Chapter 6

6. Testing and Implementation

6.1 Introduction

System Analysis and Design process including Requirement Analysis, Business Solution Options, Feasibility Study, Architectural Design was discussed in previous chapter.

Generally Software bugs will almost always exist in any software module. But it is not because of the carelessness or irresponsibility of programmer but because of the complexity. Humans have only limited ability to manage complexity. This chapter discusses about the testing of the solution and implementation methodologies.

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

6.2 Testing

Software Testing is the process of executing a program or system with the intent of finding errors. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions. Testing stages of the project can be explained as below and system was tested for all these stages.

● Component or unit testing
  - Individual components are tested independently;
  - Components may be functions or objects or coherent groupings of these entities.

● System testing
  - Testing of the system as a whole. Testing of emergent properties is particularly important.

● Acceptance testing
  - Testing with customer data to check that the system meets the customer’s needs.

6.2.1 Testing Methods and Comparison

6.2.1.1 Black Box Testing

Black Box Testing is testing without the knowledge of the internal workings of the item being tested. When black box testing is applied to software engineering, the tester selects valid and invalid input and what the expected outputs should be, but not how the program actually arrives at those outputs.Black box testing methods include equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing. This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance.

6.2.1.2 White Box Testing

White box testing (glass box testing) strategy deals with the internal data structures and algorithms. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc. These testers require programming skills to identify all paths through the software.

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

Types of white box testing includes code coverage (creating tests to satisfy some criteria of code coverage.),mutation testing methods, fault injection methods, static testing.

6.2.1.3 Comparison - Black Box Testing Vs White Box Testing

Differences between Black Box and White Box testing have been listed in below table 6.1. Based on below listed differences appreciate testing methods were used for testing.

Black Box Testing	**White Box Testing**
Designed based on the Software Requirement Specification (SRS)	Designed based on the control logic of the program unit
Tester needs no knowledge of implementation, including specific programming languages	As knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, which increases the cost
Help to expose any ambiguities or inconsistencies in the Software Requirement Specifications (SRS)	Helps in optimizing the code. It helps in removing the extra lines of code, which can bring in hidden defects.
Without clear and concise specifications, test cases are hard to design may leave many program paths untested	As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
Programmer and Tester can be independent of one another, avoiding programmer bias toward own work.	Tester might need assistance from programmer, which can cause biasness
More effective on larger units of code than glass box testing	More complicated and tedious when comes to large units of code

Table 6.1**:** Black Box Vs White Box Testing

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

6.2.1.4 Performance Testing

Performance testing checks to see if the software can handle large quantities of data or users. This is generally referred to as software scalability.

6.2.1.5 Boundary Testing

Boundary testing or boundary value analysis, is where test cases are generated using the extremes of the input domain, e.g. maximum, minimum, just inside/outside boundaries, typical values, and error values.

6.2.2 Testing Strategy and Test Plan

Black box testing and white box testing were used as the main testing techniques to test the entire system for all the stages. Although both of these testing techniques have advantages and disadvantages as discussed above, but when combined, they help to ensure thorough testing of the product.

All the functions, procedures are tested as single individual units first and after integrating individual units into one unit, integrated units were tested as a whole. When the function/procedure is tested for the first time, black box testing methodology was used. It was tested for expected results on different user inputs. If the function did not deliver the expected outcome white box testing methodology was used. So it was tested by going through each and every relevant code.

After testing for functionality of the functions/Procedures, these functions/procedures were combined and tested using black box testing methodology. If the results show any discrepancy than expected output, then again they were tested for expected results using white box testing methodology. Once the product was completed then it was tested using test data by using black box and white box approach. All the interfaces were checked using black box approach. Since validation rules have been applied at all the possible areas of user interfaces, testing of some areas were not required.

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

When preparing test data, more attention was paid to cover all the possible scenarios. For an example, it was considered values in out of the boundary, with in the boundary, less than the boundary, grater than the boundary, equal to the boundary limits when selecting test data. Further, numeric inputs were tested for some special values like zero, negative values, decimal values, integers, characters etc.

Performance testing also was done to verify whether expected numbers of users can work simultaneously in the system. This was done by creating multiple sessions in number of client machines in the LAN. Further it was checked whether the software can handle large quantities of data by loading dummy data into the system.

In case of data retrieve from database it is assured that the correct data is retrieved. When data is stored in the database, it is also tested. When deleting and updating the records in tables, it was checked whether the correct row is updated.

6.2.3 Test Cases

Two sample Test Cases have been attached in below Table 6.2 and Table 6.3 and the Test Report for those two test cases is available in the Table 6.4.

