

الآيات القرآنية

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ * خَلَقَ الْإِنْسَانَ مِنْ
عَلَقٍ * اقْرَأْ وَرَبُّكَ الْأَكْرَمُ * الَّذِي عَلَّمَ بِالْقَلَمِ * عَلَّمَ
الْإِنْسَانَ مَا لَمْ يَعْلَمْ *)

صدق الله العظيم

الآيات (1-5) من سورة العلق

الحمد لله

الحمد لله الذي إنشاء خلقه وبراً ورفع السماء وشق أسباب الثرى ولا يغيب عن
علمه ديب النمل في الليل إذا سرى

نحمده سبحانه وتعالى هجراً على الطائفين أجراً وأسبيل على العاصين شراً ...

الحمد لله الذي تسبحة البحار الزاخرات والأفهار الجاريات والجبال الراسيات ...

الحمد لله الذي دانت لعظمته الرقاب وجرى بأمرة السحاب وذلت لجبروته الصعاب ..

الحمد لله الذي انعم علينا بنعمة السمع والبصر ، نحمده حمدا يليق بجلال قدره وعظيم
سلطانه ...

الحمد لله الذي سخر لى هذا وما كنا لة مقرنين ..

وصلى الله على نبينا وحبينا محمد صلى الله عليه وسلم.

الإهداء

إلى رفاق الحياة

الذين تلاشت وتتلاشي ذاتهم جزء في ذاتنا الكل...

والي الذين يتجسدون معنا في ايقاعات شعبنا العظيم ..

وهؤلاء قلوبهم علي وطنهم وعلي شعبهم..

وهو مصدر طاقة الفكر والحياة والفن والتاريخ..

كما هو نهر انسانيتنا العظيمة الذي لا يتوقف..

إليهم جميعاً نهدي هذا البحث آمليين أن يكون دعامة حقيقية في نهضة هذا

المجال.

ونقدمه بكل التواضع مساهمة في صرح مكتبتنا العلمية السودانية

.

الشكر والعرفان

في البدء الشكر أعظمه وأجله لله سبحانه وتعالى.

إلى اللائي نشرن دعواتهن في طريقنا حتى بلغنا مآلنا عليه

..... أمهاتنا

إلى الذين تكبدوا مشاق الحياة وشقوا لنا طريق العلم

والمعرفة.....آبائنا

إلى مشرف البحث الدكتور.....د/ محمد عوض الشيخ

إلى الذين إرتوينا من فيض علمهم وتجاربهم ولم يبخلوا علينا بما

يعلمون..... أساتذتنا

إلى رفاق الدرب في العلم والمعرفة زملائنا
إلى منارة العلم وقلعة المعرفة جامعة السودان للعلوم

والتكنولوجيا

إلى كليتنا التي نفخر بها دوماً كلية علوم الحاسوب وتقانة
المعلومات

مستخلص البحث

إكتسبت الحاجة الي برامج الحاسوب أهمية كبيرة للأفراد والمؤسسات وصارت البرمجيات تؤدي أغراض كبيرة ومعقدة مما أدى للإهتمام بالجودة في البرمجيات والحاجة لتطوير طرق لاختبار البرمجيات للتأكد من صحتها وانها تؤدي الأغراض المطلوبه منها.

عملية إختبار البرمجيات تتطلب الجودة والسرعة و من مشاكلها التكلفة العالية للموارد والوقت ولذلك إتجه مهندسو اختبار البرمجيات الي تقليلها، ومن الوسائل المستخدمه في تقليل التكلفة العالية للمتطلبات والوقت هي استخدام عملية إختبار البرمجيات المحوسب وهي تركز علي قيام الحاسوب باداء بعض عمليات إختبار البرمجيات بدلا عن الانسان.

ظهور إختبار البرمجيات المحوسب أدى الي قيام أبحاث علمية استحدثت فيها أطر عملية في مجال اختبار البرمجيات المحوسب، هذا البحث ركز علي أثنتين منها وهما اطار اختبار البرمجيات المحوسب الذي يركز علي البيانات واطار اختبار البرمجيات المحوسب الذي يركز علي الكلمات المحجوزة ، وذلك بغرض تصميم إطار عمل هجين لاختبار البرمجيات المحوسب مكون من الاطارين معا لتحقيق الاستفادة القصوي منهما.

