مستخلص

هنالك العديد من الدراسات التي أجريت في مجال ضبط ومراقبة جودة البرمجيات في المؤسسات الصغيرة والمتوسطة في السودان وخلصت الي أن جميع المؤسسات الوطنيه لا تعتمد علي معايير واضحة لضبط ومراقبة جودة منتجاتها الشيء الذي يؤثر سلبا علي أداء وفعالية المنتجات البرمجية في فترة ليست بالطويلة.

تهدف الدراسة الحاليه الي تطوير منهجية بسيطة فعاله وغير مكلفة لإدارة المشاريع وضبط جودة البرمجيات في المؤسسات الصغيرة والمتوسطة المتخصصة في مجال تطوير البرمجيات في سوق العمل السوداني بصورة خاصة والدول الناميه بصورة عامة وذلك لان الشركات الصغيره والناشئة في هذه المنطقة تواجه نفس المشاكل والتحديات.

منهجيه الحل تم تنفيذها في ثلاث محاورالمحور الاول، وضعت الدراسه عده فرضيات لحل المشكله وتتمثل في حزمة من الحلول المقترحة تغطي كافة الانشطه في دورة حياه تطوير أي منتج برمجي، تم نشر إستبيان وسط قطاعات المطورين، مدراء المشاريع والباحثين في مجال هندسة البرمجيات استهدف خمسين عينه لاثبات صحة الفرضيات ومن ثم تم تحليل البيانات بناء علي مقياس ليكارت الخماسي للوصول الي نتائج واضحة ودقيقة تساعد في تصميم منهجية الحل المقترحة.

المحور الثاني، المنهجية المقترحة تم فيها دمج معيار ISO/IEC 25010:2011 لتقييم وتحسين جودة البرمجيات مع نشاطات وفعاليات المستوي الثاني من نموذج CMMI "إدارة المشاريع" فضلا عن مجموعة من الانشطه. يتم تنفيذ المنهجيه المقترحه خلال تطوير اي منتج في مسارين، المسار الاول يعرف "مسؤوليات العميل" يحدد كل التزامات العميل من اول مرحله في دورة حياه تطوير اي منتج وحتي استلام المشروع أو النظام المقترح، المسار الثاني يعرف "مسؤوليات مدير المشروع" يحدد بصورة دقيقة مسؤوليات مدير المشروع أو المؤسسة المسئولة من تطوير النظام خلال كل مراحل التطوير، تختصر دورة حياة تطوير البرمجيات في ثلاث مراحل أساسيه المرحلة الاولي، جمع البيانات وتشمل كتابه تقريرين الاول للمتطلبات الوظيفية وغير الوظيفية والثاني يستخدم لتحديد حدود المشروع التقنيه المرحلة الثانيه مرحلة التطوير وتتخللها ثلاث مراحل فرعية (تصميم، كتابه كود، الإختبار) المرحلة الاخيره المراجعه والتسليم.

المحور الثالث، يتمثل في إثبات قدرة وفعاليه الحلول المقترحة لحل المشكله وقياس مستوي نجاح الدراسه في تحقيق أهدافها، تم إرسال المنهجية المقترحة لاختبارها في بيئة عمل واقعيه ومن ثم تم نشر استبيان ثاني للحصول علي نتائج الاختبار وكانت النتائج جيدة جداً حيث أثبتت ان المنهجية المقترحة قابلة للتطبيق وتلبي احتياجات المؤسسات الصغيرة والمتوسطة، النتائج منشورة في الباب الخامس.

في الختام وضعت الدراسه توصياتها فيما يتعلق بالحلول والمنهجيه المقترحة لحل المشكله وجدوي تطبيقها وطرحت أسئله بحثيه جديده كمقترحات للدراسات اللاحقة في المجال كما ناقشت بعض التحديات التي واجهة الدراسة وكيفيه تجاوزها مستقبلاً.

Abstract

There are many studies that have been conducted ability to apply and monitor software quality standards in small and medium enterprises in Sudan and concluded that whole the national institutions do not follow clear criteria to applying and monitoring software quality standards of its products, that thing affect negatively on their products effectiveness.

The current study aims to develop a simple, efficient and inexpensive methodology for project management and software quality control in small and medium-sized enterprises specialized in the software development in Sudan's market particular and Middle East generally, because small and medium companies in this region are facing the same problems and challenges.

Proposed methodology aim to merge ISO / IEC 25010: 2011 for software quality requirement and evolution, is considered the latest standards published by International Quality Organization with activities of CMMI Level II "project management", shorten software development life cycle in three stages, initial stage for data collection, which is written reports the first one "functional and non-functional requirements report" the second report " technical project scope", development stage in three sub-phases as (design - writing code - test) , the third stage review and delivery stage, where review and test all system components with customer to make sure the whole component are working correctly and without errors Based on customer requirements and expectations. The implementation of the whole project development stages will be done in two tracks, the first track known "customer responsibility" determined customer responsibility from the first stages of development life cycle until delivered project or proposed system, the second track known "supplier responsibility" defines project manager responsibility or supplier, in each stage provide documents and reports to adjust the development process and ensure its implementation in standard.

Proposed methodology solve the problem in three basic axes, first axis prove some hypotheses will use to design proposed methodology using a questionnaire, second axis was developed proposed methodology, represents a set of solutions to problem study, third axis aim sent proposed methodology to a set of institutions for testing in a real environment and then publish questionnaire to get results.

Acknowledgments

I would like to express my gratitude to my supervisor ***Dr. Osama Mohamed Ibrahim*** for the useful comments, remarks and engagement through the learning process of this master thesis. Furthermore I would like to thank ***Arafat M. abd-algader*** for introducing me to the topic as well for the support on the way. Also, I like to thank the participants in my survey, who have willingly shared their precious time during the process of interviewing. I would like to thank my loved ones, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever.

Table of Contents

Abstract I

Acknowledgment II

List of Tables III

list of Figures IV

list of Appendix V

List of abbreviations VI

1. Chapter (1) - Introduction

1.1 Introduction 2

1.2 Research problem 3

1.3 Research objective 3

1.4 Research methodology 3

1.5 Research scope 4

1.6 Motivation of study 5

1.7 Research question 5

1.8 conclusion 6

2. Chapter (2) – literature review

2.1 Introduction 8

2.2 Basic concept of software engineering 8

2.3 Why we need software quality 9

2.4 ISO Software quality process improvement standers 11

2.5 Overview about capability maturity model (CMMI) 15

2.6 Previous study about applicable SPI in SMEs 16

2.7 Conclusion 27

3. Chapter (3) – literature review

3.1 Introduction 29

3.2 Purpose of the current study 30

3.3 Type of scientific research 30

3.4 Research methodology 31

3.5 Type of research methodologies 32

3.6 Methodology of the current study 34

3.7 Conclusion 39

4. Chapter (4) – Proposed methodology

4.1 Introduction 42

4.2 The main idea 43

4.3 Overview about CMMI level II 43

4.4 Most importance project management methodologies 44

4.5 Overview about proposal works 45

4.6 Main features of the proposal methodology 52

4.7 proposal works 53

4.8 Conclusion 75

5. Chapter (5) – Result, Discussion and validation

5.1 Introduction 77

5.2 Questionnaire number one 78

5.2.1 Degree of internal consistence and reliability 78

5.2.2 Questionnaire number one – summary 97

5.3 Questionnaire number two 102

5.3.1 Degree of internal consistence and reliability 102

5.3.2 Questionnaire number one – summary 126

5.4 Conclusion 132

6. Chapter (6) – Conclusion and future woke

6.1 Introduction 134

6.2 General – effort of study 134

6.3 Limitation 136

6.4 Recommendation 137

6.5 Future work 139

6.6 Conclusion 140

7. References 142

8. Appendices 146

List of table

Table name	page
Table 1 : Full time staff number	17
Table 2 : knowledge rate of SPI	18
Table 3 : knowledge rate of SPI standers	18
Table 4 : Knowledge rate of CMM	19
Table 5 : common problem experienced while implementing SPI standards for a large organization	20
Table 6 : Intention to follow different level of CMM	21
Table 7 : staff position in the organization	22
Table NO (5-1)	80
Table NO (5-2)	81
Table NO (5-3)	83
Table NO (5-4)	84
Table NO (5-5)	85
Table NO (5-6)	86
Table NO (5-7)	87
Table NO (5-8)	88
Table NO (5-9)	89
Table NO (5-10)	90
Table NO (5-11)	91
Table NO (5-12)	92

Table NO (5-13)	93
Table NO (5-14)	93
Table NO (5-15)	94
Table NO (5-16)	95
Table NO (5-17)	96
Table NO (5-18) Questioner number one – Hypotheses1	97
Table NO (5-19) Questioner number one – Hypotheses2	98
Table NO (5-20) Questioner number one – Hypotheses3	99
Table NO (5-21) Questioner number one – Hypotheses4	100
Table NO (5-22)	104
Table NO (5-23)	105
Table NO (5-24)	106
Table NO (5-25)	107
Table NO (5-26)	108
Table NO (5-27)	109
Table NO (5-28)	110
Table NO (5-29)	111
Table NO (5-30)	112
Table NO (5-31)	113
Table NO (5-32)	114
Table NO (5-33)	115
Table NO (5-34)	116

Table NO (5-35)	117
Table NO (5-36)	118
Table NO (5-37)	119
Table NO (5-38)	120
Table NO (5-39)	121
Table NO (5-40)	122
Table NO (5-41)	123
Table NO (5-42)	124
Table NO (5-43) Questioner number two – Hypotheses1	125
Table NO (5-44) Questioner number two – Hypotheses2	126
Table NO (5-45) Questioner number two – Hypotheses3	127
Table NO (5-46) Questioner number two – Hypotheses3	128

List of Figures

Figure name	page
Figure NO (5-1)	80
Figure NO (5-2)	81
Figure NO (5-3)	82
Figure NO (5-4)	83
Figure NO (5-5)	84
Figure NO (5-6)	85
Figure NO (5-7)	86
Figure NO (5-8)	87
Figure NO (5-9)	88
Figure NO (5-10)	89
Figure NO (5-11)	90
Figure NO (5-12)	91
Figure NO (5-13)	92
Figure NO (5-14)	93
Figure NO (5-15)	94
Figure NO (5-16)	95
Figure NO (5-17)	104
Figure NO (5-18)	105
Figure NO (5-19)	106
Figure NO (5-20)	107

Figure NO (5-21)	108
Figure NO (5-22)	109
Figure NO (5-23)	110
Figure NO (5-24)	111
Figure NO (5-25)	112
Figure NO (5-26)	113
Figure NO (5-27)	114
Figure NO (5-28)	115
Figure NO (5-29)	116
Figure NO (5-30)	117
Figure NO (5-31)	118
Figure NO (5-32)	119
Figure NO (5-33)	120
Figure NO (5-34)	121
Figure NO (5-35)	122
Figure NO (5-36)	123
Figure NO (5-37)	124
Figure NO (5-38)	125
Figure NO (5-39)	126
Figure NO (5-40)	127

List of Appendix

Appendix name	page
Appendix 1 : Proposed methodology	147
Appendix 2 : Software requirements specification (SRS)	148
Appendix 3 : simple software requirements specification	157
Appendix 4 : technical project scope form	158
Appendix 5 : Security report form	167
Appendix 6 : Checklist review form	168
Appendix 7 : project sign-off sheet	169
Appendix 8 : Questionnaire number one	170
Appendix 9 : Questionnaire number two	173

List of abbreviations

IEEE	Institute of Electrical and Electronics Engineers
ICCEEE	International conference on computer electrical and electronic engineering
ISO	International Organization for Standardization
QA	Quality assurance
QC	Quality control
SPI	Software process improvement
BSI	British Standards Institute
ANSI	American National Standards Institute
CSA	Canadian Standards Association
IEC	International Electrotechnical Commission
SQuaRE	Software product Quality Requirements and Evaluation
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
SEI	Software Engineering Institute
SPICE	Software Process Improvement Capability Determination
SME	small and medium enterprises
SRS	software requirements specification

CHAPTER (1)
INTRODUCTION

1.1 Introduction

Software production in Sudan is very limited and the percentage is about 2% of total market sales, 80% of the companies have small resources and have about 2-5 staff members [5] these factors affect to apply software processes improvement or software process improvement standards affected negatively on the quality of their products.

There is very importance software process improvement and software process standard must be available in a clear and simple way for whole team members and added to plans of each software development companies. However, the existing small and medium software development companies do not consider these standards. The main motivations of this research study the problem and propose suitable solutions.

Most of the software developers in small companies believe the software process improvement and software process standard make their job more complex. This is due to many reasons firstly most of these models are developed to meet the needs of organizations in different market and different environments and they do not consider the characteristics of small companies. Secondly, the number of staff member assigned to each project is less than the number of staff that assumed in these models. finally, the Sudanese software companies do not have staff members for quality insurance and the successful of developed software is depend on the experiences of software developers [5].

The existing literature shows a strong need to new technique or method for Software quality improvement and project management tack into account the characteristics and requirements of Sudanese software companies. Moreover, this study is going to propose a simple, efficient and inexpensive methodology aim to help software developer and project managers in Sudanese companies to guarantee success of software quality for their products.

1.2 Research problem

The existing Sudanese software development companies or small and medium enterprise generally lack to clear software process improvement and software process improvement standards used to evaluate and develop their software products.

1.3 The research objectives

this study that aim to develop a proposed methodology simple, efficient and inexpensive for project management and software quality control in small and medium-sized enterprises specialized in software development field meet their needs and use to improves their products quality, improve their team performance reflected on the time and effort required to accomplish any project, to ensure the effectiveness of the proposal solution must include the following criteria:

1- Support the best practices for software development activity.
2- Should be able to deal with various and modern development technology.
3- Provide simple and efficient documents for each step to keep progressing process standard.
4- Find easy way to convince the top management to integrate software quality standards within their institutions routine.
5- Ensures the basic software project management practices such as Project planning, Project scheduling, Risk management and management people.
6- Specify relationship between customer and project manager and roles of each one of them at the whole software development stages.

The above objectives represented a set of activities ensures to apply software process improvement and software process improvement standards on small and medium enterprise that reflected to improve their software products.

1.4 Research methodology

Take advantage of the results for previous studies as scientific and real beginning that find out the reasons about, why the software development

companies in Sudanese market does not interested to apply the software process improvement and software process improvement standard on the other hands studies try to develop a solution for the current problem .

The main stage at this study develop a methodology according to data must be collected from viewpoint of researchers, developers, project managers and each one related to software development process on the target institutions base on the basic and traditional model for software development process, collect and analyze data about time and cost estimation, design, coding, testing and whole development life cycle stages until deliver the software product to find out clear outlines about proposed solution using questionnaire called "*questionnaire number one*".

Develop proposal methodology according to data analysis and result from questionnaire number one, solutions will be tested on the real development environment on the target institutions and companies, before that should be training one employee or more at least one or two hour about how to use proposed methodology.

Questionnaire number two, using to collect data from viewpoint of project managers, system engineers, designers and developers about the effectiveness of solutions and its ability to solve the problem of their institutions and then analyzing data to got clear and accuracy results represent s a summary for study allow us to determine recommendations and how to applying software quality standards on small and medium enterprises.

1.5 Problem scope

Development a simple, efficient and inexpensive methodology includes the main activity for Project Management and software quality such as project schedule, time and cost estimation, risk management, people management, quality improvement and define clearly relationship among members of team or relationship between project manager and customer during whole the development life cycle stages used on small and medium enterprise specializing in software development on the Sudanese market.

1.6 Motivation of study

There are many factors that influenced to choice this study in this area. Where the small and medium enterprises seeking to profit quickly and reduce the expenditure for limited resources as well as lack of full-time staff to implement and monitor the software process improvement or software process improvement standards. We can summarize the most important factors that make difficult to apply these software quality standards at the national market base on study published on 2013 [5] as follows:

1. Lack small and medium enterprise for staff with experience, knowledge and specialists in the software quality standards and on the other hand it is difficult allocation staff for this mission because there are limited number of staff in small companies.

2. Top management's lack of knowledge in relation to the applying the software process improvement and software process improvement standards.

3. Lack of cooperation between the customers and project owners to presence and provide their views and suggestions as one of the most importance project management practices.

4. Project size and complexity, budget allocated, number of team members that involved to development process all these factors that directly affect on to apply software standards.

The above problems and challenges faced small and medium enterprises in the field of software development will be attention greatly in this study and try to develop solutions simple, efficient, inexpensive, applicable and positively affect the development process.

1.7 Research questions

Current study focused on addressing vital and very important subject in relation to apply software quality in small and medium-sized enterprises, raise the quality of their products to competition strongly in the field of software development with great institutions, study asked several questions as follows :

1. It is important to apply software process improvement and software process improvement standards in small and medium-sized enterprises or small and medium projects.

2. There are many project management methodologies and software quality standards, are whole these standards suitable to apply in small and medium-sized enterprises.

3. Possibility to develop a methodology for project management and software quality standard main characteristic is simple, effective and inexpensive meet needs of small and medium-sized enterprises and solve problems and challenge they face.

4. It is important to integrate software quality standard, project management with each other to take advantage of their different characteristics.

Study will be answer the above questions represent as simple contribute, support scientific research in the field of software quality control, provide a serious and effective solutions to overcome challenges and problems faced small enterprises and medium-sized particularly in the Sudanese market.

1.8 Conclusion

By the end of this chapter, which represent s the chapter one of this study that aim to develop a proposed methodology for project management and software quality control in small and medium-sized enterprises that work in the field of software development meet their needs and solve whole problems and challenges they face content, firstly discussed the research problem, why the small and medium enterprises do not apply software process improvement and software process improvement standards, research objectives aims to develop a clear mechanism to solve the problem, research methodology, ways or mechanism will be used during the study to resolve the problem, problem scope, is the small and medium-sized companies that work in the developing software finally the research questions.