Test Case ID: 1

User Interface: Login

Operation: System Login

Action	Input	Expected Output	Status
Type User Name Password and click log in button. (Correct user name and correct password)	Nilmini MniNh55	Login to the System with appropriate permissions.	Pass
Type invalid User name and keep password empty then click log in button	GP Nilmini	Display error message “Invalid User Name.” After pressing OK button, set cursor focus to Username input box.	Pass
Type invalid User name and type correct password then click log in button	GP Nilmini MniNh55	Display error message “Invalid User Name.” After pressing OK button, set cursor focus to Username input box.	Pass
Type valid User name and keep password empty then click log in button	Nilmini	Display error message “Invalid Password” Set focus on password input box	Pass
Type valid User name and invalid password then click login button	Nilmini MniNh555	Display error message “Invalid Password” Set focus on password field	Pass
Keep both user name and password empty and click login button		Display error message “User Name should not be blank.” After pressing OK button, set cursor focus to username input box.	Pass
Type both user name and password invalid and click	NNilmini MniNh555	Display error message “Invalid User Name.”	Pass

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

login button		After pressing OK button, set cursor focus to Username input box.	
Keep user name blank and type correct password and click login button	- MniNh55	Display error message “User Name should not be blank” After pressing OK button, set cursor focus to Username input box.	Pass
Keep user name blank and type incorrect password and click login button	- MniNh5555	Display error message “User Name should not be blank.” After pressing OK button, set cursor focus to Username input box.	Pass

Table 6.2 : Test Case for Test Case ID 1

Test Case ID: 2

User Interface: Add/Edit Machine Types

Operation: Add Machine Type / Modify Machine Type Description

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

Action	Input	Expected Output	Status
Input MC type which is not available at present in the table. Then press “Enter” or Press “Tab” Key	AU-100	Set focus to Description	Pass
Input a description which has not been given for any already existing Machine type	Automatic Machine	Enable true the Save button and once press the save button, save the data and disable the Save button and clear the interface	Pass
Input a description without entering a machine type	Automatic Machine	Save button should not be enable to press	Pass
Input MC type which	AU-100	Display error message “This	Pass

is already available in the table. Then press “Enter” or Press “Tab” Key		code already Exists”. Then set focus to MC type input Box. Save button should not be enable to press	
Input MC type which is not available at present in the table. Then press “Enter” or Press “Tab” Key then type a description already available for another machine type. Then press save button.	AU-1000 Automatic Machine	Display error message “This Description is already Exists for another Machine Type”. Then set focus to Description input Box. Do not allow to save	Pass
Input MC type which is not available at present in the table. Then press “Enter” or Press “Tab” Key	AU-200	Save button should not activated enabling user to save without description	Pass
Input MC type which is already available in the table and already used in a Timing sheet. Then press “Enter” or Press “Tab” Key Then Press “Delete “ Button	AU-100	Relevant Description should come to the description input box when press the “Enter” or Press “Tab” Key When “Delete” button is pressed then display error message “Delete not applicable on this record”. Then set focus to MC type input Box. Do not allow to delete	Pass
Press “Clear” Button		All the entries in input boxes should be cleared	Pass
When no any input is		The “find” button which is	Pass

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

there in the MC type input box then click the “Find” button on the top of the left side		available in the right side of the MC type input box should be enable to press.	

Table 6.3 : Test Case for Test Case ID 2

6.2.4 Test Report

Test case ID	Actual output	Status
1	❖ Login to the system when user name and password is correct. ❖ Display error message “Invalid User Name” when the user name is incorrect and set focus to user name input box. ❖ Display error message “Invalid Password” when user name is correct and password is incorrect ❖ Display error message “User Name should not be blank” when user name is blank and Set focus on user name. Display error message “Incorrect Password” when user name is correct and pass word is blank or incorrect. Set focus on password field	Pass Pass Pass Pass Pass Pass
2	❖ Display error message when try to duplicate the MC type ❖ Do not allow to delete already used MC type in order to maintain integrity of the database ❖ Do not allow to save machine type with blank description ❖ Do not allow to use already existing description for new machine type	Pass Pass Pass Pass

Table 6.4 : Test Report

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

6.3 Implementation

System implementation projects are long difficult journeys by which organisations move from an old set of technology/methods/procedures to a new one. A software implementation method is a systematic structured approach to effectively integrate software based service or component into the workflow of an organizational structure or an individual end-user. The complexity of implementing product software differs on several issues. Examples are: the number of end users that will use the product software, the effects that the implementation has on changes of tasks and responsibilities for the end user, the culture and the integrity of the organization where the software is going to be used and the budget available. It is vital to select the right strategy for implementing the application to assure successful results.

