

IMPROVEMENT ON LABEL PRODUCTION SOFTWARE QUALITY WITH TEST
CLASS STANDARDIZATION USING XP TECHNIQUE

HARVINDER KAUR A/P RAJINDER SINGH

A dissertation submitted in partial fulfillment of the
requirements for the award of the degree of
Master of Computer Science

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2018

This dissertation is dedicated with much love to my Parents, Siblings, Husband and Family for their endless support, love and encouragement.

ACKNOWLEDGEMENT

First of all, I would like to warmly express my utmost gratitude to “Waheguruji” (GOD) for His blessing and given strength during the completion of this research.

I would also like to sincerely thank my supervisor Assoc. Prof. Dr. Dayang Norhayati Binti Abang Jawawi, for her continuous motivation, advices, encouragement and support from the beginning to the end of my research studies. She has been extremely helpful and offered great guidance throughout this research. Without her continual support and guidance, this research report would not have been the same as presented here.

Besides that, my special and heartfelt thanks goes to my parents Mr. Rajinder Singh and Mrs. Kaldeep Kaur, and my siblings Daljit Kaur, Harina Kaur, Kelvinder Singh, Simran Kaur, Manisha Kaur, and Harvinder Singh Bhullar for their love and supports for always being there when I needed them throughout my studies and difficult times. Not forgetting the very special thanks to my encouraging and beloved husband Dalevynder Singh for his sacrifice, support, love, and caring inspired me to overcome all the difficulties throughout my entire academic life.

I would also like to express my gratitude to Finisar Company for providing us sponsorship, financial support for my studies and opportunity to take this Master course and accomplish it till to the end. Last but not least, special thanks to my colleagues and friends for the collaboration in knowledge sharing with each other. Not forgetting, my special thanks to our program coordinator Dr.Mohamad Shukor Talib for helping us throughout this Master program. Finally, my sincere appreciation also extends to all of the lecturers and members in Faculty of Computing, UTM who have supported me to complete my studies and research directly or indirectly.

ABSTRACT

Label production software is an “in house” automation software which contains “Test Class” with the source code of function required by the customer and product engineer to support the testing and requirements of various type of products in Finisar. It is very important to make sure the release of label production software is bug and defect free, delivered on time without any delay to avoid any aging and shipment problem, meets requirements and expectations of user and customer, and good quality through reusability. This software quality able to be improved with test class standardization here means coordinate, and organize the functions in the source code of the test class as required by the users into one standard format where it can be reusable on various product types. Current label production software contains unstable test classes with duplication functions, hard coded data and unable to be reused on various products testing which causes problem in testing of any new products with new requirement and affect the quality of the software. By studying further on the problem it is known that the test class developed though process flow diagram need to be improved first in order to develop standardize test classes. In this study it is proposed to use agile technique “Extreme Programming (XP)”. This XP technique will be implemented in the existing process flow diagram as it is able to support each of the steps required from test software request to develop test classes until testing and release of final version to production floor. By implementing XP into the existing process flow diagram it will be helpful in developing standardize test classes (STC). This STC will be measure for the reusability together with the release software version in order to prove the quality of the label production software quality is improved. In summary, label production software quality reusability can be improved with test class standardization through using the XP technique as proposed in process flow diagram.

ABSTRAK

Perisian pengeluaran label adalah perisian automasi "dalam rumah" yang mengandungi "Kelas Ujian" dengan kod sumber fungsi yang diperlukan oleh pelanggan dan jurutera produk untuk menyokong ujian dan keperluan pelbagai jenis produk di Finisar. Adalah sangat penting untuk memastikan pelepasan perisian pengeluaran label adalah berfungsi tanpa sebarang masalah, dihantar tepat pada waktunya tanpa sebarang kelewatan untuk mengelakkan masalah dalam penghantaran, memenuhi keperluan dan jangkaan pengguna dan pelanggan, dan kualiti yang baik melalui kebolehgunaan semula. Kualiti perisian ini dapat ditingkatkan dengan standardisasi ujian kelas di sini bermaksud menyelaraskan, dan mengatur fungsi dalam kod sumber kelas ujian seperti yang dikehendaki oleh pengguna ke dalam satu format standard di mana ia boleh digunakan semula pada pelbagai jenis produk. Perisian pengeluaran label semasa mengandungi kelas ujian yang tidak stabil dengan fungsi duplikasi, data berkod keras dan tidak boleh digunakan semula pada pelbagai ujian produk yang menyebabkan masalah dalam menguji mana-mana produk baru dengan keperluan baru dan menjejaskan kualiti perisian. Dengan mengkaji lebih lanjut mengenai masalah diketahui bahawa kelas ujian dibangunkan walaupun rajah aliran proses perlu diperbaiki terlebih dahulu untuk membangunkan kelas ujian piawai. Dalam kajian ini dicadangkan menggunakan teknik tangkas "Extreme Programming (XP)". Teknik XP ini akan dilaksanakan dalam rajah aliran proses yang sedia ada kerana ia dapat menyokong setiap langkah yang diperlukan dari permintaan perisian ujian untuk membangun kelas ujian sehingga ujian dan pembebasan versi terakhir ke lantai produksi. Dengan melaksanakan XP ke dalam rajah aliran proses yang sedia ada, ia akan membantu dalam membangunkan standard kelas ujian (STC). STC ini akan mengukur untuk kebolehgunaan semula bersama-sama dengan versi perisian pelepas untuk membuktikan kualiti perisian pengeluaran label bertambah baik. Secara ringkasnya, kebolehgunaan semula kualiti perisian label boleh ditingkatkan dengan standardisasi kelas ujian melalui menggunakan teknik XP seperti yang dicadangkan dalam rajah aliran proses.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	xi
	LIST OF FIGURES	xiii
	LIST OF ABBREVIATIONS	xv
	LIST OF APPENDIXES	xvi
1	INTRODUCTION	
	1.1 General Overview	1
	1.2 Background of Problem	4
	1.3 Problem Statement	9
	1.4 Aim and Objectives	13
	1.5 Scope	14
	1.6 Research Organization	15
2	LITERATURE REVIEW	
	2.1 Introduction	17
	2.2 Overview of Label Production Software	18
	2.3 Production Software Quality	19
	2.3.1 Software “Reusability”	22