CHAPTER (2)

LITERATURE REVIEW

2.1 Introduction

Software in last few decades become depend it mainly in our live, so now the software production is not like Traditional and mere programming activities become engineered under the specific practice and standards to improve the quality of a software and development effort.

This chapter summarizes. what is the software quality standers, why we need software quality standers and why we need to be sure and control software quality, the most important software quality standers for assessment and improvement, the most important practices and maturity model, Previous studies discussed problems for applying software quality standards in small and medium enterprises and studies have attempted to develop solutions for applying software standards in small and medium businesses.

2.2 Basic concepts of the software engineering

There is no specific definition of software quality to define software quality we need to define the expected characteristics or properties of high-quality software to do that we need to: [1]

- Examine the different perspectives and expectations of users as well as other people involved in. the development, management, marketing, and maintenance of the software products.
- Examine the individual characteristics associated with quality and their inter-relationship, and focus our attention on the critical characteristics of functional correctness.

We next examine the different views of quality in a systematic manner, based on the different roles, responsibilities, and quality expectations of different people, and zoom in on a small set of views.

There are five major views [22] transcendental, user, manufacturing, product, and value-based views, as outlined below:

In the ***transcendental*** view, quality is hard to define or describe in abstract terms, but can be recognized if it is Percent. It is generally associated with some intangible properties that delight users.

In the ***user*** view, quality is fitness for purpose or meeting user's needs.

In the ***manufacturing*** view, quality means conformance to process standards.

In the ***product*** view, the focus is on inherent characteristics in the product itself in the hope that controlling these internal quality indicators (or the so-called product internal.
In the ***value-based*** view, quality is the customers' willingness to pay for software.

Definition of Software quality controversial topic and there are many definitions as bellow:

- The American Heritage dictionary defines quality as "a characteristic or attribute of something".
- IEEE Glossary: Degree to which a system, component, or process meet specified requirements, and customer or user needs or expectations.
- ISO 8402: The totality of features and characteristics of a product or service that bear on its ability to satisfy specified or implied needs.

Based on the above definitions of software quality Although they differ, but they agreed about the quality standard for any software product must be achieve many characteristic and requirements functional and non-functional to ensure that the software product work correctly with high level of performance and achieve the main objective of project.

2.3 Why we need software quality

Software quality standers guide project managers, developers or others development team to make sure the software development process, progressing correctly from the early stages of the software product life cycle based on a set of steps and activities that must be followed in turn reflected on the quality of the product, development time and cost. Generally can summarize the benefits of software quality standards in the following: [2]

1. The ability to apply methodologies and procedures of the highest professional level.

2. Better mutual understanding and coordination among development teams but especially between development and maintenance teams.

3. Greater cooperation between the software developer and external participants in the project.

4. Better understanding and cooperation between suppliers and customers, based on the adoption of standards as part of the contract.

Quality assurance (QA) is a process use to ensuring that a company or organization is providing the best possible products or services. It is focuses on the end result, such as testing and focuses on enhancing and improving the process that is used to create the end result, rather than focusing on the result itself. Among the parts of the process as planning, design, development, production and service.

Quality control (QC) is a set of procedures intended to ensure that a software product or performed service adheres to a defined set of quality criteria or meets the requirements of the client or customer. QC is not identical with, quality assurance (QA). QA is defined as a procedure or set of procedures intended to ensure that a product or service under development (before work is complete) meets specified requirements. QA is sometimes expressed together with QC as a single expression, quality assurance and control (QA/QC).

In order to implement an effective quality control program, an enterprise or development team must first decide which specific standards the software product or service must meet. Then the extent of quality control practice must be determined. Real-world data must be collected for example, the percentage of units that fail and the results reported to management. After this, corrective action must be decided upon and taken for example defective units must be repaired or rejected and poor service repeated at no charge until the customer is satisfied or achieving requirement correctly. If too many unit failures or instances of poor service occur, a plan must be devised to improve the production or service process and then that plan must be put into action. Finally, the quality control process must be ongoing to ensure that remedial efforts, if required, have produced satisfactory results and to immediately detect recurrences or new instances of trouble.

2.4 ISO software procuress improvement standards

ISO (the International Organization for Standardization) and IEC (the International Electro technical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity.

ISO/IEC 9126 Software engineering — Product quality was an international standard for the evaluation of software quality. It has been replaced by ISO/IEC 25010:2011 although it is considered the most prevalent. The fundamental objective of the ISO/IEC 9126 standard is to address some of the well known human biases that can adversely affect the delivery and perception of a software development project.
These biases include changing priorities after the start of a project or not having any clear definitions of "success." By clarifying, then agreeing on the project priorities and subsequently converting abstract priorities (compliance) to measurable values, ISO/IEC 9126 tries to develop a common understanding of the project's objectives and goals. The standard is divided into four parts:

ISO/IEC 9126 1-4 was replaced by ISO/IEC 25010 – 2011 in the new model, add new characteristic and new some sub-characteristic and a number of sub-characteristic in ISO/IEC 9126 were renamed to more accurate terms. The 25010 quality standard also works a bit different than the 9126 standard. The software product quality model describes the internal and external measures of software quality. Internal measures describe a set of static internal attributes that can be measured. The external measures focus more on software as a black box and describe external attributes that can be measured.

The quality model ISO/IEC 25010 comprises the eight quality characteristics and each has set of sub-characteristic as the following: [4]

Functional Suitability

This characteristic represent s the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. This characteristic is composed of the following sub characteristics:

- **Functional completeness.** Degree to which the set of functions covers all the specified tasks and user objectives.
- **Functional correctness.** Degree to which a product or system provides the correct results with the needed degree of precision.
- **Functional appropriateness.** Degree to which the functions facilitate the accomplishment of specified tasks and objectives.

Performance efficiency

This characteristic represent s the performance relative to the amount of resources used under stated conditions. This characteristic is composed of the following sub characteristics:

- **Time behavior.** Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements.
- **Resource utilization.** Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements.
- **Capacity.** Degree to which the maximum limits of a product or system parameter meet requirements.

Compatibility

Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment. This characteristic is composed of the following sub characteristics:

- **Co-existence.** Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product.
- **Interoperability.** Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.

Usability

Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. This characteristic is composed of the following sub characteristics:

- **Appropriateness Recognizability.** Degree to which users can recognize whether a product or system is appropriate for their needs.
- **Learnability.** degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.
- **Operability.** Degree to which a product or system has attributes that make it easy to operate and control.
- **User error protection.** Degree to which a system protects users against making errors.
- **User interface aesthetics.** Degree to which a user interface enables pleasing and satisfying interaction for the user.
- **Accessibility.** Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.

Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. This characteristic is composed of the following sub characteristics:

- **Maturity.** Degree to which a system, product or component meets needs for reliability under normal operation.
- **Availability**. Degree to which a system, product or component is operational and accessible when required for use.
- **Fault tolerance.** Degree to which a system, product or component operates as intended despite the presence of hardware or software faults.
- **Recoverability.** Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system.

Security

Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. This characteristic is composed of the following sub characteristics:

- **Confidentiality.** Degree to which a product or system ensures that data are accessible only to those authorized to have access.

- **Integrity.** Degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.
- **Non-repudiation.** Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.
- **Accountability.** Degree to which the actions of an entity can be traced uniquely to the entity.
- **Authenticity.** Degree to which the identity of a subject or resource can be proved to be the one claimed.

Maintainability

This characteristic represent s the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements. This characteristic is composed of the following sub characteristics:

- **Modularity.** Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.
- **Reusability.** Degree to which an asset can be used in more than one system, or in building other assets.
- **Analyzability.** Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified.
- **Modifiability.** Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.
- **Testability.** Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.

Portability

Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another. This characteristic is composed of the following sub characteristics:

- **Adaptability.** Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.
- **Install ability.** Degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment.
- **Replace ability.** Degree to which a product can replace another specified software product for the same purpose in the same environment.

ISO/IEC 25010 is a part of the SQuaRE series of standards and was prepared by Joint Technical Committee ISO/IEC JTC 1, *information technology*, Subcommittee SC 7, *Software and System Engineering*

The SQuaRE series of standards consists of a set of divisions under the general title Software product Quality Requirements and Evaluation [4].

2.5 Overview about Capability maturity model integrity (CMMI)

CMMI is a framework of best practices. The current version, CMMI-DEV, describes best practices in managing, measuring and monitoring software development processes. The CMMI model does not describe the processes themselves; it describes the characteristics of good processes, thus providing guidelines for companies developing or honing their own sets of processes.

CMMI is the successor to CMM (Capability Maturity Model). Both CMM and CMMI were developed at the Software Engineering Institute (SEI) at Carnegie Mellon University in Pittsburgh, Pa. CMM was developed in the late 1980s, and retired a decade later when CMMI was developed. CMMI v1.02 was released in 2000. The current version, released in August 2006, is CMMI 1.2

CMMI use to help raise the level of quality of products and assist companies in better predicting the time and budget required to develop them. The models help create an environment to support these functions in a repeatable way. Continual improvement is built into the models.

Obtaining the greatest value from adopting the models' processes involves three key components:

- Understanding the new practices
- Not treating them as engraved in stone, but adapting them to the environment
- Sticking with the changes long enough for them to make a difference

CMMI aim is to improve existing software-development processes, but it can also be applied to other processes. CMMI Structure consists from five levels of process maturity defined are:

1. **Initial** - until the process is under statistical control, no orderly progress in process improvement is possible.

2. **Manage** - a stable process with a repeatable level of statistical control is achieved by initiating rigorous project management of commitments, cost, schedule, and change.

3. **Defined** - definition of the process is necessary to assure consistent implementation and to provide a basis for better understanding of the process. At this point, it is probable that advanced technology can be usefully introduced.
4. **Quantitatively Managed** - following the defined process, it is possible to initiate process measurements. This is where the most significant quality improvements begin to appear.

5. **Optimized** - with a measured process, the foundation is in place for continuing improvement and optimization of the process.

While there are many other elements to these maturity level transitions, the basic objective is to achieve a controlled and measured process as the scientific foundation for continuous improvement.

2.6 Previous study about SPI in SMEs

There are many studies researched in the ability to apply the SPI and SPI standards in the small and medium size enterprises. The most important of them, study published by A. M. Adb-algager about Problems and Future Trends of Software Process Improvement in Some Sudanese Software Organizations.

The study was conducted for 28 companies of different sizes and specialties, interview the software experts to observe how these organizations can improve their software process and then design a questionnaire in order to

show the relationship between Sudanese software organizations and software process improvement standards.
The result of study concluded that the most of the Sudanese software companies do not follow any software process improvement standards and identified the problems faced by these organizations during the implementation of software process improvement standards and shows the ability of these organizations to implement the software process improvement standards. [5]

The methodology of this study base on questionnaire that distributed among 28 respondents. The organizations chosen to complete the questionnaire are Sudanese organizations and 78% of these organizations from an Information Technology organization type and 22% of the respondents were mainly from telecommunications organization type.

The data collected by questionnaire shows 5 of 28 organizations were established before less than one year, 7 of 28 organizations were established before 1 to 3 year(s), 4 of 28 organizations were established before 4 to 6 years, 6 of 28 organizations were established before 7 to 10 years, 6 of 28 organizations were established before more than 10 years.

Most of these organizations are IT based organizations and the rest of these organizations are mainly telecommunications based organizations. In addition, as shown in Table I the number of the full time staff member in the 12 organizations are less than 10 staff member and 12 organizations are between 10 and 20 staff member and only in 4 organizations are more than 21 staff members.

TABLE I. FULL TIME STAFF MEMBER

Staff Member	Frequency	Percent	Valid Percent	Cumulative Percent
< 10	12	42.9	42.9	42.9
10..20	12	42.9	42.9	85.7
21..50	4	14.3	14.3	100.0
Total	28	100.0	100.0	

Table II shows 78% of the respondents claim to have an understanding of average or less of SPI, 4 of the 28 respondents rate their knowledge of the

SPI as extensive and 2 of the 28 respondents rate their knowledge of the SPI as very extensive.

TABLE II. KNOWLEDGE RATE OF SPI

Rate	Frequency	Percent	Valid Percent	Cumulative Percent
Very limited	6	21.4	21.4	21.4
Limited	4	14.3	14.3	35.7
Average	12	42.9	42.9	78.6
Extensive	4	14.3	14.3	92.9
Very extensive	2	7.1	7.1	100.0
Total	28	100.0	100.0	

Table III shows 24 of the 28 respondent's rate their knowledge of software process improvement standards as average or less. 2 of the 28 respondents rate their knowledge of software process improvement standards as extensive and 2 of the 28 respondents rate their knowledge of software process improvement standards as very extensive.

TABLE III. KNOWLEDGE RATE OF SPI STANDARDS

Rate	Frequency	Percent	Valid Percent	Cumulative Percent
Very limited	8	28.6	28.6	28.6
Limited	4	14.3	14.3	42.9
Average	12	42.9	42.9	85.7
Extensive	2	7.1	7.1	92.9
Very extensive	2	7.1	7.1	100.0
Total	28	100.0	100.0	

Table IV shows 26 of the 28 respondents acknowledged that their knowledge of CMM was either limited or very limited. 2 of the 28 respondents rate their knowledge of CMM as extensive. The respondents who had a less than average knowledge in SPI, SPI standard and CMM were given a lower rating to some questions because of their insufficient experience in these topics.
Moreover, 90% of the respondents think implementation of SPI standards increase software quality. But, 1 of the 28 respondents has implemented SPI standard. This result shows most Sudanese software organizations do not implement SPI standards.

TABLE IV. KNOWLEDGE RATE OF CMM

Rate	Frequency	Percent	Valid Percent	Cumulative Percent
Very limited	22	78.6	78.6	78.6
Limited	4	14.3	14.3	92.9
Very extensive	2	7.1	7.1	100.0
Total	28	100.0	100.0	

However, 57% of the respondents have the ability to implement SPI standards. 6 of the 28 respondents are not able to implement SPI standards. 6 of the 28 respondents said this question is not applicable to them due to their limited knowledge related to SPI and its standards.

The question for study has been asked; do you believe that the implementation of the software process improvement standard has an effect on Software Quality for small software scale? The respondents are of the opinion that an improvement in the quality of a final product is directly associated with the adoption of SPI standards. If the SPI standards will not be adopted surely it will result in flawed products. The 42% of the respondents approved that adopting the software process improvement standards for small-scale projects will result in no effect or in less effect. The 92.9% of the respondents believe that there is a high increase of quality when adopting software process improvement standards for medium-scale projects and 7.1% of the respondents are not sure.

All respondents believed that there was a high increase in quality when adopting software process improvement standards for large-scale projects. The findings of the effects of adopting software process improvement standards on quality and cost. Whole respondents were agreed the average number of the team members that work on software development in small-scale between 2 and 5 full time staff members. But, in medium-scale 92.9% of the respondents were decided the average size of the team members that work on software development between 5 and 10 team members and 7.1% between 2 and 5 team members. In addition, 71.4% of the respondents were agreed the average size of the team member that work on software development in large-scale between 10 and 50 team members and 28.6% between 5 and 10 team members. This result and the result in Table I shown all these organizations are a small organization because the sizes of the team member's work on software development in each project category are small.

At the same time, information has been collated from respondents shows the major problems faced by the organizations during the implementation of the software process improvement standards for different sizes of software organizations.

TABLE V. COMMON PROBLEM EXPERIENCED WHILE IMPLEMENTING SPI STANDARDS FOR A LARGE ORGANIZATIONS

Common problem	Frequency	Percent	Valid Percent	Cumulative Percent
Lack of skilled people who can follow standards	6	21.4	21.4	21.4
Lack of top management support	12	42.9	42.9	64.3
Project size/complexity	4	14.3	14.3	78.6
Project team size	2	7.1	7.1	85.7
Lack of skilled – lack of customer collaboration	2	7.1	7.1	92.9
Lack of skilled – lack of top management - lack of customer collaboration	2	7.1	7.1	100.0
Total	28	100.0	100.0	

The result shows that the lack of skilled people who can follow standards is the major factor in both small and medium organizations. 21% of the respondents agreed that the major problem in implementation of the software process improvement standards for small and medium organizations is the lack of customer collaboration and 14% of the respondents agreed the problems in implementation of the software process improvement standards for small and medium organizations is a lack of skilled people who are unable to follow and apply standards.

Table V shows 42.9% of the respondents agreed that the major problem in implementation of the software process improvement standards for large organizations is the lack of top management support. 21.4% of the respondents decided the problems in implementation of the software process improvement standards for large organizations are lack of skilled people who are unable to follow and apply standards and 14.3% of the respondents decided that are project size and project complexity problems. And 21.3% of the respondents agreed that the problem in implementation of the software process improvement standards for large organizations are the number of staff for each project and combination of more than one reason such as lack of skilled people and lack of customer collaboration problems as shown in Table V.

The 35.7% of the respondents are not sure that which processes are most suitable for software process improvement standards in different kinds of

organization. This results because the most respondents do not have knowledge of SPI and SPI standards.

In addition, 35.7% of the respondents have belief in another SPI Model for software process improvement standards in different kinds of organization. This results because the most respondents are small organizations and the respondents tack in their account Sudanese organization properties. 28% of the respondents believe that CMM, CMMI and ISO 9000 are most suitable SPI standards.

For medium organizations 4 of the 28 respondents follow different level of CMM by 50% and 4 of the 28 respondents follow different level of CMM by 75%. For large organizations 8 of the 28 respondents follow different level of CMM by 75%.

The intention to follow different levels of CMM for different kinds of organizations is measured. 71.4% of the respondents did not answer this question because most of the respondents do not have knowledge about CMM standards.