6.3.1 Implementation Strategy

Since the software application consists of three modules as per in the high level architectural design, the implementation was done using iterative, incremental approach. Phase wise implementation process enables to execute by incrementally aligning the product with the end-user. Phase wise implementation process is shown in Figure 6.1

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

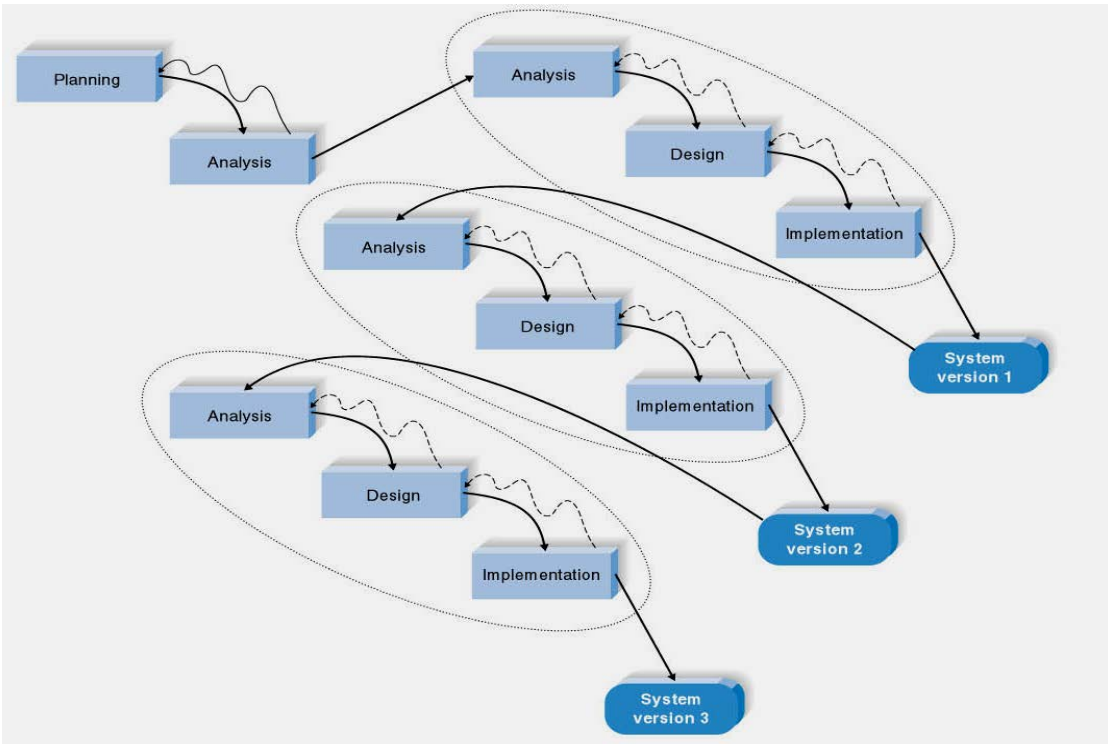


Figure 6.1 Phase Implementation Process

Comparison of the ways that can be used in implementing or introducing a new system is shown in below Table 6.5

Method	Main Advantages	Main Disadvantages
1. Immediate cutover Straight from old system to new system on a single date	Rapid, low cost	High risk. Major disruption if serious errors with the system
2. Parallel running Old system and new system run side-by-side for a period	Low risk than immediate cutover	Slower and higher cost than immediate cutover
3. Phased Implementation Different modules of the system are introduced sequentially	Good compromise between methods 1 and 2	Difficult to achieve due to interdependencies between modules
4. Pilot System Trial implementation occurs before widespread deployment		Has to be used in combination with the other methods.

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

Table 6.5 : Comparison of Software Implementation Methods

Direct Implementation

With this method of implementation the users stop using the manual system and start using the computer system from a given date.

The advantage of this method is that it is less costly in effort and time than any other method of implementation. The disadvantage of this method is that if problems occur the users do not have any alternative apart from returning to a manual system which may prove difficult if it has been discontinued.

Parallel Running

With parallel running, the new system is introduced alongside the existing system. With parallel running both systems (manual and computer, or old computer and new computer system) will be in operation at the same time. This has the advantage that the results from the new system can be compared with those of the old system.

However, it has the major disadvantage that each job is done twice and therefore it means a lot of extra work for the users.

Phased Implementation

The phased implementation method enables us to break our project in to smaller milestones. Major disadvantage is difficult to achieve due to interdependencies between modules

Pilot system

The pilot system of implementation involves the new system being fully implemented for a short period to establish any problems with the system. The method of implementation should cause the least amount of cost and time.

After evaluating above two methods, it was identified that the Phased method can be used since there were no interdependencies of each module of the new system.

