



Faculty of Information and Communication Technology

**AUTOMATED ACCEPTANCE TESTING TOOL FOR
REQUIREMENTS VALIDATION**

Nor Aiza binti Moketar

Master of Computer Science in Software Engineering and Intelligence

2013

BORANG PENGESAHAN STATUS THESIS

JUDUL : AUTOMATED ACCEPTANCE TESTING TOOL FOR
REQUIREMENTS VALIDATION

SESI PENGAJIAN : 2013 - 2014

Saya : NOR AIZA BINTI MOKETAR

(HURUF BESAR)

Mengaku membenarkan tesis Sarjana ini disimpan di Perpustakaan Fakulti Teknologi
Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hak milik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.

4. ** Sila tandakan (/)

_____ SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

_____ TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

_____ TIDAK TERHAD

(TANDA TANGAN PENULIS)

(TANDA TANGAN PENYELIA)

Alamat Tetap: _____

Dr. Massila binti Kamalrudin

Nama Penyelia

Tarikh: _____

Tarikh: _____

CATATAN : * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana (PS).

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

**AUTOMATED ACCEPTANCE TESTING TOOL FOR
REQUIREMENTS VALIDATION**

Nor Aiza binti Moketar

A thesis submitted

**in fulfillment of the requirements for the degree of Master of Computer Science in
Software Engineering and Intelligence**

Faculty of Information and Communication Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2013

DECLARATION

I declare that this thesis entitle “Automated Acceptance Testing Tool for Requirements Validation” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : Nor Aiza binti Moketar

Date :

APPROVAL

I hereby declare that I have read through this project report and in my opinion this project report is sufficient in terms of scope and quality for the award of the degree of Master of Computer Science (Software Engineering and Intelligence).

Signature :

Name : Nor Aiza binti Moketar

Date :

DEDICATION

This thesis is dedicated to my husband Muhamad Hafiz bin Harun, my daughter
Nur Hani Sufia, my parent, and all my family members.

ABSTRACT

Requirements validation is an important process that determine the client-stakeholders' needs and expectation of a product are sufficiently correct and complete. Various requirements validation techniques such as requirements review, prototyping, model-based and testing-based requirements validation can help to evaluate the correctness and quality of requirements. Yet, most of these techniques are tedious, expensive and time consuming. As a result, most project participants sometimes are reluctant to invest time and effort in the requirements validation process. Moreover, there is also lack of tool support that promotes the effective collaboration among client-stakeholders in the process. Motivated from these conditions, we have developed a lightweight approach for requirements validation with automated acceptance testing tool. We have evaluated the tool's usability via substantial usability study, expert review and comparative studies. The results were positive and demonstrate that our tool is able to facilitate the requirements validation process, and establish client-stakeholders collaboration in order to get the right requirements at the earlier stage of software development project.

ABSTRAK

Pengesahan keperluan adalah proses yang penting untuk menentukan kehendak dan jangkaan pelanggan dan pihak berkepentingan adalah tepat dan lengkap. Terdapat pelbagai teknik pengesahan keperluan seperti kajian keperluan, prototaip, pengesahan berasaskan model dan ujian yang boleh membantu dalam menilai ketepatan dan kualiti sesuatu keperluan. Namun, kebanyakan teknik ini sangat remeh, mahal dan mengambil masa yang lama. Kesannya, kebanyakan anggota/ahli sesuatu projek enggan melaburkan masa dan usaha dalam proses pengesahan keperluan. Tambahan pula, peralatan sokongan dalam membantu meningkatkan keberkesanan kerjasama antara pelanggan dan pihak berkepentingan dalam proses ini juga adalah terhad. Ini telah memberi motivasi kepada kami untuk membina suatu pendekatan yang ringan untuk memudahkan proses pengesahan keperluan dengan memperkenalkan perisian automatik untuk ujian-penerimaan pengguna. Kami telah menjalankan kajian kebolegunaan, kajian pakar dan juga kajian perbandingan dengan perisian sedia ada untuk menilai tahap kebolegunaan perisian ini. Hasil kajian ini adalah positif dan menunjukkan bahawa perisian kami dapat membantu dan memudahkan proses pengesahan keperluan dan meningkatkan kerjasama antara pelanggan dan pihak berkepentingan dalam memperoleh keperluan yang tepat di peringkat awal projek pembangunan perisian.