For small organizations Table VI shows 4 of the 28 respondents follow different level of CMM by 25%, 2 of the 28 respondents follow different level of CMM by 50% and 2 of the 28 respondents follow different level of CMM by 75%.

TABLE VI. INTENTION TO FOLLOW DIFFERENT LEVEL OF CMM

Percent	Frequency	Percent	Valid Percent	Cumulative Percent
75%	2	7.1	11.1	11.1
50%	2	7.1	11.1	22.2
25%	4	14.3	22.2	44.4
Other	10	35.7	55.6	100.0
Total	18	64.3	100.0	
Missing	10	35.7		
Total	28	100.0		

Summary of the main question, Table VII shows 4 of the 28 respondents are programmer or developer, 2 of the 28 respondents are software architect, 6 of the 28 respondents are software engineer, 2 of the 28 respondents are consultants, 2 of the 28 respondents are project manager, 6 of the 28

respondents are executive manager, 4 of the 28 respondents are programmer or developer and project manager, and 2 of the 28 respondents are others.

TABLE VII. STAFF POSITION IN THE ORGANIZATION

X	Frequency	Percent	Valid Percent	Cumulative Percent
Programmer/ Developer	4	14.3	14.3	14.3
Software engineer	2	7.1	7.1	21.4
Software engineer	6	21.4	21.4	42.9
Consultant	2	7.1	7.1	50.0
Project manager	2	7.1	7.1	57.1
Executive manager	6	21.4	21.4	78.6
Other	2	7.1	7.1	85.7
Programmer – project manager	4	14.3	14.3	100.0
Total	28	100.0	100.0	

Summary of the study:

More than 90% of the respondents think implementation of software process improvement standards increase software quality. But, still most of Sudanese software organizations are not implementing any software process improvement standards. However, most respondents are able to implement software process improvement standards. Respondents agreed that adopting software process improvement standards for a small-scale project will result in an increase in cost. Conversely, when it came to medium and large-scale projects the costs started to decrease for adopting software process standards. 57.1% of the respondents for small-scale project believe that there is a decrease in quality when adopting software process improvement standards. Although, the 92.9% of the respondents of medium-scale projects believe that there is a high increase in quality when adopting software process improvement standards and all respondents believed that there is a high increase in quality when adopting software process improvement standards for large-scale projects. The most common problems experienced while implementing software process improvement standard for different kinds of software organizations are: lack of skilled people who are unable to follow standards, lack of top management support, lack of customer collaboration, project size/complexity, project Team size and the cost. 35.7%

of the respondents believe other SPI Model is the most suitable of the software process improvement standards for the different kinds of organization. This results because the most respondents are small organizations and the respondents tack in their account Sudanese organization properties.

Previous study about software process improvement methodologies for small and medium enterprises:

In this section, we will look to a study about *The* Impact of Software Process Improvements in Small and Medium Scale Enterprises, the main objective of study is proposes new software process model that can be used in small and medium enterprises. A new model is proposed based on the traditional software process development models such as Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI), and International Organization for Standardization (ISO) and Software Process Improvement Capability Determination (SPICE) [6].
Study revealed there are basically large gains to be made within the industry by the wider implementation of SPI, but as yet the use of models such as CMMI within small organizations has been limited. There is a general agreement that they cannot be applied unmodified to small organizations. Many researches were carried out in order to determine what modifications must be made to the model, to make it effective and efficient in these development environments. Tailoring is needed in specific areas, such as documentation, management, review, resources and training. Major improvements can be achieved by improving the technical issues of the process rather than organizational issues, and proposed a model that integrate CMM with the ISO 9001 and ISO 9000-3 models.

A proposed model will provide an answer for existing issues like. How can SPI be used to the organization's business goals? How can a software measurement be used effectively within the SPI programmed? How can we assess the effectiveness of the SPI programmed, so that the management can see the return on their investment?

The study pointed out there is fundamental operational differences between small and large organizations the Small organizations are concerned with practice and the large organizations with processes. There are six main key

points to software process improvements in small and medium enterprises and they are [23]:

(1) Senior management support
(2) Adequate staffing
(3) Applying project management principles to process improvement
(4) Integration with ISO 9001
(5) Assistance from process improvement consultants
(6) Focus on providing value to projects and to the business

There is study identified seven small organization challenges to follow software process improvements [24]:

(1) Handling requirements.
(2) Generating documentation.
(3) Managing projects.
(4) Allocating resources.
(5) Measuring progress.
(6) Conducting reviews.
(7) Providing training.

It is quite difficult for any SMEs to select an improvement approach, and to apply it in their organization without the external help from the consultants [25]. Some of the shortcomings faced by SMEs are: Excessive documentation, Extensive number of Specific Practices (SP), Requirement of extensive resources, High training costs, Practices independent of project type, Lack of guidance in satisfying project and development team needs and Many of the smaller companies oppose the CMMI model due to the expensive compliance effort, both in time and money.

The summary of All the above mentioned SPI methodologies are divergent in characteristics; it is required to find out some significant and common attributes so that we can find a comparative view of all the selected approaches. Kautz et al concluded in their findings that primary lesson for the small organizations, which wish to perform improvement activities, is that it makes sense to use a structural model to organize the process. They further suggests the secondary lesson is that the model should be adjusted to the particular conditions of the organizations and the third lesson is that it make sense to perform the improvement activities as a project with clearly

assigned and documented roles, responsibilities and resources. He further pointed out the significance of factors to be studied further like management support and commitment, project planning and organization, education and training, assessment, monitoring and evaluation, staff involvement, support and knowledge transfer by external consultants, usability and validity of the introduced changes and cultural feasibility for process improvement in software SMEs.

The proposed methodology can be used for future work which is aimed at helping small and medium enterprises to implement and improve their software processes. To help small and medium scale industries, we need to analyze and find the characteristics of these enterprises. To check the software process improvements success factors, we need to determine the software process activities. To improve the software business processes, select the most appropriate software process models that are used in SMEs. Select the most suitable SPI traditional models, Compare these models with the software process models and figure out the missing activities. Then, modify the software process activities to attain all process areas of SPI model depending on the activities of other SP models. After the completion of these modifications, determine the new SP model requirements and conduct administrative questionnaires on small and medium enterprise to check whether the new modified model meets their requirements. We will get an overall idea after the analysis of the questionnaires. Then we can finalize the final requirements needed for implementing the SPI model for the small and medium enterprises.

Studies try to adopt CMMI:

There are many studies try to adopting CMMI with in small and medium enterprise easy and efficient, **Adopting CMMI for Small Organizations** Sponsored by the U.S. Army Aviation and Missile Research, Development & Engineering Center (AMRDEC) Software Engineering Directorate (SED).[7]

Pilot Project Purpose, a joint project performed by the partnership between the SEI and AMRDEC SED to establish the technical feasibility of developing guidance and other special purpose transition mechanisms to support adoption of CMMI by small and medium enterprises (SMEs).

Pilot Project Goals:

1. Exercise at least 3 CMMI Process Areas in a small company
2. Work with at least 2 companies
3. Codify recommendations for how to package, sell, appraise, train, implement CMMI for SME's to extent reasonable based on our pilots
4. Be able to articulate business case for small companies similar to those in HSV to adopt CMMI.
5. Generate "follow-on" path to extend initial pilots
6. Percent project results at SE2 2004 and the Annual CMMI User Technology Conference
7. Provide SEI CMMI/SCAMPI projects w/appropriate change requests/feedback

Summary and Recommendations: (Lessons Learned)

1. Small Business needs to realize pay off quickly
2. Customer driven requirements are significant (de)motivator
3. Small businesses do not have staff dedicated solely to CMMI implementation customer requirements take priority and can cause delays
4. There is not a lot of functional organization to leverage from in a small business
5. CMMI is easier to interpret for product development than for services Small Businesses are typically more service oriented
6. “The customer rules” Many small organizations adopt/adapt their business practices directly from their customers or primes

2.8 Conclusions

By the end of this chapter, we have overview about what software quality from different view or perspectives, the basic concepts of software engineering related to the software process improvement (SPI) and software process improvement standard (SPI stander), the purpose and benefit for using SPI and SPI standard, the most important institutions and organization responsible to develop quality standards for software such as IEEE, ISO and IEC, the most important quality models and overview about CMMI, study about the ability to apply SPI with in Sudanese organization and the problem faced and finally studies that try to solve the problem and propose quality model appropriate with small and medium enterprise.

CHAPTER (3)

RESEARCH METHODOLOGY

3.1 Introduction

Research in common it's devote effort to a search for knowledge. Once can also define research as a scientific and systematic search for pertinent information on a specific topic. On other hand can define research is an art of scientific investigation and some people consider research as a movement, a movement from the known to the unknown [8], actually it is a voyage for discovery.

Research is an academic activity should be used in a technical sense. According to Clifford Woody "*research comprises defining and redefining problems, formulating hypothesis or suggested solutions; collecting, organizing and evaluating data; making deductions and reaching conclusions; and at last carefully testing the conclusions to determine whether they fit the formulating hypothesis*". [8]

The purpose of research is to discover answers to questions through the application of scientific procedures. The main aim of research is to find out the truth which is hidden and which has not been discovered as yet. Though each research study has its own specific purpose, we may think of research objectives as falling into a number of following broad groupings: [8]

1. To gain familiarity with a phenomenon or to achieve new insights into it (studies with this object in view are termed as *exploratory* or *formulate* research studies).
2. To portray accurately the characteristics of a particular individual, situation or a group (studies with this object in view are known as *descriptive* research studies).
3. To determine the frequency with which something occurs or with which it is associated with something else (studies with this object in view are known as *diagnostic* research studies).
4. To test a hypothesis of a causal relationship between variables (such studies are known as *hypothesis-testing* research studies).

3.2 Purpose of the current study

Purpose of this study is two main parts the first one to get a master's degree in software engineering and the second part is solving problem faced small and medium enterprises and how to develop software project base on software quality model, using proposed methodology content of a set of activities and steps simple, efficient and inexpensive assume resolve the problem.

3.3 Types of scientific research

There are several types of the scientific research depends on the field, type of the preliminary information and data available to researcher and the expected results or outputs. Basic types of search as follows: [8]

1. Descriptive vs. Analytical: Descriptive research includes surveys and fact-finding enquiries of different kinds. The major purpose of descriptive research is description of the state of affairs as it exists at Percent.

2. Applied vs. Fundamental: Research can either be applied (or action) research or fundamental (to basic or pure) research. Applied research aims at finding a solution for an immediate problem facing a society or an industrial/business organization, whereas fundamental research is mainly concerned with generalizations and with the formulation of a theory.

3. Quantitative vs. Qualitative: Quantitative research is based on the measurement of quantity or amount. It is applicable to phenomena that can be expressed in terms of quantity. Qualitative research, on the other hand, is concerned with qualitative phenomenon, i.e., phenomena relating to or involving quality or kind.

4. Conceptual vs. Empirical: Conceptual research is that related to some abstract idea(s) or theory. It is generally used by philosophers and thinkers to develop new concepts or to reinterpret existing ones.

This study can be classified under the category of Applied research as it seeks to find a applicable solution for real problem in the field of software engineering and specifically focus on small and medium enterprises challenges which do not have sufficient budget for the implementation for software quality model, which can be expensive in many cases, working to find information on a large scale and considered it as the Scientific contribution when has been proven hypotheses and provide a solution acceptable and appropriate.

3.4 Research methodology

Methodology it is necessary for a researcher to design solution for the problem chosen. It is important for researcher to know not only the research methods necessary for the research under taken but also the methodology for achieving the goal of study.

Research methodology is a collective term for the structured process of conducting research. There are many types of methodologies used in various types of research and the term is usually considered to include research design, data gathering and data analysis and also referred to as research methods, usually encompasses the procedures followed to analyze and interpret the data gathered. These often use a range of sophisticated statistical analyses of the data to identify correlations or statistical significance in the results. Generally used with in academic research for test hypotheses or theories.

Main part of the research methodology is how the research is conducted. This is called the study design and typically involves research conducted using questionnaires, interviews, observation and/or experiments.

Objective, represent active research can be difficult to conduct because tests can usually normally only be conducted on a small sample, this means that researchers need to have a very detailed understanding of the types and limitations of research methodologies which they are using.

In the field of scientific research, there is a difference between the two fundamental concepts in any process of scientific research, research methods and research methodology as below [7]:

1. Research methods, are the various procedures, schemes and algorithms used in research. All the methods used by a researcher during a research study are termed as research methods. They are essentially planned, scientific and value-neutral. They include theoretical procedures, experimental studies, numerical schemes, statistical approaches, etc. Research methods help us collect samples, data and find a solution to a problem. Particularly, scientific research methods call for explanations based on collected facts, measurements and observations and not on reasoning alone.

2. Research methodology is a systematic way to solve a problem. It is a science of studying how research is to be carried out. Essentially, the procedures by which researchers go about their work of describing, explaining and predicting phenomena are called research methodology. It is also defined as the study of methods by which knowledge is gained. Its aim is to give the work plan of research.

3.4 Type of research methodologies

Research methodologies can be quantitative or qualitative. Ideally, comprehensive research should try to incorporate both qualitative and quantitative methodologies but this is not always possible, usually due to time and financial constraints.

Quantitative research is based on the measurement of quantity or amount. Here a process is expressed or described in terms of one or more quantities. The result of this research is essentially a number or a set of numbers. Some of the characteristics of qualitative research/ method as below: [7]

1. It is numerical, non-descriptive, applies statistics or mathematics and uses numbers.
2. It is an iterative process whereby evidence is evaluated.
3. The results are often presented in tables and graphs.
4. It is conclusive.
5. It investigates the what, where and when of decision making.

Qualitative research is concerned with qualitative phenomenon involving quality. Some of the characteristics of qualitative research/method as below [7]:

1. It is non-numerical, descriptive, applies reasoning and uses words.
2. Its aim is to get the meaning, feeling and describe the situation.
3. Qualitative data cannot be graphed.
4. It is exploratory.
5. It investigates the why and how of decision making.

Quantitative or qualitative. Depend on the preferences and abilities, and the suitability of particular approaches of the topic and the researcher need to be able to justify why was chosen to use quantitative or qualitative methodology.

Appropriate method for this study is a quantitative methodology which Depend on the measurement and mathematical statistics way, Selection of this method is essentially based on the subject and the field of the study and on the other hand there are important factors directly influenced in the decision-making concerning about this choice outlined in the following:

1. The study essentially depends on the results of previous study regarding the level and the ability of small and medium enterprises in the Sudanese market commitment to the standards of software quality. All the results were in the mathematical form clear and precise that makes it very easy for researchers and people interested in field of study to understand and use the result, accordingly it hard for any subsequent study to publish their results outside this context.

2. The mathematical nature of the study field (Software Engineering), which that the results be represent ative in the mathematical and statistical form more convincing than result that depend on the non-numerical or descriptive, on the other hand reflecting the level of effort and the work done by researcher to publish accurate and reliable results.

3. Publishing the results of study in the mathematical and statistical form that make the it better and support their argument and to ensure about achieving its goals that represent ed to develop a scientific and practical solutions for small and medium enterprises to apply the software quality standards with an easy and simple practices to convincing developers and systems engineers to those institutions.

4. Publishing the results of study mathematical and statistical form that make it easy to investigated and positively affects about making the right decision for using the results and recommendations of the study or not, on the other hand supports and helps significantly subsequent studies on the same field.

Choose a search methodology correctly and then how to collect and analyze data is the first step to ensure the success of the study and achieve their goals and help us answer the research questions, which depend on proper planning and progress correctly and the level of effort.

3.5 Methodology of the current study

Based on lengthy discussions with experts, specialists and academics in the field of software engineering and a previous study tried to develop solutions quality model are suitable to work in small and medium enterprises [6], which concluded that software quality models may not be suitable for every institution does not choose any model must in the first place to study and learn organization's needs clearly and then determine its appropriate quality model, or even merge with some models. Study Hypotheses, developed a methodology for small and medium enterprises consisting of a set of steps and activities and is working to integrate ISO quality model of assessing and improving software quality ISO / IEC 25010 with the level II of CMMI in one model simple, easy to use and inexpensive can positively affect in a way to improve the software quality in these institutions and is reflected to reduce the time and resources required to develop any software project.

Answering the questions and test the basic premise of study, if it is proven it is very useful for solving a fundamental problem facing small and medium

enterprises (SMEs) specialized in web development and how to keep those institutions to use software quality improvement and software quality improvement standers. The study impose that it is possible to develop a software quality model meet all small and medium enterprises (SMEs) needs and positively impact on the level of their software project quality in the simple and efficient to use framework and the most important thing must be inexpensive based on a previous study to develop model achieving above characteristics must be developed on the needs of those institutions and this leads us to these characteristics and requirements of this model must be collected from developers, system engineers and project managers.

3.5.1 Collecting data

Dealing with any real life problem it is often found data at hand are inadequate and hence, it becomes necessary to collect data that are appropriate.

Primary data can be collected either through experiment or through survey and observes some quantitative measurements, with the help of which examines the truth contained in study hypothesis. But in the case of a survey, data can be collected by any one or more of the following ways: [8]

1. By observation: This method implies the collection of information by way of investigator's own observation, without interviewing the respondents. The information obtained relates to what is currently happening and is not complicated by either the past behavior or future intentions or attitudes of respondents. This method is no doubt an expensive method and the information provided by this method is also very limited. As such this method is not suitable in inquiries where large samples are concerned.

2. Through personal interview: The investigator follows a rigid procedure and seeks answers to a set of pre-conceived questions through personal interviews. This method of collecting data is usually carried out in a structured way where output depends upon the ability of the interviewer to a large extent.

3. Through telephone interviews: This method of collecting information involves contacting the respondents on telephone itself. This is not a very widely used method but it plays an important role in industrial surveys in developed regions, particularly, when the survey has to be accomplished in a very limited time.

