Guidelines for the Development of Automated Test Case Incident Reporting Tool(s)

Name: Bushra Areeb Fatimah

Registration No: 366-FBAS/MSSE/F12

Supervisor: Dr. Rizwan Bin Faiz

Co-Supervisor: Mr. Muhammad Nasir

Department of Computer Science Faculty of Basic and Applied Sciences International Islamic University Islamabad

2015

1. Computer software

بسم الله الرحمن الرحيم

Department of Computer Sciences & Software Engineering

International Islamic University Islamabad

Final Approval

Dated: 4th Sep, 2015

It is certified that we have read the project report submitted by Miss Bushra Areeb Fatimah (366-FBAS/MSSE/F12) and it is our judgment that this research project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the MS Degree in Software Engineering.

Committee

External Examiner
Dr. M. Zohaib Zafar Iqbal
Assistant Professor,
FAST University, Islamabad
zohaib.iqbal@nu.edu.pk

Internal Examiner
Mr. Zulqarnain Hashmi
Lecturer.
Department of Computer Sciences,
IIUI, Islamabad.
zulqarnain@iiu.edu.pk

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD

Supervisor
Dr. Rizwan-Bin-Faiz
Assistant Professor
RIU, Islamabad
rizwan.faiz@riu.edu.pk

A Dissertation Submitted To

Department of Computer Science,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the

Degree of MS Software Engineering

Declaration

We hereby declare that this Thesis 'Framework on the development of Automated Test Case Incident Reporting Tools" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this research with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor Dr. Rizwan Bin Faiz and co-supervisor Mr. Muhammad Nasir. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from any of the training institute or educational institutions, we shall stand by the consequences.

bushra.

Bushra Areeb Fatimah
366-FBAS/MSSE-F12

Declaration

We hereby declare that this Thesis "Framework on the development of Automated Test Case Incident Reporting Tools" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this research with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor Dr. Rizwan Bin Faiz and co-supervisor Mr. Muhammad Nasir. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from any of the training institute or educational institutions, we shall stand by the consequences.

bushra.

Bushra Areeb Fatimah
366-FBAS/MSSE-F12

Acknowledgement

"IN THE NAME OF ALLAH THE MOST MERCIFUL AND THE MOST BENIFICIANT"

All praise to Almighty Allah, the source of all knowledge, who gave us the understanding, courage and patience to complete this research project.

I extend a deep respect for my loveable mother whose blessings and aspirations always inspired me, it is because of her untiring efforts that I am at this position today.

I express my profound regards to my supervisor, Dr. Rizwan Bin Faiz, with deep gratitude for his constant help, invaluable guidance, encouragement and enthusiasm throughout this project. His enthusiasm, integral view on research and his mission for providing high quality work has made a deep impression on me. Without his help this thesis would not have been possible.

I would like to thank Mr. Muhammad Nasir for his valuable time and suggestions who monitor my work and took effort in reading and providing me with valuable comments. I also want to offer heartfelt thanks to the many hundreds of the respondents who have participated in our online survey. Their helpful feedback helped us shape and sharpen the proposed framework

Specially, I want to acknowledge the prayers, endless love, efforts and support of my family and friends especially to my mother who is truly my motivation and inspiration without which I was unable to reach this stage. My deep gratitude goes to my family for their well wishes, moral support and continues encouragement. I am especially indebted to my friends Samia Rafique and Asma Habib for their moral support.

To mention all those whom I am indebted is impossible. Their aspiration, suggestion, assistance, guidance and for everything they have contributed I am sincerely thankful.

Bushra Areeb Fatimah

366-FBAS/MSSE-F12

Abstract

Test case incident reporting is the process of reporting incidents after the termination of testing which involves recording of actual results against expected and other related information which explains reasons of test case failure. It is the most evolving, observable and a challenging task in testing as it has strong impact on testing efficiency as it helps in future maintenance therefore reducing cost and improving quality of software.

Aim of current research is to deliver a framework for the development of automated test case incident reporting tool(s). Although there are many commercial and open source test case incident reporting tools available but their feature set are misaligned with the current requirements of software industry. This is highlighting the necessity to develop such a framework which provides architectural foundation for test case incident reporting tool(s) in order to define their scope with respect to phases and its feature set instead of implementing features which may belong to different phases. Proposed framework classifies feature set in to core phases of test case incident reporting tool(s) and is thoroughly evaluated through well reputed international organizations which are CMMI level 5 and ISO 9001, 90003 certified and also by highly experienced (up to 15 years) software and quality engineers.

Therefore there exists a real need to have a test case incident reporting framework which classifies most comprehensive set of features in to core phases of test case incident reporting.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

FBAS	Faculty of Basic And Applied Sciences
IIUI	International Islamic University Islamabad
ATCIR	Automated Test Case Incident Reporting
SQA	Software Quality Assurance
CMMI	Capability Maturity Model Integration
ISO	International Organizations for Standardizations

CHAPTER 1

INTRODUCTION

1. Introduction

This chapter gives an overview of automated testing, incident reporting, research problem, research aims and objectives, problem domain, research methodology and research process. At the end of this chapter, thesis breakdown in to chapters is given.

1.1. Background

Software testing is a process of executing a software system with the intension of finding errors [17] which results in high reliability of software. It defines whether a program matches its specifications and executes in its intended environment or not [16]. Hence testing confirms reliability in software by recognizing errors. It is the evaluation process in which tester determined the efficiency and correctness of system attributes and features to meet intended work [11]. The test process are performed in parallel with software development and determines whether the activity conform to the requirements and satisfies its intended use and user needs or not [12].

Tester can achieve the high test coverage by using automated testing tools [18]. Testing tools are the software products which are used to automate or semi automate the software testing methods or process which includes test case designing, execution and incident reporting [5, 19]. As a result of which not only cost and efforts are reduced but also the software quality is improved. Since automation improves the overall process of software testing [20]. Among all testing processes test case incident reporting is a challenging task as it has strong impact on effectiveness and efficiency of the whole software testing process [21]. Automated test case incident reporting tools record and track the status of incidents, occur during the automatic execution of the test cases. It often has opportunities of recording, fixing, re-testing of incidents, and reporting capabilities [13]. Its aim is to record anomalies of all the discovered errors and differences between actual and expected outcomes for the successful completion of testing process [14]. Development of automated test case incident reporting tools represent a great challenge for the software engineers as they implement variety of features but these tools are not according to the current requirements of academia. Industries are also gradually implementing them but these tools are misaligned with the requirements of software industry. These so far developed tools do not follow any standard, framework or set of guidelines of software testing and are not mature enough [21]. According to one survey conducted on tool usage of testing practices rate for incident reporting is only 30% [25]. Therefore it is the need of both academia and software industry, to tune them accordingly.

The goal is, therefore, to perform a systematic mapping review of test case incident reporting tools, in order to attain evidence that which automated test case incident reporting tools are available and what features they should include. A detailed multidimensional analysis will be conducted of industry survey and tool evaluation criteria so as to identify gaps and clusters. This will help us in establishing framework for the development of automated test case incident reporting tools which will be a milestone contribution for the development of future automated test case incident reporting tools. This framework will be for those people who will develop automated test case incident reporting tools and also for those who will use them.

1.2. Motivation behind the Study

Our proposed research is motivated by automated test case incident reporting tools. In present there are large numbers of tools for automated test case incident reporting which are developed according to one's own need having no consensus in there parameters [24]. These tools implement various features but they are not following any common standard that's why there exists a gap between academia and software industry. Therefore there is a need to bridge the gap by providing a framework to software industry as which set of parameters/features should be implemented by automated test case incident reporting tools. This motivates us to conduct this research which will lay out a road map for the development of automated test case incident reporting tools. It is guideline for those people who will develop automated test case incident reporting tools as well as for those who use them.

1.3. Aims and Objectives

The aim of our research is to develop framework for the development of automated test case incident reporting tools which provides foundation to those who develop them. We will achieve this objective in several steps firstly we will identify list of open source and commercial tools, after that tool evaluation criteria will be identified which include standard presented by IEEE, ISO, IEC etc. Then a software industry survey will be carried out to identify list of common referred commercial and open source automated test case incident reporting tools, limitations of common referred tools and set of desired features that should be implemented by automated test case incident reporting tools. Finally we will conduct detailed multi-dimensional analysis of industry survey and tool elevation criteria so as to identify gaps and clusters. This will lead us to establish framework for the future development of automated test case incident reporting tools.

1.4. Problem Domain

The most common referred automated test case incident reporting tools developed already are different from each other and there is no framework exists in literature which provides a common set of guidelines which a tool should follow to minimize the bridge between academia and industry. It also takes time and effort to know which tool is the best to use. This study will provide a comprehensive framework for the development of automated test case incident reporting tool so that a standard tool can be developed which is not only meeting the needs of industry but also the academia.

1.5. Research Question

We intend to address following research question in our proposed research.

RQ.1. What is the identification of maximum set of feature(s) and core phases of test case incident reporting tool(s)?

RQ.2. What is the classification of feature set in to core phases?

RQ.3. How can we apply quality criteria on the proposed framework?

1.6. Expected Outcomes

The main outcome of this research work is:

1) Framework for the development of test case incident reporting tool(s)

This framework provides the architectural foundation for the development of tools to be used for test case incident reporting, its usage is that in future which ever tool of test case incident reporting is developed this framework will set an architectural foundation for it as tools are to be used in reality for test case incident reporting whereas framework provides a layout to develop them.

1.7. Research Methodology

Investigative and prescriptive methodologies in nature are used in this research study.

1.8. Research Process

Figure 1 elaborates the research process of the research.

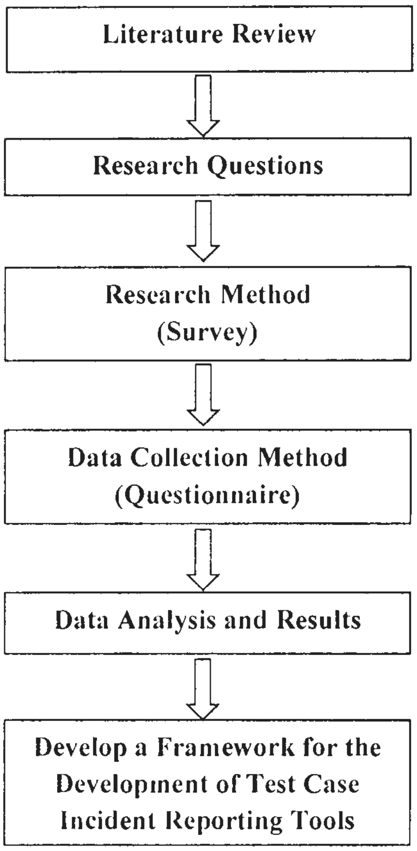
Literature Review
Research Questions
Research Method (Survey)
Data Collection Method (Questionnaire)
Data Analysis and Results
Develop a Framework for the Development of Test Case Incident Reporting Tools

Figure 1.1 The Research Process

1.8.1. Source of Data

The data will be collected from practitioners of software development organizations in different countries of the world. The type of targeted population can be in-house software development organization or the organizations that produce the software for external parties. The organizations can be small, medium or large.

1.8.2. Research Methods

We will use the survey method to collect data. The reason behind using survey method is that surveys have many advantages for example through survey large amount of data can be collected in relatively short period of time, surveys are

CHAPTER 2

LITERATURE REVIEW

2. Introduction

In order to obtain information about which parameters, frameworks, techniques or approaches for test case incident reporting are available, it is important to look at literature survey to identify gaps and clusters.

This chapter explores major frameworks and approaches as discussed in literature in terms of automated incident reporting, their features/parameters and limitations. The purpose was to investigate how comprehensively frameworks were proposed and to find the parameters which need to be verified from industry for test case incident reporting frameworks. At the end it presents gap analysis for future research and improvements are suggested.

2.1. Literature Review

J. Tang [1] proposed "an adaptive model for test automation" with the help of which all Software Testing Lifecycle (STLC) activities, including test case design, test case execution, and test case incident reporting, can be performed efficiently. This model contains requirement agent, construct agent, execution agent and report agent. Report agent creates defect report and analyzing test results. It used query model to query the defects data and test log beside with test reports are generated. However this research was constrained due to following reasons: There was no justification why test results are analyzed and why defect reports are created and this study only reviewed academia and lack industrial feedback.

A. Safana et al. [2] proposed a "SpiraTeam Tool" which has the capability to identify incidents, manage test cases, requirement and associated traceability. In this tool, incidents which can be bugs, issues, enhancements, limitations, change requests, and risks are created, edited, assigned, tracked, managed and closed. They further elaborated automation of testing activities test case requirements and defect can be linked together, processes and real time results can be associated and a central repository of results can be built [2]. However this research was constrained due to lack of industrial feedback.