ولتصميم اطار العمل الهجين لاختبار البرمجيات المحوسب هنالك متطلبات يجب تحقيقها وهي المتطلبات ذات المستوي العالي واهمها هو سهولة الاستخدام ، القابلية للاصلاح والتنفيذ التلقائي للاختبار واستخراج

التقارير، ولتحقيق كل منها كانت هنالك متطلبات اخري أكثر تفصيلا، وفي هذا البحث صمم إطار العمل الهجين لاختبار البرمجيات المحوسب الذي يحقق تلك المتطلبات وللتأكد من صحة الإطار نُفذت نسخ تجريبية بدليل للتنفيذ علي مجموعة مختلفة من البرامج وقد نجح دليل التنفيذ في إثبات جدوي الاطار.

بالرجوع الي البحث والنتائج المستخلصة من دليل التنفيذ توصل البحث الي أن إطار العمل الهجين لاختبار البرمجيات المحوسب المكون من اطار اختبار البرمجيات المحوسب الذي يركز علي البيانات واطار اختبار البرمجيات المحوسب الذي يركز علي الكلمات المحجوزة في هذا البحث يحقق المتطلبات المرجوة منه ، مما يعني أنه يوفر طريقة عملية لاختبار البرمجيات المحوسب وان تنفيذه يؤدي الي زيادة كفاءة عملية إختبار البرمجيات المحوسب ، وبالتالي التأثير الايجابي علي الجودة في البرمجيات.

Abstract

Software systems are getting more and more important for organizations and individuals at the same time software's growing bigger and more complex, this lead to importance of software quality; the need for better quality means more pressure for software testing and for test engineers taking care of it; also the needs to decrease the amount of resources needed for the testing operation is rising.

One attractive solution to this problem is test automation, i.e. allocating certain testing tasks to computers; researches were done in test automation for defining approaches and frameworks, this dissertation focuses on two of the test automation frameworks; data-driven test automation framework and keyword-driven test automation framework, in order to construct a hybrid test automation framework.

For constructing the test automation hybrid framework the research surveys on its requirements, it started by defining the high-level requirements; it's the most important requirements it identified as ease-of-use, maintainability, the ability to automatically execute tests and report results; more detailed requirements were derived from the high-level requirements.

Design and implementation was done for fulfilling those requirements it

presents the hybrid test automation framework, to validate and to ensure the feasibility of the framework against its requirements, implementation pilots were done in an execution prototypes, and it validates the framework.

Based on the pilot results, the overall hybrid test automation framework design and implementation found to be feasible; the most interesting finding is that the implementation of the hybrid test automation Framework will provide test automation methods that lead to increase the efficiency of the software test automation which affect software quality.

Contents

أَلَيَات الْقِرَانِيَة.....	i
أَلْحَمْد لِلّٰه.....	ii
الإِهْدَاء.....	iii
الشُّكْر وَالْعِرْفَان	iv
مُسْتَخْلَص الْبَحْث	v
Abstract.....	vi
Contents.....	vii
List of Figures	xii
List of Tables.....	xiii
List of Codes.....	xiii
List of Images.....	xiii
Terms.....	xiv

Chapter 1 Introduction

1.1 Introduction	1
1.2 Software Testing Background	1
1.3 The Problem Statement	1
1.4 The Proposed Solution of the Problem	2
1.4.1 The Research Objective.....	2

1.4.1.1 Steps Towards Fulfill the Objective	3
1.4.2 The Research Scope and Limits.....	3
1.5 Methodology	4
1.6 Overview of the Research.....	4
1. 6.1 The Research Outline	5
1.7 Chapter Summary	5

Chapter 2 Software Testing Background

2.1 Introduction	6
2.2 Software Testing Background	6
2.3 Objectives of Testing.....	6
2.4 Test Cases.....	6
2.4.1 Selection of Test Cases.....	6
2.4.2 Software Test Oracles.....	7
2.5 Testing Levels	7
2.5.1 Low Level Tests	8
2.5.1.1 Unit Testing.....	8
2.5.1.2 Integration Testing.....	8
2.5.2 High Level Tests.....	9
2.5.2.1 System Testing	9
2.5.2.2 Acceptance testing.....	12
2.6 Testing Versus Test Automation	12
2.6.1 Manual Testing.....	12
2.6.1.1 Cost of Manual Testing	12
2.6.1.2 Where to Keep Your Manual Tests	12
2.6.1.3 When to Focus on Manual Testing	13
2.6.1.4 The Problem with Manual Testing:	13
2.6.2 Manual and Automated Testing.....	13
2.6.3 Automated Testing	14
2.6.3.1Researches on Software Test Automation	14
2.7 Chapter Summary	14