2.4	Test Classes Standardization with Refactoring	23
2.5	Related work with other industry	25
2.5.1	IBM	26
2.5.2	INFOSYS	27
2.5.3	INTEL SHANNON	29
2.5.4	Comparison on Related Work with Other Industry	30
2.6	Discussion on Agile Technique	33
2.6.1	Comparison between Agile Technique SCRUM and XP	37
2.6.2	Related study on Agile Technique with Software Quality	39
2.7	Summary	45
3	RESEARCH METHODOLOGY	
3.1	Introduction	46
3.2	Research Process Flow Chart	46
3.2.1	Identify the problem occur with Label production software	48
3.2.2	Studying existing process flow diagram	50
3.2.3	Investigate and Identify suitable technique to standardize Test Class and improve label production software quality	56
3.2.4	Propose and enhance process flow diagram with suitable Technique (Agile Technique)	57
3.2.5	Testing the release label production software branch “Reusability”	58
3.2.6	Evaluate the label production by software quality factor “Reusability”	60
3.2.7	Result and Analysis	64
3.2.8	Research report and documentation	65
3.3	Case Study	65
3.4	Summary	67

4	PROCESS FLOW DIAGRAM IMPLEMENTING XP	
	4.1 Introduction	68
	4.2 Enhanced Process Flow Diagram with XP	68
	4.2.1 Proposed Process Flow Steps Implementing With “XP” Agile Technique	72
	4.3 Discussion on Difference between Existing and Enhance Process Flow Handling TSR Request	80
	4.4 Summary	92
5	STANDARDIZE TEST CLASS GENERATED BY XP IMPLEMENTATION	
	5.1 Introduction	93
	5.2 Standardize Test Classes through Process Flow Diagram	93
	5.2.1 “XP” Practices Used In Order To Standardize Test Classes	94
	5.2.2 Three Main Steps Produce Standardize Test Class Implemented with “XP” Practices	96
	5.3 Discussion on Process Flow Handling the Standardize Test Class	102
	5.4 Case Study on Standardize Test Class	106
	5.5 Overall Discussion on Standardize Test Class (STC) Achievement	115
	5.6 Summary	116
6	EVALUATION AND TESTING RESULTS	
	6.1 Introduction	117
	6.2 Reusability Measurement	117
	6.2.1 Test_SmartEEPROM_WriteChecksum	118
	6.2.2 Test_SmartEEPROM_UpdateTestData	125
	6.2.3 Test_SmartEEPROM_WriteCustSN	133
	6.3 Comparison Result	140
	6.4 Result and Analysis	142
	6.5 Summary	143

7	CONCLUSION AND FUTURE WORK	
	7.1 Summary	144
	7.2 Research Contribution	147
	7.3 Future Work	148
	REFERENCE	149
	APPENDIX A	152
	APPENDIX B	153
	APPENDIX C	157
	APPENDIX D	159

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	Comparison of the Quality Models according to Gregor study	21
2.2	Overall summary of issue occurred, technique used and result achieved for respective companies	31
2.3	Overall comparison on how the techniques are tested and what is being measured for respective companies	32
2.4	Comparison of techniques with general features in agile methodologies	36
2.5	Comparison between SCRUM and XP discussed against the Key Elements	37
2.6	Quality Attributes and their measure	43
2.7	Quality before and after applying CI technique	44
3.1	The software quality factors with agile technique	58
3.2	The software quality factors “Reusability” and descriptions	61
3.3	Input and parameter measured to calculate the percentage of reusability	62
3.4	Reusability calculated metrics and definition	63
3.5	The quality attribute that will be compared for before and after the process flow diagram enhanced	64
3.6	Table contains the case study and brief description	66
4.1	Difference between each steps handle in original and XP release process Flow Diagram	89
5.1	Difference way used by developers handling each steps before and after the XP practices implemented	104

5.2	Difference between the ways in the Original and XP Practices process flow diagram handling Test Classes Types according to TSR requirement	105
6.1	Release Software with more than once similar standardize test class tested on each five product	121
6.2	V1, V2, V3 and V4 Release software quality measurement through reusability of the release version with nine different products	122
6.3	V1, V2, V3 and V4 Release software quality measurement through reusability of the release version with nine different products units tested	123
6.4	Release Software with more than once similar standardize test class tested on each two product	129
6.5	V1, V2 and V3 Release software quality measurement through reusability of the release version with seven different products	130
6.6	V1, V2 and V3 Release software quality measurement through reusability of the release version with seven different products units tested	131
6.7	Release Software with one similar standardize test class tested on each three product	135
6.8	V1, V2 and V3 Release software quality measurement through reusability of the release version with eight different products	136
6.9	V1, V2 and V3 Release software quality measurement through reusability of the release version with eight different products units tested	137
6.10	Shows the Comparison and Analysis result for Standardize test classes developed through Original and XP release Process flow	140
6.11	Analysis result by comparing original and XP according to Analysis Attributes	142