S. Aljahdali et al. [3] evaluated system components by verifying that the system fulfills specified sets of requirement. This involved test management and control activities which include test planning, test execution and defect management. Defect management is an important activity on which extra attention has to pay. It includes reporting and tracking of defects, defect types, defect severity level, analysis of defects and defects resolution. Each defect whether it is minor or major should be reported because it improves test conditions and future testing. They also suggested that at the end of texting cycle minor defects can be ignored unless they are critical. [3] However this research was constrained due to lack of industrial feedback and the parameters selection criteria for ATCIR identification was also not justified.

Vos T.E.J et al. [4] proposed "a methodological framework for evaluating software testing techniques and tools". The steps of the framework includes: objective of the tool that define the research question for tool evaluation; nature of the tool comprising prerequisites, results, performed operations and tool license; subjects that who apply the tool; objects are those on which tool has to be used; variables involves which data to collect which invokes efficiency, effectiveness and satisfaction of the tools; protocol i.e. how to execute the study and threats for the validation of the study performed. Framework was applied on two industrial case studies and one academic environment but still there is need to evaluate it on more industrial projects to make this study more perfect. However this research was constrained due to limited scale industrial evaluation.

J. Lee et al. [5] conducted "a survey on software testing practices" in which weakness of software testing methods and tools are identified. The survey was divided into three sections including introduction of company, testing environment and test process. The questions in test process are based upon the current practices of software testing methods and tools which include test estimation; test planning, test execution, defect management and defect reporting were selected from ISO/IEC standard 29119. Survey was sent to the companies whose names are mentioned on Fortune 1000 companies website and it was also found that tools usage is higher in defect management and reporting as compared to other processes, maximum practitioners used tools for defect recording, tracking and reporting but usage of these tools is low, there is difficulty in owning, tools used in a limited manner and there is also a need for guidance to evaluate methods and tools. However this research was constrained due to following reasons: this survey only discuss phases of testing process at an abstract level instead of going into detail of ATCIR which can be further explain by sub parameters propose only abstract level The focus of this study was only on the generic software testing processes.

Xiaojun et al. [6] developed "an intelligent general-purpose automated software testing suit" with two tools Harness and Logscanner. Logscanner is an analysis tool which functions in takeing log and dictionary files as input. It not only scans the log files but also summarize the filtered log error messages before producing a report which concludes errors. It can be used to detect and report critical errors automatically and periodically during the testing, or search for specific errors in a large log file during a problem investigation process. However this research was constrained due to lack of industrial feedback and the criteria for the selection of parameters was also not defined.

Hongbo et al. [7] proposed a framework for automated software system RunTool which include test case design, test case execution and log analyser. In test case report result of the tests are classified, test report generated and test results are analysed, classified, recorded and reported. The RunTool framework has five levels including case level, step level, execution level, result level and test report level; all layers are integrated with log level. The test report level starts when the system began to run, it receives results from each layer and then the logs are displayed. Further Hongbo et al. also compared RunTool with WinRunner tool which take lots of human resources to write test cases and TestPartner tool which depend on record reply, if there is no equipment to record, the test cases cannot be written. However this research was constrained due to lak of industrial feedback and the selection of parameters was also not defined. There has to be some criteria based on which selection can be done.

Jonathan et al. [8] evaluated more than hundred academic papers on defect reporting by assessing software defects efficiently for analyzing the latest advancement in refining reliability of the software. The research was categorized into the areas of automation of defect detection and defect fixing and attributes, quality and triaging of defect reports whereas automation of defect detection and defect fixing requires more research. However this research was constrained due to lack of industrial feedback. Selection criteria for tools were also not defined.

Maen Hammad et al. [9] suggested 'a visualization approach for bug reports' to help testers to understand status of submitted reports by modelling different report statuses and their relationships. The visualization includes the developers who handle the bugs and their roles by providing valuable information related to status of bugs, developers information and status of particular bugs. However this research was constrained due to lack of industrial feedback and the selection of parameters was also not defined.

T. Abdou et al. [10] proposed a framework for the testing process of open source software as their development process is different from other software process. For this purpose testing process of NetBeans, Mozilla and Apache HTTP servers studied and compared with ISO/IEC standard software testing process which shows similarities between two activities. The study also highlighted differences and improvement of open source software testing process. However this research was constrained due to lack of industrial feedback and the selection of tools criteria is also not defined.

2.2. Conclusion

By analyzing the existing literature we can see that features set of common referred automated test case incident reporting tools is not organized into phases. Besides proposed frameworks are based upon academic reviews and are not validated from software industry. Since choice of feature set to be implemented by automated test case incident reporting tool are not validated by software industry, therefore desired set of features to be implemented by current tools may not be aligned with current requirements of software industry. So, there is no state of the art in any one framework which is acknowledged by all software industry. Beside this the design methodology of developed framework is controversial i.e. existing framework development is based on weak evidence; some frameworks are developed based upon the comparison of randomly selection of two tools, some frameworks propose feature sets but no justification is given on which basis these are suggested. That's why we designed a framework and implements quality criteria on it.

Our work provides solution involves the industrial feedback. All findings are verified from industry. Our work is a step towards to bridge the gap between software industry and academia. It is noticed that reviewed frameworks are domain specific whereas we focused on independent of domain that is why our framework is generic and we have come up such diversified phases of automated test case incident reporting.

Therefore there is a need to bridge the gap between software industry and academia by providing a framework validated from software industry as which set of parameters/features should be implemented by automated test case incident reporting tools.

CHAPTER 3

RESEARCH METHODOLOGY

3. Introduction

This chapter emphasizes on the research methodology including research approach, research method, survey design technique, data collection method comprising data collection technique, structure and explanation of our questionnaire used in the development of framework of automated test case incident reporting tools. The purpose of this chapter is to highlight how our research process has been conducted and why does the particular research method more appropriate.

3.1. Research Methodology

This section explains and justifies research process which includes research approach, research method, and survey design technique for the development of framework of automated test case incident reporting tools. Diagrammatical presentation of research process is as follows:

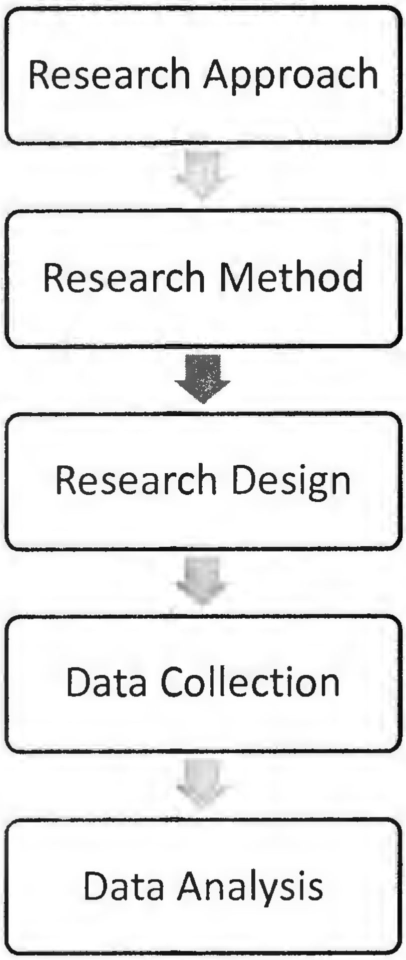

Figure 3.1 Research Methodology

3.1.1. Research Approach

In literature two approaches, deductive and inductive are used by researchers [15]. In deductive approach hypothesis, theories or suggestions are created from theoretical framework and then experiments are carried out to accept or reject them. It is "top down" approach because researcher starts working from more general information to more specific. They start from broad field of information and work down to specific conclusion. While in inductive approach researcher interpret the phenomenon which is based upon the respondent response [23]. It is "bottom up" approach which establishes a new theory from data. This approach used systematic tools including questionnaire, interview and theories. The analysis of research approach provides a new theoretical framework to understand the phenomenon. In our research work inductive approach is used.

3.1.2. Research Method

In literature three research methods case study, experiment and survey are discussed [15]. Researchers use these methods according to their research objective as each method is different from another. Case study and experimental research methods are used to answer "how" and "why". However, surveys are used to find the answer of "what" and "which". Our research deals with "what" and "which" that's why survey is used to answer the research question. Surveys also have many advantages such as large amount of data can be collected in relatively short period of time and they are cheap as compared to other data collection techniques.

3.1.3. Research Design

Our instrument of survey is a questionnaire which was designed based upon feature set of updated open source and commercial tools which are most frequently referred in literature. Below table shows gaps and clusters in the feature set of tools. Clusters in feature set means some features are repeatedly implemented by all tools whereas, gaps refers to the fact that some are not. Thus there is need develop such framework which associates comprehensive feature set(s) against core phases of test case execution tool(s) as per current software industry requirements. Besides such gaps and clusters of feature set are used to design the both open ended and close ended questions in the survey.

Our survey was designed upon the guidelines of Pfleeger and kitchenharm [37-40]. It was distributed to the software and quality engineers through personal contacts and social sites. 133 responses were considered from Pakistan, India, USA, UK, Italy, Egypt, Switzerland, Spain, Brazil, Singapore, Belarus, Germany, France, Belgium, and Pakistan. Our survey encompasses various phases of test case execution, each of which was further explained through comprehensive feature(s) set. Phases and feature(s) set were identified through literature review which includes ISO/IEC 29119 standard and common referred open source and commercial tools.

Common Referred Tools	**Features of Common Referred Tools**				
Tools	**License**	**Identify Incidents**	**Report/ Update Incidents**		**Communicate status to stakeholder**
			Summative evaluation	Formative evaluation	
JMeter [32] **2014**	Open Source	-	-	Yes	-
Selenium [33] **2013**	Open Source	-	-	Yes	Yes
FitNesse [34] **2013**	Open Source	-	Yes	-	-
Bugzilla Testopia [35] **2013**	Open Source	Yes	-	Yes	-
SpiraTest [35][36] **2014**	Commercial	Yes	-	-	-
Squish [37] **2013**	Commercial	Yes	-	Yes	-

Table 3.1 Gap Analysis

3.1.3.1. *Population and Sample Size:*

Population of our survey was software quality department people. Survey was sent through emails and social site to the respondents and it was assured to the respondents that their data would be kept confidential only aggregated results will be published. Most of the survey participants were intended to response questionnaire only because they were encouraged that results shared with them when it would completed. The participant companies are must be user of automated software testing tools and the respondents requirements are as follows:

- Have at least one year of experience in software testing

- Be currently involved in software testing activities
- Can be national and international

3.2. Data Collection Method and Sources

In literature data collection method is divided into two groups; primary data and secondary data [8]. Primary data collection is done by the researchers themselves using experiments, questionnaires or interviews to address the problem for explicit purpose while secondary data is collected through literature survey which is already published in research papers, journals, conference proceeding papers.

3.2.1. Primary Data

Our primary data was collected globally from national and international software organizations to elicit data through questionnaire which was launched online through "Google Docs" service. Almost 10 countries and more than 153 respondents were engaged in our primary data collection and their responses were very valuable for our research work.

3.2.2. Secondary Data

A comprehensive literature survey had performed for the collection of secondary data. For this purpose number of research paper were reviewed which were gathered from the digital libraries including IEEE Xplore, Science Direct, Springer link, ACM, CiteSeerX and Google scholar by using different search strings. Selection of each research paper was evaluated on the basis of its relevancy and quality aspects. The objective of literature survey was to identify common referred automated test case incident reporting tools and their features and to evaluate different tools' features, techniques, approaches, framework, and guidelines so that recommended practices of test case incident reporting can be identified.

3.3. Questionnaire Preparation

A well designed effective questionnaire which meet research objective was prepared keeping following important points in mind:

- Simple English language was used which is understandable for every respondent
- Questionnaire was not lengthy so the respondents not take it as a fatigue.

- Maximum closed-ended questions were added to facilitate the respondent whereas a few open-ended questions were also added to get respondent's opinion and suggestions.
- All technical terms were explained with in questionnaire.
- Online questionnaire was designed which was feasible for respondents to response as it did not require any form to download.
- Online URL would be shared through e-mail and social networks.

3.3.1. Questionnaire Structure

The questionnaire was comprised of three parts:

1) Respondent introduction
2) Organization information
3) Test case incident reporting tools information and required features

3.3.2. Questionnaire Explanation

In our questionnaire we have asked opinions of respondents and to make the comparison of their response we have used percentage. Percentage was calculated by the formula P=F/N*100 where P is percentage, F is frequency and N is the total number of responses.

3.3.2.1. Demographic Details

In order to ensure the information of the respondents which are involved in our survey, the information about respondent name, experience, designation and email address was asked.

3.3.2.2. Organizational Details

In order to identify quality standard of the organization, the information about organization CMMI level, ISO certification, country and number of people working in organization was enquired.