ACKNOWLEDGEMENT

Alhamdulillah, all praise and gratefulness is due to Allah. Thank you Allah for giving me good health and strength throughout my journey in completing this study.

I would like to express my deepest appreciation to my supervisor Dr. Massila Kamalrudin for the supervision and constant support. This thesis would not have been possible without her valuable help, support, motivation, enthusiasm and patience in guiding me to the success of this study.

I take this opportunity to thank all my lecturers, my colleagues and friends who have provided assistance and moral support at every occasion of the study. A big thank you goes to our expert reviewers who have evaluated our tool and also to all those who filled in the questionnaire.

Last but not least, my deepest gratitude and truthful love goes to my beloved husband, my daughter, my parents and all my family members for their endless love, prayers, and encouragement. Thank you very much.

LIST OF TABLES

TABLE	TITLE	PAGE
Table 1.1	Top 10 Factors Contributes to IT Project Failure – Chaos Report	3
Table 1.2	Evaluation Result of The Automated Acceptance Test Tool	15
Table 5.1	The Extracted EUC Model	26
Table 5.2	Summary of Tools Comparison	30

LIST OF FIGURE

FIGURE	TITLE	PAGE
Figure 2.1	V-Model of Software Testing	9
Figure 3.1	Research Component	17
Figure 3.3	Research Flow	18
Figure 4.1	The Overview of Our Approach	21
Figure 4.2	High-Level Architecture of Our Prototype Tool	22
Figure 4.3	UI View of Our Prototype Tool	23
Figure 4.4	The UI View of Workable UI and The Related Abstract Test Cases	23
Figure 4.5	The UI View of Testing Result Report	24
Figure 5.1	Usability Evaluation Results	27

Table of Contents

DECLARATION	iv
APPROVAL	v
DEDICATION	vi
ABSTRACT	vii
ABSTRAK	viii
ACKNOWLEDGEMENT	ix
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER 1.....	1
INTRODUCTION	1
1.1 Research Background	1
1.1.1 What is Requirements?	1
1.1.2 Requirements Validation	2
1.2 Problem Statement.....	3
1.3 Research Questions.....	5
1.4 Research Objective	6
CHAPTER 2.....	7
LITERATURE REVIEW AND RELATED WORKS.....	7
2.1 Requirements Validation Techniques.....	7
2.2 Acceptance Testing.....	8
2.3 Automated Acceptance Test Tool/Framework.....	10
2.4 Related Works on Automated Acceptance Testing Tool	11
2.5 Preliminary Study and Results	13
2.6 Discussion and Summary	16

CHAPTER 3.....	17
RESEARCH METHODOLOGY.....	17
3.1 Research Design	17
CHAPTER 4.....	20
IMPLEMENTATION	20
4.1 Our Approach	20
4.2 Architecture and Implementation	21
4.3 Tool Usage Example.....	22
4.4 Summary.....	24
CHAPTER 5.....	25
EVALUATION	25
5.1 Usability Criteria for Usability Evaluation.....	25
5.2 Survey Method.....	26
5.3 Survey Result and Analysis	27
5.4 Expert Review	28
5.5 Tools Comparison.....	29
5.6 Survey Validity Assessment.....	31
5.7 Summary.....	31
CHAPTER 6.....	32
CONCLUSION AND FUTURE WORK.....	32
6.1 Summary of Research Contributions.....	32
6.2 Conclusion	32
6.3 Future Work.....	33
6.4 Summary.....	33

REFERENCES	35
APPENDIX A: CONSENT FORM (STUDENT)	38
APPENDIX B: SURVEY QUESTIONNAIRE	39
APPENDIX C: INTERVIEW QUESTIONNAIRE	40

CHAPTER 1

INTRODUCTION

This research examines one of the most important phases of Requirements Engineering (RE) process, which is requirements validation. It is an important stage to ensure that we have the right requirements from client-stakeholders before moving to software development process. Getting the right requirements is very important to guarantee the success of the software project. This chapter describes our research background, problem statement, research questions and objectives.