First, implementation was done after completing 1st module and then other two modules were implemented in another two stages. After demonstrating the product for the relevant users, they were given opportunity to do a test run. First, system was tested by the users individually and then users were asked to test the system in multi-user environment for the entire cycle. Then based on the user feedback, minor adjustments were done. Once users confirmed that all the functionality is available, parallel run was done for a week. After identifying that there are no any operational issues and all the system outputs can be generated as expected, live run was started.

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

In the live implementation, all the users were given user accounts and passwords. Since user rights have been linked with the users account, access were set to activate at the time of login to the system automatically only for relevant functionalities of the relevant user. They have been given the facility to change their password at any time as they wish. For some areas, Audi trail has been introduced to identify who has done what, at what time on what date.

After moving to the live run, necessary actions were taken to take backups of the database every day in the night in order to use if there are any data losses or hard ware failure.

6.3.2. User Training

Before starting the implementation, the key users of the system were trained by people who really understand the system. Since the system is less likely to work successfully if the users resent it, or do not believe that it will help them, it was make arrangements to involve key users in major decisions, and keep the ultimate users well informed about what is happening, and any positive impact on their jobs, especially any current problems that the system will remove. This helped to create an atmosphere of interest and acceptance.

6.3.3. User guide

User guides were written in plain English rather than technical language. The guide covered how to run the system, how to enter data, how to modify data, how to set parameters, how to use utilities and how to save and print reports.

Further, a list of error messages and advice on what to do if something goes wrong also were included in the user guide.

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

6.3.4 Program Deployment

The deployment of the software was organized in very simple smarter manner. As standard practice, an executable setup CD was created including all the ActiveX controls and .dll files including other necessary supporting components. Using this setup CD, system can be installed in any client computer very easily.
In order to implement modifications, upgrades etc easily, the application files were kept in one place in the application server. After installing the setup in the client machine, user has to just create a short cut to the application file in order to run the system. This will improve the maintainability.

6.3.5 Problems Faced

- At the beginning, there was a resistance from the employees to move to the new increment system because this is a completely different system compared to the exiting system. This situation had to be handled carefully by having many discussions and by giving complete understanding to employees.

- There was a big resistance from the production supervisors when the new data collection form was introduced for collecting the shift wise daily production transaction data. This was kind of an additional work for them compared to the work load done by them earlier. To manage this situation, higher management involvements were obtained.

- Before the new implementation, all standard and job timing sheet were in hard form. But with the new implementation, it was converted into soft form. Additional effort had to be taken to change the users to new practice.

6.4 Sample User Interfaces

In order to demonstrate the user friendliness of the system few screen shots have been shown below.
Further some other user interfaces are included in the *Appendix G.*