4. By mailing of questionnaires: The researcher and the respondents do come in contact with each other if this method of survey is adopted. Questionnaires are mailed to the respondents with a request to return after completing the same. It is the most extensively used method in various economic and business surveys.

5. Through schedules: Under this method the enumerators are appointed and given training. They are provided with schedules containing relevant questions. These enumerators go to respondents with these schedules. Data are collected by filling up the schedules by enumerators on the basis of replies given by respondents. Much depends upon the capability of enumerators so far as this method is concerned. Some occasional field checks on the work of the enumerators may ensure sincere work.

In this study I was adopted on the personal interview and some existing methods of modern technique communication, for example design a soft copy questionnaire and send invitations by e-mail and publish invitations through social networks to ensure greater coverage and included some tools and feature to ensure the correction of the questionnaire process and avoid errors that be occur to reach true and accurate results.

Based on factors from previous study [5] that effect negatively for applying the software process improvement and software process improvement standards on small and medium enterprises in Sudanese market specialize on software development. The proposed methodologies try to develop solution for this problem. Primarily the main component of proposal solution will be as a set of steps and activates for all development process represent the software process improvement (SPI) to achievement the objectives of the study.

3.5.2 Questionnaire number one

Collecting information about proposed methodology from viewpoint of researchers, developers, project managers and each one related to software development process on the institutions target through the basic and traditional model for software development process from collect and analyze data, time and cost estimation, design, coding and testing until deliver the software product and finally find out what are the difficulties they face in each of these steps and then analyze the data for giving clear outlines about the method of solution.

3.5.3 Questionnaire number one - Analyze data

Data will be analysis closely and related operations such as establishment of purposeful and usable categories, the application of these categories to raw data through tabulation, drawing statistical inferences and then transform the categories of data are into readout that may be tabulated and counted and used to develop proposal methodology.

3.5.4 Develop a solution

Develop proposal methodology according to result of the questioner number one, solution will be **tested on the real environment** by contacting to a set of institutions and companies that targeted before. Pot the methodology on test and before that should be training one employee or more at least one or two hour about how to use it. Follow the activity of methodology throw all software development process and provide any support and answer all queries for best practice and achieving goals.

3.5.5 Questionnaire number two

Send proposal methodology for testing in the real live and then collecting information about its efficiency and ability to resolve problems and to be applicable or not. I will design a simple questionnaire into groups and categories, each category in a set of questions formulating in a clear, simple and easy to read and understand for respondents, define the relationship between category clearly, include all the ideas that relate to the assessment proposal activity, can also obtain notation or sufficient information when we needed to make some modified for methodology.

List categories are arranged it basically on research questions, hypotheses and as well as taking into account objectives of the study and whether the proposal work can actually solve the problems that faced the small and medium enterprise for applying software quality standards that published on previous studies.

People Target of the questionnaire were project managers, system engineers, designers and developers or all Jobs that are involved in the software development process during all the stages from system analysis to deliver the project and at this stage it is not necessary to take the information and questionnaire of categories that are outside the scope of the institution, such as researchers and other that are interested in the science of software engineering simply and because the solution is develop and directed for people directly effect on the software development process and they are know their needs and better able to assess the proposed work.

Previously I was noted that the questionnaire will be in two copies (soft and hard) and the personal interviews will be the main method for collecting data and will use other methods for contacting with respondents such as send the questionnaire invitation via email and social networks personally to make the range wider, taking into account the correction and accuracy of the questionnaire process on the other hand there are great benefit If outsource developers and systems engineers in the dissemination the questionnaire on the company's target range and reflected clearly on the accuracy of the results.

3.5.6 Questionnaire number two - Analyze data

Analyzing data and preview the result considered one of the last stages in the study and transfer raw data in to readable form in preparation for submission, is worth mentioning that all the studies are similar in the data collection process, but the Appears difference about how to Percent data that has been analyzed from one type of research to another. Studies depends on the quantitative methodology displayed the results in the mathematical form by using tables or diagrams make it easy to read and understand that make it best suited for this study.

The main task of this stage Sort and classify data in the mathematical form and then apply various statistical operations using **SPSS** to get a clear and accurate results that will be compared with the hypothesis and whether it supports or not, percentage and difference between two.

After completion analyzing data we have got clear and accuracy results represent s a summary whole the effort and work that has been done through study and displayed results, determine recommendations for applying software process improvement and software process improvement standards during application development process.

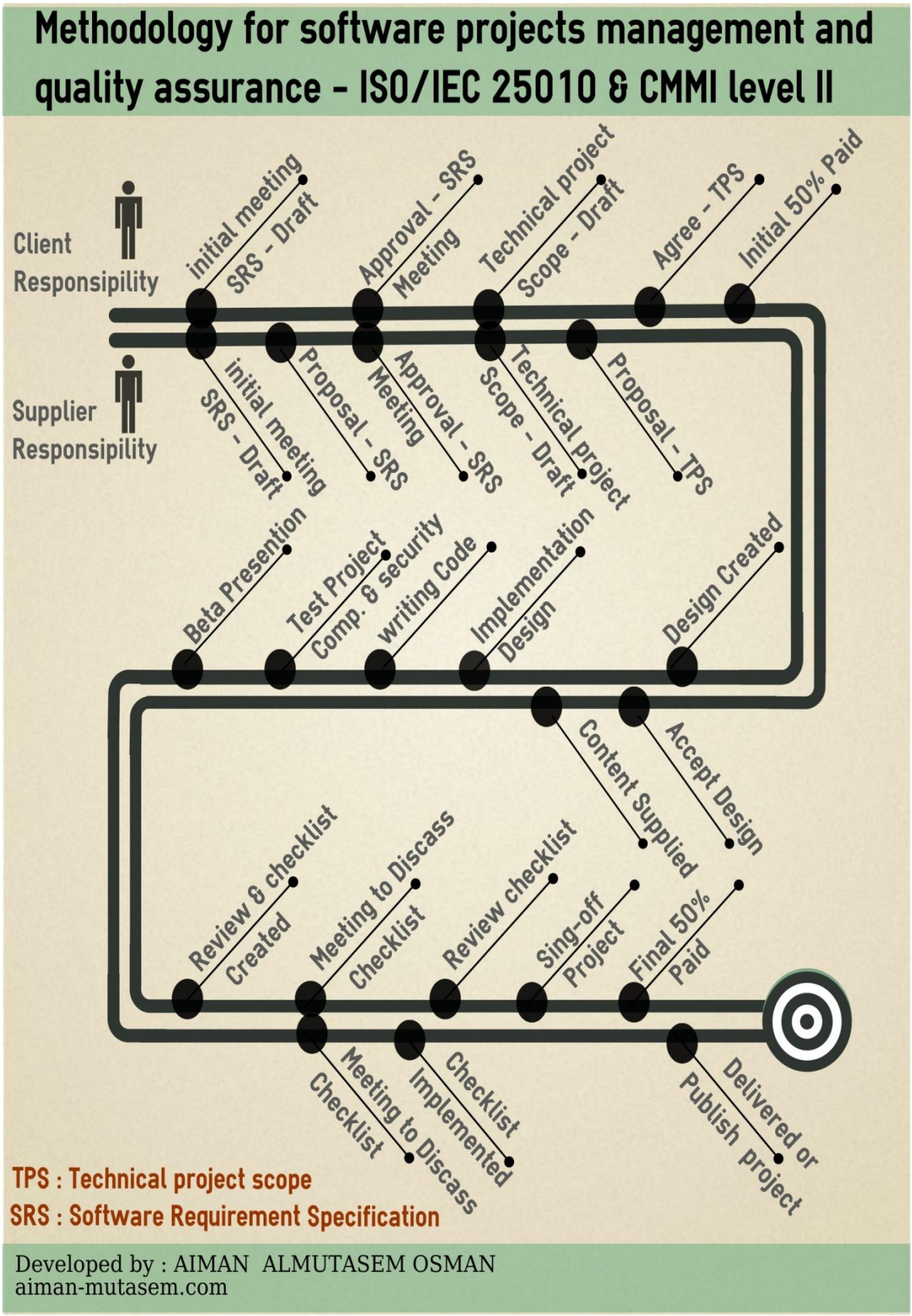
finally, I will try to preview the contribute or any something new can found during the research and how to impact the results for target companies on the quality of their software development process also display shortcomings and limitation on the study and outline about how to be avoided in subsequent studies. I hope very much that the oldest simple scientific contribution in this area and on other hand open door for new questions and new studies. Concluded refine the future work that should be done later.

3.5 Conclusion

By the end of this chapter, covered overview about main concept and types of research, theoretical and main types of research methodology, research methods, main hypothesis of this study, develop proposal methodology, questionnaire number one collecting information about the components of proposed methodology from viewpoint of researchers, developers, project managers and analyze data and then make any modification on the proposed work, prove the hypothesis of the study by sending the proposal methodology to a number of target companies for testing it in the real live with real development process , questionnaire number tow collecting information and give feedback from peoples who deal with the proposal methodology and their assessment of the level of successful proposal solutions for their problems about how to apply software process improvement and software improvement standards and then find out the result after analysis and the last point in this chapter the concludes discuses how to determine recommendations and future works.

CHAPTER (4)

PROPOSED METHODOLOGY


Methodology for software projects management and quality assurance - ISO/IEC 25010 & CMMI level II
Client Responsipility
Supplier Responsipility
initial meeting
SRS - Draft
Approval - SRS
Meeting
Technical project
Scope - Draft
Agree - TPS
Initial 50% Paid
Proposal - SRS
Proposal - TPS
Beta Presention
Test Project
Comp. & security
writing Code
Implementation
Design
Design Created
Content Supplied
Accept Design
Review & checklist
Created
Meeting to Discass
Checklist
Review checklist
Sing-off
Project
Final 50%
Paid
Delivered or
Publish project
TPS : Technical project scope
SRS : Software Requirement Specification
Developed by : AIMAN ALMUTASEM OSMAN
aiman-mutasem.com

4.1 Introduction

Most of the software development companies around the world are considered as small and medium enterprise (SMEs) and represented the backbone of the national and international economy most of them Lack for software process improvement and software process improvement standards specifically the national companies because Lack the knowledge and resources to implement the software quality standards [5], which is expensive in most often and on the other hand More than 90% of the SMEs think implementation of software process improvement standards increase software quality.

There are many factors that influenced to choice this study in this area. Where the small and medium enterprises seeking to profit quickly and reduce the expenditure for limited resources as well as lack of full-time staff to implement and monitor the software process improvement or software process improvement standards. We can summarize the most important factors that make difficult to apply these software quality standards at the national market base on study published on 2013 [5] as follows:

1. The most important factor, lack small and medium enterprise for staff with experience, knowledge and specialists in the software quality follow the standards, where their salaries significant burden on small businesses and on the other hand it is difficult to disperse the staff for this task after training for the limited number of employees in the organization and the limited number of employees and the limited number of employees who work in each project.

2. Top management's lack of knowledge in relation to the applying the software process improvement and software process improvement standards or lack of interest, it is difficult to apply the standards of software quality in small and medium enterprises if they are not supported by top management's.

3. Lack of cooperation between the customers and project owners with employees and people who response to develop their projects to be member of team of a working group commits them to co-exist and provide their view and suggestions as one of the requirements of project management practices.

4. Project size and complexity, the budget allocated for the project, the number of team members that involved to development process all these factors that directly affect on to apply software process improvement and software process improvement standards, small and medium enterprises do not have the time and resources to do so.

The above problems and challenges faced small and medium enterprises in the field of software development will be attention greatly in this study and try to develop solutions simple, efficient, inexpensive, applicable and positively affect the development process.

4.2 The main idea

At this study I will develop proposal methodology based on ISO / IEC 25010 software quality assessment and improve developed by international organization for standardization ISO until Percent this study is the latest and strongest software standers was develop in 2011 within a large family known as SQuaRE Software product Quality Requirements and Evaluation, which consists of a series of parts starting from ISO / IEC 2500n to ISO / IEC 25099 - 25050 are specializing in software quality.

4.3 Over view about CMMI level II

A Capability Maturity Model Integrity (CMMI) is a model of mature practices in a specified discipline, used to improve and appraise a group's capability to perform that discipline. CMMI-based process improvement benefits include improved schedule and budget predictability, improved cycle time, increased productivity, improved quality (as measured by defects), increased customer satisfaction, improved employee morale, increased return on investment, decreased cost of quality. On the other hand integrated vision of improvement for all elements of an organization, efficient, effective improvement and appraisal across multiple disciplines but most of (SMEs) belief that the CMMI develop for large companies and apply it costly.
There are five level of CMMI, this study will be focus about level II that care about project management process and activities are documented, standardized, and integrated into a family of standard software processes for the organization.

The main characteristic and activity that should be done during the CMMI level II is:

1. Processes should be easier to understand.
2. Managers and team members spend less time explaining how things are done and more time does.
3. Projects are better estimated, better planned, and more flexible.
4. Quality is integrated into the project.
5. Costs may go up initially, but do go down over time.
6. There may be more documentation and paper.

The current study care about how to develop methodology for software process improvement and software process improvement standers using to solve the problem face small and medium enterprise about how to apply software quality standards and in particular, companies specializing in the web development by mixing the CMMI level II with ISO/IEC 25010 software quality model for assessment and improvement on the other hand develop set of activities and steps to keep project development process easy, more efficient, more conformable quality standers and reduce the cost and money.

4.4 Most importance project management methodologies

Study will take project management and software quality as the main goals by describe every phase, activity and task, helping developers, project managers and everyone within the development project team to apply best practice approach to managing their projects and how deliver their projects on time within budget, how to initiate, plan, execute and close projects properly. Manage time, cost, quality, change, risk and the other challenge that related about how to manage staff, customers and suppliers.

There are many type of methodology using to manage software project, which has proved successful in many fields, particularly software development, however the problem lies in don't give information for developers or project manager about what to do exactly technically to keep your process conformed to quality standers. Where they give guidelines and directives without preferences thus is customized to experienced staff and need more time for training. The most famous as below:

4.4.1 Agile

Software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Referred as "*The Agile movement proposes alternatives to traditional project management. Agile approaches*

are typically used in software development to help businesses respond to unpredictability" [26].

The main priority is to satisfy the customer needs through incremental delivery approach, requirements change is managed and control even if it is late in development, the iterative approach in agile development control change for the customer's competitive advantage .Delivering build by build to the customer, each individual build ranging from a couple of weeks to a couple of months.

4.4.2 Scram

Methodology, process framework that has been used to manage complex product development since the early 1990s. Scrum is not a process or a technique for building products rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and development practices so that you can improve.

4.4.3 XP

Extreme Programming is a system of practices that a community of software developers is evolving to address the problems of quickly delivering quality software, and then evolving it to meet changing business needs. It has already been proven at many companies of all different sizes and industries worldwide.

The above project management methodology has proved a great success, but need to be modified to fit the needs of (SMEs). The most important features of the proposed methodology it is directed to Web applications that make positively affects the development of effective solutions.

4.5 Overview about proposal works

Proposed methodology develop based on the two tracks, the first track it is the customer responsibility and the second track supplier responsibility (peoples produce a product for a software) determines the responsibility of each one of them in every stage of the software development process to ensure the final product meets the customer needs, expectations and required level of quality.

Track one customer responsibility determining functional and non-functional requirements, descript how the system should be after the process will be done, performance of the system on the future and then follow up requirement implementation and give his opinion clearly, accurate and receives reports from each stage to be abreast for the level of development progressing to make any adjustments promptly before move to another stage.

Track two define supplier responsibility, project manager (PM) or suppliers and his responsibility from initial stages until deliver project, all this development activity will be done by a set of reports template or documents will be develop during current study and integration with the effort from the previous studies, represent as bellow:

1. Determining project requirements and manage project.
2. Develop project plan and monitor project implementation.
3. Control and manage project team.
4. Process measurement and analysis data for each stage.
5. Quality assurance based on the software quality assessment and improvement ISO / IEC 25010.
6. Identify a clear mechanism to communicate with the customer and make sure of his consent to each output of each stage to ensure the success of the project.

Overall the proposed methodology suggests the implementation of the project in three phases, not far from the traditional software development practices like water flow model as follows:

1. The **initial** stage, collect information and cost estimation.
2. The **development** stage, develop the Front-end and back-end of the project.
3. The **review** and **delivery** stage, review and deliver the project.

4.5.1 The initial stage

Collect information and cost estimation. This stage very important stage determines the success of each development process, there are information will be collected to guarantee success of project such as functional requirements, determine the level of quality required, estimate the cost and time required to complete the project, identify the risks and how to manage risks, develop and implementation plan. Whole this tasks must be done

collaboratively between the customer and project manager, each according to his responsibility, divide on two main phase the first one *written requirements report* and the second phase determine the *technical requirements* of the project, as follows:

Write requirements report, requirements Report use to identification the functional and non-functional requirements basic on the customer needs accurate, clear and unambiguous based on a specific template report called software requirements specification report (SRS), activity for writing requirements report should be as follow:

1. Customer provide project manager the first draft or simple list for his main requirements, PM use Customer list to decisions what is applicable and what isn't, called *the initial requirements meeting*.
2. PM coordinated requirements clearly and detailed, identify the priority of each required according to customer needs and then sent report to customer (Proposal SRS).
3. Customer receives a report from project manager after formulated and arranged to make any observation or acceptance then a signature as a measure of acceptance and reliance, must be accept the final report on meeting if it possible called *the final requirements meeting*.

By the end of this stage we have got requirements report clear, regular and agreed between the project manager and the customer.