1.1 Research Background

1.1.1 What is Requirements?

Requirements are captured at the first stage of Requirements Engineering process. It is a formal expression of client-stakeholders' need and expectation of a system to satisfy their business objectives. It describes "how the system should behave, constraints on the system's application domain information, constraints on the system operation or specification of a system property or attribute" [1]. It is a foundation of the system that needs to be well understood and agreed by client-stakeholders in order to built the right product that reflect their requirements.

Capturing the correct, complete and unambiguous requirements are very critical in requirement engineering process to guarantee the success of a software projects. Errors in requirements may have unacceptable consequences in the software development. Thus, it is vital to have a requirements validation to detect errors in the requirements specification before moving to the next phase of software development life cycle.

1.1.2 Requirements Validation

Requirements validation is the last phase of requirements engineering process, after the requirements elicitation and specification process. This phase is to ensure the right requirements are captured from the client-stakeholders. The main objective of requirements validation is to confirm that requirement is correct, complete, consistent and agreed by client-end users before the implementation [2]. It is an important process as it contributes to detect possible defects or errors in the requirements specification and to eliminate those errors before moving to the later phase of software development. The cost of the changes to fix errors in software will be increases and also more time consuming in later phases of the software life cycle than correcting them during the requirement phase.

There are various requirements validation techniques applied in the industry such as requirements review, inspections, prototyping, model-based, requirements testing and viewpoint-oriented requirements validation [2][3][4]. Each of them has their own advantages and disadvantages depending on the organizations' needs to fit with their project requirements, schedule and budget. Selecting the appropriate requirements validation techniques for validating requirements is very important in order to detect errors at the early stages of software development. This will help in completing the projects within schedule, budget and according to desired functionality [2].

1.2 Problem Statement

Incomplete requirements, poor user's input and stakeholder's conflicts are commonly known to be the major root causes to impaired software projects. A software project is considered as an ultimate failure when the intended users refuse to use the system. One of the reasons for their refusal to use the system is that the system does not meet their needs and expectations [5][6].

A recent comprehensive study by Nasir and Shahibudin [7] shows that the most critical factor in determining success of a software project is the precise requirements and specification. The same have been reported in the Standish Group Chaos Report [8], which described the top ten factors of impaired IT projects as in the Table 1.1 below.

No.	Project Impaired Factors	% Of Responses
1	Incomplete Requirements	13.1
2	Lack of User Involvement	12.4
3	Lack of Resources	10.6
4	Unrealistic Expectations	9.9
5	Lack of Executive Support	9.3
6	Changing Requirements and Specifications	8.7
7	Lack of Planning	8.1
8	Didn't need it any longer	7.5
9	Lack of IT Management	6.2
10	Technology Illiteracy	4.3
11	Other reasons	9.9

Table 1.1: Top 10 Factors contributes to IT Project Failure – Chaos Report

These studies and many other research shows that a poor up-front definition of the requirements is the major cause of impaired software project. The cost and time of changes to fix software errors in the later stage is much higher and time consuming compared to correcting them in the requirements stage. Therefore, it is vital to have an early

requirements validation to detect errors in the requirements specification before moving to the next phase of software development life cycle.