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	The process flow diagram being used in Production software release	12
2.1	The agile development life cycle	34
2.2	XP architecture and structure	38
2.3	SCRUM architecture and structure	38
2.4	Simple agile methods life cycle	40
2.5	Agile techniques and Quality assurance	40
3.1	Research Framework process flow	47
4.1	Enhance blocked diagram with agile techniques	71
4.2	Five main steps with details explanation with XP techniques implementation	79
4.3	Standardize Test Class (STC) Handle with Original Process Flow Diagram	91
4.4	Standardize Test Class (STC) Handle with Proposed XP Process Flow Diagram	91
5.1	Sample of Specification file setting for “Test_SmartEEPROM_UpdateTestData”	109
5.2	Passing qualification result for standardizes test class Case 1	110
5.3	Sample of Specification file setting for “Test_SmartEEPROM_UpdateTestData”	113
5.4	Passing qualification result for standardizes test class Case 2	114
6.1	Testing result of test class “Test_SmartEEPROM_WriteChecksum”	119

6.2	Testing result of test class “Test_SmartEEPROM_WriteChecksum” reused more than once	120
6.3	Reusability Measurement chart for software version with standardize test class “Test_SmartEEPROM_WriteChecksum”	124
6.4	Testing result of test class “Test_SmartEEPROM_UpdateTestData” using one standardize test class	126
6.5	Testing result of test class “Test_SmartEEPROM_UpdateTestData” reused more than once	127
6.6	Reusability Measurement chart for software version with standardize test class “Test_SmartEEPROM_UpdateTestData”	132
6.7	Testing result of test class “Test_SmartEEPROM_WriteCustSN”	134
6.8	Reusability Measurement chart for software version with standardize test class “Test_SmartEEPROM__WriteCustSN”	138
6.9	Reusability Measurement chart for standardize test class	141

LIST OF ABBREVIATIONS

ASD	-	Adaptive Software Development
CI	-	Continuous Integration
CMM	-	Capability Maturity Model
DSDM	-	Dynamic Systems Development Method
FDD	-	Feature-Driven Development
PQM	-	Project Quality Metrics
RAD	-	Rapid Application Development
RL	-	Reuse Level of release software version
RP	-	Reuse Level for product tested on each release software version
RUP	-	Rational Unified Process
STC	-	Standardize Test Class
SVC	-	Software Version Control
TSR	-	Test Software Request
UAT	-	User Acceptance Testing
UP	-	Unified Process
XP	-	Extreme Programming

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Gantt Chart	161
B	Software Reusability Calculation Metrics	162
C	Table Data For Test Class Tested Once On Specification File	166
D	Code Standards Document Attached In Performe & New Process Flow Attached In Performe	168

CHAPTER 1

INTRODUCTION

1.1 Overview

Over the years, automation software has become an essential input for the operations across all industries and sectors. This software has been used in testing and verifies the products being produced by the manufacturing industries. In order to make sure the process flow and quality of the software is efficient and productive, it is important to improve and maintain the quality of the software and this can be done through “Software Quality”. According to IEEE Standard (version 610.12-1990) defines software quality as "The degree to which a system, component, or process meets specified requirements” and “The degree to which a system, component, or process meets customer or user needs or expectations”.

On the other hand, according to International Software Testing Qualifications Boards (ISTQB) software quality is defined as "The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs". This software quality has its own requirements and evaluation for products which contains the functional suitability, reliability, performance efficiency, operability, security, compatibility, maintainability, reusability and transferability. This quality requirement will be very helpful in improving the reusability of test classes in “Production Software” to test the transceiver modules which is being produces and shipped by Finisar.

Production software or known as “PARTEST” is an automation software used to test the transceiver produced in Finisar. Transceiver or known as transmitter or receiver is a device which combines transmission and receiving capability on shared circuitry. This automation software “PARTEST” is an internal “in house” software created and handled by developers team in Finisar and released to production for testing various types of fiber optic transceiver products. This software is not only used in Finisar “Ipoh” but also used in other Finisar site such as “Wuxi”, ”Shanghai”, and “Sunnyvale”.

The same automation software contains test classes which is used to run various type of test which start from Initial Test, Final Test, ER Temp Comp, Label Test, Barcode Test and many more. This test being run is not a small test it is quite big and each test contains many sub test which need to be fulfilled before proceed to next step. All the test being run is also belong to different developing area and different production software branch. One of the areas which will be covered in this study is Labeling area which used “Label production software”.

“Label Production Software” has its own release branch to run the entire test in production floor of labeling area. The release branch is “18.000.xx” and this software version has been used in many sites of Finisar such as “Shanghai”, “Wuxi”, “Ipoh” and “Sunnyvale”. It plays an important role in supporting various products testing and customer requirements. It is very important to make sure the release of label production software test classes is bug and defect free, delivered on time without any delay to avoid any aging and shipment problem, meets requirements and expectations of user and customer, and is maintainable. This is because if the new release label production software test classes have bugs and failures for all the products then it will affect the yields and performance of the products.