Technical Project specification, Software Technical specification use to determine the technical requirements of project and punctuated by several importance tasks at this stage determined date to start development process, estimate the total time needed to complete the project, estimate the total cost, resources needed to complete the project, assess the level of risks and how to manage risks and then project plan.
Software Technical specification activity will be done by a set of template reports and sequence of stage as below:

1. Based on the one report template called Software Technical specification project manager will be identifying the technical requirements that raise efficiency operation and extent of compatibility with software quality standard, time to complete the project and the financial cost, must be determined the final technical specification on meeting called technical requirements meeting.

2. Customer receives the final technical report from the project manager to make any observation or acceptance and then signature as a measure of acceptance and reliance.

By the end of this stage project manager will be obtained the reports are clear and agreed upon form functional and non-functional requirements and other reports about risk management, project plan and then identify activity to time management mechanism as Milestone and preparation everything to starting development process.

Paid mechanism**,** The most project management methodologies don't care or don't determine paid mechanism or method and leave it to agreement between the customer and supplier responsible to develop software but in this proposed methodology requires that, the customer pays the value of 30% to 50% of the total cost to start development process and anything that will be ensures in the final contract and payment dates.

4.5.2 Development stage

At this stage must build the structure of the system and all its components according to reports and documents which are agreed with the customer. Include tow sub stages, **design** and **writing code** as follows:

4.5.2.1 Design stage

Initial design and develop the Front-end of the project. At this stage start to implement the first component of the project, which shows the general shape of the project, the basic forms and pages to perform tasks of the system requirements, at this stage must pay attention to customer satisfaction as an essential element to continuing. The design process is completed correctly through the following steps:

1. Initial phase in which they are designing a prototype of the system will be develop by using any tools that specializing to prototype development or Photoshop.
2. Send the prototype to customer for evaluate and express an opinion about the prototype in terms of colors and tools distribution, at the same stage the customer sent the system content like usage agreement or information about the organization Etc.

3. Implementation the final design for project or the Front-end that agreed with the customer by using the techniques deployed in the field, such as HTML5 and CSS3, taking into account the concepts of SEO in the design phase.

By the end of this phase will be build the Front-end of the system that agreed to the customer and may be need to make some modification to design on the writing code stage but do not affect the shape of the basic design concepts and taking into account the quality of the design at this stage.

4.5.2.2 Writing code stage

Develop back-end of the project, at this stage developer start to writing code for project components and taking in their account time scheduling and level of quality for each project tasks according to project plan, effort division and the rules that govern members of the team in the implementation of project, which will be detailed later in this chapter. The responsibilities of project manager at this stage manage and control development process and evaluation progressing process. Preferably technically works must be centralized, that means all works must be on the main server to help development team to communicate between themselves and on the other hand with project owner.

Finish development touches, after the end of the stage of writing code for all project component according to requirements report and make sure it works correctly and without errors, customer expectation and make sure about the good integration between back-end and front-end within a single system effectively and then make some final touches to the project based on tests and observations of the development team as ending of the current stage.

Security test, the most importance challenge for development a good software product it is the threats or security weaknesses is one of the biggest challenges faced by the development team and other people responsible for information security so at this stage we will identify a set of steps and activities to test the product that has been developed to discover any security issue or vulnerability and then fixed issues. To test the product use manual testing to found the logical threads, on the other hand use the security scanners free or commercial copy like Burst or IBM AppScan to discover a long list of security issues. The main important vulnerability which should be focus at this stage the TOP 10 OWASP provided by the open web

application security project [9] but there are hundreds of security issue and considers this the most dangerous and the popular and foams and then are commitment to removed any security threat has been discovered immediately.

Beta Presentation, after confirming the development of the whole components of the software project correctly, without errors and make some adjustments there are a meeting holding called *Beta Presentation* brings all team members except the customer or project owner to discuss what was developed specifically and matched the current output with the predictable output according to requirements report, software quality standards and then testing usability of the software product. The meeting is considered as a procedure for completion of the software development phase and preparing of the review and delivery stage.

By the end of development stage the team will be completed develop the project correctly and without errors according to a requirements report, advanced level of protection and insurance it at *Beta Percentation* then the final works is ready for submission to customer to get his final opinion or observation.

4.5.2.3 The review and delivery stage

Review and deliver project component. the end of development stage determined by the project manager after development team make sure about all the component has been done and project is ready to susceptible to customer. At this stage project manager must be officially inform the project owner or customer about the complete of the development process, preferably commitment to schedule time specified in advance and that its end is expected delivery of the project.

Review and checklist created, at this stage, customer or project owner will be chick project components and make sure its work correctly, expected and compared project component with its functional needs and determine the level of compatibility. Customer write all his observation about whole project component on report called a *Checklist review form* and then send the report to PM contains his observations about project for spelling mistakes and grammatical, links, project style, usability and accessibility, etc each element on reports has an identification number and order of priority, as final assessment to discussed with project manager.

Meeting to discuss checklist, project manager received *Checklist review form* from customer and must be commit and his team to rework for all customer observation immediately, at this stage there is very importance meeting to discuss checklist called checklist discuss meeting between the customer and project manager to understand the customer observation clearly and re-work implementation.

Checklist implementation, at this stage project manager must be received the final checklist form customer contain tasks that need to be modified and Review, which was agreed upon in advance between customer or project owner, which are often simple form or report explore location of each task or defect and then project manager verified of the whole tasks if they are exist or not and writing notes about each task.

Project Manager held a small meeting with development team and explains each task and discusses the proposed amendment by team members and at the same meeting makes a plan to complete works that identified in checklist and determine the role of each member in the implementation process.

Implementation plan that are developed by project manager carefully and effectiveness way one by one task until the last one in the list, taking into account the time schedule.

After complete the implementation process of the checklist should be checked again to ensure the whole tasks have been implemented successfully and then project manager call customer or project owner and inform him about the work is done and must be review or verify it again by himself.

Review checklist, at this stage, customer makes sure all the tasks in the checklist are complete and implementation correctly by providing the customer form or a simple report contains all tasks arranged by the degree of priority and importance to the customer or supplier. Whole the answer must be simple and directly about fixed and complete the task or not.

Sign-off the project, after the customer verified whole issues and items on the checklist are fixed and implemented correctly must sign a simple form called *Project Sign-out Sheet* acknowledging that the project developed correctly, without errors, as expected, according to project requirements

specification document, project technical specification and ISO/ICE 25010 standard for software quality assessment and improvement.

Final ratio paid, at this methodology indicated earlier it is care about payment management process as an integral part of the project management process and at this stage customer is obliged to pay the residual value of the project cost, which be agreed upon in advance and must be between 70% - 50% as a prerequisite to complete the publishing process.

Published, after making sure that the customer complete the residual value of project paid it was also agreed upon beforehand all the project files are uploaded on the internet server for actual operation. Is delivered customer project files in case he/she wanted to publish the project or the system on the own special server or other host and assist in the publishing process if requested.

In brief in the preceding paragraphs making an overview about all stages of the proposed methodology, consisting of four basic stages as follows:

1. The initial stage.
2. The development stage.
3. The review and delivery stage.

In subsequent section of this chapter I will give accurate detail for each procedure at each stage, and the documents required to ensure the process will be easy, simple and according to specific criteria and standers must be adhered to ensure the quality of software product.

4.6 Main features of the proposal methodology

One of the most important challenge of the current study, how to develop methodology used to manage projects development process and improve the quality of the software meet the needs of small and medium enterprises simple, ease of use and inexpensive. Some of the most important features of the proposed methodology:

1. Simple, easy to use and inexpensive.
2. Designed to solve the problems and challenges faced small and medium enterprises to apply software quality standards.
3. Apply level tow of the CMMI form “project management”.
4. Integration of CMMI with ISO/IEC 2510.

5. Measurement and Analyze data for stages.
6. Develop reports and document to mange development process.
7. Based institutional and clear standards to manage development process.

Study will prove above values through a set of tools that are outlined in the previous section (Research methodology) and solving the problems faced small and medium enterprises about how to apply software quality standards and software quality management.

4.7 Proposal Works

Subsequent chapters I will explain in more detail the proposed method and include documents, measurement and analysis tools use in each stage to ensure the development.

4.7.1 The initial stage

Collect information, requirements, technical specification, cost and time estimation according to specific software quality standards. This stage very important stage and its determines success of each development process stage, functional requirements, determine the level of quality, assessment budget and number of employees required to develop project, identify the risks and how to manage it, develop and implementation plan. There are two main phases at this stage the first one written requirements report and the second phase determines the technical specification of the project.

4.7.1.1 Software requirements specification

Requirements Report called software requirements specification (SRS) use to identification the functional and basic requirements accurate, simple and unambiguous, explains why product is needed and describes what the finished product will be like the biggest part of the requirements document is the formal list for requirements.

Requirements include description the system properties, specifications and how the system must be work, constraints placed upon the development process, should not include design solutions except for interface requirements often include embedded design [10].

Requirements come from end users, customers, and sometimes from developers. End users tend to state requirements in descriptive or narrative terms which might need to be broken down into individual requirement statements.

It's important to have users, customers, and sometimes developers contribute to the requirements document to create a fuller description for system. The practice including these groups also helps to ensure that everyone is in agreement about and what is to be done before development begins [10].

At this study will write software requirements specification according to recommended practice for software requirements specification (IEEE) [10] and characteristic of ISO / IEC 25010 stander.

This clause provides background information that should be considered when writing SRS. This includes the following: [10]

Nature of the SRS is a specification for a particular software product, program, or set of programs that performs certain functions in a specific environment. The SRS may be written by one or more represent atives of the supplier, one or more represent atives of the customer, or by both. The basic issues that the SRS writer(s) shall address are the following:

1. *Functionality.* What is the software supposed to do?
2. *External interfaces.* How does the software interact with people, the system's hardware, other hardware, and other software?
3. *Performance.* What is the speed, availability, response time, recovery time of various software functions, etc.?
4. *Attributes.* What is the portability, correctness, maintainability, security, etc. considerations?
5. *Design constraints imposed on an implementation.* Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

Environment of the SRS It is important to consider the part that the SRS plays in the total project plan. The software may contain essentially all the functionality of the project or it may be part of a larger system. In the latter

case typically there will be an SRS that will state the interfaces between the system and its software portion, and will place external performance and functionality requirements upon the software portion. Of course the SRS should then agree with and expand upon these system requirements.

Since the SRS has a specific role to play in the software development process, the SRS writer(s) should be careful not to go beyond the bounds of that role. This means the SRS the activity of the write requirements report should be as follow:

1. The customer provide project manager the first draft or clear list for his functional and main requirements and negotiated , called *the initial requirements meeting* and must be meeting if it is possible.
2. The project manager formulate requirements clearly a coordinated and detailed, determining the priority of each requirement easily understood and then sent it to the customer.
3. The customer receives a report from the project manager after formulated and arranged to make any observation or acceptance and then the signature as a measure of acceptance and reliance, called *the final requirements meeting* and must be meeting if it is possible.

By the end of this stage we have got requirements report clear, regular and agreed between the project manager and the customer.

4.7.1.2 Characteristics of a good (SRS)

The main characteristics of software requirements specification should be as follow:

1. **Correct,** An SRS is correct if, every requirement stated there in is one that the software shall meet. There is no tool or procedure that ensures correctness. The SRS should be compared with any applicable superior specification, such as a system requirements specification, with other project documentation, and with other applicable standards, to ensure that it agrees. Alternatively the customer or user can determine if the SRS correctly reflects the actual needs. Traceability makes this procedure easier and less prone to error.

2. **Unambiguous,** An SRS is unambiguous if, every requirement stated therein has only one interpretation. As a minimum, this requires that each characteristic of the final product be described using a single unique term.

3. **Complete,** An SRS is complete if, it includes the following elements:
    - All significant requirements, whether relating to functionality, performance, design constraints, attributes, or external interfaces. In particular any external requirements imposed by a system specification should be acknowledged and treated.
    - Definition of the responses of the software to all realizable classes of input data in all realizable classes of situations. Note that it is important to specify the responses to both valid and invalid input values.
    - Full labels and references to all figures, tables, and diagrams in the SRS and definition of all terms and units of measure.

4. **Consistency** refers to internal consistency. If an SRS does not agree with some higher-level document, such as a system requirements specification, then it is not correct. A must for not conflict between the requirements of SRS.

5. **Ranked for importance** and/or stability An SRS is ranked for importance and/or stability if each requirement in it has an identifier to indicate either the importance of stability of that particular requirement. Typically, all of the requirements that relate to a software product are not equally important. Some requirements may be essential, especially for life-critical applications, while others may be desirable. There are factors effect on the ranked for importance :

    Degree of stability

    One method of identifying requirements uses the dimension of stability. Stability can be expressed in terms of the number of expected changes to any requirement based on experience or knowledge of forthcoming events that affect the organization, functions, and people supported by the software system.

Degree of necessity

Another way to rank requirements is to distinguish classes of requirements as essential, conditional, and optional.

- *Essential.* Implies that the software will not be acceptable unless these requirements are provided in an agreed manner.
- *Conditional.* Implies that these are requirements that would enhance the software product, but would not make it unacceptable if they are absent.
- *Optional.* Implies a class of functions that may or may not be worthwhile. This gives the supplier the opportunity to propose something that exceeds the SRS.

6. **Verifiable,** An SRS is verifiable if, every requirement stated there in is verifiable. A requirement is verifiable if, and only if, there exists some finite cost-effective process with which a person or machine can check that the software product meets the requirement. In general any ambiguous requirement is not verifiable.

7. Modifiable, An SRS is modifiable if, and only if, its structure and style are such that any changes to the requirements can be made easily, completely, and consistently while retaining the structure and style. Modifiability generally requires an SRS to
   - Have a coherent and easy-to-use organization with a table of contents, an index, and explicit cross-referencing.
   - Not be redundant (i.e., the same requirement should not appear in more than one place in the SRS).
   - Express each requirement separately, rather than intermixed with other requirements.

The above criteria ensure to write a good SRS covers all the requirements of the customer and positively affects in the development process and customer satisfaction.

Software development process should begin with project manager and customer to agreement on what is the software completed must do. Agreement to (SRS) should be jointly prepared it is important because usually either the customer or supplier is qualified to write a good SRS alone because the following factors:

1. Customers usually do not understand the software design and development process well enough to write a usable SRS.
2. Suppliers usually do not understand the customer's problem and field of endeavor well enough to specify requirements for a satisfactory system.

A special situation exists when a system are being defined congruent specification as functionality, interfaces, performance, and other attributes and constraints of the software are not predefined, but rather are jointly defined and subject to negotiation. This makes it more difficult but no less important, to meet the characteristics a good (SRS).

SRS evolution, SRS may need to evolve as the development of software product progresses. It may be impossible to specify some details at the time project is initiated (e.g., it may be impossible to define all of the screen formats for an interactive program during the requirements phase). Additional changes may ensue as deficiencies, shortcomings, and inaccuracies are discovered in the SRS. There are two major considerations in this process are the following: [10]

1. Requirements should be specified as completely and thoroughly as is known at the time, even if evolutionary revisions can be foreseen as inevitable. The fact that they are incomplete should be noted.
2. A formal change process should be initiated to identify, control, track, and report projected changes. Approved changes in requirements should be incorporated in the SRS in such a way as to
   - Provide an accurate and complete audit trail of changes.
   - Permit the review of current and superseded portions of the SRS.

Prototyping is used frequently during the requirements portion of a project. Many tools exist that allow a prototype, exhibiting some characteristics of a system, to be created very quickly and easily. Prototypes are useful for the following reasons:

1. Customer may be more likely to view the prototype and react to it than to read the SRS and react to it. Thus, the prototype provides quick feedback.
2. Prototype displays unanticipated aspects of the systems behavior. Thus, it produces not only answers but also new questions. This helps reach closure on the SRS.

3. An SRS based on a prototype tends to undergo less change during development, thus shortening development time.

Embedding design in the SRS, requirement specifies an externally visible function or attribute of a system. A design describes a particular subcomponent of a system and/or its interfaces with other subcomponents. The SRS writer(s) should clearly distinguish between identifying required design constraints and projecting a specific design.

The SRS should specify what functions are to be performed on what data to produce what results at what location for whom. The SRS should focus on the services to be performed. The SRS should not normally specify design items such as the following:

1. Partitioning the software into modules.
2. Allocating functions to the modules.
3. Describing the flow of information or control between modules.
4. Choosing data structures.

Writing a requirements report to meet whole the customer functional and non-functional needs, describes all the components of the proposed system clearly and accurately, meets the needs of the end user, some characteristics of the software quality recommended by the developers and software quality model. This study develops SRS template according to recommended practices for software requirements specifications (IEEE) [10], Software requirements specification [11] and ISO/IEC 25010 software quality for assessment and improvement [1].

4.7.1.3 Software requirements specification section

Proposal software requirements specification (SRS) content seven sections the first one ***Introduction*** content project overview and project scope, ***General description*** content project state, project user characteristics, project assumption ***project requirements*** content functional requirements, user interface requirements, performance efficiency, operability, security, compatibility, Maintainability and transferability ***product user senior and use case*** - ***Appendix*** - ***References*** and the last one ***index*** , for more details proposal software requirements specification at Appendix 1 .

4.7.1.4 Technical project scope

Main activities in the initial stage collect information and specify customer requirements on two sub stages first one software requirement specification

(SRS) has been discussed in detail. The second sub stage write report about technical project scope includes project objective, project owner, sponsors and stakeholders, specify high-level project requirements, milestones and outcomes, project constraints, cost estimates and other activity to make project more manageable. Typical components of a project scope statement include a project objective, justification, product description, expected outcomes, assumptions and limitations. Written a technical project scope on the follow activity:

1. Project manager wrote the first Draft about technical project scope according to information derived from the customer. Preferably in direct meeting called a *technical project scope meeting* to discuss technical scope in details.
2. Project manager formulate requirements clearly, coordinated and detailed for each section to be easy to understand and then sent it to the customer.
3. Customer receives a final document from project manager to make his observation or acceptance and then the signature as acceptance and reliance.