3.3.2.3. Tools Section

In this section we proposed features of tools which are presented in literature and at the end of every question, an open ended question was requested for respondents' suggestions for extra features.

3.3.3. Questionnaire Administration:

Respondents were not required to answer all survey questions and only 20-30 minutes were required to fill the survey form. Respondents could also save their survey responses which were managed internally by survey website.

Questionnaire was shared with target respondents using URL through social networks including email, Facebook groups and LinkedIn. It is assured to target respondents that all the information provided will only be used for research purposes and will not be disclosed or shared with any individual or organization under all circumstances.

The time period of conducted survey was from June 22, 2014 to November 11, 2014. The total number of submitted responses was 154. Once the data collection was completed, survey data was downloaded from the survey website using Excel format.

CHAPTER 4

RESULTS AND ANALYSIS

4. Introduction

This chapter focuses on the detailed analysis of the survey conducted national and international level organizations for the implementation of the development of automated test case incident reporting tools. The survey was based on gap analysis identified in literature review in chapter 2. Demographic details of respondent and variables selected for analysis are also explained in this chapter. Results are presented in the graph form with explanation.

4.1. Demographic Details of Survey

The survey form was sent to more than 200 national and international organizations including Pakistan. USA. India, UAE, UK, Australia, Saudi Arabia, Switzerland, Denmark, Singapore. Ireland, Norway, Malaysia and South Korea. Out of which we received response from 154 organizations including Netsol Technologies, IBM System, Huawei UK ltd, NorthBay Solutions Pvt. Ltd., Bentley Systems, LMKR, Macrosoft, Nextbridge (Pvt) Ltd and many others. The designations of respondents include Software Quality Assurance and Automation Engineer, Quality Engineer, Quality Management Specialist, Quality Advisor, Software Quality Assurance Analyst, Software Test Engineer. QA Manager, Test Manager, Software Test Engineer, SQA Analyst, QA Specialist, Principal Quality Engineer, and Software Test Automation Engineer etc. Some responses from the results are also discarded to make our analysis more authentic as these were not proper filled up.

4.2. Research Analysis

Rigorous data analysis of the survey reveals core phases as mentioned in section 4.3.1 till 4.3.2 each of which is further explained through maximum feature(s) set. Proposed framework for test case incident reporting tool(s) is then evaluated through rigorous quality criteria, which includes CMMI, ISO certification and experience of respondents.

4.2.1. Certifications of Organization

Responded organizations were certified by CMMI (Capability Maturity Model Integration) at level 4 and 5 and by ISO (International Organizations for Standardizations) 9001 and 90003 certified. Through these certifications of organizations credibility can be established which made worth of feedback obtained.

4.2.2. Experience of Respondents

Survey was sent to the people having different years of experienced people and the minimum experience required to fill the questionnaire is 1 year. Mostly respondents of the survey are highly experienced. The experience is further divided into three categories i.e. high, medium and low based upon the response range. The categorization is as follows:

High ≥ 8 years

5 years ≤ Medium ≥ 7 years

1 ≤ Low ≥ 4 years

4.3. Detailed Analysis

Different phases of automated test case incident reporting are identified and against each phase sub factors are identified. Each sub factor is then analyzed through different variables including CMMI level, ISO certification, organization size and respondents' experience which are endorsed by software industry.

4.3.1. Analyze Test Results

The output of test case execution that is test results are analyzed to establish the difference between expected and actual results.

4.3.1.1. Analyze Test Results

4.3.1.1.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.1.1.1.1. CMMI Level

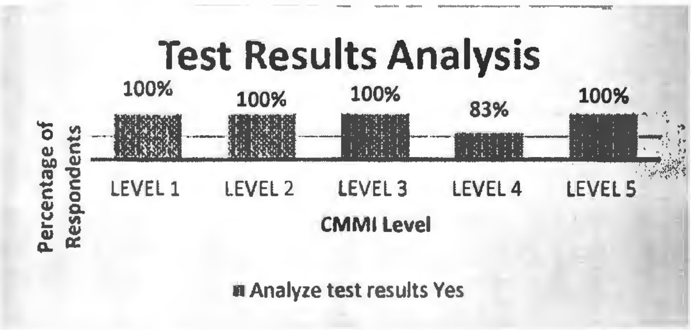

Figure 4.1 Test Result Analysis – CMMI Level

The above chart explains that CMMI Level 1, 2, 3 and 5 organizations 100% support analysis of test result. Whereas, CMMI level 4 organizations supports up to 83%.

4.3.1.1.1.2. ISO Certification

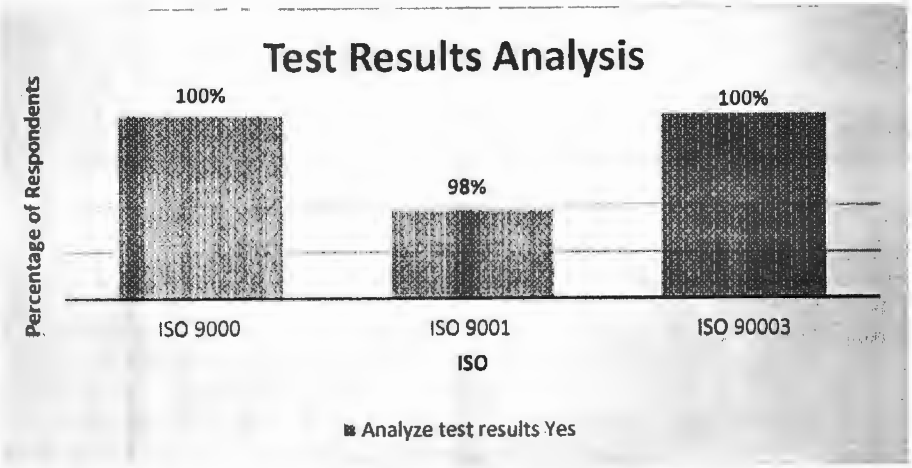

Figure 4.2 Test Result Analysis – ISO Certification

The above chart explains that ISO 9000 and ISO 90003 organizations 100% support implementation of test results analysis. However, ISO 9001 organizations support up to 98%.

4.3.1.1.2. Organization Size Analysis

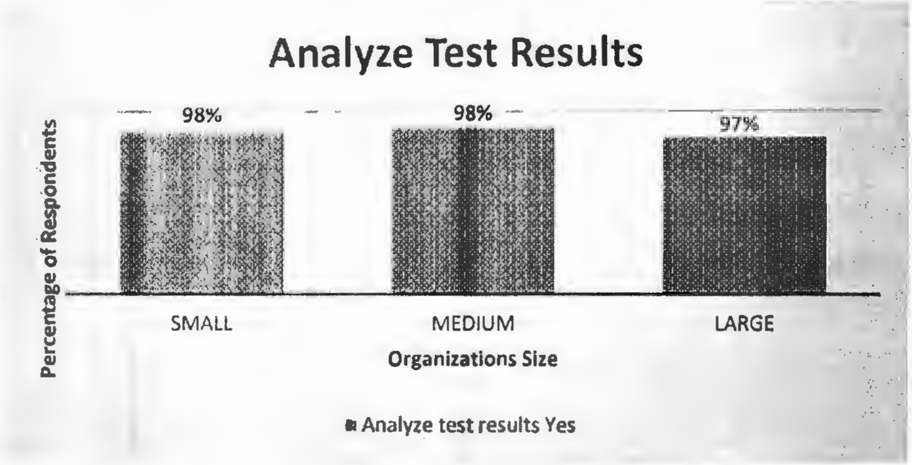

Figure 4.3 Analyze Test Results - Organization Size

The above chart explains that 98% of small and medium size organizations support implementation of test result analysis whereas, large organizations supports up to 97%.

4.3.1.1.3. Experience of Respondent (in years) Analysis

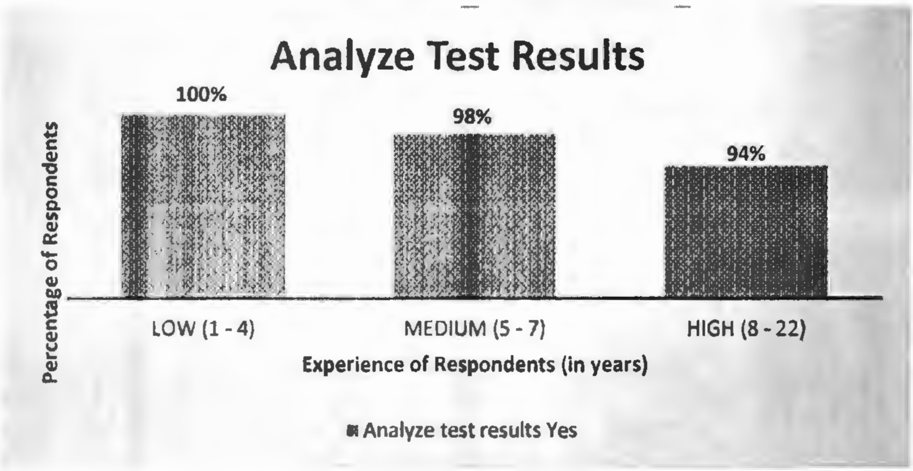

Figure 4.4 Analyze Test Results – Respondents' Experience

The above chart explains that respondents with low experience i.e. from 1 to 4 years, 100% support implementation of test result analysis. However, respondents with medium i.e. 5 to 7 years and high i.e. 8+ years of experience support up to 98% and 94% respectively.

4.3.1.2. Update Incidents

4.3.1.2.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.1.2.1.1. CMMI Level

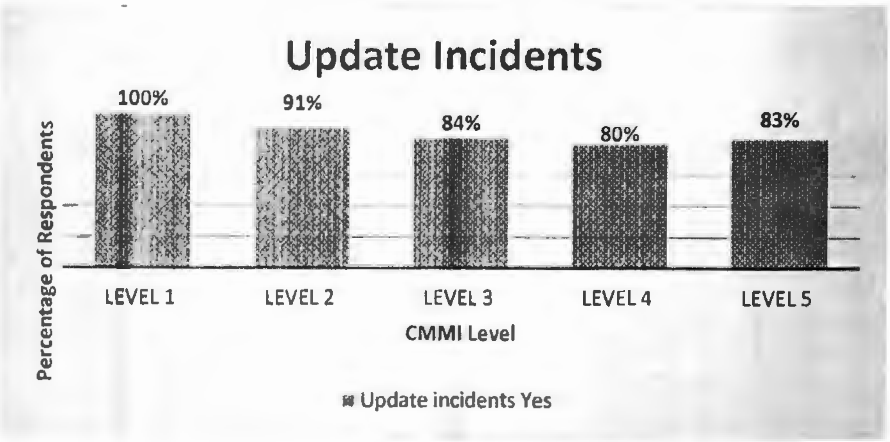

Figure 4.5 Update Incidents - CMMI Level

The above chart explains that CMMI Level 1 organizations 100% support update incidents. Whereas, CMMI level 2, 3, 4 and 5 organizations support up to 91%, 84%, 80% and 83% respectively.

4.3.1.2.1.2. ISO Certification

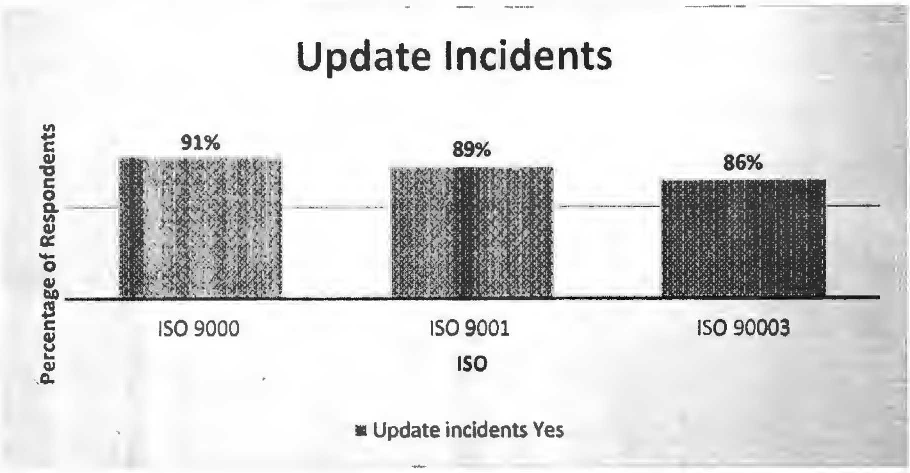

Figure 4.6 Update Incidents - ISO Certification

The above chart explains that ISO 9000, ISO 9001 and ISO 90003 organizations support implementation of update incidents up to 91%, 89% and 86% respectively.

