

Evaluation

7.1. Introduction

Previous chapter I have discussed the key implementation methods for this system. This chapter is dedicated for the verification and validation of the product. This will gives some outline of methods of testing and detailed test cases for system testing.

7.2. Verification and validation

From the customers point of view this is the most important part of the software designed life cycle. With regard to the production confirmation system it's a very vital to validate and verify with the objectives it designed. Since this will be intend to use in an apparel production plant to capture the production in/out it's should be well accurate. In any non-conformity of the system could result the whole production facility stand still.

According to the software engineering literature definitions for verification and validation are as follows.

Verification: Are we building the software right

Validation: Are we building the right software

Verification of the software deals with the software development methodology. Software should meet its specification. Verification confirms

the software product meets its specifications like functional and non functional requirements. Validation confirms beyond the specification requirements. Generally the specification may not contain all the customer requirements. But by validation process we must ensure the software is produced up to the satisfaction of customer.

By performing above procedures, we can confirm the software product is ready to use. It developed the confidence in both customer and developer. Verification and validation can be tested from below methodologies.

- Software inspection or peer review
- Software testing

7.3. Software inspection or peer review

This methodology to confirm the software product is development based on the design documents, code developed during the system analysis and design. This can be performed without executing the software. Therefore it's also called as static analysis.

This methodology can be performed in all the steps of software design process. Coding review is also a part of this process. This testing methodology doesn't confirm that the software actually meets its specifications. Since its static analysis we cannot run it and get results. It only confirms the correspondence between the design documents and the software product. This is not demanding run able software to carry out this process. And there is no way of validate emergent properties like performance and reliability here.

7.4. Software testing

Software testing can be performed for the development completed software product or for the component that has been developed or to a prototype.

Therefore, if the waterfall model used, then the testing process will be carryout only after the end of development. But these can be performed before the end of development, in iterative model as the prototype is available before the product completion and in component base model for the components that are ready before hand.

In general there are 2 types of tests carried out in the software testing.

1. Validation testing

Validates with the specification and confirmed the product met its design objectives.

2. Defect testing

Try to find any defect in the system rather validates with its design objectives. By inserting fault data or following wrong precedence try to find any bug in the system.

In the small and middle scale software development testing can be done in a one go after the design step is completed or the prototype/component is developed. But in Large systems the testing process may carry out in each level of the development (eg: after component design, after the integration or interfacing done, after full system design and etc). Since the production confirmation system is not a mega size development it doesn't needs series of testing.

7.5. My testing approach

There are numerous test methods for software system testing. Depend on the requirement test methodology decided. Software system may comprise with many components and all the components are developed and integrated before the testing carried out. Below are the most common methodologies, I have taken consider in the testing my system.

7.5.1. Integration testing

When the system is designed from components this kind of a testing can be carried away. By incrementally link components and thoroughly testing them this testing can be done. In this process access to the system code is required.

7.5.2. Release testing/Black-box testing

With the completed software product the test is carried out. In this approach system is considered as black box and passes the input to system and get the expected output. If the expected output doesn't come it will be reported as a bug in the system. Refer to the figure 7.1.

To perform this test the system knowledge or software codes not needed. It needs some test cases that defined the input, output and instructions to give the input.



Figure 7.1 Black-box testing

When there are number of scenarios the test cases can be segregated base on the scenario. And then the test case can be performed for each scenario separately.

7.5.3. Performance testing

This will be test the system compatibility with emergent properties like performance, reliability etc. Before this type of test carried out the system should be tested for its main functionality.

Since my system can be separated into different business scenarios like bundle guide creation, UPC assignment, Line In etc release testing/black box testing can be performed. Integration testing cannot be performed due to the dependability of all the components.

7.6. Test cases

Below is the list of use cases needs to be tested.

1. Bundle guide creation
2. UPC and MR material assignment
3. Line In
4. Rejects update

I will be including the bundle guide creation test case here and rest will be included in to appendix E.

1. Bundle guide creation

Test Case ID	1			
Tested Component	Bundle guide creation			
Tested Area	Functionality			
Purpose	Create bundle guide when sales order line item plant and module is given			
Prerequisites	<p>Cut Fabric + RM have been received to plant according to Requirement</p> <p>Line wise daily requirement should be pre identified.</p> <p>Sewing Module wise efficiency should be considered.</p> <p>Sewing Production Order and Packing Production Order has to be created</p> <p>Material Requisition can only be generated after the respective Bundle Guide is Created.</p> <p>Material Requisition can only edited before the bundle guide is confirmed for the Line In Operation</p>			
Test Case Description				
No.	Test Case	Test data	Expected output	Result
1	Load sewing data	Sales order: 40677 Line item :10 Module SLK07 Plant 5200	System will load all the sales order relevant data + available prod. orders	Pass
2	Prod. Order selection	Select the prod. Ord from drop down menu Prd.Ord 1338619	Then the available sizes will display with open quantities	Pass

3	SAVE : Press save button	After adjusting the size qty and fill the set no cut no(not necessary) press saved button	You will be get the bundle no (combination of sales order+ line item + sequence)	Pass
	Test Results :		Pass	

Table 7.1 Test case for bundle guide creation

7.7. Summary

I have discussed in this chapter the evaluation and testing. Identified as suitable testing mechanism for this development is black box testing. Then I produce a one test case. Next chapter will be the conclusion and further work.



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk