FUNCTIONAL TESTING OF CoRNEA BUSINESS APPLICATIONS SYSTEM (CoRNEA BIZ-APPS)

NUR HAKIMAH BT HUSIN

UNIVERSITI TEKNOLOGI MALAYSIA

FUNCTIONAL TESTING OF CoRNEA BUSINESS APPLICATIONS SYSTEM (CoRNEA BIZ-APPS)

NUR HAKIMAH BT HUSIN

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Computer Science (Real Time Software Engineering)

Centre for Advanced Software Engineering
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia

NOVEMBER 2006

ACKNOWLEDGEMENT

Praise to Allah SWT who has bestowed me the ability to complete this work. My sincere appreciations to my supervisors, lecturers, colleagues, family and Custommedia Sdn. Bhd. for all the helps and supports towards accomplishment of this paper.

ABSTRACT

Recent software system contains lot of functions to provide various services. According to this tendency, it is difficult to ensure software quality and to eliminate crucial faults in the system developed. The purpose of testing in any software development process is to ensure the system is released with defect free and meeting the purpose of the system requirement specifications. Thus, Testing is a critical component of a mature software development process. It is one of the most challenging and costly process activities, and in its fullest definition it provides strong support for the development of high quality software. Software testing process has gone through evolution that can be traced back by examining changes in the testing process model and the level of professionalism over the years. Various testing process models, methodologies and techniques have been invented to ensure software testing achieve its objectives. Testing of CoRNEA Business Application (CoRNEA Biz-Apps) system has been implemented based on Framework One software development process. With the adoption of suitable test process and incorporating testing best practices into implementation of testing activities would result with a high confidence in releasing the system tested into implementation stage of software development lifecycle.

ABSTRAK

Sistem perisian kini mempunyai banyak fungsi yang menyediakan pelabagai servis mengikut kecenderungan ini, adalah susah untuk memastikan kualiti perisian dan menghapuskan kesalahan penting dalam sistem yang dibangunkan. Tujuan pengujian dalam proses pembangunan perisian adalah untuk memastikan setiap sistem dikeluarkan dengan tiada kecacatan dan mencapai tujuan sistem sebagaimana dispesifikasikan. Oleh itu, pengujian adalah satu komponen penting dalam sistem pembangunan perisian matang.Ianya adalah salah satu aktiviti proses yang paling mencabar dan melibatkan kos yang mahal, dan dalam definisi penuh ia menyediakan sokongan yang kuat dalam membangunkan perisian yang berkualiti tinggi. Proses pengujian perisian telah melalui proses evolusi yang mana dapat dilihat dengan meneliti perubahan dalam model proses pengujian dan tahap professional. Pelbagai model proses, cara-cara dan teknik pengujian telah dicipta dalam memastikan pengujian perisian mencapai objektifnya. Pengujian ke atas sistem CoRNEA Business Application (CoRNEA Biz-Apps) telah dilaksanakan berdasarkan proses pembanunan Framework One. Dengan mengguna pakai proses pengujian yang sesuai dan mengadaptasi praktis terbaik pengujian dalam melaksanakan aktiviti pengujian menyebabkan keyakinan yang tinggi bagi memindahkan sistem yang diuji ke tahap pelaksanaan dalam kitaran pembangunan perisian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

CASE	– Center for Advanced Software Engineering
CoRNEA Biz-Apps	– CoRNEA Business Applications
CUSTOMMEDIA	– Custommedia Sdn Bhd
F1	– Framework One
MSC	– Multimedia Super Corridor
OO	– Object-Oriented
SAD	– Software Architecture Document
SDLC	– Software Development Lifecycle
PTC	– Public Task Center
ST	– System Test
IT	– Information Technology
URS	– User Requirement Specification
UT	– Unit Testing
UIT	– Unit Integration Test

CHAPTER 1

INTRODUCTION

This Chapter is divided into two subsections which is Company Background and Project Background. Company Background contains information on company where the project was done. Project background describe on the type and nature of the project that was carried out during the assignment.

1.1 Company Background

Custommedia Sdn. Bhd. (CUSTOMMEDIA) is an established wholly Malaysian owned solution company. The company specialized in software engineering solutions for its wide-ranged clientele. Incorporated in 1990, our core objective is to develop software solutions that address specific and unique requirements of clients in all sectors of industries, government agencies and educational institutions.

The organization is an ISO 9001 certified MSC (Multimedia Super Corridor) status company, which continuously looking forward to expand their business and cooperation using local and international expertise. Since incorporated, CUSTOMMEDIA has been moving forward rapidly and accumulated a vast and complete range of experience, expertise and skills in software development and engineering spanning across various platforms, tools, and methodologies. These expertise and skills are deployed in solution engineering, product development, system supports and system maintenance or to provide professional services.

They believes that their comprehensive quality driven process delivers software solutions that are built-to-fit. Their process determines the success and quality of their software products and services; technology is only the enabler. Any technology can be adopted towards a solution.

1.2 Background of Project

The project to be accomplished during internship rendered around software testing of CoRNEA Business Application (CoRNEA Biz-Apps). The full testing lifecycle was performed on the CoRNEA Biz-Apps system to ensure only a quality product is released into implementation stage of software development lifecycle. The testing operation is carried over in iterative software development environment. The testing of CoRNEA Biz-Apps concerned on functionality of the system to be tested and also its conformance to standard deployed by the system.

CoRNEA Biz-Apps is a component of CoRNEA system or CoRNEA Framework. The broad goals of the CoRNEA Framework are to offer a comprehensive administrative automation with intranet and internet based capabilities to:

- reduce workload and saves time and resources, leaving more time to focus on improving standards in teaching and learning (i.e. focus on the learner).
- enhance leaner performance by quickly analyzing and detecting problems improving communications, and helping institutions make better decisions with accurate real-time data.
- allow institutions to communicate directly and securely with guardians through the institution’s education portal.

The objective of the CoRNEA Framework is to efficiently and effectively manage the resources and processes required to support the Administration, and the Teaching & Learning functions of the educational institution. The main focus of the CoRNEA Framework is to provide a single enterprise solution for the planning, implementing, monitoring and evaluation of programs and activities of the educational institution system.

The CoRNEA Framework is to support, manage the day-to-day and ad-hoc school teaching and learning processes. The processes includes generating class schedules, tracking assets and facilities, managing library items, recording student and staff attendance, managing hostel boarders, and managing fee collections and expenditures.

The CoRNEA Framework solution architecture comprises of two major components:

- CoRNEA Foundation
- CoRNEA Business Applications (CoRNEA Biz-Apps).

The CoRNEA Framework’s components are integrated and somewhat modular.

- The CoRNEA Foundation (including the CoRNEA Essential Application) are mandatory components of CoRNEA Framework
- The CoRNEA Biz-Apps are optional components offering additional functionalities to complete the CoRNEA Framework as a single enterprise school management solution.

The integration between the sub components will be achieved through component integration implementations. The applications will be integrated through one single integration interface so that data can be transferred from one application to the other in a transparent way. Access to any of the applications in CoRNEA Foundation and CoRNEA Biz-Apps is done through the Personal Task Center (PTC), which is verified by the Authentication Engine.

In addition to integration of sub component , the integration is also achieved through low-level component integration and database integration implementations. Information from all applications will be available online because components share the same database.

The main goal of CoRNEA Biz-Apps is to provide for the additional functionalities to complete the CoRNEA Framework as a single enterprise solution for school management. The additional functionalities enable the school to manage all aspects of school administrative work, including fee collection and expenditures management, facilities management, staff appraisal and performance, hostel residence management and library resource management. The CoRNEA Biz-Apps will be connected to the CoRNEA Framework through the CoRNEA Personal Task Centre.

1.3 Objective

The project started by identifying the objectives to be achieved during the project assignment. The objectives were identified within the scope of software engineering body of knowledge. The objectives of the project that have been recognized are as follows:

i. To design Unit Integration Test of CoRNEA Business Application
ii. To execute Unit Integration Test of CoRNEA Business Application
iii. To design system test for Cornea Business Application
iv. To execute system test of Cornea Business Application

1.4 Scope

Software testing covers a wide scope in software engineering that contains various level, methods and techniques of implementing it. Thus, the scope for the project are to be identified to serve as boundary for testing operation to be done in order to achieved the stated objectives. The identified objectives are to be achieved within following scopes:

i. Test execution of unit integration will be done based on the division of modules assigned by company.
ii. All test execution of unit integration test of application will be based on existing test procedure where any deviations from it will be updated accordingly.
iii. A new set of test procedures are to be developed for new system to be tested.
iv. Study on User Requirement Specifications (URS) and Software Architecture Design (SAD) of CoRNEA
v. Area of testing covers system functionality and meeting system requirements and standard.
vi. All activities and deliverables of testing process are based on testing stage of Framework One software process framework which is employed by company throughout CoRNEA development.
vii. Testing will be done in iterative software development environment, where the testing stage spans throughout phases of software development lifecycle.