Project manager must be specifying project scope no matter, project timeline, project goals and allocating project resources. These are activities help PM deliver project on time and on budget as below:

Scope, One of the first priorities in software development process implementation the full scope of the project according to the end goal of the project is to be 'functional requirements ', given budget to the software project's completion and what departments will need to be involved. A plan is written detailing key players of the project, such as the project manager, his contact information, the approach that will be used to begin the project and the roles and key responsibilities of the software development team and whole technical specification in the technical project specification document.

Time, during time management activities, the needs of the software creation and development are broken down into sections and divided among team members for completion. The tasks are to be completed by certain times, one step at a time, or are to be completed by a deadline with different teams working on separate tasks at the same time. Each software creation project is different. Staff numbers are reviewed, special skills sets analyzed and teams

created based on strengths. Each step of the project will be put against a time constraint with each section having its own deadline.

Integration, Another team is put together specifically to discuss and analyze the full feasibility and benefit of completing the project. This team will interview end users who interact with the company's current software applications. Information technology employees who maintain the current company network will discuss any issues that may arise with the deployment of the new software, such as managing downtimes, and which method is best for testing.

Quality, Quality management takes place before, during and after the creation and deployment of the new software. Charts will be made and the software will be tested against business expectations, such as whether the software processes data and stores it correctly, or retrieves and displays correct information to the user. End users will provide feedback and comments about the user-friendliness of the software and any glitches they may experience. Quality assurance involves monitoring the financial costs of building the software, such as the hours put in by programmers and the project management team. The project manager will measure the successfulness of the software creation project by analyzing final reports.

The current Study proposes a simple document for managing a technical specification stage at appendix 1:

4.7.2 The development stage

Consist of three sub stages design, coding and then testing at design phase develop project Front-end.

4.7.2.1 Design stage

Consider as the first phase at project implementation, which shows the general project shape, basic forms and pages to perform tasks or requirements and at this stage must pay attention to customer satisfaction. Design phase must be completed correctly through the following steps:

1. The first step designing a prototype of the system will be develop by using any tools that specializing in the prototype development or Photoshop.

2. Send prototype to customer for evaluate and provide his observation about prototype for example colors, fonts and tools distribution if the customer accept prototype at the same stage sent system content like usage agreement or information about the organization Etc.
3. Implementation the final design Front-end that agreed by customer using development techniques such as HTML5 and CSS3, taking into account the SEO concepts.

4.7.2.2 Tips to design a perfect web form layout

There are some tips to design a perfect website layout '*front-end*', cover what must believe every new website builder working within a small and medium enterprise should know and do before starting a new project, and what they should pay attention to during the process.

These principles cover not only design aspects such as landing page design but also how to manage design step. Useful for designer on small and medium enterprise to creating professional website layouts, as follow:

1. Put your thoughts on paper first, Very initial sketches. Don't use any tool like *Photoshop* before giving any thought to the problem they are trying to solve. Design is about solving problems and those problems can't be resolved through without a good layout and a clear hierarchy. Think about the content, the layout and the functionality before starting to build HTML stricter.

2. Start sketching a top level framework, Sketching a basic wires will help designer to resolve problems and to structure the layout. The first thing must be do is come-up with a top level framework that solves all the design problems. The framework is the User Interface that surrounds the content and helps to perform actions and navigate through it.

3. **Choose your typography**, Exploring different typefaces and colors is part of the discovery phase of a project. I would recommend not using more than two different typefaces in a website but it really depends on the nature of the project to use more or less.

4. **Select your colors theme**, Use a limited set of colors and tones to guard against visual overload. Throughout the process of choosing a set of typefaces to use and must start exploring what colors you will use in the user interface, backgrounds, and text.

5. **Divide the layout**, the simpler the structure of the site, the easier it is for users to navigate. It needs a reason and a final outcome for the user. The layout needs to help the content highlighting and what is the most important part in the website. In reality there shouldn't be too many call outs on a page so everything should drive to that final "What can I do here".

6. **Rethink the established**, Shape the way users browse the internet, it's up to us to decide how many steps a simple action will take and how efficient website will be. Design patterns and conventions are there because they work but sometimes they are there because no one spent enough time evaluating them or rethinking them. It's important to rethink the established interactive patterns on any component and to see how we can improve them.

7. **Pay attention to the details**, Game work in progress: detail view this statement has been overused lately but it's not always visible in the final product. Depending on the concept behind the project. It could be a small interaction, an unexpected animation or an aesthetic touch like a little gradient in a button or a subtle stroke around a box in the background. But overall this touch is essential and also natural if you really enjoy what you do.

8. **Design the best case scenario but prepare for the worst**, Bear in mind how your layout will work on different devices and screen sizes. As designers our job is to solve problems through different constraints. With web design, the constraints range from conceptual and technical issues to content related issues. For instance a user could be using a really small screen and check the site when there is barely any content on it so it looks broken. The study proposes to build website based on responsive design.

9. **Presentations**, The best design can be ignored or thrown away if don't Percent it properly. Always keep in mind that what is totally clear to you might not be that clear to somebody seeing your design for the first time. There is important concepts must be care about it on design step reflect on the project efficiency and quality as Search Engine Optimizations (SEO), security on user side.
10. **Search engine optimization or SEO** is a set of rules that can be followed by web site owners to optimize their web sites for search engines and thus improve their search engine rankings.

In addition it is also a great way to increase the quality of their web sites by making them user- friendly, faster and easier to navigate. SEO can also be considered as a framework since the whole process has a number of rules (or guidelines), a number of stages and a set of controls.

4.7.2.2 Search engine optimization

In today's competitive market SEO is more important than ever. Search engines serve millions of users per day looking for answers to their questions or for solutions to their problems. SEO can help business grow and meet the business objectives. Search engine optimization is essential because:

- The majority of search engines users are more likely to choose one of the top 5 suggestions in the results page.
- SEO is not only about search engines but good SEO practices improve the user experience and usability of a web site.
- User's trust search engines and having a presence in the top positions for the keywords the user is searching increases the web site's trust.
- SEO is also good for the social promotion of the web site. People who find the web site by searching Google or Yahoo are more likely to promote it on social networks.
- SEO is also important for the smooth running of a big web site. Web sites with more than one author can benefit from SEO in a direct and indirect way. Their direct benefit is increase in search engine traffic and their indirect benefit is having a common framework (checklists) to use before publishing content on the site.
- SEO can put you ahead of the competition. If two web sites are selling the same thing, the search engine optimized web site is more likely to have more customers and make more sales.

SEO above definition may sound complicated so in simpler terms Search Engine Optimization is a way to improve web site so that it will appear closer to the top positions in the search results of search engines.

When you perform a search on search engine the order by which the returning results are displayed, is based on complex algorithms. These algorithms take a number of factors into account to decide which web site should be shown in the first place, second place etc.

In general, search engines are trying to find the most important pages of the web and show those first when a user enters a search query. One of the factors to determine the position a web page will appear in the results is the number of incoming links.

Incoming links are a signal of trust and depending from where the links are coming, they can greatly affect your ranking position (either positively if the links are coming from well-known and trusted sites or negatively if they are paid links, article directories, link farms etc.).

SEO is one of the tools available in the Internet Marketing arsenal. It is not Internet Marketing as such but it can be part of your overall **Internet Marketing campaign** which normally includes other things like social media promotion, content strategy etc.

To sum it up, Search engine optimization or SEO is a way to optimize the web site so that search engines will understand it better and give the higher rankings. It is important since a good SEO approach can drive more traffic to the web site, blog or on-line store and gain more customers, make sales and fulfill business purpose.

By the end of this phase will be build the Front-end of the system that agreed to the customer and may be need to make some modification to design on the writing code stage but do not affect the shape of the basic design concepts and taking into account the quality of the design at this stage.

4.7.3 Code writing

Writing the first lines of code, implementation project structure and components, through several activities related project management such as time management - budget - team work and requires considerable effort from the project manager and developers should be taking into account the three basic concepts at this stage writing code standards, defiance programming and central implementation.

4.7.3 .1 Defensive programming

Defensive programming design intended to ensure the continuing function of a piece of software under unforeseen circumstances. Defensive programming techniques are used especially when a piece of software could be misused, improve software and source code, in terms of:

- General quality - reducing the number of software bugs and problems.
- Making the source code comprehensible - the source code should be readable and understandable so it is approved in a code audit.
- Making the software behave in a predictable manner despite unexpected inputs or user actions.

Overly defensive programming however introduces code to prevent errors that can't happen, but needs to be executed on runtime and to be maintained by the developers, thus increasing the runtime and maintenance costs. There is also the risk that the code catches or prevents too many exceptions. In those cases, the error would be suppressed and go unnoticed, while the result would be still wrong. There are some defensive programming techniques as follow:

- *Intelligent source code reuse*, if existing code is tested and known to work, reusing it may reduce the chance of bugs being introduced. However, reusing code is not always a good practice, particularly when business logic is involved. Reuse in this case may cause serious business process bugs.
- *Legacy problems*, before reusing old source code, libraries, APIs, configurations and so forth, it must be considered if the old work is valid for reuse, or if it is likely to be prone to legacy problems. Legacy problems are problems inherent when old designs are expected to work with today's requirements, especially when the old designs were not developed or tested with those requirements in mind.

- *Low tolerance against "potential" bugs*, Assume that code constructs that appear to be problem prone (similar to known vulnerabilities, etc.) are bugs and potential security flaws. The basic rule of thumb is: "I'm not aware of all types of security exploits. I must protect against those I *do* know of and then I must be proactive!"

4.7.3 .2 Roles for writing code

Most serious software development projects use coding guidelines these guidelines are meant to state what the ground rules are for the software to be written: how it should be structured and which language features should and should not be used.

The benefit of existing coding guidelines is therefore often small, even for critical applications. A verifiable set of well-chosen coding rules could, however, make critical software components more thoroughly analyzable, for properties that go beyond compliance with the set of rules itself.

There are very importance's roles must be used during code writing stage summarize by *Rules for Developing Safety Critical Code* [12] as below*:*

1. Restrict all code to very simple control flow constructs – do not use *goto* statements, *setjmp* or *longjmp* constructs, and direct or indirect *recursion*.
2. All loops must have a fixed upper-bound. It must be trivially possible for a checking tool to *prove* statically that a preset upper-bound on the number of iterations of a loop cannot be exceeded. If the loop-bound cannot be proven statically, the rule is considered violated.
3. Do not use dynamic memory allocation after initialization. This rule is common for safety critical software and appears in most coding guidelines.
4. No function should be longer than what can be printed on a single sheet of paper in a standard reference format with one line per statement and one line per declaration. Typically, this means no more than about 60 lines of code per function
5. The assertion density of the code should average to a minimum of two assertions per function. Assertions are used to check for anomalous conditions that should never happen in real-life executions. Assertions must always be side-effect free and should be defined as Boolean tests. When an assertion fails, an explicit recovery action must be

taken, e.g., by returning an error condition to the caller of the function that executes the failing assertion. Any assertion for which a static checking tool can prove that it can ever fail or never hold violates this rule. (I.e., it is not possible to satisfy the rule by adding unhelpful "assert (true)" statements.)

6. Data objects must be declared at the smallest possible level of scope. This rule supports a basic principle of data-hiding. Clearly if an object is not in scope, its value cannot be referenced or corrupted.
7. The return value of non-void functions must be checked by each calling function, and the validity of parameters must be checked inside each function.
8. Simple macro definitions. Token pasting, variable argument lists (ellipses), and recursive macro calls are not allowed. All macros must expand into complete syntactic units. The use of conditional compilation directives is often also dubious, but cannot always be avoided. This means that there should rarely be justification for more than one or two conditional compilation directives even in large software development efforts, beyond the standard boilerplate that avoids multiple inclusion of the same header file. Each such use should be flagged by a tool-based checker and justified in the code.
9. The use of pointers should be restricted. Specifically, no more than one level of dereferencing is allowed. Pointer dereference operations may not be hidden in macro definitions. Function pointers are not permitted.
10. All code must be compiled, from the first day of development, with all compiler warnings enabled at the compiler's most pedantic setting. All code must compile with these setting without any warnings. All code must be checked daily with at least one, but preferably more than one, state-of-the-art static source code analyzer and should pass the analyses with zero warnings.

There are several very effective static source code analyzers on the market today, and quite a few freeware tools as well. There simply is no excuse for any software development effort not to make use of this readily available technology.

For the importance of the above rules project manager must be remind the developers of these rules before and during development stage and should be tested by the expert's people.

4.7.3 .3 Software project security

Develop a secure software product the most important thing and one of the biggest challenges faced development team. By the end of development stage development team must be insure the software product is secure. The current study identifies the acceptance security level and recommended tools and technology to discover any security issue or vulnerability.

Insecure software is undermining financial, healthcare, defense, energy, and other critical infrastructure. As digital infrastructure gets increasingly complex and interconnected, the difficulty of achieving application security increases exponentially. Can no longer afford to tolerate relatively simple security problems like those presented in this OWASP Top 10 [9].

Identify acceptance security level by specify the most important issues according to OWASP that make developers learn from the mistakes of other organizations. Executives should start thinking about how to manage the risk that software applications create in their enterprise.

The OWASP Top 10 focuses on identifying the most serious risks for a broad array of organizations. For each of these risks, provide generic information about likelihood and technical impact using the following simple ratings scheme, which is based on the OWASP Risk Rating Methodology.

OWASP Top 10 Application Security Risks – 2013:

1. **Injection**
   Injection flaws, such as SQL, OS, and LDAP injection occur when untreated data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
2. **Broken Authentication and Session Management**
   Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
3. **Cross-Site Scripting (XSS)**
   XSS flaws occur whenever an application takes entrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

4. **Insecure Direct Object References**
   A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
5. **Security Miss configuration**
   Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
6. **Sensitive Data Exposure**
   Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
7. **Missing Function Level Access Control**
   Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.
8. **Cross-Site Request Forgery (CSRF)**
   A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
9. **Using Components with Known Vulnerabilities**
   Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known

vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

10. **Un-validated Redirects and Forwards**
    Web applications frequently redirect and forward users to other pages and websites, and use untreated data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Devoid any software product of the above vulnerabilities represent s an acceptable level of protection but does not consider whole risks and challenges can face the software, there are more than 500,000 software vulnerability registered [9]. The top 10 were classified based on the first stumbling block proliferation and the possibility of independence, and discovered the extent of its impact.

Software security testing use to discover software vulnerabilities and repair to not allow any threatened can be exploited by hackers. There are two main techniques the first one by using software tools known a security scanners considered the most effective, easily and more prevalent, use to discover a long list of threats and vulnerabilities, the second techniques using manual testing is a very effective way to discover the bugs that are not detectable by scanners such as the logical bugs.

There are many software security scanners that are used to discover security risks which proved very effective in the performance of their work, including commercial and open source copy, in this study we will discuss follow:

- **Burp Suite** is an integration of tools that work together to perform security tests on web applications. It is also a platform for attacking applications on the web. Burp Suite contains all the Burp interfaces and tools made for speeding up and facilitating the process of application attacks. . The various tools of Burp work together seamlessly to allow identified findings and share information within one of the tools to form the foundation of attacks using a different tool. When putting a web application to the test, Burp Suite helps the penetration tester through the process starting from identifying vulnerabilities all the way to the mapping and exploitation phase. Understanding the framework of Burp Suite will help you know when to use which feature with what scenario.

- **OWASP ZAP** is an open source tool designed to help security professionals as well as developers to find out the security vulnerabilities Percent in the web application. One of the most important features is Free Web Application Scanners is that it deserves special attention. Unlike other tools in the market, ZAP helps security newbie's to learn and apply the application security skills in an easy way. ZAP is open source and completely free to use, which also means that users have the opportunity to implement changes which they think would add value to the tool. ZAP comes equipped with many features which can be used to test the overall strength of a web application.

Template for security report at appendix 5:

4.7.4 Beta Presentation

Beta Presentation, very important meeting between software development team members, before product delivered, it is intended to make sure the software product has been developed correctly and covers whole the customer requirement according to the software requirement and specification report, technical project specification with the specific level of quality and standards from the initial stages of development process. The most important agenda of the beta Presentation meeting as follows:

- Ensure the whole project components have been implemented according to the priority of each requirement and report (SRS).
- Ensure the whole project components of the project are working correctly and without errors.
- Make sure to achieve the desired level of quality.
- Ensure the level of protection in the software product achieve the required level (secure).

By the end of this meeting development team members and project manager will be sure about software project that will be delivered it is develop according to the specific requirements, pre-defined criteria and customer expectations.

4.7.3 The review and delivery stage

The end of the development stage determined by the project manager after development team make sure all the component has been developed and project is ready to deliver. At this stage project manager must be officially inform project owner or customer about complete development process, preferably commitment to schedule.

4.7.3.1 Review and checklist created

Review and checklist created, at this stage, customer or project owner will be chick project components and make sure its work correctly, expected and compared project component with its functional needs and determine the level of compatibility. Customer write all his observation about whole project component on report called a *Checklist review form* and then send the report to PM contains his observations about project for spelling mistakes and grammatical, links, project style, usability and accessibility, etc each element on reports has an identification number and order of priority, as final assessment to discussed with project manager. There is sample for *Checklist* at appendix1:

4.7.3.2 Checklist discussion meeting

After the customer specify his observations by using checklist form must be deliver to project manager, and preferably at meeting face-to-face called *checklist discussion meeting* it is intended to receive customer observations one by one, discussed, obliged to fixed whole the issues immediately and then give the customer a general idea about the proposed solution to their observations on the project checklist review report.

