure that the system is useful. Verification is concerned with whether the system is well-engineered and error-free.” [8]

5.3.2 - Testing Process

- **Unit testing**

Single units/components of software are tested in unit testing, which is a type of software testing. The objective is to validate that each unit of the software functions as designed. A unit is the tiniest piece of software that can be tested.

- **Integration Testing**

After unit testing, integration testing is performed, in which individual units are grouped together and tested as a group. This testing level's goal is to show problems in the interfaces and interactions between integrated components. Depending on the definition of a unit, either white-box or black-box testing can be used.

- **System Testing**

System testing is a type of testing that is done on an entire integrated system to see if it meets its criteria. The black box testing approach is used to test the system's functionality from end to end during system testing.

- **Acceptance Testing**

This was the last stage of the testing process before the system was put into service. Instead of using fictional test data, the system was tested with data provided by the system procurer. This is a product beta test conducted by genuine end-users.

- **Regression Testing**

Regression testing is a sort of software testing used to ensure that a recent program or code modification hasn't broken current features. The major goal of this testing is to ensure that the system's existing functional and non-functional features have not been harmed by the recent changes.

5.4 - Proof of testing of work

The most significant component of this phase is creating test scenarios. Test cases that are well-planned should be able to validate system module functionality. All forms of system requirements must be validated by system test cases. The essential components of the test cases built for this system are listed below.

- **Test case for Customer Registration**

Test No	Test Case	Expected Result	Status
1	Submit the form without filling in the mandatory fields.	Prevent submitting and show error messages	Pass
2	Use already exists username and email address	show error message	Pass
3	Invalid email insert	show error message	Pass
4	Incorrect Password length insert	show error message	Pass
6	Invalid phone number insert	show error message	Pass
7	Enter the numbers in the text fields	show error message	Pass
8	Invalid zip inserts	show error message	Pass
9	Submit the form with correct data	Redirect to the login page to login to the dashboard	Pass
10	Without login URL browsing to the customer dashboard	Redirect to login page	Pass

Table 5. 1 – Test Case for Customer Registration

- **Test case for Customer Login**

Test No	Test Case	Expected Result	Status
1	Enter valid username and password	Successful login to the system	Pass
2	Enter username only	show error message	Pass
3	Enter only password	show error message	Pass
4	Enter invalid username and password	show error message	Pass

Table 5. 2 – Test Results for Customer login

- **Test case for Checkout Process**

Test No	Test Case	Expected Result	Status
1	Submit the form without filling in the mandatory fields.	Prevent submitting and show error messages	Pass
2	Enter the numbers in the text fields	show error message	Pass
3	Invalid email insert	show error message	Pass
4	Invalid phone number insert	show error message	Pass
5	Invalid zip inserts	show error message	Pass
6	With out check the delivery check box	Show error message of delivery form	Pass
7	Without selecting payment method	Show error message	Pass
8	Change the delivery value related to the province	Show delivery charges and calculate to total	Pass
9	Submit the form with correct data	direct to the invoice page	Pass
10	Without login URL browsing to the checkout and invoice	Redirect to the cart page	Pass

Table 5. 3 – Test Result for Checkout Process

5.5 - User Acceptance Testing

“User Acceptance Testing (UAT), also known as beta or end-user testing, is defined as testing the software by the user or client to determine whether it can be accepted or not. This is the final testing performed once the functional, system and regression testing are completed.

The main purpose of this testing is to validate the software against the business requirements. This validation is carried out by the end-users who are familiar with the business requirements.

UAT, alpha and beta testing are different types of acceptance testing.

As the user acceptance test is the last testing that is carried out before the software goes live, obviously this is the last chance for the customer to test the software and measure if it is fit for the purpose. “[9]

5.6 - User Evaluation

User evaluation is evaluation based on user input, that is, an evaluation that includes those for whom the system has suggested users. Probable methods, observational methods, questionnaires, interviews, and physiological monitoring methods are all examples of user evaluation procedures. Different sorts of users with varying levels of access privileges were selected and included in the sample, and user evaluation questionnaires were distributed to gather input.