Most of the problem and issue occur in this new release label production software is caused by the failure in the test classes and compatibility error with new and existing products being tested. “Test class” contains source code of methods and function which is required by the customer and product engineer for testing their products. This test classes is written in

VB.NET and is used in each sub test of Main test. This test classes is runs and called according to how it set in the specification file.

All of this test classes is handled by different developers groups of test development team. Because of each developer has its own style of coding , different product has different type of requirement, different customer has different requirement and currently the changing format of Finisar and Customer Serial Number requirement causes this test classes to be unstable and prone to failure. This defect will cause the quality of the label production software to be reduced. So, it is important to improve the quality of the new release label production software with “Test Class Standardization”.

Standardization here means coordinate, and organize the functions in the source code of the test class as required by the users into one standard format where it can be used by various product types. This is to help to avoid duplication functions exist in more than one test classes and hard coded data exist in the source code of test classes. The test class standardization can be done by creating generic test classes and refactoring the test class source code with generic function which can be used by various customer and product type for testing. Through refactoring the test class source code, the software quality can be achieved by measuring reusability of the test classes and software being released to production.

By standardizing the test classes no matter it is existing test classes or new test classes, it can help in making the test classes and software to be more feasibility and manageable to support various customer and product type with different and same requirement with generic test classes and flexible software version of label branch. This will help user to just specify the same generic test classes in specification file for the entire product which has the same calculation or functions to be used.

1.2 Background of Problem

In Finisar Company, automation software has been running for around 15 years. “Automation Software” it is a general technology terms used to describe any process, test or flow being automated through the use of computers and software. This software is being run with use of equipment’s such as machines, various types of boards, power supply and many more. It is quite big software and has been used to support various testing processes, productivity and quality of all the products which is being produced in company.

Automation software is being maintained by all the developers from various sites of Finisar. Previously it used VB6 programming language and now it is migrated to VB.NET programming language. This software is called as “PARTEST” and it is being actively run in production floor where each process contain its own version and branch to support each of the process flow. Mainly, all the process flows play an important role in producing the products and each of this process has its own functions and task in testing the products.

“Label Process” is one of the important processes and without this the product delivery is uncompleted. This process supports all the labelling process and test include Label test, Custom Label Test, Label 2 Test and Barcode Test. Each of these tests has its own requirement and function which is supported and completed through “Label Production Software” with branch “18.000.xx”. This software supports all the testing of the requirements and updating the data of the product as requested by customer before being delivered.

Moreover, the testing is done through test classes specify in specification file for each product. Each test classes have different requirement and functions. So, to support each of this test classes whenever there is any issue, error or bug occurs it will be fixed in the label software branch and provided with higher running version once qualified working. It means that this label production software will be keep on upgraded to higher version to support the entire

existing product type, new product type and customer special request as stated in the “Minipip” document. This requirement from customer will be depend on their product type and customer type. Different customer will have different requirement and sometimes they will be sharing same requirement.

So, whenever there is a new request from customer a new test class will be created to support the requirement and this new test class creation lead to upgrading of software version. There is exists test classes with same requirement but cannot be used because of the hard coded data in the test classes which is not supported by new request. This brings to different customer with same requirement will be using different version of software because the test class that created only support for one customer and product type.

Sometimes it is quite hard to maintain various version of software which is being run in production and it can lead to use wrong version of software for the required customer and product. Because of this situation, production folks will always complain and prefer to have standardized of software version. They prefer to install and maintain one ECO software version that is flexible enough to support the product on the tester.

Moreover, there is no any proper documentation for the developer to follow when they are dealing with the code and this causes the code of the test classes created to be messy, not user friendly and not standardize. In order to make the code to be understandable and improve the software quality, a proper documentation and flow is required to be followed. This flow will be very helpful in test class standardization and refactoring the code to support the software reusability.

Besides Finisar, there are many other companies and industries have improved the quality of the software being used in producing and testing their products. “IBM” is one of the companies that has implement quality management across the products and application life

cycle, minimize code defects to improve the software quality and reduced the development cost (Mishiev and Omer, 2009). At first it was quite hard for IBM to maintain the software quality because development cost continue to rise, and the growing needs to get products delivered to market with accelerated development schedules. So, by catching defects as early as possible in the development cycle using static and dynamic analysis to improve the overall code quality, and adopt quality strategies can help in minimize risk in software development projects and significantly reduce the development costs.

IBM has start managing quality at the beginning of the projects rather than trying to build it in at the end and this way has improved operational predictability, resulting in lower costs and improved quality without time penalties. By adopting such a life cycle approach, quality improvements can help deliver a quantifiable return on investment and satisfying business objectives. Quality strategies implementation can help in maintaining the functionality, reliability, and performance of products and applications in development and production. IBM has also used agile methodologies for process and it helps to smooth out the complexities of software development and this development technology help deliver ship ready code.

Other than that, as the pace of business is getting faster as game changers “Infosys” company is looking forward to deploy new features rapidly, resulting in frequent application releases as opposed to the earlier one time release scenario (Rao, 2017). Consequently, the time taken for a feature to be implemented from ideation to production deployment is shrinking fast. This has resulted more frequent application releases and the change has drastically impacted the dynamics of the development life cycle in “Infosys”. Previously the company uses traditional approach where a development team deploys a feature and then passes it on to a separate operation team causes the delays involved in acknowledging, testing and deploying the application in traditional manner increase the time to market of the feature and quality issues on the application being released. This has been solved by implementing “DevOps” methodology which is a blend of development and operations has helped to provide an understanding that helps design applications for rapid delivery.

The core tenet of the “DevOps” practice is Continuous Delivery (CD) and Continuous Integration (CI) under the agile technique has help in automated deployment and verification of an application across a set of environments. This automation not only will reduce the manual errors but also allows quick, reliable and repeatable deployment of rapidly developed code. In case of legacy systems, which have evolved a period of time without any consideration of automation, the adoption of the “DevOps” approach has helped in large scale refactoring or redesigning. For Infosys, by adopting the “DevOps” methodology with agile technique (Continuous integration) in a legacy system project has involves three key aspects such as standardization, automation and shifting left. All these three keys has helped in improve the quality, performance and reusability of the system, reduce the risk and help in reducing the cost of fixing the bugs as early as possible.

Moreover, there is one more company which has face some problem with their software development is “Intel Shannon” (Fitzgerald et al., 2006). There was many problem occur with the software being released and produced by this company such as highest defect density, less accuracy to development schedule, low quality and many more. Previously, this company has been assessed at Level 2 on the Capability Maturity Model (CMM) and this has led to some discipline in the development process and the rapid time to market pressure. This pressure and issues has led this company to consider agile methods. “Intel Shannon” has been deploying a range of agile methods such as XP for technical engineering aspects of software development and SCRUM for the project planning and tracking.

In “Intel Shannon” the XP practices used are pair programming, testing, metaphor, collective ownership, refactoring, coding standards and simple design. While the SCRUM practices used are scrum planning, scrum sprints, and architecture post game closure sessions. The development statistics of this company has reported that both of this agile methodology can deliver the quality software within schedule. Indeed the country manager for Intel in Ireland has also identified Intel Shannon’s delivery of extremely high quality software within schedule as the “key competitive edge” for Intel Shannon. Both of these techniques are release parallel together with CMM Level 2 and this has not cause any perturbation in relation to the CMM certification level. From this it is known that agile methods used in “Intel Shannon” have been

quite compatible and helpful in improving the quality of the software being produces and help in zero defect.

Furthermore not only Finisar but all the companies and industry have gone through many difficulties in handling the quality of the software being produced by the respective company. There are many unwanted bugs and issues occur in software being produces and this can cause problem in business. To overcome all this it is very important to choose the suitable technique which can help in improve the quality of software. All the three industry “IBM”, “Infosys”, and “Intel Shannon” has use “Agile Technique” in solving the entire software problem that being faced and improve the software quality.

“Agile technique” is a software development methodology with a set of lightweight methods and includes XP, Scrum, feature-driven development, and more. This methodology is used for modelling and documenting software system based on best practices. Agile methods use various tactics and try to overcome the limitations of the dynamic nature of systems development projects. This method addresses the inherent problems of traditional systems development using two scientific concepts such as “empirical process control” and “soft-systems thinking” (Javanmard and Alian, 2015).

According to Hossain, Kashem and Sultana (2013) an agile technique aims to develop and implement software quickly in close cooperation with the customer in an adaptive way and enhance software quality. Agile technique is being used when there is a new changes needed to be implemented. The freedom agile gives to change is very important and new changes can be implemented at very little cost because of the frequency of new increments that are produced. These are the reason why agile technique is flexible enough and very useful and helpful for industry. Agile software development methods were developed to provide more customer satisfaction, to shorten the development life cycle, to reduce bug rates, and to accommodate changing business requirement during the development process.

Other than “Agile Technique”, there is one more techniques can be used to develop and modify the test classes to improve the software quality such as “Traditional Methodology”. A traditional software development methodology is classified as heavyweight methodologies and contains software methodologies such as waterfall, Rational Unified Process (RUP), Rapid Application Development (RAD), Spiral Model and others (Leau et al., 2012). These methodologies are based on a sequential series of steps like requirements definition, solution building, testing and deployment. Traditional software development methodologies require defining and documenting a stable set of requirements at the beginning of a project (Leau et al., 2012). The traditional software development methods are dependent on a set of predetermined processes and on-going documentation which is written as the work progresses and guides further development.

In Summary, for Finisar the quality of the label production software can be improved with standardizing the test class and to do it is important to choose a useful technique which will be very helpful. The technique that is chosen to standardize the test class in improving the label production is “Agile Technique”. It is a very useful methodology to be adopted in the modern software development process to replace the traditional heavyweight development life cycle.

1.3 Problem Statement

The current issue that faced with label production software is related to the test classes being coded is unstable and causes the quality of the software to be affected. This developed test classes is unable to be reused to support the testing of various product types where it means that one test class can be only used for one specific products and not able to be reused on more than one product types. It is very hard to maintain the code of test classes and quality of label production software which is developed to be supportive for all the products that are running with the label production software version. Current release Label production software branch 18.000.xx has many unpredicted error occur when being tested with existing and new products.

The error occurs is related and caused by the test class being used. From what known this test classes are being handled by different developers and each developer has its own style of coding and creating the test classes which causes the test classes to be inflexible and prone to failures. There is no any standard document or proper process flow with suitable technique to be followed and referred during development of test classes. The problem occurs in Finisar with label production software that needs to be solved are all related to test classes.

One of the problem is there exist hardcoded data in the code of existing test classes and this causes whenever there is a new request from user this require to create a new test class because existing test classes not able to support the requirement and not able to be reused on another product with similar requirement. This problem has cause for duplication function with same requirement to be coded in more than one test classes. Besides that, there is number of software version release in Label Production to support various product type and requirement. This is happen because of new requirement from customer required Partest update, if existing test class is not able to support the testing.