Chapter 3 Automation of Software Testing

3.1 Introduction	15
3.2 Test Automation Background.....	15

3.2.1 Objectives of Test Automation	15
3.2.1.1 The Benefits of Test Automation.....	15
3.3 Which Tests to Automate	17
3.3.1 Limitation of the Scope	18
3.4 Attributes of Test Automation	18
3.5 Cost of Automated Testing.....	18
3.6 Disadvantages of Automation Testing.....	19
3.7 Chapter summary.....	19

Chapter 4 The Proposed Framework Requirements

4.1 Introduction	20
4.2 High Level Requirements	20
4.3 Framework Capabilities.....	20
4.3.1 Executing Tests Unattended	20
4.3.2 Starting and Stopping Test Execution.....	20
4.3.3 Handling Errors	21
4.3.4 Verifying Test Results	21
4.3.5 Assigning Test Status	21
4.3.6 Handling Expected Failures.....	21
4.3.7 Detailed Logging	22
4.3.8 Automatic Reporting	23
4.4 Modularity	23
4.4.1 Linear Test Scripts.....	23
4.4.2 Test Libraries and Driver Scripts.....	24
4.5 Data-Driven Testing	24
4.5.1 Editing and Storing Test Data.....	25
4.5.2 Processing Test Data	26
4.5.3 Advantages of Data-driven scripts.....	27
4.5.4 Disadvantages of Data-driven scripts	27
4.6 Keyword-Driven Testing.....	27
4.6.1 Editing and Storing Test Data.....	28
4.6.2 Processing Test Data	28
4.6.3 Keywords in Different Levels.....	28
4.6.4 Advantages of Keyword-driven scripts.....	30
4.6.5 Disadvantages of keyword-driven scripts	30
4.7 Other Implementation Issues	30

4.7.1 Implementation Language	30
4.7.1.1 System Programming Languages	31
4.7.1.2 Scripting Languages	31
4.7.2 Implementation Technique	31
4.7.2.1 Manual Programming	32
4.7.2.2 Capture and Replay.....	32
4.7.3 Testware Architecture.....	32
4.7.3.1 Version Control	32
4.7.3.2 Coding and Naming Conventions	32
4.7.3.3 Documentation.....	33
4.8 Testability	33
4.8.1 Control.....	33
4.8.2 Visibility	33
4.9 Roles.....	34
4.9.1 Test Automation Manager	34
4.9.2 Test Automation Architect	34
4.9.3 Test Automators	34
4.9.4 Test Designer.....	35
4.10 Detailed Requirements	35
4.11 Chapter Summary	35

Chapter 5 The Proposed Framework Design and Implementation

5.1 Introduction	36
5.2 Framework Structure	36
5.2.1 Test Design System	36
5.2.2 Test Monitoring System	36
5.2.3 Test Execution System	37
5.2.4 Test Data Parser.....	37
5.3 The Framework View	37
5.4 Presenting and Processing Data- Driven Test Data	38
5.4.1 Presenting Test Cases	38
5.4.2 Using Test Data	40
5.4.3 Data Driven Testing Example.....	40
5.5 Presenting and Processing Keyword-Driven Test Data	41
5.5.1 Presenting Test Cases	41
5.5.2 Presenting User Keywords	41

5.5.3 Using Test Data	42
5.5.4 Keyword Driven Testing Example	43
5.6 Chapter Summary	43

Chapter 6 Implementation Pilot

6.1 Introduction	44
6.2 Implementation Decisions	44
6.2.1 Technical Decisions.....	44
6.2.2 Decisions Regarding the Pilot.....	44
6.3 Implementing Reusable Framework Components	45
6.3.1 Test Data Parser.....	45
6.3.2 Logger	46
6.4 Data-Driven Windows Application Testing	46
6.4.1 Test Data.....	46
6.4.2 Driver Script	46
6.4.3 Test Library	47
6.4.4 Test Log.....	47
6.5 Keyword-Driven Windows Application Testing	47
6.5.1 Test Data.....	47
6.5.2 Driver Script	47
6.5.3 Test Library	47
6.5.4 Test Log.....	48
6.6 Keyword-Driven Web Testing.....	48
6.6.1 Test Data.....	48
6.6.2 Driver Script	48
6.6.3 Test Library	49
6.6.4 Test Log.....	49
6.7 The Pilot Results.....	50
6.8 Chapter Summary	50