Figure 5.1 depicts an example evaluation questionnaire for testing user satisfaction, figure 5.2 depicts a user acceptance test evaluation sample, and figure 5.3 depicts the system's user feedback results analysis diagram.

User Feedback Form – Computer Hardware Purchasing and Troubleshooting Assistant Management System

Role of User:

Test Date:

Mark (✓) inside the box. Please provide a more comprehensive response with your honest option.

No	Description	Exceptional	Satisfied	Bad
1	User interface design			
2	system's response time			
3	Form validation			
4	Understandability of error messages			
5	Interaction with navigation links and buttons			
6	system's responsiveness			
7	Report generating capability			
8	Recognizing the system's design			
9	Efficiency of the functionalities			
10	Overall impressions of the system			

Thank You

Figure 5. 1 – User Feedback Form

**User Feedback Form – Computer Hardware Purchasing and Troubleshooting
Assistant Management System for U-star Digital**

Role of User: Customer.....

Test Date: 26/03/2022.....

Mark (✓) inside the box. Please provide a more comprehensive response with your honest option.

No	Description	Exceptional	Satisfied	Bad
1	User interface design	✓		
2	system's response time	✓		
3	Form validation		✓	
4	Understandability of error messages		✓	
5	Interaction with navigation links and buttons		✓	
6	system's responsiveness	✓		
7	Report generating capability		✓	
8	Recognizing the system's design		✓	
9	Efficiency of the functionalities		✓	
10	Overall impressions of the system	✓		

Thank You

Figure 5. 1 – User Acceptance Test Evaluation Sample

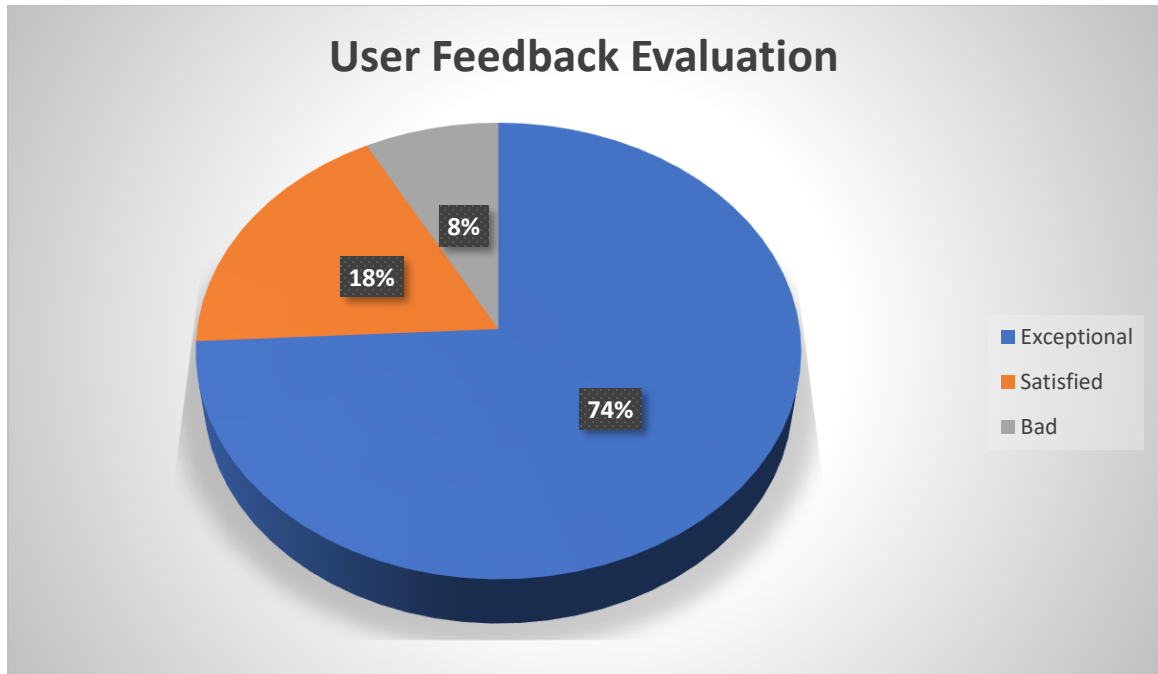


Figure 5. 3 – Summery of User Feedback Evaluation