User interface of the Main Menu is shown in the Figure 6.2

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

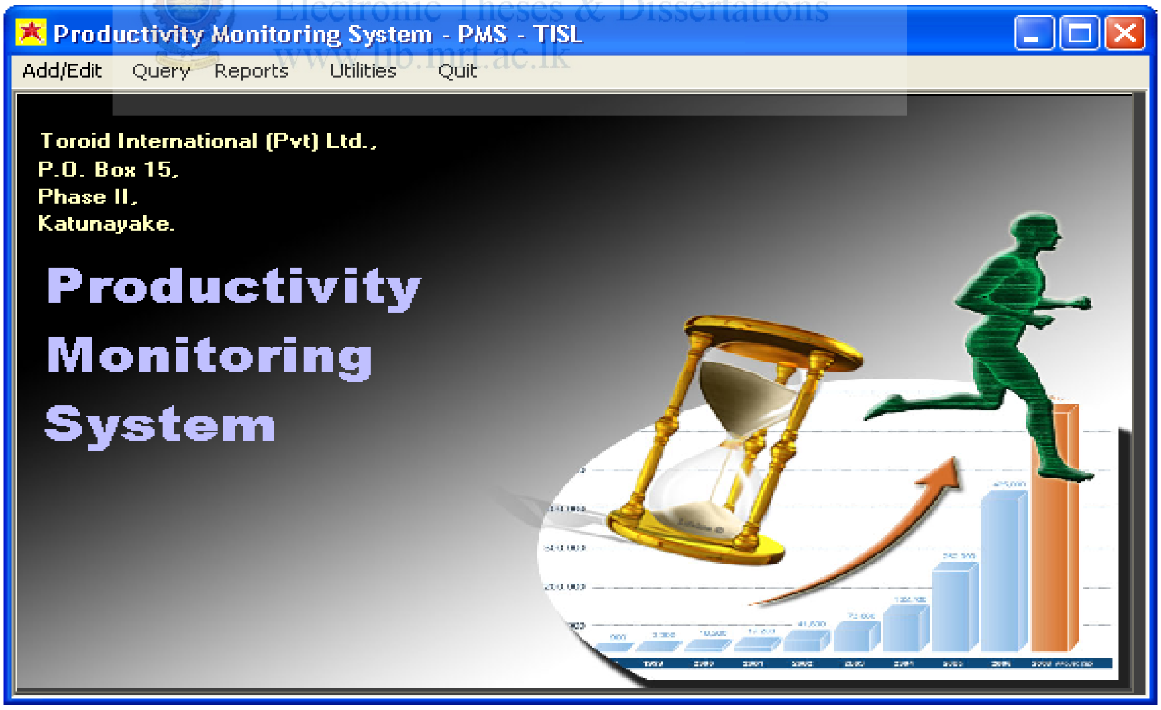
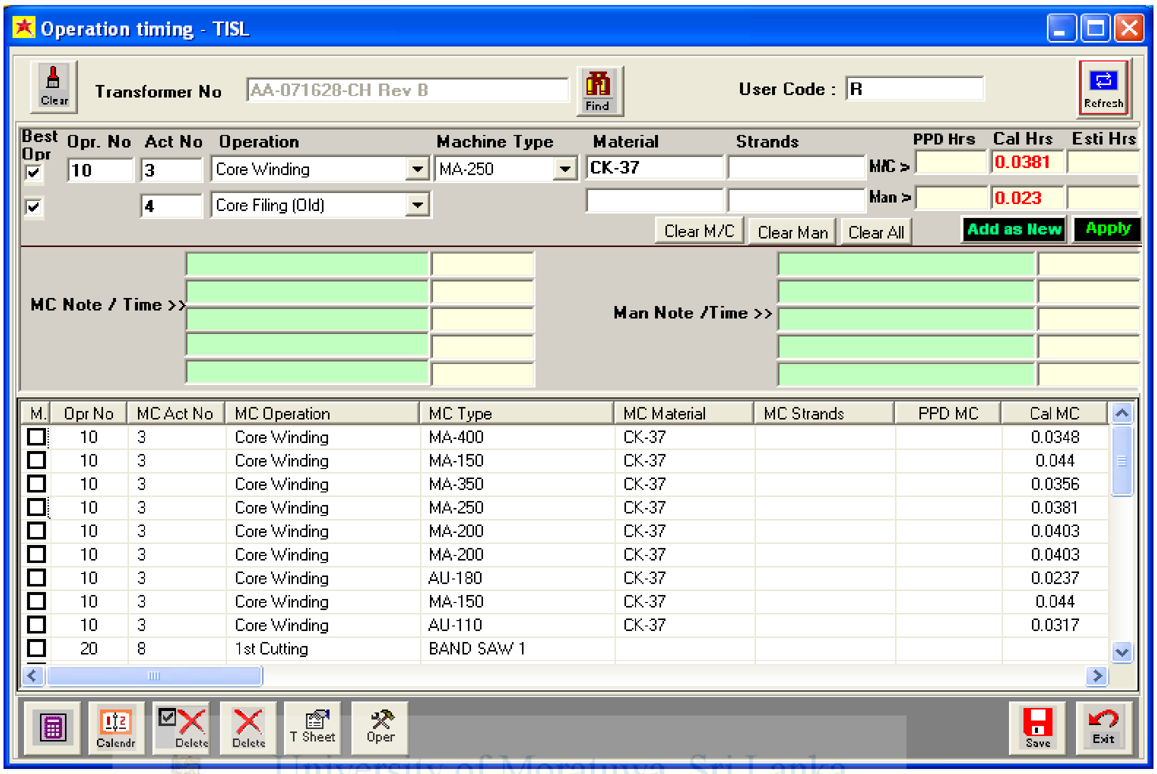


Figure 6.2 : User Interface of Main Menu Screen

User interface of the Transformer Timing Data Entering is shown in Figure 6.3



Operation timing - TISL

Clear | Transformer No | AA-071628-CH Rev B | Find | User Code : | R | Refresh

Best Opr	Opr. No	Act No	Operation	Machine Type	Material	Strands		PPD Hrs	Cal Hrs	Esti Hrs
✓	10	3	Core Winding	MA-250	CK-37		M/C >		0.0381	
✓		4	Core Filing (Old)				Man >		0.023	

Clear M/C | Clear Man | Clear All | Add as New | Apply

MC Note / Time >> | Man Note /Time >>

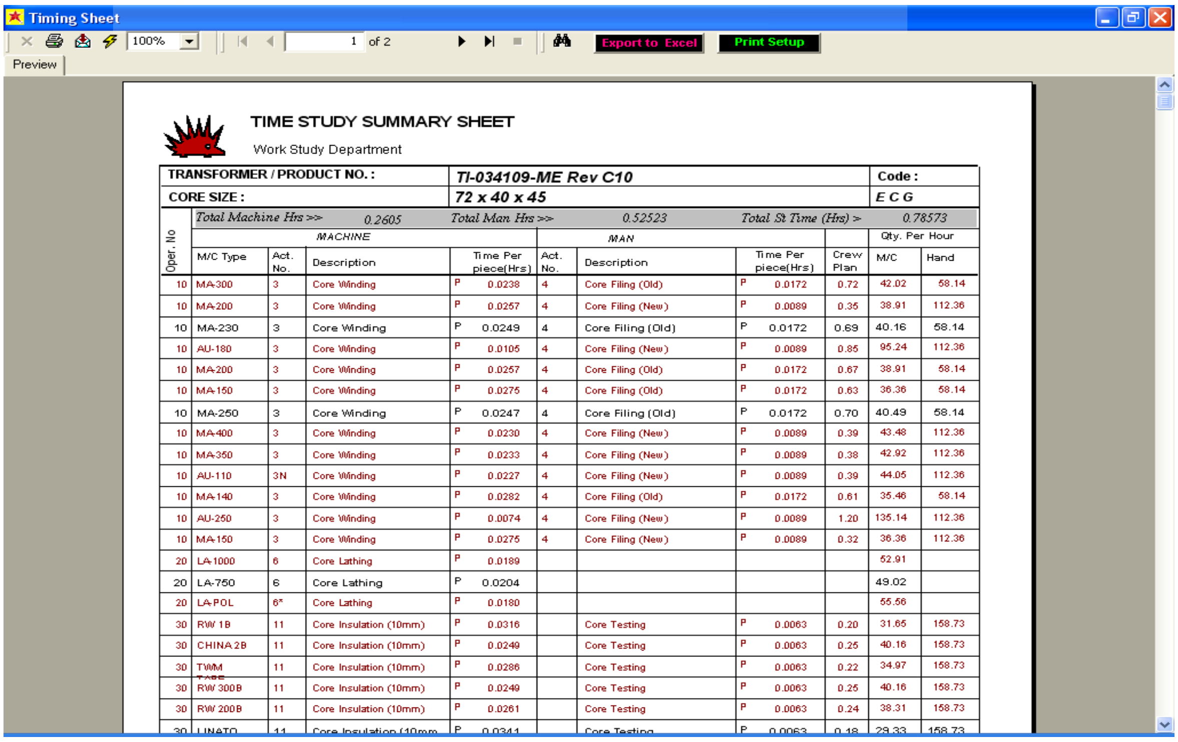
M.	Opr No	MC Act No	MC Operation	MC Type	MC Material	MC Strands	PPD MC	Cal MC
☐	10	3	Core Winding	MA-400	CK-37			0.0348
☐	10	3	Core Winding	MA-150	CK-37			0.044
☐	10	3	Core Winding	MA-350	CK-37			0.0356
☐	10	3	Core Winding	MA-250	CK-37			0.0381
☐	10	3	Core Winding	MA-200	CK-37			0.0403
☐	10	3	Core Winding	MA-200	CK-37			0.0403
☐	10	3	Core Winding	AU-180	CK-37			0.0237
☐	10	3	Core Winding	MA-150	CK-37			0.044
☐	10	3	Core Winding	AU-110	CK-37			0.0317
☐	20	8	1st Cutting	BAND SAW 1				