Chapter 7 Results, Lessons Learned and Limitations

7.1 Introduction.....	51
7.2 Evaluating the Dissertation Against its Requirements.....	51
7.3 Results.....	51
7.3.1 Feasibility of the Framework.....	52
7.3.2 The Framework Benefits	52

7.4 Learned Lessons.....	53
7.5 Limitations	53
7.6 Chapter Summary	53

Chapter 8 Conclutions and Future Work

8.1 Introduction.....	54
8.2 Future Work	54
8.3 conclusion	54
9. References.....	55
10. Appendices	

List of Figures:

No.	Topic	Page No.
1.1	Different Available types of test automation frameworks	1
1.2	The Aims and the topics of the dissertation	4
2.1	V Model	7
4.1	Handling expected failures	21
4.2	Two test log examples	22
4.3	Example of test report	23
4.4	Linear test scripts	24
4.5	Driver scripts and a test library	24
4.6	Data-driven approach	25
4.7	Data-driven test data file	25
4.8	Keyword-driven test data file	28
4.9	Handlers for keywords in Figure 4.8	28
4.10	Keyword-driven test data file with higher level keywords	29
4.11	Creating higher level keywords in test library	29
4.12	Creating higher level keywords in test design system	29
5.1	High level view of the hybrid framework	38
5.2	Detailed view of the framework	38
5.3	data-driven input file Example	39
5.4	Data-driven test log example	41
5.5	Keyword-driven test cases and user keywords	42

5.6	User login screen	43
5.7	Using keyword-driven test data log	43
6.1	Simple Calculator System	45
6.2	Test data for data-driven Calculator pilot	46
6.3	the test log created for data driven for Calculator Example	46
6.4	Test cases for keyword-driven Calculator pilot	47
6.5	Test log from keyword-driven Calculator example	48
6.6	Test cases and user keywords for keyword-driven web pilot	48
6.7	Test log of keyword-driven web pilot (connection Not Succeeded)	49
6.8	Test log from keyword-driven web pilot (Succeeded)	49

List of Tables

No.	Topic	Page No.
1.1	issues excluded from the scope	3
4.1	High level requirements for test automation frameworks	20
4.2	Suggested logging levels	22
4.3	Detailed requirements for a Hybrid test automation framework	35
6.1	Pilot Results against the Detailed Requirements	50
7.1	Evaluating the dissertation against its requirements	51
7.2	evaluating the framework against high level requirements	52

List of Codes

Listing No.	Topic	Page No.
4.1	Simple data-driven parser	26
5.1	Using data-driven test data	Appendix A
5.2	simple Calculator example	Appendix A
5.3	simple Calculator Data-driven test example	Appendix B
5.4	java code that implement login screen	Appendix C
5.5	Using keyword-driven test data for login	Appendix D

5.6	Using keyword-driven test data for calculator	Appendix E
6.1	data-driven testing Calculator Example	Appendix F
6.2	keyword-driven testing for Calculator examples	Appendix G
6.3	keyword-driven testing web Example	Appendix H

List of Images

Image 5.1: Simple data-driven parser out put26

Terms

Data-Driven testing

A scripting technique that stores test inputs and expected outcomes as data, normally in a tabular format, so that a single driver script can execute all of the designed test cases.

Defect

Introduced into software as the result of an error, a flaw in the software with potential to cause a failure, also called Fault or, informally, bug. (Craig and Jaskiel, 2002)

Driver

A software module that invokes and, perhaps, controls and monitors the execution of one or more other software modules. (IEEE Std 610.12-1990)

Driver Script

A test script that drives the test execution process using testing functionality provided by test libraries and may also read test data from external sources called a control script.

Error

A mistake, misconception, or misunderstanding on the part of a software developer. (Burnstein, 2003)

Expected Failure

Occurs when a test case which has failed previously fails again similarly.

Expected Outcome

Outputs and data states that should result from executing a test. (Fewster and Graham, 1999)

Keyword-Driven Testing

A test automation approach where test data and also keywords instructing how to use the data are read from an external data source, when test cases are executed Keywords are interpreted by a test library this is called by a test automation framework. (Mosley and Posey, 2002)

Library

A controlled collection of software and related documentation designed to aid in software development, use or maintenance. (IEEE Std 610.12-1990)

Oracle

A document or piece of software that allows test engineers or automated tests to determine whether a test has been passed or not. (Burnstein, 2003)

System Under Test (SUT)

The entire system or product to be tested.
(Craig and Jaskiel, 2002)

Test Case

A set of inputs, execution preconditions and expected outcomes developed for a particular objective such as to, exercise a particular program path or to verify Compliance with a specific requirement.