So, every time when update the test classes or create new test classes require new software version upgrade and new version to be build and release to production. This has also causes difficulties to standardize one software version to be used in labelling production. Moreover, there is different type of bugs and error occurred for different and existing running products when there is new software updated and release to production for new product support. This entire problem is occur because of the test classes being used and as it is known that this test classes are developed by developers and some of the developer are not aware about the process flow and there is no proper documentation or techniques to be referred and used by developers during test class development.

Overall, this entire problem occur has affected the software quality such as reusability of the software. Not only quality of the software is affected but it also have affected the product and production line performance by affecting the yield performance of the product being tested, aging issue in production area for the failing products cause by bugs occur in the software used

for testing and this will bring to missed shipment. It is very important to investigate and find suitable techniques which can be used to improve the test classes and software quality which is affected by the bugs and failure occurs in test classes being used. This technique can be used and implemented in the “Process flow” which will be used by the test and development team in Finisar in test class development.

“Process flow” contains the flow and steps being taken before the release of new software version to production. It contains the steps which start before developing or modifying the code until it release to the production flow to be used by products tested in production area. This is a just a general flow and it does not contain any proper document or technique which can be referred and used by developers. From the flow of the process flow diagram, it is known that during the development and modifies test class there is no any standard documentation, code pattern and technique to be followed and used. All the developer will use their own coding styles to develop and modify the test classes. This will cause high chances for bugs and software failure to occur because of the unstable test class being used to test the products.

The process flow diagram being used to update the production software and release the new version to production is as shown in Figure 1.1 below.

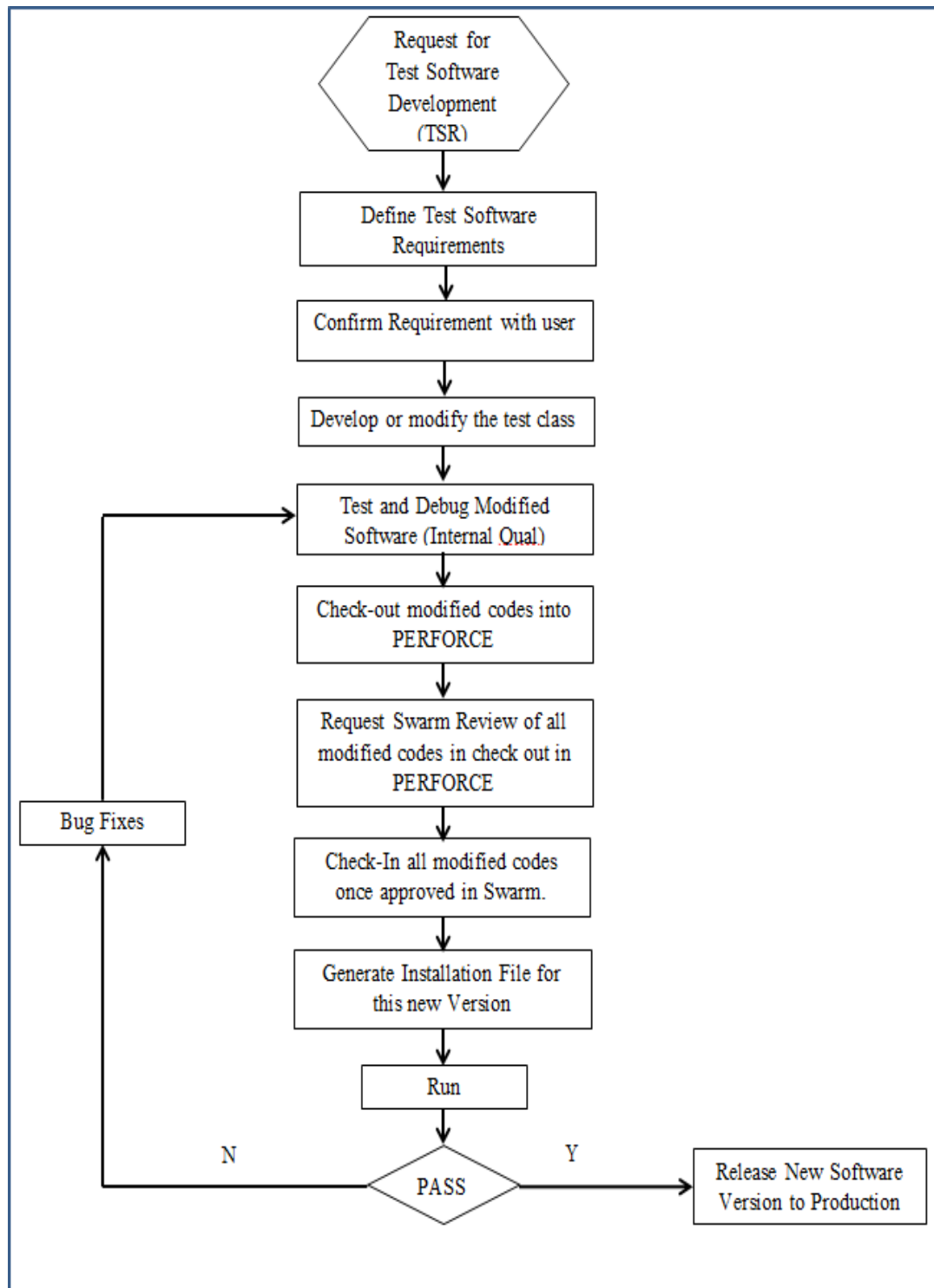


Figure 1.1: The process flow diagram being used in Production software release

Therefore, the research questions that need to be solved in this study are as listed below:

1. What are the suitable agile techniques that can be used to enhance the process flow in order to improve the software quality and process flow of label branch?
2. What are the suitable agile techniques that can be used in standardizing test class to improve the software quality of label branch?
3. Is the tested standardize test classes is able to support various type of products?
4. Is the standardize test class is reusable enough to be used by any product type and able to improve the software quality factor reusability?