1.5 Deliverables

At the end of the project assignment, there are list of deliverables that are expected as projects outcomes. Identifying the deliverables to be produced by the end of project is based on the objectives and scope of project assignment carried out during project completion. The following deliverables are identified:

i. Unit Integration Test (UIT) Procedure

Unit Integration Test Descriptions is the final outcome of the design unit integration test activity performed during the project assignment. Unit Integration Test Procedure contains UIT test cases, UIT test cases expected results, test procedures, test model, and the sample test data. Besides that, this document also has a test traceability matrix map against use case realization. This deliverables is produced with adherence to IEEE Standard for Software Test Documentation. Tailoring has been made to the format of deliverable, with no sections or contents required by the standard are ignored. Following the IEEE standard, test specification can be presented in three types of document known as test design specification, test case specification, and test procedure specification. In the testing stage of Framework One, the deliverable required in representing test specification is known as Test Procedure that combined all the outcome of test design, test case, and test procedures specification into a single document.

ii. System Test Procedure

Similar to UIT Procedure document, System Test Procedure represent the system test design specification, system test case specification, and system test procedure specification in a single deliverable. System test procedure different with the UIT Procedure in terms of the scope each section or contents is identified. As for UIT procedures, the contents of this deliverable met the standard of IEEE Standard for Software Test Documentation.

iii. Unit Integration Test (UIT) Result

IEEE Standard for Software Test Documentation outlined that the test reporting can be represented by four types of documents known as test item transmittal report, test log, test incident report, and test summary report. The first document is ignored as it deem unnecessary in the current environment. The test log, and test incident report are used to record the outcome of testing execution. Unit Integration Test Result then documented based on the record in test log and test incident report to produce a comprehensive deliverable of test execution activity that contains the status of each test case after test execution performed. A test summary report is produced in separate document by the Quality Control Lead at the end of each test iteration. This document is not covered within the scope of this project.

iv. System Test Result

System Test Result has similar contents and purpose with UIT result. The difference is, it is used to document system test execution result where each system test cases are recorded with the status appropriate test status.

1.6 Project Plan

In order to achieve the stated objectives, project plan for the project needs to be developed. It contains the resource requirements, strategy, and schedule to successfully achieve the objectives. The project plan was prepared based on company project plan to ensure the project can be executed within the time given. Since CoRNEA Biz-Apps contains many phases, the strategy of which phases to enter software development lifecycle is up to management of the company.

The resource requirement of project plan concerned on the location and accessibility of input and output document in performing test activity. Some test activities' outcome needs to be recorded in the system where the license of using them is limited. Thus, access time to the system is granted based on team in the organization. Mostly input documents are stored in a server where accessed to it is unlimited. But accessing that document directly from the server slows down the process. An option is to have a copy of the document on working environment. This option requires a regular checking to the server in order not to work on wrong documents as the new version is baseline.

The strategy in test activity completion is to ensure any changes on the system development schedule timeline are updated as it affected the planned activity as a whole. Regular check up on the source document is needed in order not to work on wrong version of documents. For any task assigned in team shall be discussed as a teamwork as to be clear on subdivision of task. This is required to avoid redundant task done among team member. For team based task assignment, the collaborative way of executing and documenting the work needs to be clearly defined as to avoid rework because of different method of documenting the outcome.

In order not to miss the deadline and keep up with other activities that runs concurrently, a schedule for testing activities are needed to be prepared based on project timeline. Please refer to Gantt Chart for schedule of the project in Appendix A

References:

1. Auri Marcelo Rizzo Vincenzi1, Jos´e Carlos Maldonado,M´arcio Eduardo Delamaro, Edmundo S´ergio Spoto, and W. Eric Wong, *Component-Based Software: An Overview of Testing,* Springer-Verlag Berlin Heidelberg 2003

2. Bor-Yuan Tsai*, Simon Stobart, Norman Parrington And Barrie Thompson, *Iterative design and testing within the software development life cycle*, Software Quality Journal **6**, (1997) 295–309

3. B. Hailpern, P. Santhanam, *Software Debugging, Testing, And Verification***,** IBM Systems Journal, Vol 41, **4** No 1, 2002

4. Carol S. Smidts, Donald W. Sova, *A Comparison of Software-Testing Methodologies,* 1995 Proceedings Annual Reliability And Maintainability Symposium

5. David B. Brown, Saeed Maghsoodloo.and Wllliam H. Deason, *A Cost Model For Determining The Optimal Number Of Software Test Cases***,** IEEE Transactions On Software Engineering. Vol. 15. No. 2. February 1989

6. David Gelperin And Bill Hetzel, *The Growth Of Software Testing*, June 1988 Volume 31 Number 6 Conmunications of the ACM

7. D.P. Kelly, R.S. Oshana, *Improving software quality using statistical testing techniques,* Information and Software Technology 42 (2000) 801–807

8. Dick Hamlet, *An essay on software testing for quality assurance – Editor's introduction,* Annals of Software Engineering 4 (1997)