4.3.1.2.2. Organization Size Analysis

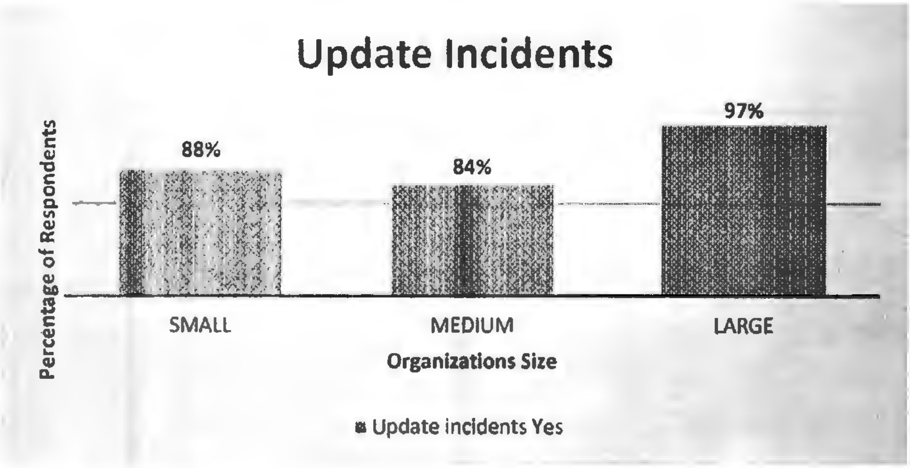

Figure 4.7 Update Incidents - Organization Size

The above chart explains that 88% of small size, 84% of medium size and 97% of large size organizations support implementation of update incidents.

4.3.1.2.3. Experience of Respondent (in years) Analysis

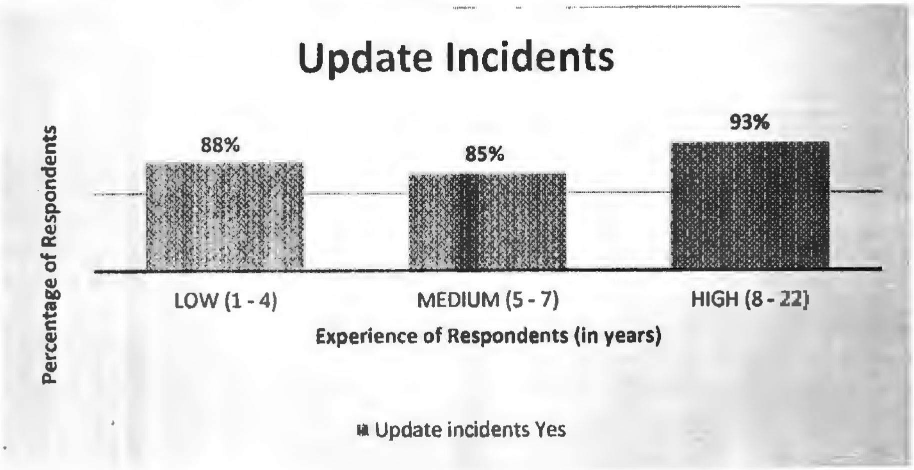

Figure 4.8 Update Incidents – Respondents' Experience

The above chart explains that respondents with high experience 93% support implementation of update incidents. However, respondents with low and medium experience support up to 88% and 85% respectively.

4.3.1.3. Determine that Incident requires Reporting

4.3.1.3.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.1.3.1.1. CMMI Level

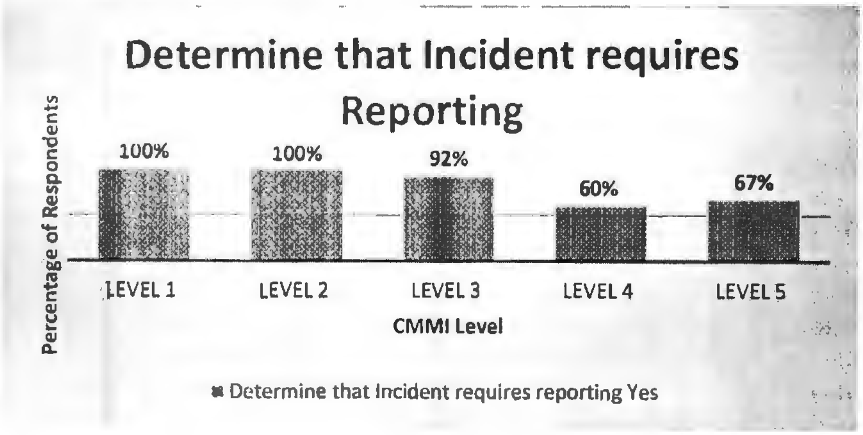

Figure 4.9 Determine that Incident requires Reporting - CMMI Level

The above chart explains that CMMI Level 1 and 2 organizations 100% support that incidents require reporting whereas , CMMI level 3, 4 and 5organizations supports up to 92%, 60% and 67% respectively.

4.3.1.3.1.2. ISO Certification

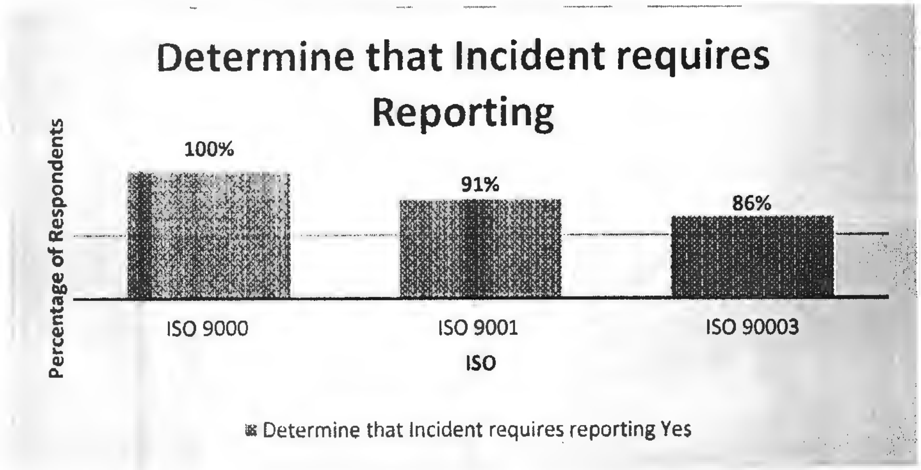

Figure 4.10 Determine that Incident requires Reporting - ISO Certification

The above chart explains that ISO 9000 organizations 100% support that incidents require reporting, however, ISO 9001 and ISO 90003 organizations support up to 91% and 86% respectively.

4.3.1.3.2. Organization Size Analysis

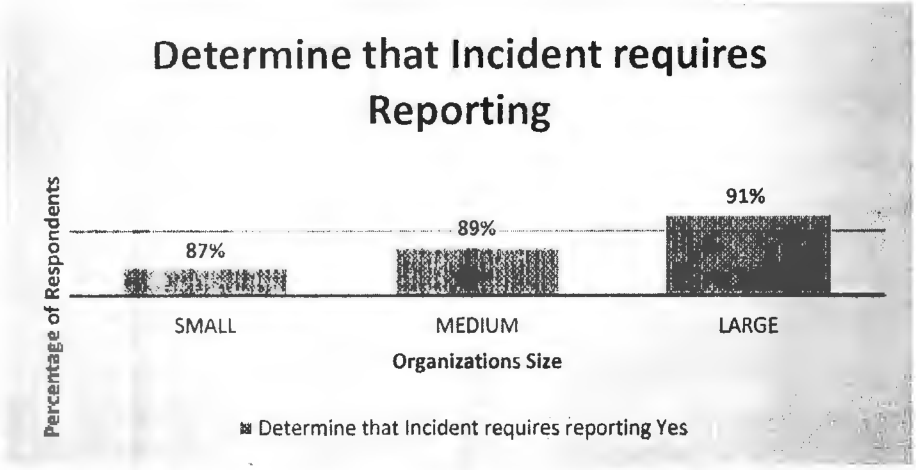

Figure 4.11 Determine that Incident requires Reporting - Organization Size

The above chart explains that 87% of small, 89% of medium and 91% of large size organizations support that incidents require reporting.

4.3.1.3.3. Experience of Respondent (in years) Analysis

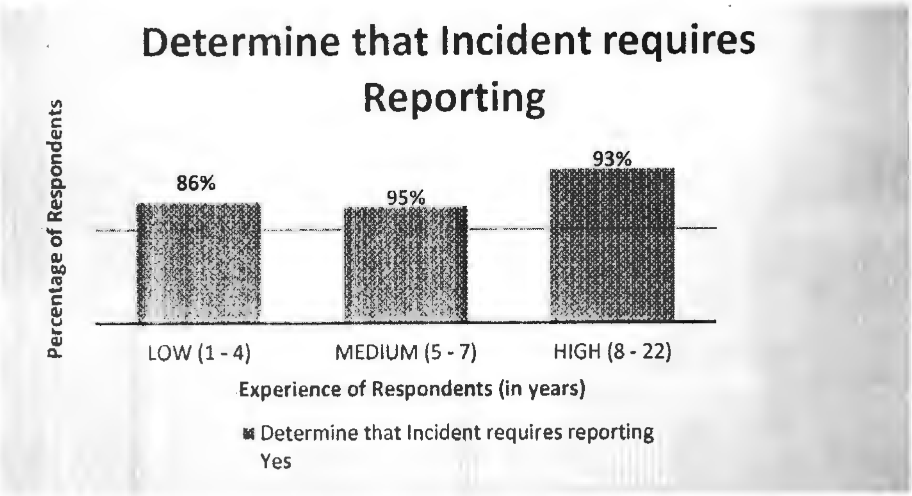

Figure 4.12 Determine that Incident requires Reporting - Respondents' Experience

The above chart explains that respondents with high experience 93% support that incidents require reporting. However, respondents with low and medium experience support up to 86%, and 95% respectively.

4.3.1.4. Assign Action Items for Resolution

4.3.1.4.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.1.4.1.1. CMMI Level

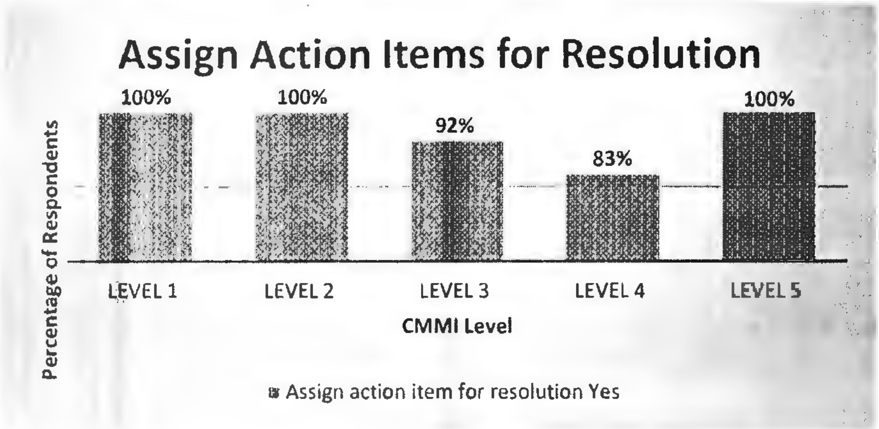

Figure 4.13 Assign Action Items for Resolution - CMMI Level

The above chart explains that CMMI Level 1, 2 and 5 organizations 100% support that action items should be assigned for resolution. Whereas, CMMI level 3 and 4 organizations supports up to 92% and 83% respectively.

4.3.1.4.1.2. ISO Certification

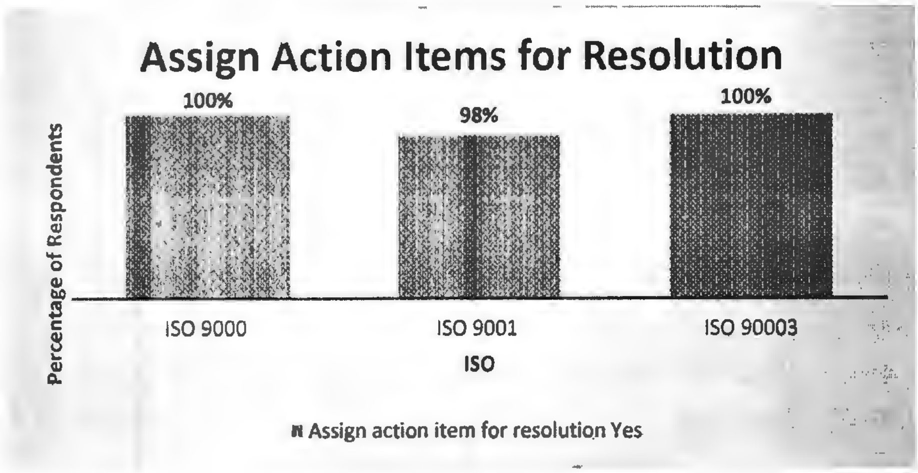

Figure 4.14 Assign Action Items for Resolution - ISO Certification

The above chart explains that ISO 9000 and ISO 90003 organizations 100% support that action items should be assigned for resolution. However, ISO 9001 organizations support up to 98%.