Requirements validation is aimed to confirm that requirement is correct, complete, consistent and agreed by client-end users before the implementation [2]. It is an important process as it contributes to detect possible defects or errors in the requirements specification and to eliminate those errors before moving to the later phase of software development. Unfortunately, it is often neglected and informally conducted on ad hoc bases without any proper procedure due to ignorance of the tasks, lack of resources and limited schedule or project funding [9]. Moreover, there is also lack of effective tool support that promotes the collaboration among client-stakeholders in requirements validation process. The client-stakeholders involvement and collaboration during requirements validation process is very important to ensure every party have the same understanding on the requirements.

1.3 Research Questions

The key research questions in this study is:

“Can automated acceptance testing tool help in providing better support in requirements validation process?”

Based on this main research question, we have formulated a few smaller research questions as follows:

- What are the limitations in the current requirements validation techniques applied in the industry?
- Why is acceptance testing is important for requirements validation in software development project?
- What are the challenges in the manual acceptance testing process?
- How is the automated acceptance testing tool help to improve the manual acceptance testing process and enhance the requirements quality?

1.4 Research Objective

The main objective in this study is to propose a new approach for requirements validation process in order to get the best quality of requirements to drive the software development project. This study also aims to provide the following:

- To find the limitation in the current requirements validation techniques.
- To develop a better support in requirement validation by providing automation to acceptance test.
- To compare and evaluate the efficacy and usability of the new approach with the current available automated acceptance-testing tools as requirements a validation approach.

CHAPTER 2

LITERATURE REVIEW AND RELATED WORKS

This chapter describes the research foundation in this study. We starts with some literature review on the existing requirements validation techniques in general and find the problems and limitation for each technique. Further, we investigate the current acceptance testing process as well as the implementation of automated acceptance test tools as a requirements validation approach in the industry.

2.1 Requirements Validation Techniques

There are numerous requirements validation techniques that commonly use in the industry, such as requirements review, inspections, prototyping, model-based, requirements testing and viewpoint-oriented requirements validation [2][3][4]. Each of them has their own advantages and disadvantages depending on the organizations' needs to fit with their project requirements, schedule and budget.

Requirements reviews involve a group of people from both the client-stakeholders and developer to read and analyze the requirements. Reviews help both parties to identify problems and discuss on possible solution to resolve the problems. However, this technique is not effective in identifying defects, time and resources consuming as it involves people from different organisation.

Model-based requirements validation technique uses UML model such as use case diagram, class diagram and sequence diagram to validate requirements. It also provides an overview of the entire system and consistency of the requirements. However, this

techniques is time consuming and very technical, which difficult for the client-stakeholders to understand the requirements.

Studies show that requirements prototyping and testing-based requirements validation are both effective techniques to identify defects. Requirements prototyping is found beneficial as it help the user to visualize the requirements by providing a prototype of the system. The prototype also reusable in other activities such as system designing and user interface development [3].

Testing based requirements validation helps in defining test cases to ensure each requirement are testable. It is an effective way of exposing requirements problems like incompleteness, inconsistency and ambiguity through suggesting possible test of the requirements [2]. The test cases are also reusable in the testing phase. However, both of these techniques are expensive and time consuming as it requires resources and effort to develop the prototype as well as writing the test cases.

2.2 Acceptance Testing

Requirements engineering and software testing are two software development components that is closely related to the verification and validation process. Both components aim to support the development phase in order to build a product that will meet the client-stakeholders need and expectations. Typically, there are four stages of testing involved in software development project, which includes unit, integration, system, and acceptance level testing as depicted in a V-model as shown in Figure 2.1 [10]. The V-Model of software development shows the test activities beginning in parallel with the corresponding development activities [11]. Our focus in this study is on the validating the user requirements with acceptance testing.

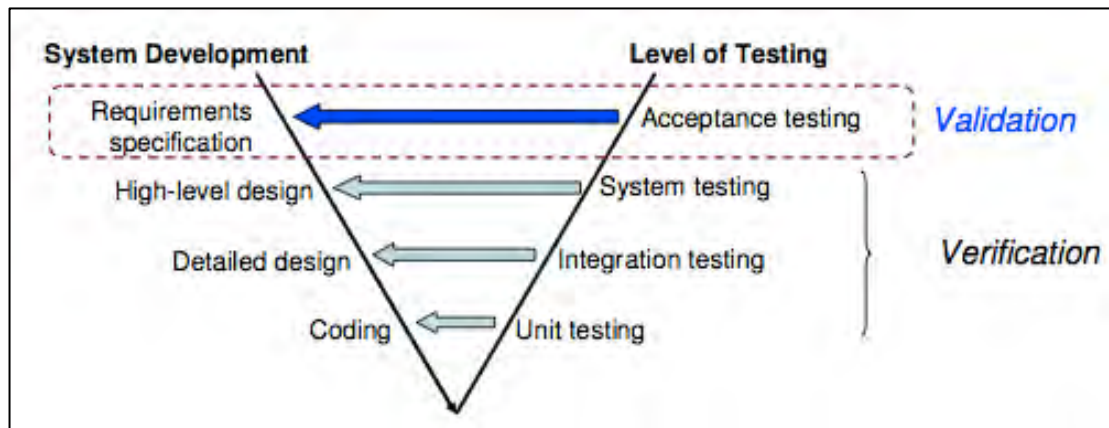


Figure 2.1. V-Model of Software Testing

Acceptance tests or business-facing test is also known as a “functional”, “customer”, or “story” test [12]. Acceptance tests are used to identify that the software to be developed will meet the customer’s needs and interests. It is a validation activity that should be performed by the customers to ensure that the intended software is acceptable and covers all of the main functionalities as in the initial requirements. The term “acceptance test” is often confused with the “user acceptance test (UAT)”, which is generally conducted at the last phase of SDLC for the purpose of a final validation of the newly developed software. In contrast, acceptance test or business-facing test is done at the initial phase of SDLC. Commonly used to measure the business value of a requirement, it is a validation (testing) procedure conducted at a higher level between the business logic and the user interface or directly with the user interface [12][13].

However, in many cases the customers refuse to do the acceptance test as they lack of knowledge, motivation and time constraint. Therefore, in the real software development, the implementation of the acceptance tests is often being left to the sole initiative of the developers. As a result, the acceptance tests are often expressed in an overly technical notation, resulting in difficulties for the customers and domain experts to review the requirements. Moreover, manual acceptance testing processes are generally tedious, expensive and time consuming [13]. Therefore, this test is often neglected and done poorly

by the development team. Sensitive with these problems, some researchers have made the efforts to develop automated acceptance test frameworks.

2.3 Automated Acceptance Test Tool/Framework

The automation of acceptance tests seems as a promising initiative to ease and improve the tedious manual process [13]. An automated acceptance-testing framework can provide significant value to projects by involving the customer to work together with the development team [14]. The customer is responsible to write the acceptance tests in the language that they understand. This task is usually done with the collaboration with the Business Analyst and the development team. This is important to ensure that all business stakeholders and the development team have the same understanding on the requirements. Plus, the defined test criteria can served as a “contract” between the development team and the business stakeholders. Acceptance test also can helps to identify incomplete or ambiguous requirements.

There are a lot of open source tool available for automated acceptance testing. Cucumber [15] and FitNesse [16] are among the most popular automated acceptance test framework. Concordion [17] is another good tool that almost similar to Cucumber. Basically, all of these tools have similar flow/framework where it starts with the customers write the specification or acceptance criteria and test scenarios in natural language commonly in the form of tables or “*Given-When-Then*” format. Then, the developers need to write the fixture code to execute the test and finally produce the test results. These tools are found beneficial as it allows the non-technical users to implement this tool, as it does not require any programming skills to specify the acceptance tests. The ease of writing and reading the requirements specification also draws the business stakeholders to collaborate in the process.