Calendr | Delete | Delete | T Sheet | Oper | Save | Exit

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

Figure 6.3 : User Interface of Transformer Timing Data Entering

User interface of the Timing Sheet Report is shown in Figure 6.4



Timing Sheet

100% | 1 of 2 | Export to Excel | Print Setup

Preview

TIME STUDY SUMMARY SHEET

Work Study Department

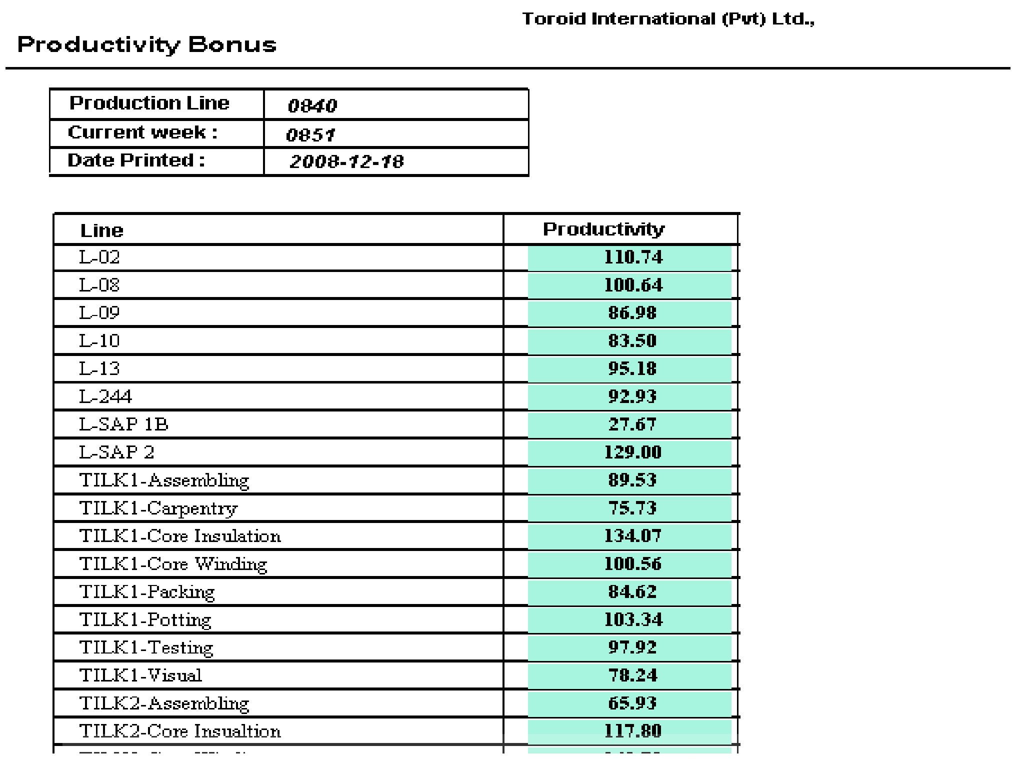
TRANSFORMER / PRODUCT NO. :	*TI-034109-ME Rev C10*	Code :
CORE SIZE :	*72 x 40 x 45*	*E C G*

Total Machine Hrs >> 0.2605 | *Total Man Hrs >>* 0.52523 | *Total St Time (Hrs) >* 0.78573

Oper. No	MACHINE				MAN				Qty. Per Hour	
	M/C Type	Act. No.	Description	Time Per piece(Hrs)	Act. No.	Description	Time Per piece(Hrs)	Crew Plan	M/C	Hand
10	MA-300	3	Core Winding	P 0.0238	4	Core Filing (Old)	P 0.0172	0.72	42.02	58.14
10	MA-200	3	Core Winding	P 0.0257	4	Core Filing (New)	P 0.0089	0.35	38.91	112.36
10	MA-230	3	Core Winding	P 0.0249	4	Core Filing (Old)	P 0.0172	0.69	40.16	58.14
10	AU-180	3	Core Winding	P 0.0105	4	Core Filing (New)	P 0.0089	0.85	95.24	112.36
10	MA-200	3	Core Winding	P 0.0257	4	Core Filing (Old)	P 0.0172	0.67	38.91	58.14
10	MA-150	3	Core Winding	P 0.0275	4	Core Filing (Old)	P 0.0172	0.63	36.36	58.14
10	MA-250	3	Core Winding	P 0.0247	4	Core Filing (Old)	P 0.0172	0.70	40.49	58.14
10	MA-400	3	Core Winding	P 0.0230	4	Core Filing (New)	P 0.0089	0.39	43.48	112.36
10	MA-350	3	Core Winding	P 0.0233	4	Core Filing (New)	P 0.0089	0.38	42.92	112.36
10	AU-110	3N	Core Winding	P 0.0227	4	Core Filing (New)	P 0.0089	0.39	44.05	112.36
10	MA-140	3	Core Winding	P 0.0282	4	Core Filing (Old)	P 0.0172	0.61	35.46	58.14
10	AU-250	3	Core Winding	P 0.0074	4	Core Filing (New)	P 0.0089	1.20	135.14	112.36
10	MA-150	3	Core Winding	P 0.0275	4	Core Filing (New)	P 0.0089	0.32	36.36	112.36
20	LA-1000	6	Core Lathing	P 0.0189					52.91	
20	LA-750	6	Core Lathing	P 0.0204					49.02	
20	LA-POL	6*	Core Lathing	P 0.0180					55.56	
30	RW 1B	11	Core Insulation (10mm)	P 0.0316		Core Testing	P 0.0063	0.20	31.65	158.73
30	CHINA 2B	11	Core Insulation (10mm)	P 0.0249		Core Testing	P 0.0063	0.25	40.16	158.73
30	TWM TAPE	11	Core Insulation (10mm)	P 0.0286		Core Testing	P 0.0063	0.22	34.97	158.73
30	RW 300B	11	Core Insulation (10mm)	P 0.0249		Core Testing	P 0.0063	0.25	40.16	158.73
30	RW 200B	11	Core Insulation (10mm)	P 0.0261		Core Testing	P 0.0063	0.24	38.31	158.73
30	LINATO	11	Core Insulation (10mm	P 0.0341		Core Testing	P 0.0063	0.18	29.33	158.73