4.3.1.4.2. Organization Size Analysis

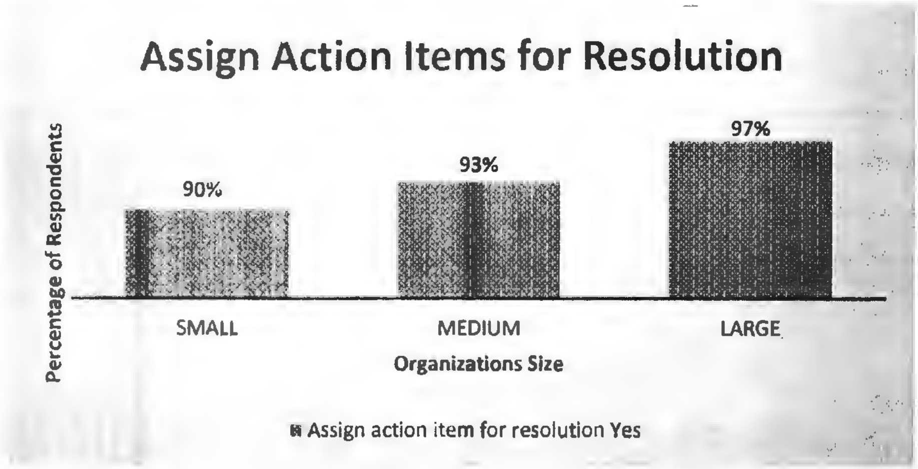

Figure 4.15 Assign Action Items for Resolution - Organization Size

The above chart explains that 90% of small, 93% of medium and 97% of large size organizations support that action items should be assigned for resolution.

4.3.1.4.3. Experience of Respondent (in years) Analysis

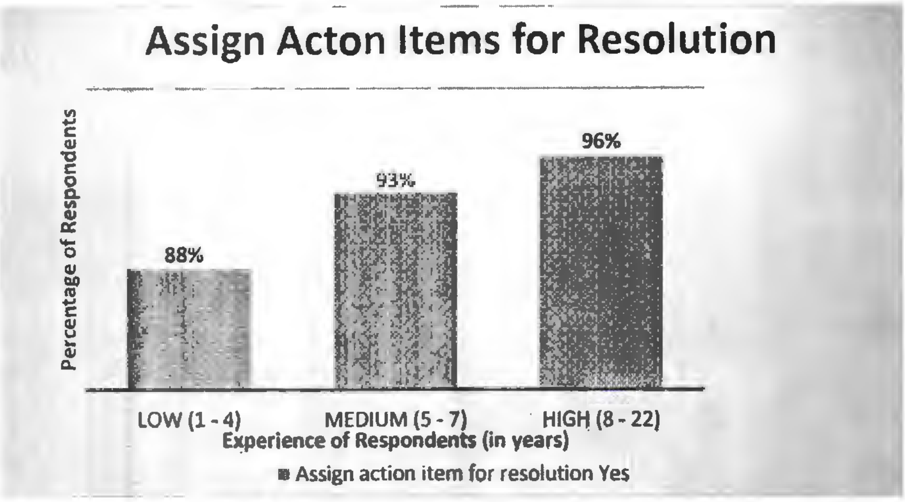

Figure 4.16 Assign Action Items for Resolution - Respondents' Experience

The above chart explains that respondents with high experience 96% support that action items should be assigned for resolution. However, respondents with low and medium experience support up to 88% and 93% respectively.

4.3.2. Create/Update Incidents

A report created / updated in which incidents' occurrence, nature, and status is recorded.

4.3.2.1. *Identify Incidents*

4.3.2.1.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.2.1.1.1. CMMI Level

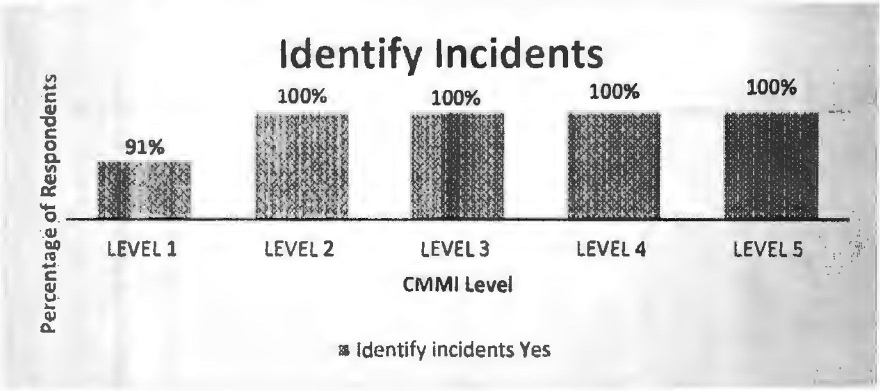

Figure 4.17 Identify Incidents - CMMI Level

The above chart explains that CMMI Level 2, 3, 4 and 5 organizations 100% support identify incidents. However, CMMI level 1 organizations supports up to 91%.

4.3.2.1.1.2. ISO Certification

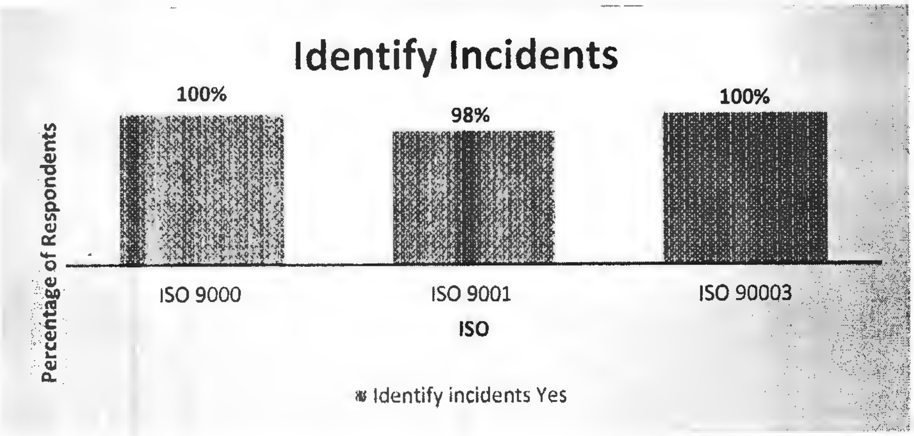

Figure 4.18 Identify Incidents - ISO Certification

The above chart explains that ISO 9000 and ISO 90003 organizations 100% support identify incidents. However, ISO 9001 organizations support up to 98%.

4.3.2.1.2. Organization Size Analysis

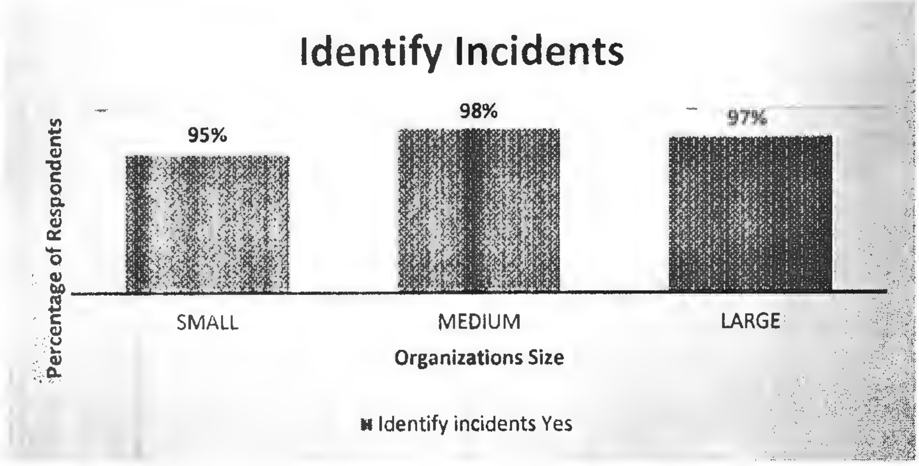

Figure 4.19 Identify Incidents - Organization Size

The above chart explains that 95% of small, 98% of medium and 97% of large size organizations support identify incidents.

4.3.2.1.3. Experience of Respondent (in years) Analysis

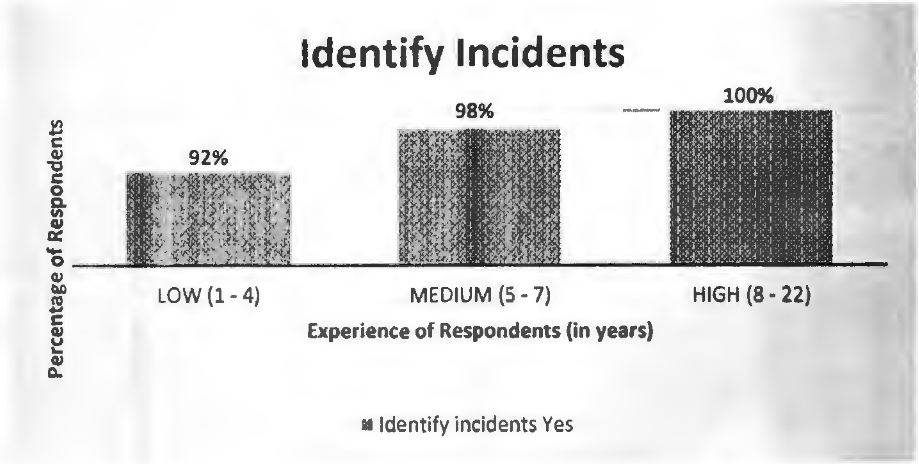

Figure 4.20 Identify Incidents - Respondents' Experience

The above chart explains that respondents with high experience 100% support identify incidents. However, respondents with low and medium experience support up to 92% and 98% respectively.

4.3.2.2. Report Incidents

4.3.2.2.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.2.2.1.1. CMMI Level

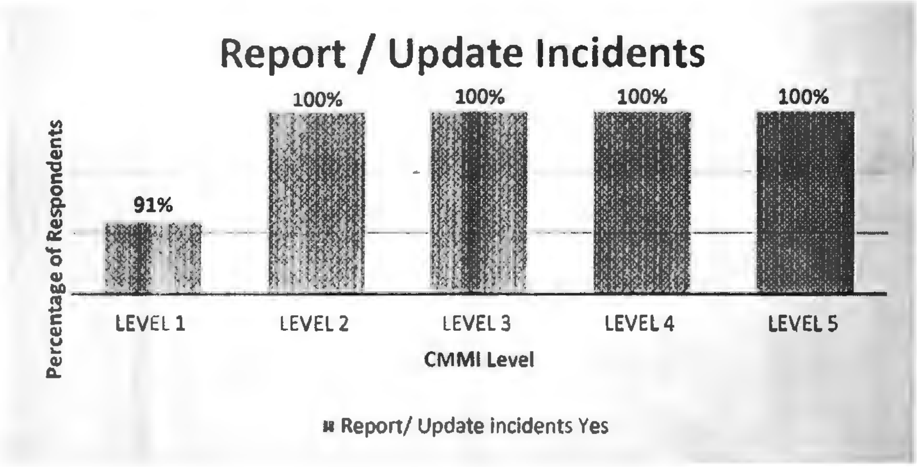

Figure 4.21 Report/Update Incidents - CMMI Level

The above chart explains that CMMI Level 2, 3, 4 and 5 organizations 100% support report/update incidents. Whereas, CMMI level 1 organizations supports up to 91%.

4.3.2.2.1.2. ISO Certification

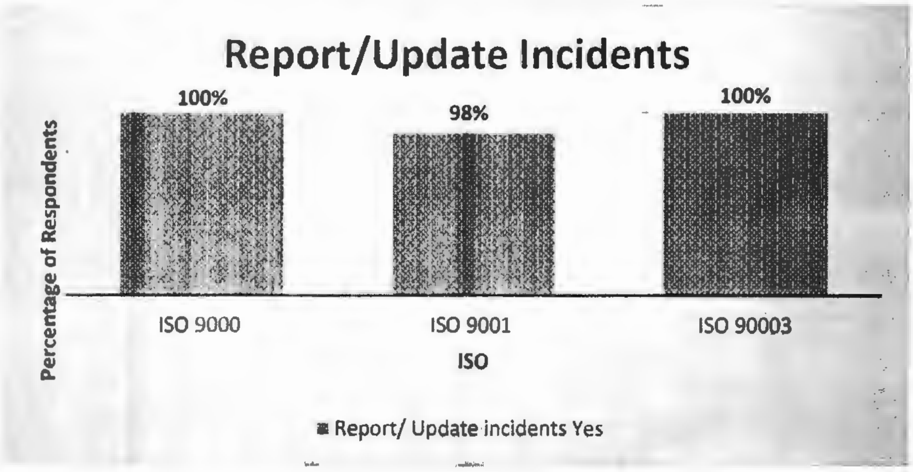

Figure 4.22 Report/Update Incidents - ISO Certification

The above chart explains that ISO 9000 and ISO 90003 organizations 100% support report/update incidents. However, ISO 9001 organizations support up to 98%.