4.7.3.3 Checklist rework

When project manager receive the final checklist report from customer and verified the whole report items if they exist or not and writing notes about each issues must be held a small meeting with the development team explains each issue and discuss the proposed solution with them, at the same meeting make a plan to complete the work and determine the role of each member in the rework process.

Project manager monitor implementation rework project plan carefully and effectiveness one by one issues until the last one in the list, taking into account the time schedule.

After Completion of the rework process checklist must be checked again to ensure that whole tasks have been implemented successfully and then project manager call or inform customer or project owner about work is done and must be review or verify it by himself.

4.7.3.4 Review checklist

At this stage customer make sure whole issues in the checklist are fixed and implementation correctly using a simple report called *Review Checklist Form* get its data from previous report '*project checklist form*' delivered to project manager and must be content a simple and direct answer for each item in the list about are the issue fixed or not. The template of review checklist at appendix 1:

4.7.3.5 Sign-off the project

When the customer verified the whole issues and items on the checklist are fixed and implemented correctly must sign a simple form called *Project Sign-out Sheet* acknowledging that the project developed correctly, without errors, as expected, according to project requirements specification document, project technical specification and ISO/ICE 25010 standard for software quality assessment and improvement. A project sign-off procedure usually takes place during the contract closure phase of a project – the project executors should provide results of their work to the customers for examination, and then after all necessary testing has been completed, should get a formal statement from the customers to verify that their job has been successfully done. The template of project sign-off appendix1.

4.7.3.6 Final ratio paid

At the current study and proposal methodology indicated earlier it is care about payment management process as an integral part of the project management process and at this stage the customer is obliged to pay the residual value of the project cost, which be agreed upon in advance and be between 70% - 50% as a prerequisite to complete the publishing process.

4.7.3.7 Published

After making sure the customer complete the residual value of the project paid it was also agreed upon beforehand all the project files are uploaded on the internet server for actual operation. Is delivered customer project files in case he/she wanted to publish the website or the system on the own special server or other host and assist in the publishing process if requested.

4.8 Conclusion

By the end of this chapter, I was cover, overview about study main idea, over view about CMMI level II, most importance project management methodologies (Agile, Scrum, XP), overview about proposal work (initial stage, development stage, Review and delivered stage), main feature about proposal methodology, initial stage (software requirements specification – SRS, SRS section, technical project scope), development stage (design stage, tips to design a prefect web form layout, search engine optimizations), code writing (defensive programming, rules for writing code, software project security, beta Presentation), Review and delivery stage (review and checklist created, checklist desiccation meeting, checklist rework, review checklist, sign-off the project, final ratio paid, publish project or system).

CHAPTER (5)

RESULTS, DISCUSSION AND VALIDATION

5.1 Introduction

In this chapter I will discuss data analytics and result for questionnaires (questionnaire number one – questionnaire number two), I was send invitations (soft copy) by e-mail and publish invitations through social networks to ensure greater coverage and included some tools and feature for avoiding errors to reach truth and accurate results.

Questionnaire number one, collecting information about proposed methodology from viewpoint of researchers, developers, project managers and each one related to software development process on the institutions target through the basic and traditional model for software development process from collect and analyze data, time and cost estimation, design, coding and testing until deliver the software product and finally find out what are the difficulties they face in each of these steps and then analyze the data for giving clear outlines about the method of solution.

Develop proposal methodology according to result of the questioner number one, solution will be tested on the real environment by contacting to a set of institutions and companies that targeted before. Pot the methodology on test and before that should be training one employee or more at least one or two hour about how to use it.

Questionnaire number two, Send proposal methodology for testing in the real development environment and then collecting data about its efficiency and ability to resolve problems. Questionnaire into groups and categories, each category in a set of questions formulating in a clear, simple and easy to read and understand for respondents, define the relationship between category clearly, include all the ideas that relate to the assessment proposal activity, can also obtain notation or sufficient information when we needed to make some modified for methodology.

People Target by questionnaire number two project managers, system engineers, designers and developers or all Jobs that are involved in the software development process during all the stages from system analysis to deliver the project.

After completion analyzing data, we have got clear and accuracy results represent s a summary for study allow us to determine recommendations and how to applying software quality standards on small and medium enterprises.

5.2 Questioner number one

5.2.1 Degree of internal consistency and reliability

Scale: all variables

Case processing summary

	Number	Percent
Valid	50	100.0
Excluded	0	0
Total	50	100.0

Reliability statistics

Alpha	N of items
.985	24

The above table display truth and the internal consistency using SPSS program for test the questions and items study, alpha value reached to (0.985) this means the degree of validity and reliability of this study is very high and this enables us to analyze data and get correct and truthful results.

Analysis based on Likart Scale

A method of ascribing quantitative value to qualitative data, to make it amenable to statistical analysis. A numerical value is assigned to each potential choice and a mean figure for all the responses is computed at the end of the evaluation or survey. Used mainly in training course evaluations and market surveys, Likert scales usually have five potential choices (strongly agree, agree, neutral, disagree, strongly disagree) but sometimes go up to ten or more as below :

Value	Weighted average
Strongly agree	From 5.00 to 4.20
agree	From 4.19 to 3.40
Neutral	From 3.39 to 2.60
Disagree	From 2.59 to 1.80
Strongly disagree	From 1.79 to 1.00

Variable 1-1:

Apply software quality standards in small and medium enterprises ensure the quality of their products.

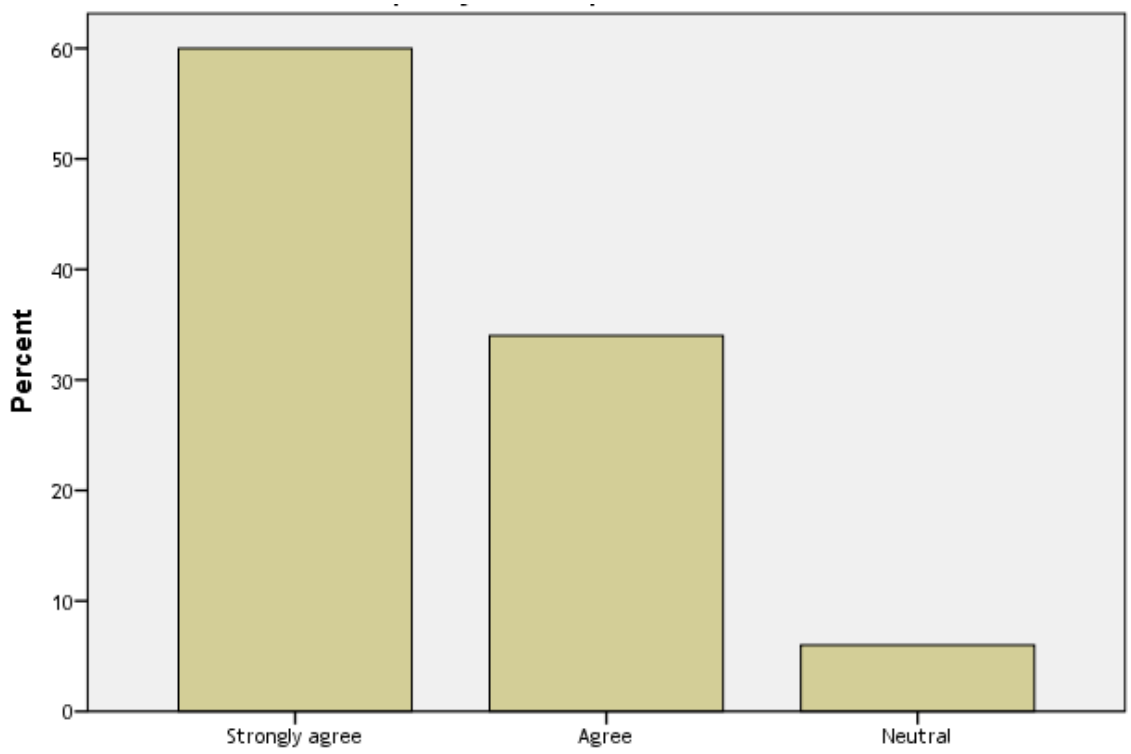
Table NO (5-1)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1	Frequency	30	17	3	0	0	4.54	.6.13	Strongly agree
	Percent	60.0	34.0	6.0	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees

Figure NO (5-1)



Apply software quality standards in small and medium enterprises ensure the quality of their products.

Variable number 1-2:

Financial costs and qualified staff it is the main factors that affect to apply software quality standards in small and medium enterprise.

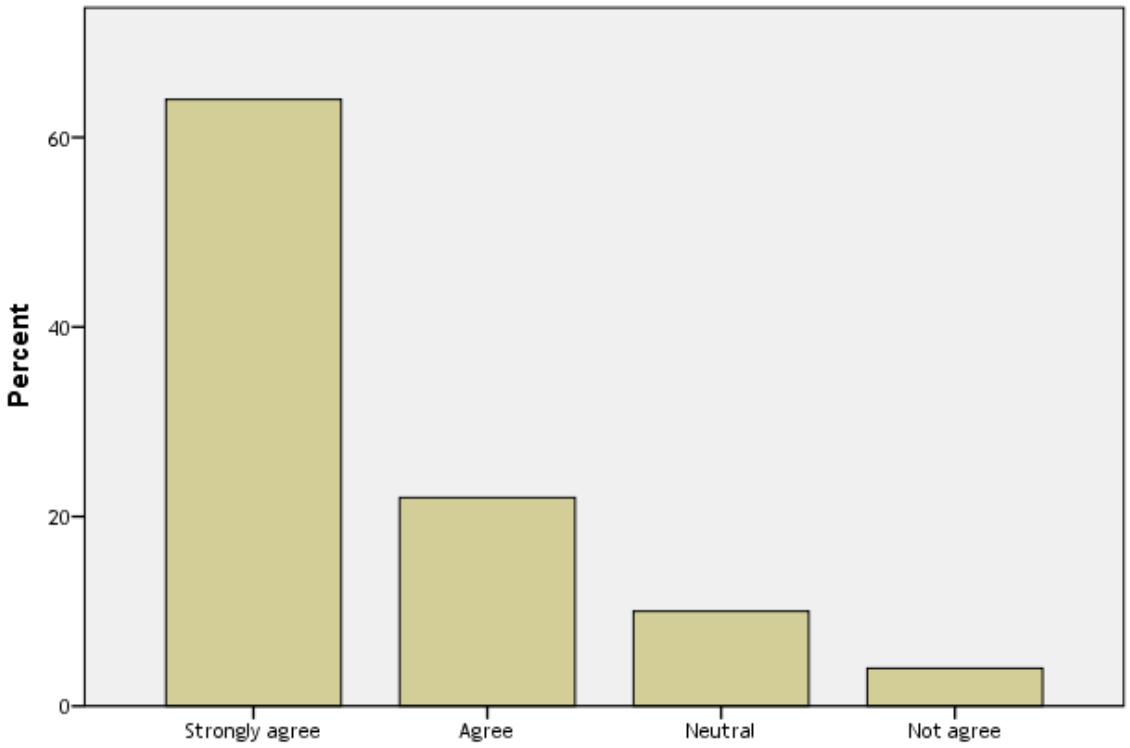
Table NO (5-2)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable#2	Frequency	32	11	5	2	0	4.46	.838	Strongly agree
	Percent	64.0	22.0	10.0	4.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees

Figure NO (5-2)



Financial costs and qualified staff it is the main factors that affect to apply software quality standards in small and medium enterprise.

Variable number 1-3:

Small and large companies have the same ability to apply the software quality standards.

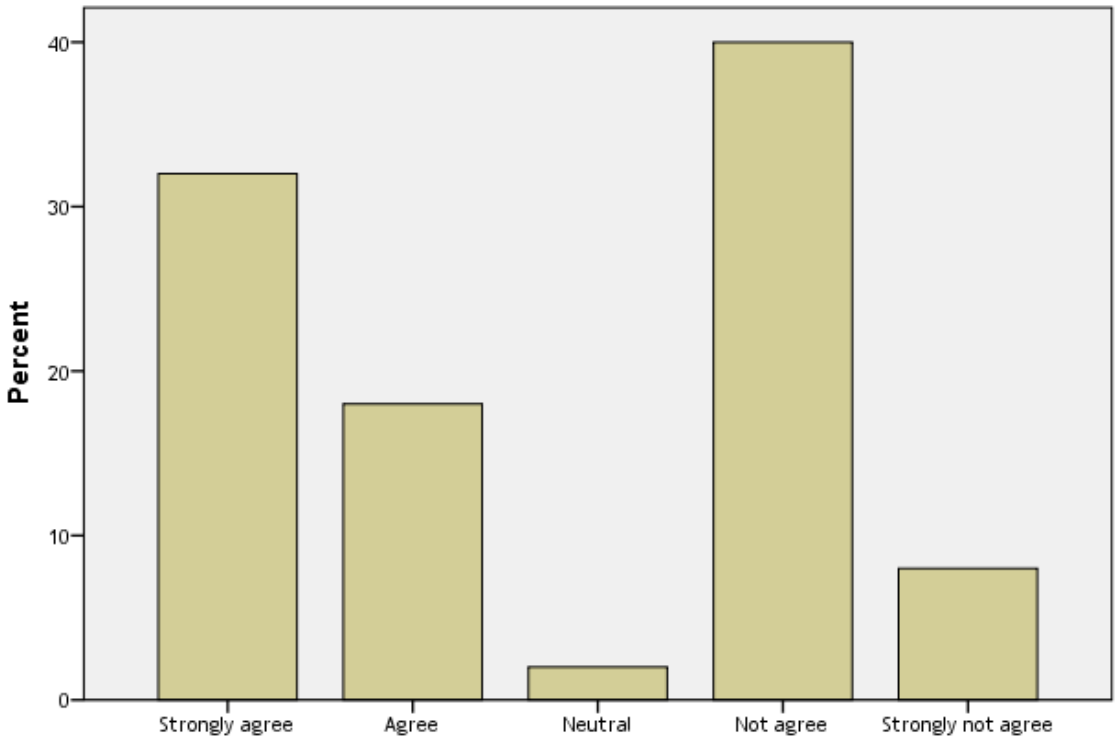
Table NO (5-3)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #3	Frequency	16	9	1	20	4	3.26	1.468	Neutral
	Percent	32.0	18.0	2.0	40.0	8.0			

Source: Case Study 2014

The above table Display the result of variable is: Neutral

Figure NO (5-3)



Small and large companies have the same ability to apply the software quality standards.

Variable 1-4:

Complexity of any software quality standards or project management methodology effect to applicable.

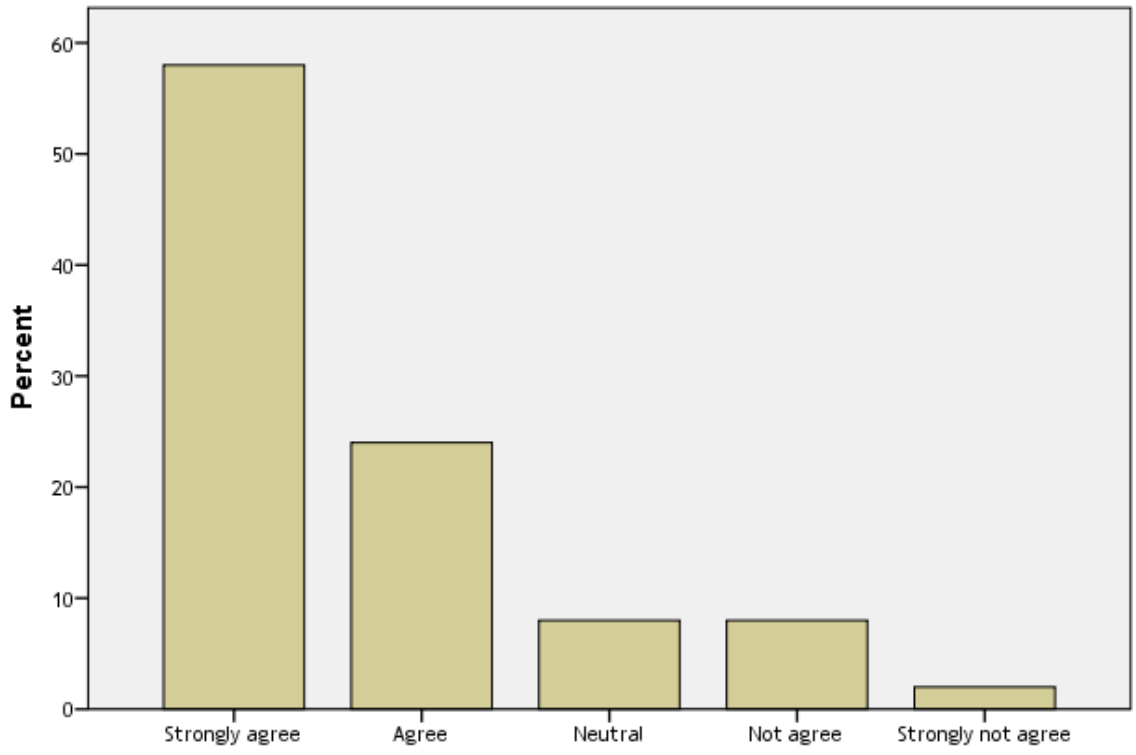
Table NO (5-4)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviati	Result
Variable #1-4	Frequency	29	12	4	4	1	4.28	1.051	Strongly agree
	Percent	58.0	24.0	8.0	8.0	2.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree

Figure NO (5-4)



Complexity of any software quality standards or project management methodology effect to applicable.

Variable 1-5:

Each software quality standards and project management methodologies satisfy the needs of small and medium enterprises and applicable.