1.4 Aim and Objectives

The aim of this research is to propose an appropriate technique and “process flow” for standardizing the test classes in order to improve the quality and performance of software. There exist many techniques of Agile such as Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Dynamic System Development Method (DSDM) and many more to be used but it is important to a chose a technique which can be very helpful (Javanmard and Alian, 2015). So, the main concern is on choosing a suitable technique which can be used to support the test class standardization and improve the label production software quality.

Furthermore, this research consists of set of objectives that need to be achieved and lead to the research process. The main objectives are:

1. To investigate and identify the suitable agile technique to be used in standardizing test class and process flow of label branch to improve the software quality.
2. To propose and enhance the process flow with suitable agile technique to help in easier maintenance of the code, lower risk of errors and improve software quality.
3. To test the new released label production software branch with standardize test classes and evaluate the software quality factor reusability to make sure it is reusable enough to be used by any product type.

1.5 Scope

In this research, the boundary of the research is defined. The following are significant.

- i. This research focused on refactoring and standardization test class on label production software.
- ii. This research focused on the test classes which are covered in the branch “18.000.xx” and created using Microsoft Visual Studio 2015 (VB.NET 2015).

- iii. The research focused on proposed technique to be implemented in the existing process flow to improve the software quality.

- iv. The research focused on software reusability to support various types of product in label production process in Finisar.

1.6 Research Organization

This research made up of seven chapters. The thesis outline is as follows:

Chapter 1, it discuss on the overview of label production software quality with test class standardization, background of problem, problem statement, aim and objectives, and scopes.

Chapter 2 is a literature review of current techniques can be used in standardizing the test classes for to improve label production software quality. Various approaches and techniques compared to determine weaknesses and strength among them. Also, related work and previous attempt to improve reusability and loosely coupled especially in label production system has been clarified.

Chapter 3 describes the methodology of this research which include research process, explained in details on the sequences of process, research framework, details explanation on the original process flow diagram and case study which will be referred during testing and evaluation.

Chapter 4 will discuss in details the enhanced process flow diagram with XP. Each steps involved in the process flow diagram will be discussed in more details with the XP technique to be implemented. There will be also included discussion on the difference between original and new proposed XP release process flow diagram in handling TSR request and developing standardize test classes.

Chapter 5 will include the discussion on the standardize test class developed from the enhanced process flow diagram with XP agile technique. It will also include the discussion on the difference between original flow against XP release flow in handling each TSR request to developed standardize test classes. It will also include some samples of case study in order to show how this standardize test class created from the new process flow and able to be reusable to support various product testing.

Chapter 6 is the evaluation and final testing results. It will include the discussion on the reusability measurement use on each standardize test classes and release software version. to prove the software quality is able to be improved. There will be also discussion on the comparison and analysis result done in order to prove that the attributes used in analysis is able to be improved and compared against original and new proposed XP release standardize test classes.

Chapter 7 provides the conclusion which include the details discussion on the overall summary of this research and its contribution to knowledge in addition to future work for further study.

REFERENCES

- Anderson Jr, J. L. from NSWC Corona Division, (2004), “*Using Software Tools And Metrics To Produce, Better Quality Test Software*” in Autotestcon 2004. Proceedings Conference on 20-23 Sept, Pages 293-297
- Ali, A., Jawawi, D. N. A., Isa, M.A., and Babar, M.I., (2016), “*Technique for Early Reliability Prediction of Software Components Using Behaviour Models*”, Software Reliability Prediction, PLOS One, pages 1-24
- Almseidin, M., Alrfou, K., Alnidami, N., and Tarawneh, A., (2015), “*A Comparative Study of Agile Methods: XP versus SCRUM*”, International Journal of Computer Science and Software Engineering (IJCSSE), Volume 4, Issue 5, ISSN 2409-4285, pages 126-129
- Al-Tarawneha, M.Y, Abdullah, M.S, and Alic, A.B.M, (2010), "A Proposed Methodology for Establishing Software Process Development Improvement for Small Software Development Firms", Applied Science Division, College of Arts and Sciences, University Utara Malaysia, Published by Elsevier Ltd., Procedia Computer Science 3 (2011) 893–897
- Baxter, S.M., Day, S.W., Fetrow, J.S., and Reisinger, S. J., (2006), “*Scientific Software Development Is Not an Oxymoron*”, PLOS Computational Biology, Volume 2, Issue 9, e87, Pages 0975-0978
- Brian, F., Gerald, H. and Kieran C., (2006), “*Customizing agile methods to software practices at Intel Shannon*”, Operational Research Society, European Journal of Information Systems, Pages 200-213
- Chavoya, A., Lopez-Martin, C., Andalon-Garcia, I. R., and MedaCampana, M. E., (2012), “*Genetic Programming as Alternative for Predicting Development Effort of Individual Software Projects*”, Genetic Programming Estimates Development Effort, PLOS One, Volume 7, Issue 11, e50531, pages 1-10
- Dantas, F. and Garcia, A., (2010), “*Software Reuse versus Stability: Evaluating Advanced Programming Techniques*”, Brazilian Symposium on Software Engineering, IEEE Computer Society, Pages 40 -49