4.3.2.2.2. Organization Size Analysis

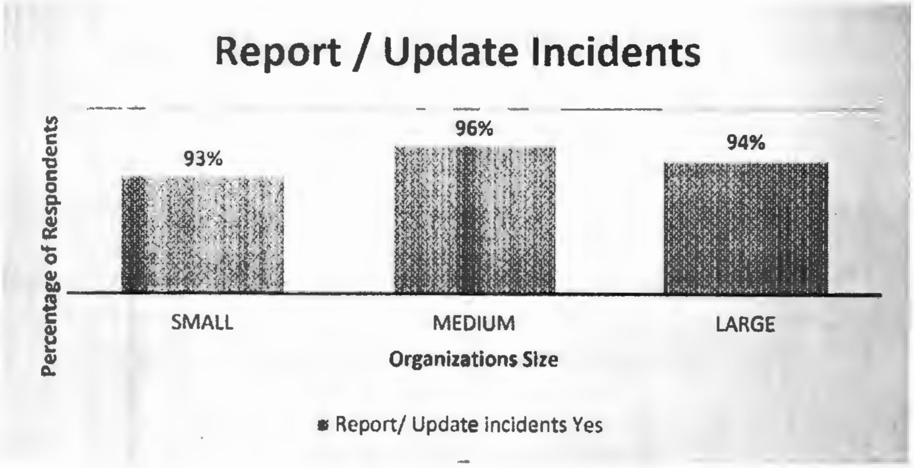

Figure 4.23 Report/Update Incidents - Organization Size

The above chart explains that 93% of small, 96% of medium and 94% of large size organizations support report/update incidents.

4.3.2.2.3. Experience of Respondent (in years) Analysis

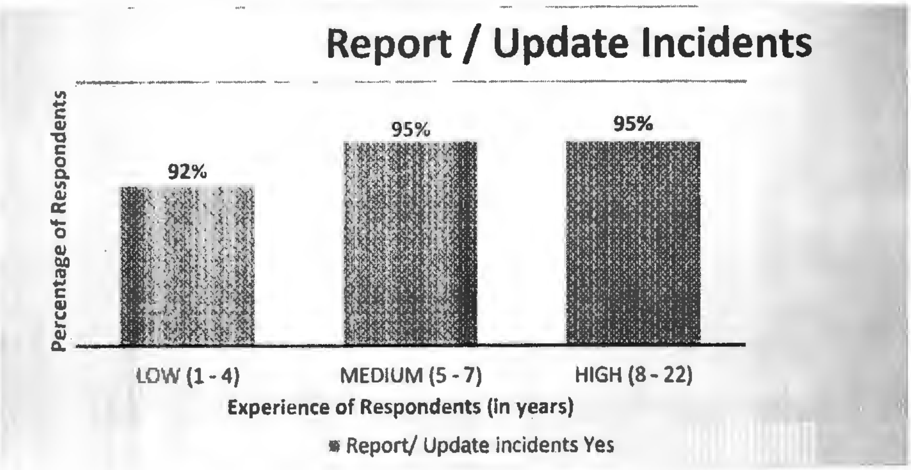

Figure 4.24 Report/Update Incidents - Respondents' Experience

The above chart explains that respondents with high and medium 95% support implementation of report/update incidents. However, respondents with low experience support up to 92%.

4.3.2.3. Quality of Incident Reporting

4.3.2.3.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.2.3.1.1. CMMI Level

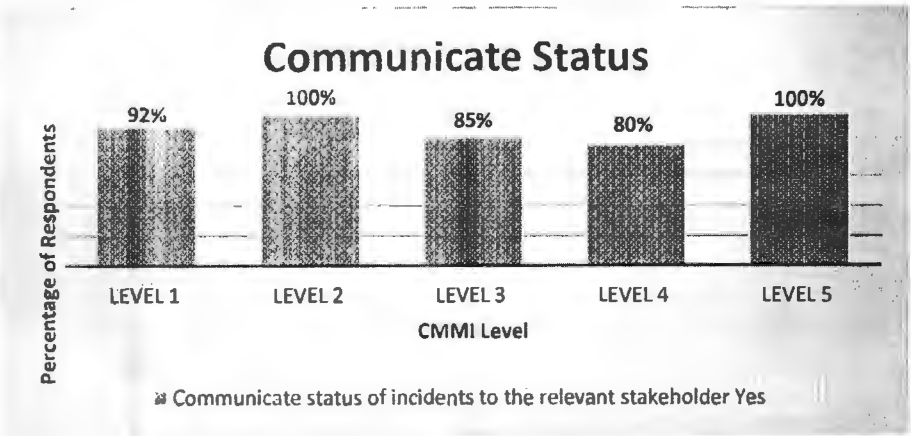

Figure 4.25 Quality of Incident Reporting - CMMI Level

The above chart explains that CMMI Level 2 and 5 organizations 100% support implementation of quality of incident reporting i.e. communicate status of incidents to the relevant stakeholder whereas, CMMI level 1, 3 and 4 organizations supports up to 92, 85% and 80% respectively.

4.3.2.3.1.2. ISO Certification

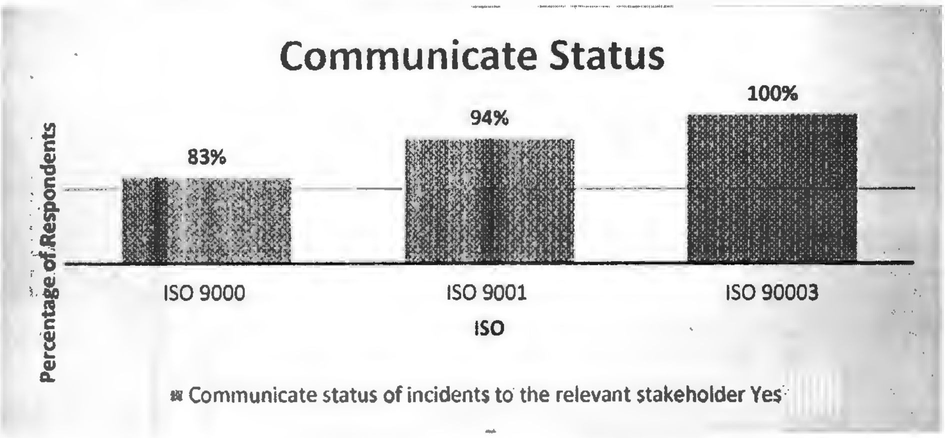

Figure 4.26 Quality of Incident Reporting - ISO Certification

The above chart explains that ISO 90003 organizations 100% support implementation of quality of incident reporting i.e. communicate status of incidents to the relevant stakeholder. However, ISO 9000 and ISO 9001 organizations support up to 83% and 94% respectively.

4.3.2.3.2. Organization Size Analysis

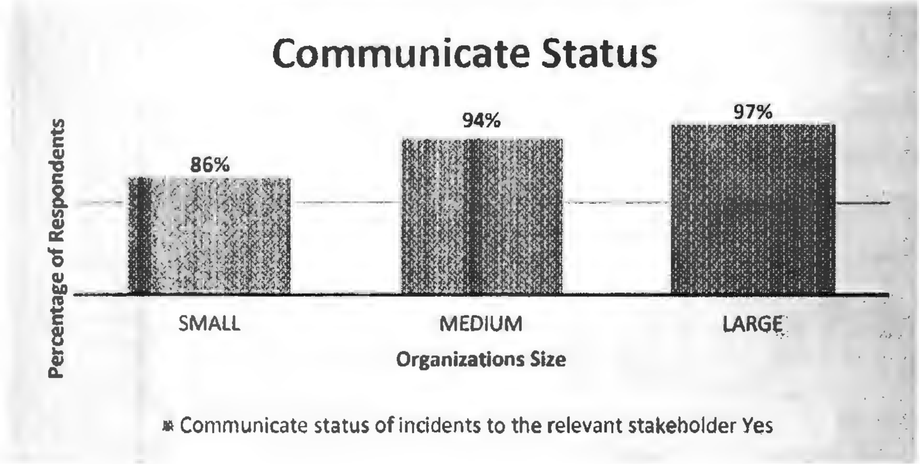

Figure 4.27 Quality of Incident Reporting - Organization Size

The above chart explains that 86% of small, 94% of medium and 97% of large size organizations support implementation of quality of incident reporting i.e. communicate status of incidents to the relevant stakeholder.

4.3.2.3.3. Experience of Respondent (in years) Analysis

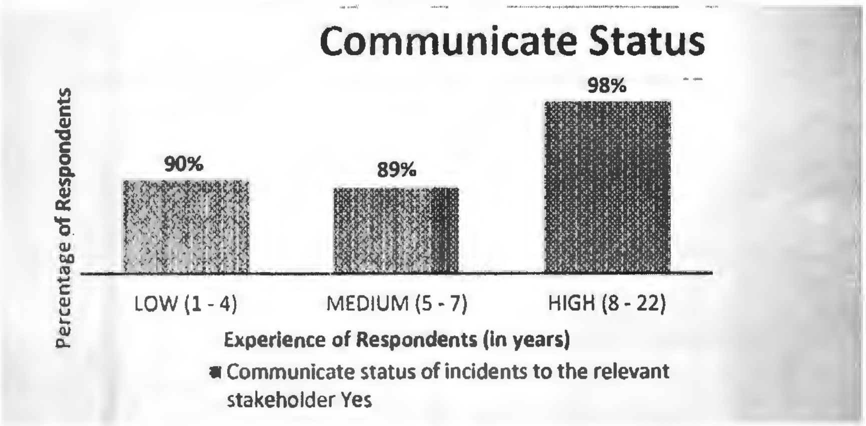

Figure 4.28 Quality of Incident Reporting - Respondents' Experience

The above chart explains that respondents with high experience 98% support implementation of quality of incident reporting i.e. communicate status of incidents to the relevant stakeholder. However, respondents with medium and low experience support up to 89% and 90% respectively.

4.3.2.4. Incident Reporting

4.3.2.4.1. Certification Wise Analysis

Certification wise analysis includes CMMI, ISO and in-house certifications analysis.

4.3.2.4.1.1. CMMI Level

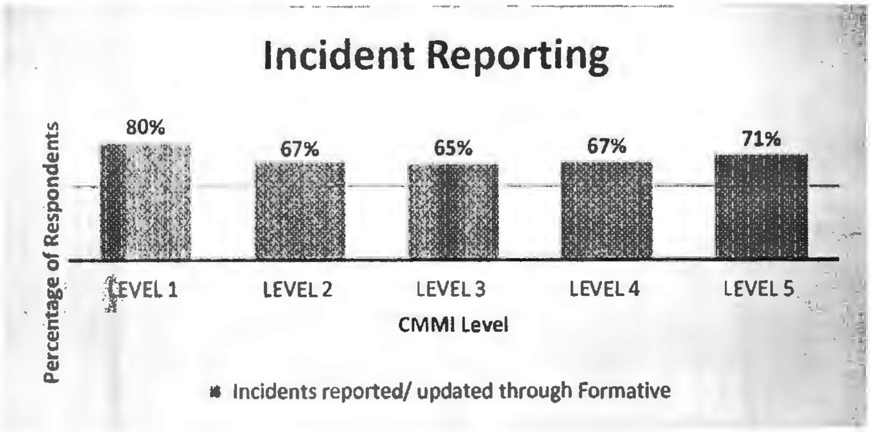

Figure 4.29 Incident Reporting - CMMI Level

The above chart explains that CMMI Level 1, 2, 3, 4 and 5 organizations support Incident reporting through formative evaluation 80%, 67%, 65%, 67% and 71% respectively. Whereas, incident reporting through summative evaluation is supported 20%, 33%, 35%, 33% and 29% by CMMI level 1, 2, 3, 4 and 5 organizations respectively.