9. Filippo Ricca and Paolo Tonella, *Testing Processes of Web Applications,* Annals of Software Engineering 14, 93–114, 2002

10. Ghaffari Abu Leeny Chacko Jo˜ao W. Cangussu, *Software Test Process Control: Status and Future Directions*, Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04), 2004 IEEE

11. Harrine Freeman, *software testing*, IEEE Instrumentation & Measurement Magazine September 2002

12. Hye-Jung Jung, Won-Tae Jung, and Hae-Sool Yang, *A Study on the Standard of Software Quality Testing***,** Springer-Verlag Berlin Heidelberg 2006

13. Ilene Burnstein, Taratip Suwanassart, Robert Carlson, *Developing A Testing Maturity Model For Sofiware Test Process Evaluation And Improvement,* Computer Science Dept., Illinois Institute of Technology, Chicago

14. James Lyndsay, *The Importance of Data in Functional Testing***,** Workroom Productions 2001

15. JoaÄo W. Cangussu, Raymond A. DeCarlo, Aditya P. Mathur, *A Formal Model of the Software Test Process*, IEEE Transactions On Software Engineering, Vol. 28, No. 8, August 2002

16. John E. Bentley, *Software Testing Fundamentals—Concepts, Roles, and Terminology*

17. Juhani Snellman, *Development Process for Automating Software Testing*, University Of Helsinki Department of Computer Science, 2005

18. Kim Man Lui and Keith C.C. Chan, *Test Driven Development and Software Process Improvement in China*, Springer-Verlag Berlin Heidelberg 2004

19. Mark R. Blackburn, Robert Busser, Aaron Nauman,, *Removing Requirement Defects and Automating Test*, 2001, Software Productivity Consortium NFP

20. Masayuki Hirayama, Tetsuya Yamamoto, Osamu Mizuno, And Tohru Kikuno, *Analysis of Software Test Item Generation - Comparison between High Skilled and Low Skilled Engineers* **,** Proceedings of the 12th Asian Test Symposium (ATS'03)

21. Mikko Karinsalo and Pekka Abrahamsson, *Software Reuse and the Test Development Process: A Combined Approach*, Springer-Verlag Berlin Heidelberg 2004

22. N. Malevris, *E.* Petrova, *On the Determination of an Appropriate Time for Ending the Software Testing Process*, 0-7695-0825-1/00 2000 IEEE

23. Ossi Taipale 1, Kari Smolander 1,2, Heikki Kälviäinen, *Factors Affecting Software Testing Time Schedule*, Proceedings of the 2006 Australian Software Engineering Conference (ASWEC'06), 2006 IEEE

24. P. Dasiewicz, *Formal Approaches to Software Testing***,** Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering

25. R. Black and G. Kubaczkowski, "*Mission Made Possible,*" Software Testing and Quality Engineering magazine**,** Volume 4, Number 4.

26. Rex Black, *Critical Software Testing Processes*, Journal of Software Testing Professionals, 2000

27. Ram Chillarege, *Software Testing Best Practices*, IBM Research - Technical Report RC 21457 Log 96856 4/26/99

28. Sharon Waligora Richard Coon, *Improving the Software Testing Process in NASA's Software Engineering Laboratory*

29. Software Engineering Technical Committee of the IEEE Computer Society, *IEEE Standard for Software Test Documentation*, 1998 Institute of Electrical and Electronics Engineers, Inc.

30. S. H. Kan, J. Parrish, D. Manlove, *In-process metrics for software testing*, IBM Systems Journal, Vol 40, 220 No 1, 2001

31. S. Murugesan, *Attitude Towards Testing: A Key Contributor to Software Quality*, CASE Research Group, Faculty of Business and Technology University of Western Sydney, Macarthur; NSW 2560, Australia

32. Sreedevi Sampath, Valentin Mihaylov, Amie Souter, Lori Pollock, *Composing a Framework to Automate Testing of Operational Web-Based Software,* Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04), 2004 IEEE

33. Staffan Pernler & Niclas StAhl, *An Automated Environment for Software Testing and Reliability Estimation*, Ericsson Telecom AB, Transport Network Systems

34. T. Downs, P. Garrone, *Some New Models of Software Testing with Performance Comparisons,* IEEE Transactions On Reliability, Vol. 40, No. 3, 1991 August

35. Walid Kobrosly and Stamatis Vassilliadis, *A Survey of Sofware Functional Testing Technique*, 1988 IEEE

36. Yuri Chernak, *Validating and Improving Test-Case Effectiveness,* I E E E, January / February 2001

37. Yutao He and Herbert Hecht, Yutao He and Herbert Hecht, *Measuring and Assessing Software Test Processes Using Test Data*, 0-7695-0927-4/00, 2000 IEEE