- Galster, M., Weyns, D., Tofan, D., Michalik, B., And Avgeriou, P., (2014), “*Variability In Software Systems - A Systematic Literature Review*” In IEEE Transaction O Software Engineering, Vol.40, No. 3, Pages 282-306
- Galina, M. and Adeel, O., (2009), “*The top three secret to successful agile software development*”, Agile software development White Paper, January, Copyright IBM Corporation, Pages 1-12
- Garousi, V. and Feldere, M., (2016), “*Developing, Verifying, and Maintaining High-Quality Automated Test Scripts*” in IEEE Software, May/June, Published by the IEEE computer Society, Pages 68 -75
- Gregor Panovski, Dipl.ing. , (2008), “*Product Software Quality*”, Master’s Thesis, Eindhoven University of Technology in Netherlands, Department of Mathematics and Computing Science, pages 1-103
- Hamdan, S., and Alramouni, S., (2015), "A quality framework for software continuous integration", 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015, Elsevier, Procedia Manufacturing 3 (2015), pages 2019 – 2025
- Hanley, K. (2010), “97 Things Every Programmer Should Know”, Collective Wisdom from the Experts, Copyright 2010, Printed in the United States of America, Published by O’Reilly Media, Inc. 1005 Graven stein Highway North, Sebastopol CA 95472, Pages 1-229
- He, P., Li, B., Ma, Y., and He, L., (2013), “*Using Software Dependency to Bug Prediction*”, Hindawi Publishing Corporation, Mathematical Problems in Engineering, Volume 2013, Article ID 869356, pages 1-12
- Highland, F. and Kornman, B., (1993), “*The use of cleanroom methodology for knowledge based application development*”, Artificial Intelligence for Applications, 1993. Proceedings. Ninth Conference on 1-5 March 1993, pages 361-367
- Hossain, A., Kashem, Dr. Md. A., Sultana, S., (2013), "*Enhancing Software Quality Using Agile Techniques*", IOSR Journal of Computer Engineering (IOSR-JCE) Volume 10, Issue 2 (Mar. - Apr. 2013), Pages 87-93
- Huang, C.Y., Leung, H., Leung, W.H.F, and Mizuno, O., (2012), “*Software Quality Assurance Methodologies and Techniques*”, Hindawi Publishing Corporation, Advances in Software Engineering, Volume 2012, Article ID 872619, pages 1-2
- Kari, A.B., Poonam, c., Valerie, H., Allan, P., Brian, S. Jim, V., and Bruce, V., (2008), “*Minimizing code defects to improve software quality and lower development costs*”,

- Development Solutions White Paper, October, Copyright IBM Corporation, Pages 1-12
- Leau, Y.B., Loo, W.K., Tham, W.Y., and Tan, S.F, (2012), “*Software Development Life Cycle AGILE vs. Traditional Approaches*”, International Conference on Information and Network Technology, October, School of Engineering and Information Technology University Malaysia Sabah, Malaysia, Vol.37, Pages 162-167
- Nguyen, T.T., (2013), “*Improving Software quality with Programming Patterns*”, *Graduate Theses and Dissertations*, Paper 13576, Doctor Of Philosophy, Digital Repository, Iowa State University, Ames, Iowa, Pages 1 -137
- Nithin, A., and Ruchira, A., (2014), “*DevOps for legacy systems - The demand of the changing applications landscape*”, Computer Science and Information Technology (CS&IT), USICT, Dwarka, Delhi, India, Pages 37 -46
- Noble, K., and Anju, S., (2013), “*Effect of Refactoring on Software Quality*”, Paper 13576, Doctor Of Philosophy, Digital Repository, Iowa State University, Ames, Iowa, Pages 1 -137
- Ray, M., and Mohapatra, D.P., (), “*Prioritizing Program Elements: A Pretesting Effort to Improve Software Quality*”, ISRN Software Engineering, International Scholarly Research Network Volume 2012, Article ID 598150, 20 pages
- Richard A. Baker, Jr. from Schneider Automation. Inc, (1997), “*Code Reviews Enhance Software Quality*”, Proceedings of the 1997 International Conference, 17-23 May, Pages 570-571
- Sejdinovic, A., and Atachi, F.,(2006), “*Methods for software quality improvement*”, A study performed upon the C10v2 system for Metrima AB, May, Department of Computer Science and Electronics, IDE, Malardalen University, Vasteras, Pages 1-81
- Sukhpal, S., and Inderveer, C., (2012), “*Enabling Reusability in Agile Software Development*”, International Journal of Computer Applications, July, Thapar University, Patiala, Volume 50, Pages 33-40
- Sarbjeet, S., Sukhvinder, S., and Gurpreet, S., (2010), “*Reusability of the Software*”, International Journal of Computer Applications, October, Sri Sai College of Engineering and Technology, Pathankot, India, Volume 7, Pages 38-41
- Zulkefli, H.S., (2004), “*Engineering Maintenance System (EMESYS) Maintenance Module Using .Net Platform*”, Master Degree Award, August, Centre for Advanced Software Engineering, Universiti Teknologi Malaysia, Pages 1-106