4.3.2.4.1.2. *ISO Certification*

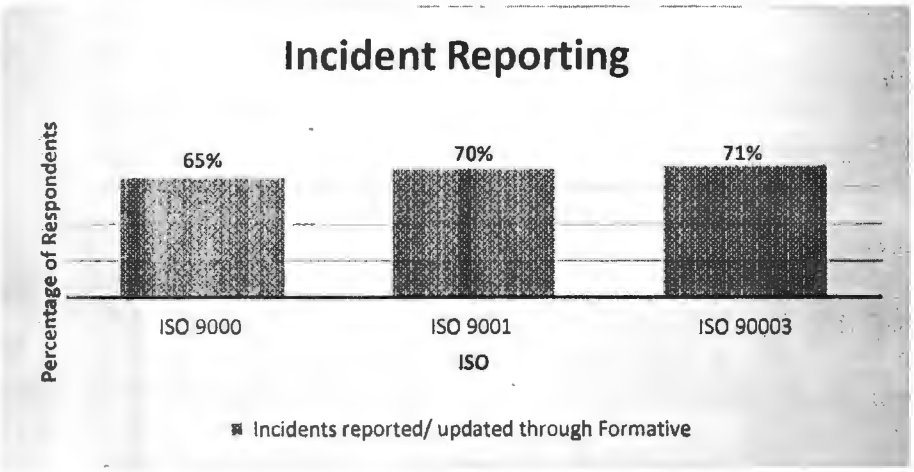

Figure 4.30 Incident Reporting - ISO Certification

The above chart explains that ISO 9000, ISO 9001 and ISO 90003 organizations support incident reporting through formative evaluation 65%, 70% and 71% respectively. Whereas, incident reporting through summative evaluation is supported 35%, 30% and 29% by ISO 9000, ISO 9001 and ISO 90003 organizations respectively

4.3.2.4.2. Organization Size Analysis

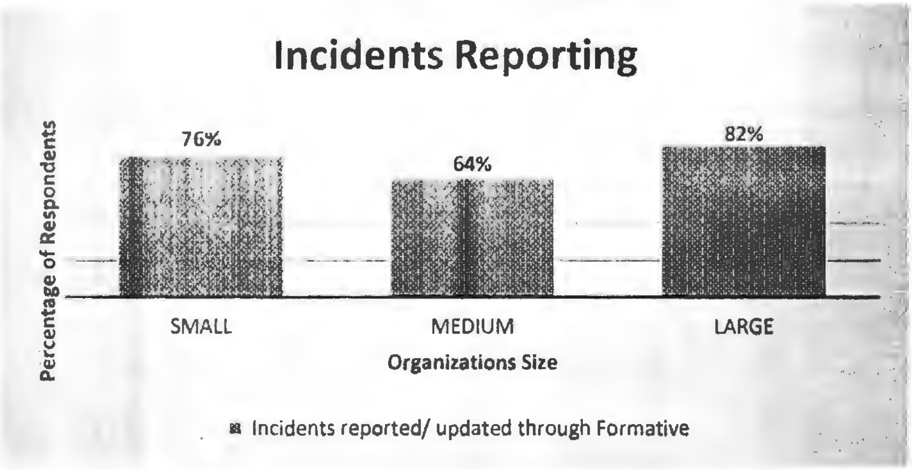

Figure 4.31 Incident Reporting - Organization Size

The above chart explains that 98% of small, 64% of medium and 82% of large size organizations support incident reporting through formative evaluation whereas, incident reporting through summative evaluation is supported 24%, 36% and 18% by small, medium and large size organizations respectively.

4.3.2.4.3. Experience of Respondent (in years) Analysis

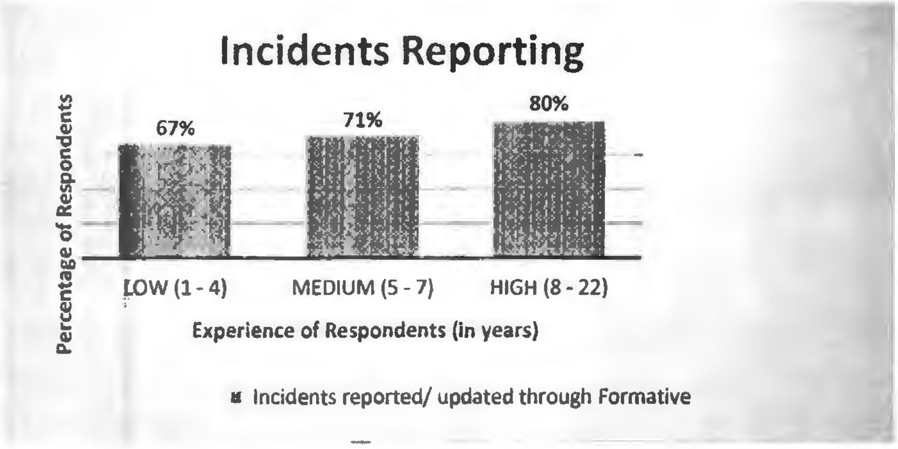

Figure 4.32 Incident Reporting - Respondents' Experience

The above chart explains that respondents with low, medium and high experience support incident reporting through formative evaluation 67%, 71% and 80% respectively. Whereas, incident reporting through summative evaluation is supported 33%, 29% and 20% by low, medium and high experiences respectively.

CHAPTER 5

RESULTS AND FINDINGS

5. Introduction

This chapter highlights proposed guidelines for the development of automated test case incident reporting tools after analysis of survey results by highlighting on each research question. In section 5.1 answer of research question 1 and question 2 is explained that is identification of feature sets and their classification into core phases. In section 5.2, research question 3 is answered that is quality criteria is applied on proposed framework. For this purpose industrial feedback ranges from 40% to 100% is categorized in to high, medium and low parameters.

5.1. Identification of Feature Sets and Core Phases and their Classification

In this section core phases and feature sets are identified and classification of core phases into feature set is discussed. This section highlighted the answers of RQ.1 and RQ.2 which are:

RQ.1. What is the identification of maximum set of feature(s) and core phases of test case incident reporting tool(s)?

RQ.2. What is the classification of feature set in to core phases?

Classification of Feature Sets in to Core Phases	**Core Phase I**				**Core Phase II**			
	Analyze Test Results				**Create/Update Incidents**			
Feature Sets	Analyze Test Results	Update Incidents	Determine that Incident requires Reporting	Assign Action Items for Resolution	Identify Incidents	Report Incidents	Quality of Incident Reporting	Incident Reporting by Formative Evaluation

5.2. Quality Criteria on Proposed Framework

In this section answer of RQ.3. i.e. How can we apply quality criteria on the proposed framework? is discussed. After the identification of feature sets and core phases and their classification quality criteria is applied on the proposed framework to establish credibility of feature sets and core phases. This section suggests that a comprehensive feature set is mandatory for test cases incident reporting. In below tables high, medium, low are the level of support of different organization to apply feature set of test case incident reporting in their organization. Each level have specific ranges i.e.

100 < High> 80

80 < Medium>60

60 < low> 40

5.2.1. Analyze Test Results

It describes the test results, output of test case execution which reveals the difference between actual and expected results.

Core Phase I		**Implication of Quality Criteria on the Proposed Framework**			
		CMMI	**ISO**		**Respondent Experience**
		Level 5	**9001**	**90003**	**High**
Analyze Test Results	Analyze Test Results	medium	high	high	High
	Update Incidents	high	high	high	High
	Determine that Incident requires Reporting	medium	high	high	high
	Assign Action Items for Resolution	high	high	high	high

Table 5.1 Analyze Test Results

Table 5.1 shows that above phase and its feature(s) set was highly endorsed by ISO 9001 and 90003 certified organizations and highly experiences quality engineers. Besides it was highly and medium endorsed by CMMI level 5 organizations. Thus above phase and its feature(s) set integrated in the proposed framework.

5.2.2. Create / Update Incidents

This phase document occurrence, nature, and status of all incidents in a report.

Core Phase II		Implication of Quality Criteria on the Proposed Framework			
		CMMI	ISO		Respondent Experience
		Level 5	9001	90003	High
Create/Update Incidents	Identify Incidents	High	high	high	high
	Report Incidents	High	high	high	high
	Quality of Incident Reporting	High	high	high	high
	Incident Reporting by Formative Evaluation	medium	medium	medium	high
	Incident Reporting by Summative Evaluation	Low	low	low	low

Table 5.2 Create / Update Incidents

. Table 5.2 shows that above phase and its feature(s) set was highly endorsed by highly experiences quality engineers. Besides it was highly and medium endorsed by CMMI level 5 and ISO 9001 and 90003 certified organizations. Thus above phase and its feature(s) set integrated in the proposed framework.

5.3. Proposed Framework

Several number of test case incident reporting tools, which implement a verity of feature, are used in industry to improve the quality of software product. However, feature set of current tools are misalign with current requirement of software industry. This is because current tools do not any framework or a set of guidelines validated as per current needs of software industry. We therefore propose such a framework for test case incident reporting tool(s) whose phases and their feature set are designed based upon current industrial feedback.

Framework for test case incident reporting tool(s) as illustrated in the figure below has core phases which are further explained through feature set that can be implemented for small, medium and large scale organizations. Besides they are evaluated through rigorous quality criteria i.e. highly experienced software and quality engineers and organizations which are CMMI level 5 and ISO certified. Using this framework the developers and organization will choose suitable phase(s) which are further explained through comprehensive feature set for the development of test case incident reporting tools. The proposed framework consists of following phases. The arrows denote the relationship between each phase.

- Testing results obtain from test execution phase are analyzed.
- Incidents report are created/ updated which records nature, status and occurrence of incidents.

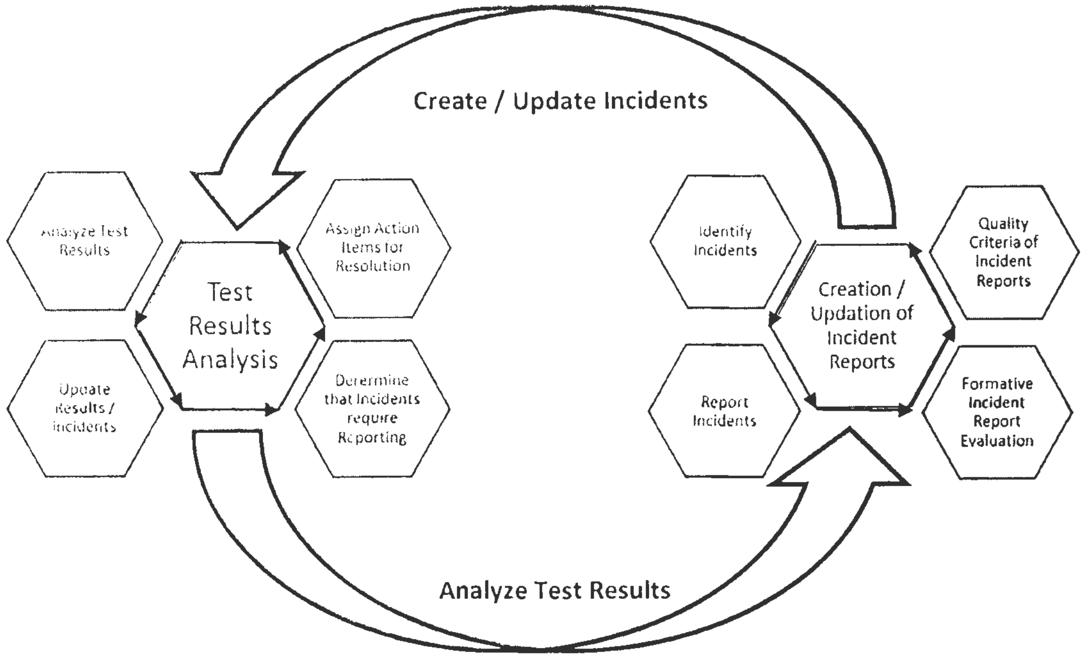
Create / Update Incidents
Analyze Test Results
Assign Action Items for Resolution
Test Results Analysis
Update Results / incidents
Determine that Incidents require Reporting
Identify Incidents
Creation / Updation of Incident Reports
Quality Criteria of Incident Reports
Report Incidents
Formative Incident Report Evaluation
Analyze Test Results

Figure 5.1 Proposed Framework for Test Case Incident Reporting

Such framework will bridge the gap between academia and industry. it provide architectural foundation for test case incident reporting tools in order to define their scope with respect to phases and its feature set instead of implementing features which may belong to various phases.

CHAPTER 6
CONCLUSION

6. Conclusion

Test case incident reporting is an important part of software testing that's why number of tools had been developed to improve quality of software and to minimize cost and efforts. Each tool is different from another as they are not following any set of guidelines and it also takes time and effort to know which tool is the best to use. This study contributes to research by providing a comprehensive framework which improve the test case incident reporting tools development. This framework is derived from detailed literature review and a global survey of software industry so that desired feature the real need of software quality department people can be identified and existing features and frameworks can be evaluated. In addition we have tried to discover the long term vision about automated software test case incident reporting tools. So that software quality engineers can manage and improve their software testing process (tools). After detailed frequency analysis it is concluded that these parameters/ feature sets gains consistent support from the high experienced respondents, CMMI level 3, 4 and 5, ISO 9000 and 9003 certified and large scale organizations so it should be integrated in the proposed framework. This framework bridge up the gap between academia and industry and sets foundation for all those tools which are developed for automated test case incident reporting. It also provides a layout to those who develop automated test case incident reporting tools in future. This will improve the development of test case incident reporting tools.