Figure 6.4 : User Interface of Timing Sheet Report

User interface of the Productivity Bonus Report is shown in Figure 6.5



Toroid International (Pvt) Ltd.,

Productivity Bonus

Production Line	0840
Current week :	0851
Date Printed :	2008-12-18

Line	Productivity
L-02	110.74
L-08	100.64
L-09	86.98
L-10	83.50
L-13	95.18
L-244	92.93
L-SAP 1B	27.67
L-SAP 2	129.00
TILK1-Assembling	89.53
TILK1-Carpentry	75.73
TILK1-Core Insulation	134.07
TILK1-Core Winding	100.56
TILK1-Packing	84.62
TILK1-Potting	103.34
TILK1-Testing	97.92
TILK1-Visual	78.24
TILK2-Assembling	65.93
TILK2-Core Insualtion	117.80

Figure 6.5 : User Interface of Productivity Bonus Report

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

User interface of the Management Dash Board is shown in Figure 6.6

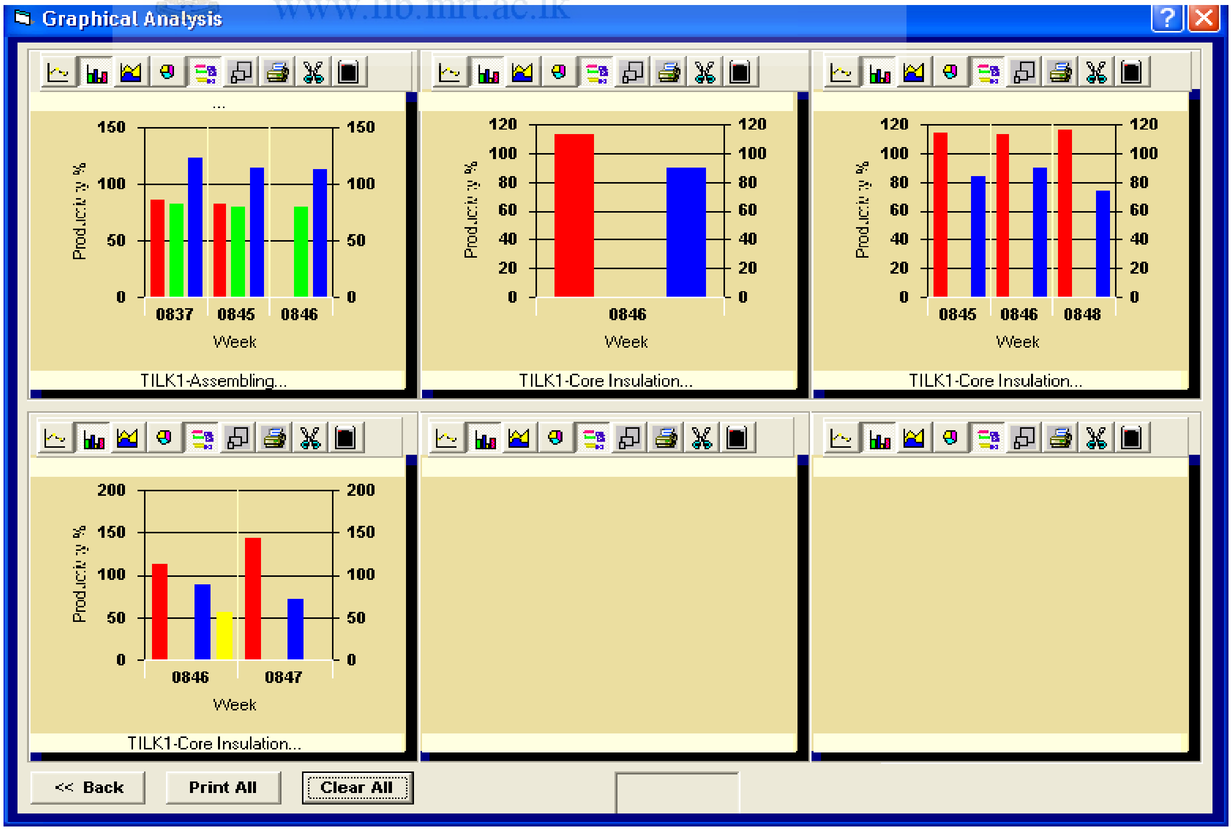


Figure 6.6 : User Interface of Management Dash Board

6.5 Summary

This chapter discusses about Testing Methods, Testing Strategy, and Test Plans of the solution and implementation methodologies with implementation strategy. Further, problems faced in the application development process also have been discussed in this chapter.

Once the software implementation was done, the next main activity was to evaluate the implemented software. In other words it should be evaluated whether the software meets all the requirements identified at the initial stage of finalizing SRS and the provided functionality works properly. The next chapter discusses about performing the software evaluation.

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