This research can be extended through development of a rigorous research instrument which involves such commercial tools that are not published in academia. The research work can further be extended through a correlation analysis of phases and its set of parameters against small, medium or large scale organizations, type of certification i.e. CMMI and ISO and respondents experience. Besides it can be validated by developing a tool which is following proposed framework and then it will compare with the existing most common referred tool which can be evaluated based upon quality criteria.

REFERENCES

References

[1] Jingfan Tang, "Towards Automation in Software Test Life Cycle Based on Multi-Agent," *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on* , vol., no., pp.1,4, 10-12 Dec. 2010

[2] Safana, Al; Ibrahim, S., "Implementing Software Test Management Using SpiraTeam Tool," *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on* , vol., no., pp.447,452, 22-27 Aug. 2010

[3] Aljahdali, S.; Hussain, S.N.; Hundewale, N.; Poyil, AT., "Test Management and Control," *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on* , vol., no., pp.429,432, 22-24 June 2012

[4] Vos, T.E.J.; Marin, B.; Escalona, M.J.; Marchetto, A., "A Methodological Framework for Evaluating Software Testing Techniques and Tools," *Quality Software (QSIC), 2012 12th International Conference on* , vol., no., pp.230,239, 27-29 Aug. 2012

[5] Lee, J.; Kang, S.; Lee, D., "Survey on software testing practices," *Software, IET* , vol.6, no.3, pp.275,282, June 2012

[6] Xiaojun Wu; Jinhua Sun, "The Study on an Intelligent General-Purpose Automated Software Testing Suite," *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on* , vol.3, no., pp.993,996, 11-12 May 2010

[7] Hongbo Yu; Yihua Lan; Haozheng Ren, "The Research about an Automated Software Testing System RunTool," *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on* , vol., no., pp.1,4, 28-29 May 2011

[8] Strate, J.D.; Laplante, P.A., "A Literature Review of Research in Software Defect Reporting," *Reliability, IEEE Transactions on* , vol.62, no.2, pp.444,454, June 2013

[9] Maen Hammad; Somia Abufakher; Mustafa Hammad, "A Visualization Approach for Bug Reports in Software Systems." *International Journal of Software Engineering & Its Applications*, vol.8, no.10, pp. 37,46, 2014

[10] Abdou, T.; Grogono, P.; Kamthan, P., "A Conceptual Framework for Open Source Software Test Process," *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual* , vol., no., pp.458,463, 16-20 July 2012

[11] Jiantao Pan, 1999. "Software testing" [online]. Available : http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/, cited on 25th January 2014, at 2:07 pm

[12] IEEE STD 829-2008

[13] Available: http://qatestlab.com/knowledge-center/software-testing-glossary/incident-management-tool/ , cited on 7th February 2014, at 10:30 am

[14] "Software and systems engineering Software testing Part 2: Test processes," ISO/IEC/IEEE 29119-2:2013, pp. 1-68, 2013.

[15] Wohlin, Claes, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012

[16] Whittaker, J.A., *What is software testing? And why is it so hard?* Software, IEEE, 2000. 17(1): p. 70-79.

[17] Pressman, R.S. and D. Ince, *Software engineering: a practitioner's approach*. 2000: McGraw-Hill New York.

[18] Rafi, Dudekula Mohammad, Katam Reddy Kiran Moses, Kai Petersen, and Mika V. Mäntylä. "Benefits and limitations of automated software testing: Systematic literature review and practitioner survey." In *Proceedings of the 7th International Workshop on Automation of Software Test*, pp. 36-42. IEEE Press, 2012.

[19] Abran, Alain, James W. Moore, Pierre Bourque, Robert Dupuis, and L. Tripp. "Guide to the software engineering body of knowledge, 2004 version." *IEEE Computer Society* 1 (2004).

[20] Kasurinen, Jussi, Ossi Taipale, and Kari Smolander. "Software test automation in practice: empirical observations." *Advances in Software Engineering* 2010 (2010).

[21] Vos, Tanja EJ, Arthur I. Baars, Felix F. Lindlar, Peter M. Kruse, Andreas Windisch, and Joachim Wegener. "Industrial scaled automated structural testing with the evolutionary testing tool." In *Software Testing, Verification and Validation (ICST), 2010 Third International Conference on*, pp. 175-184. IEEE, 2010.

[22] Wohlin, Claes, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. "Experimentation in software engineering". *Springer Science & Business Media*, 2012.

[23] Runeson, Per, Martin Host, Austen Rainer, and Bjorn Regnell. "Case study research in software engineering: Guidelines and examples." *John Wiley & Sons*, 2012.

[24] Garousi, Vahid, and Junji Zhi. "A survey of software testing practices in Canada." *Journal of Systems and Software* 86, no. 5, 1354-1376, 2013.

[25] Ng, S.P., Murnane, T., Reed, K., Grant, D., Chen, T.Y.: 'A preliminary survey on software testing practices in Australia'. *Proc. Conf. Australian Software Engineering*, pp. 116–125, 2004.

[26] Pfleeger, Shari Lawrence, and Barbara A. Kitchenham. "Principles of survey research part 2: designing a survey." *Software Engineering Notes* 27, no. 1 (2002): 18-20.

[27] Kitchenham, Barbara A., and Shari Lawrence Pfleeger. "Principles of survey research: part 3: constructing a survey instrument." *ACM SIGSOFT Software Engineering Notes* 27, no. 2 (2002): 20-24.

[28] Pfleeger, Shari Lawrence, and Barbara A. Kitchenham. "Principles of survey research: part 1: turning lemons into lemonade." *ACM SIGSOFT Software Engineering Notes* 26, no. 6 (2001): 16-18.

[29] Kitchenham, Barbara, and Shari Lawrence Pfleeger. "Principles of survey research part 4: questionnaire evaluation." *ACM SIGSOFT Software Engineering Notes* 27, no. 3 (2002): 20-23.

[30] Kitchenham, Barbara, and Shari Lawrence Pfleeger. "Principles of survey research: part 5: populations and samples." *ACM SIGSOFT Software Engineering Notes* 27, no. 5 (2002): 17-20.

[31] Kitchenham, Barbara, and Shari Lawrence Pfleeger. "Principles of survey research part 6: data analysis." *ACM SIGSOFT Software Engineering Notes* 28, no. 2 (2003): 24-27.

[32] J. Ernits *et al.*, "Model-Based Testing of Web Applications Using NModel," *Testing of Software and Communication Systems*, Lecture Notes in Computer Science M. Núñez, P. Baker and M. Merayo, eds., pp. 211-216: Springer Berlin Heidelberg, 2009.

[33] D. Amalfitano "Chapter 1 - Testing Android Mobile Applications: Challenges, Strategies, and Approaches," *Advances in Computers*, M. Atif, ed., pp. 1-52: Elsevier, 2013.

[34] S. Hammond, and D. Umphress, "Test driven development: the state of the practice," in Proceedings of the 50th Annual Southeast Regional Conference, Tuscaloosa, Alabama, pp. 158-163, 2012.

[35] Fomin, S., "Test management with Testopia — missing link?," *Software Engineering Conference in Russia (CEE-SECR), 2009 5th Central and Eastern European* , vol., no., pp.253,258, 28-29 Oct. 2009

[36] Safana, Ahmed Ibrahim, and Suhaimi Ibrahim. "Implementing Software Test Management Using SpiraTeam Tool." In *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*, pp. 447-452. IEEE, 2010.

[37] Khandkar, Shahedul Huq, S. M. Sohan, Jonathan Sillito, and Frank Maurer. "Tool support for testing complex multi-touch gestures." In *ACM International Conference on Interactive Tabletops and Surfaces*, pp. 59-68. ACM, 2010.

APPENDIX

Appendix A

Survey to find the Guidelines for the Development of Automated Test Case Incident Reporting Tool(s)

I am Bushra Areeb Fatimah, Reg # 366-FBAS/MSSE/F12, bushra.msse366@iiu.edu.pk, from International Islamic University Islamabad. This questionnaire is part of my MS research thesis in (Software Engineering) under the supervision of Dr. Rizwan Bin Faiz, rizwan.faiz@riphah.edu.pk, who is a Higher Education Commission (HEC) Approved Supervisor. He is currently working as an Assistant Professor in Faculty of Computing in RIPHAH International University, Islamabad.

As part of our research this questionnaire is requested to be filled up by various software development organizations across the globe. We confirm that all the information provided by respondents will only be used for research purposes and will not be disclosed or shared with any individual or organization under all circumstances.

The objective of research is to propose minimum set of parameters that should be implemented by Automated Test Case Incident Reporting Tool. Automated Test Case Incident Reporting is a process that records and tracks the status of incidents; occur during the automatic execution of the test cases. It often has opportunities of recording, fixing, re-testing of incidents, and reporting capabilities.

Respondent Introduction

1. Name of Respondent

2. Gender of Respondent*Required

○ Male
○ Female

3. Designation of Respondent*Required

4. Experience of Respondent (in years)*Required

5. E-mail of Respondent

Organization Information

6. Name of your organization:

7. Name of country in which your organization is located*Required

8. Number of employees working in your organization:*Required

Choose an item.

9. Number of people working in quality department:* Required

Choose an item.

10. Which among below mentioned quality standards does your organization belongs to?

10.i. CMMI process maturity level

Choose an item.

10.ii. ISO standard

Choose an item.

10.iii. Any other please specify

Test Case Incident Reporting Tool

11. Which automated tool(s) does your organization use for test case incident reporting?

□ 1. Jmeter

□ 1.1. Jmeter 2.9

□ 1.2. Jmeter 2.10

□ 1.3. Jmeter 2.11

□ 2. Selenium

□ 2.1.Selenium 2.30

□ 2.2. Selenium 2.40

□ 2.3. Selenium 2.50

□ 3. FitNesse

□ 3.1. FitNesse 20140418

□ 3.2. FitNesse 20140201

□ 3.3. FitNesse 20131110

□ 4. Squish

□ 4.1. Squish 5.0.2

□ 4.2. Squish 5.0.3

□ 4.3. Squish 5.1

□ 5. Bugzilla Testopia

□ 5.1. Bugzilla Testopia 2.3

□ 5.2. Bugzilla Testopia 2.4

□ 5.3. Bugzilla Testopia 2.5

□ 6. Spira Test

□ 6.1. Spira Test 4.0

□ 6.2. Spira Test 4.2

Test Case Incident Reporting Tool

12. Please specify any other tool and its version used for test case incident report in your organization?

12.i. What is the type of automated test case incident reporting tool(s) as specified in Q.11.?

Choose an item.

12.ii. At which level of testing your organization used that tool as specified in Q.11.

Choose an item.

12.iii. Which type of testing is performed using that tool as specified in Q.11 ?

□ 1. Functional Testing

□ 2. Performance Testing

□ 2.1. Load Testing

□ 2.2. Stress Testing

□ 2.3. Volume Testing

□ 3. Regression Testing

□ 4. Acceptance Testing

□ 4.1. Alpha Testing

□ 4.2. Beta Testing

□ 5. Security Testing

□ 6. Usability Testing

□ Other: □

How are Incidents reported in your organization?

Below are few possible activities through which test case incident reports can be created. Each activity involves certain set of tasks through which the activity can be accomplished.

Analyze Test Results

(Test result is output of test case execution which manifests the difference between actual and expected results.)

13. How are test results analyzed in your organization?

	Yes	No
Analyze test results	○	○
Update incidents	○	○
Determine that Incident requires reporting	○	○
Assign action item for resolution	○	○

Please specify any other way of analyzing test results?

□

Create/Update Incident Reports

(Incident Report is the documentation of the occurrence, nature, and status of an incident)

14. How are incident reports created/updated in your organization?

	Yes	No
Identify incidents	○	○
Report/Update incidents	○	○
Communicate status of incidents to the relevant stakeholder	○	○

14.i. How are incidents reported/update in your organization?

○ Through summative evaluation (Critical incidents can be collected which will enable us to make statements such as "x percent of the users found feature y in context z was helpful/ unhelpful."

○ Through formative evaluation (Contextual data can be collected around each incident so that the tester can place the critical incidents in scenarios)

Please specify any other way of creating/updating incident reports?

Suggestions for Improving Questionnaire

Please suggest how can we improve our questionnaire?

Confirmation Page