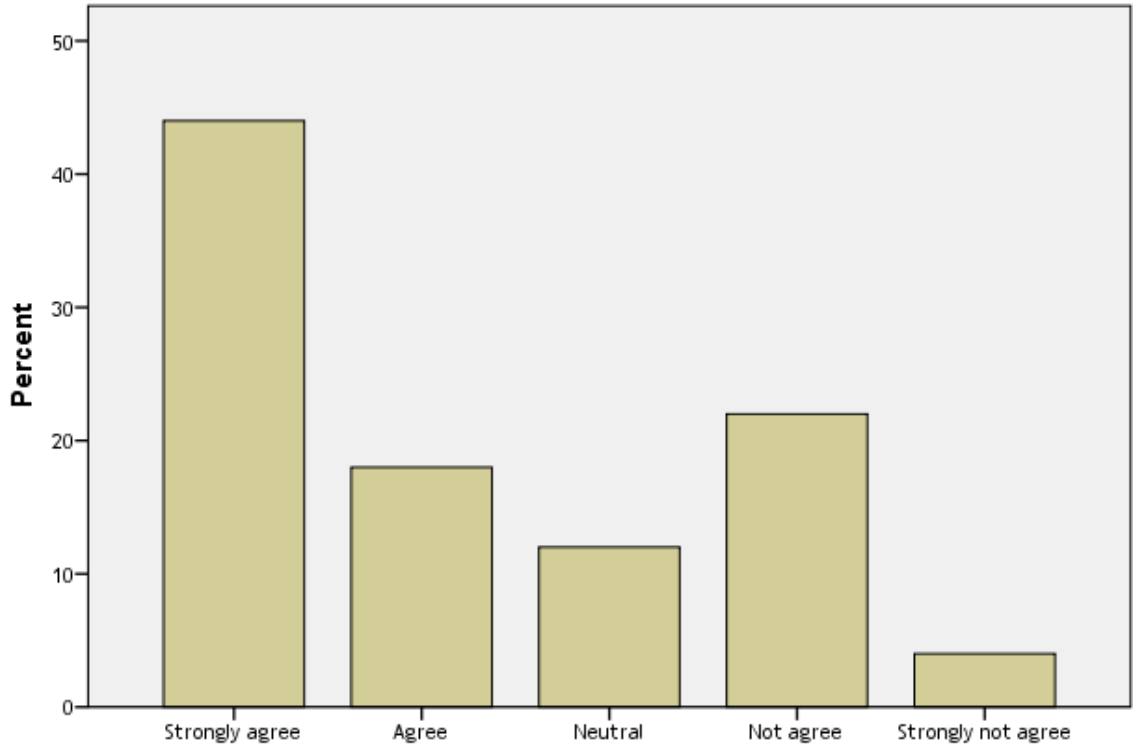
Table NO (5-5)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-5	Frequency	22	9	6	11	2	3.76	1.333	Agree
	Percent	44.0	18.0	12.0	22.0	4.0			

Source: Case Study 2014

The above table Display the result of variable is: Agree

Figure NO (5-5)



Each software quality standards and project management methodologies satisfy the needs of small and medium enterprises and applicable.

Variable 1-6:

Merge two or more software quality standards or project management methodologies with each other to take advantage of their different characteristics in one model ensures the effectiveness of the resulting model.

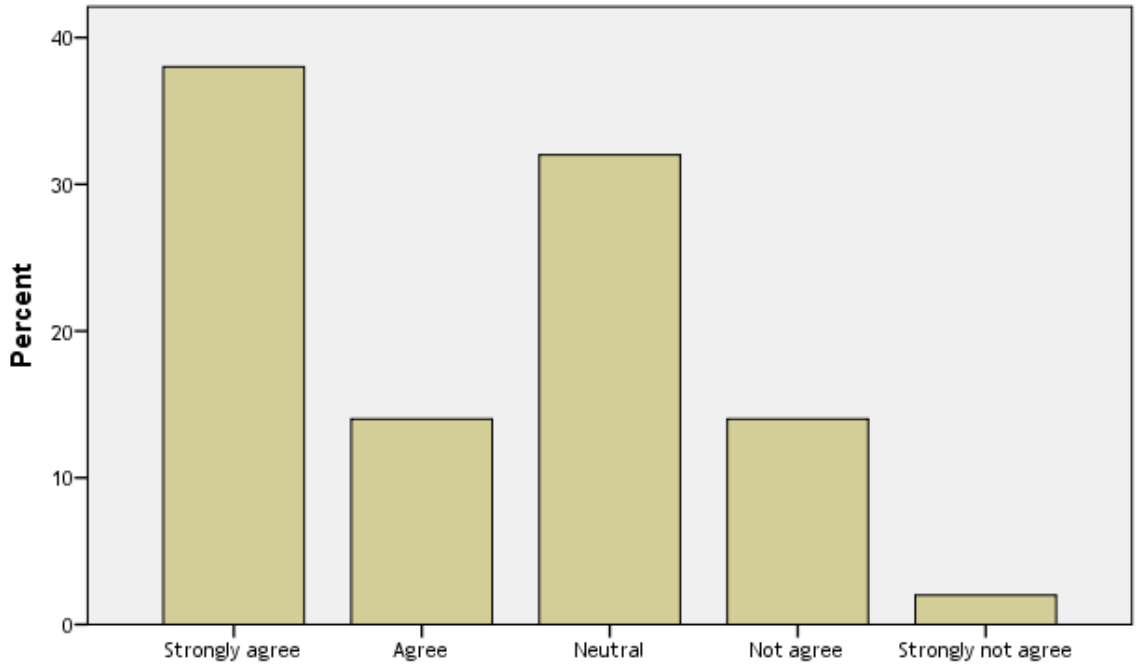
Table NO (5-6)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable# 1-6	Frequency	19	7	16	7	1	3.72	1.179	Agree
	Percent	38.0	14.0	32.0	14.0	2.0			

Source: Case Study 2014

The above table Display the result of variable is: Agree

Figure NO (5-6)



Merge two or more software quality standards or project management methodologies with each other to take advantage of their different characteristics in one model ensures the effectiveness of the resulting model.

Variable 1-7:

Integrating software quality standards characteristics with the software development process from initial stages (writing requirements report) ensures the quality of software product.

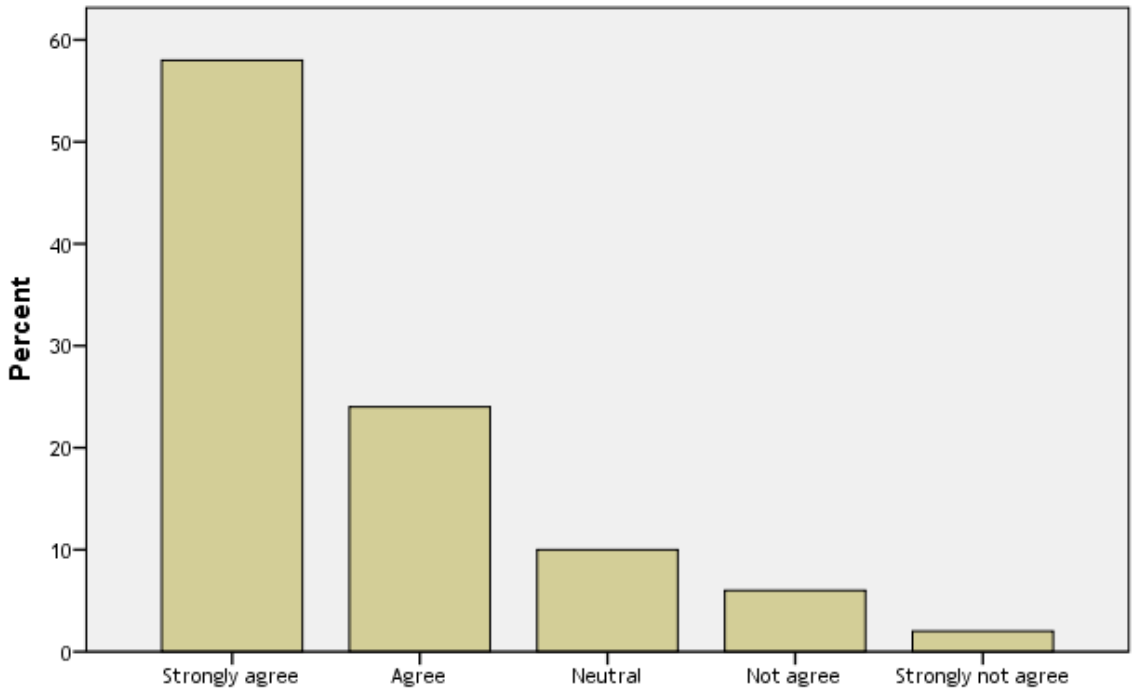
Table NO (5-7)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-7	Frequency	29	12	5	3	1	4.30	1.015	Strongly agree
	Percent	58.0	24.0	10.0	6.0	2.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree

Figure NO (5-7)



Integrating software quality standards characteristics with the software development process from initial stages (writing requirements report) ensures the quality of software product.

Variable 1-8:

Monitor apply software quality standards characteristics during development process ensures the quality of the final product.

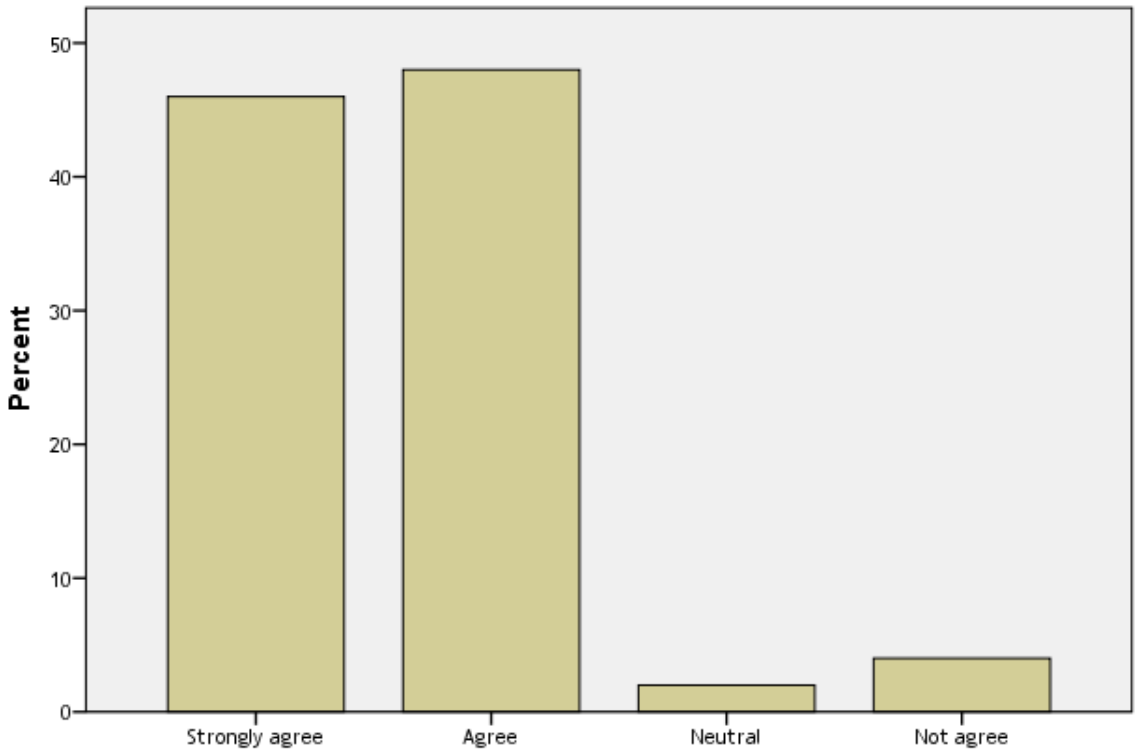
Table NO (5-8)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable# 1-8	Frequency	23	24	1	2	0	4.36	1.015	Strongly agree
	Percent	46.0	48.0	2.0	4.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-8)



Monitor apply software quality standards characteristics during development process ensures the quality of the final product?

Variable 1-9:

Develop technical recommendations based on the software quality standards characteristics to be implemented during development process, helps to ensure the quality of software product.

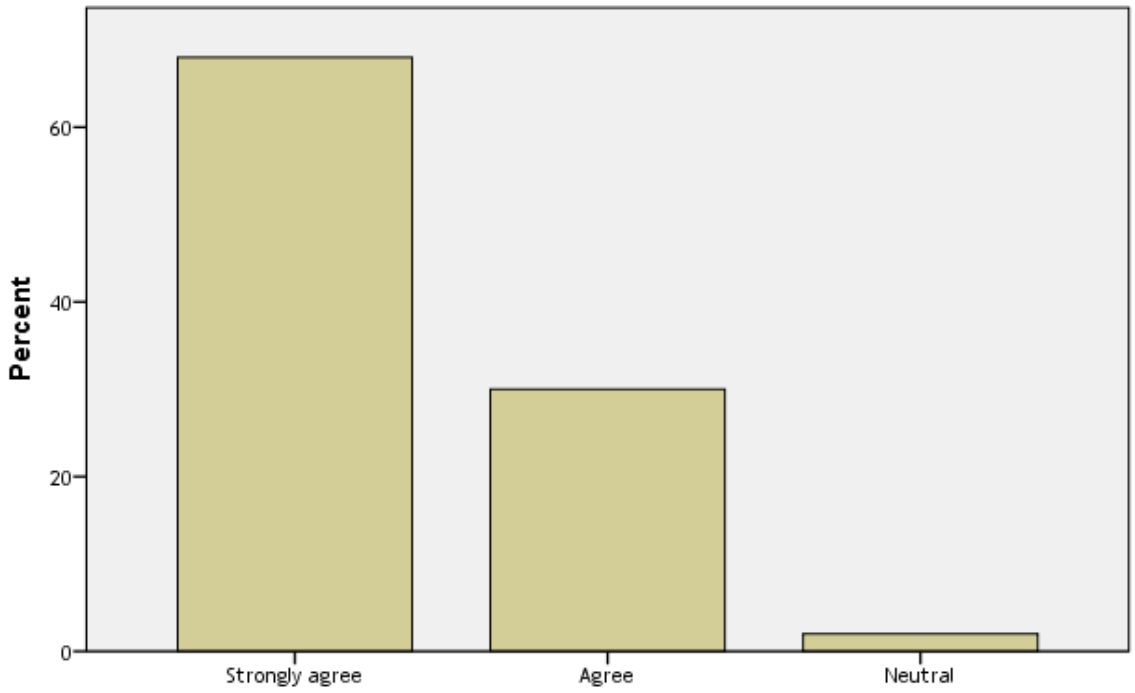
Table NO (5-9)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-9	Frequency	34	15	0	0	1	4.66	0.697	Strongly agree
	Percent	68.0	30.0	0.0	0.0	2.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree

Figure NO (5-9)



Develop technical recommendations based on the software quality standards characteristics to be implemented during development process, helps to ensure the quality of software product.

Variable 1-10:

It is not important develop a small software project by using iterative methodology. "Divide the software project to several parts then develops and delivers each one separately".

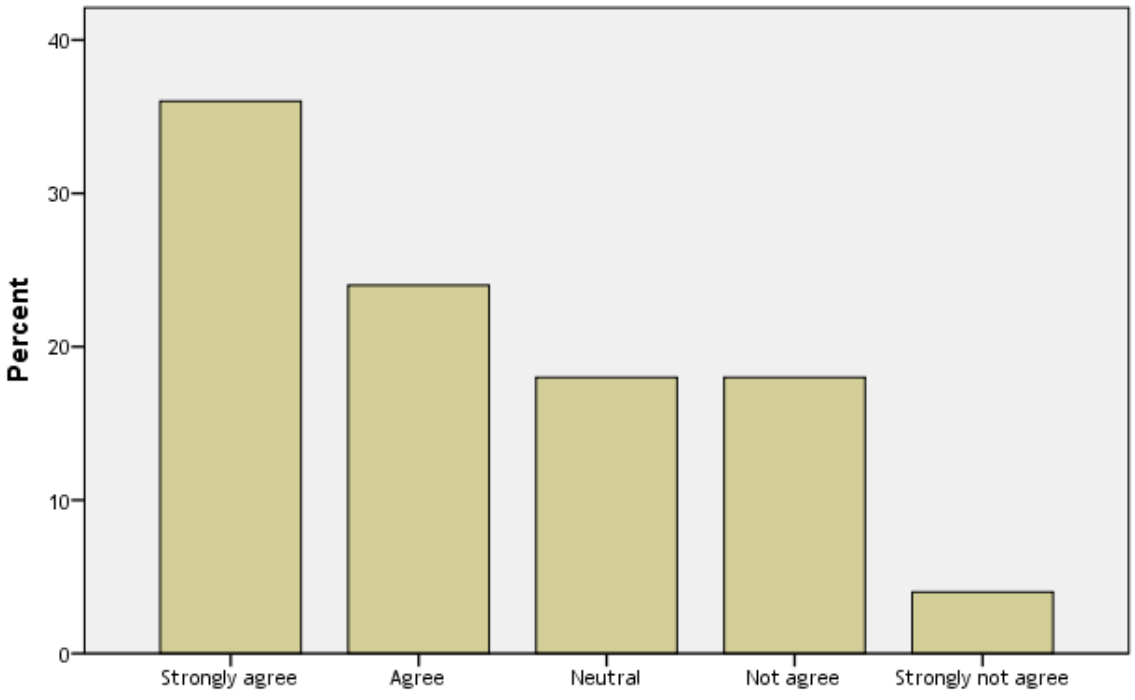
Table NO (5-10)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviati	Result
Variable#10	Frequency	18	12	9	9	0	3.70	1.249	Strongly agree
	Percent	36.0	24.0	18.0	18.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-10)



It is not important develop a small software project by using iterative methodology. "Divide the software project to several part then develop and deliver each one separately"

Variable 1-11:

Develop a methodology for improving software quality and project management based on the needs of the institutions makes it easy to apply.

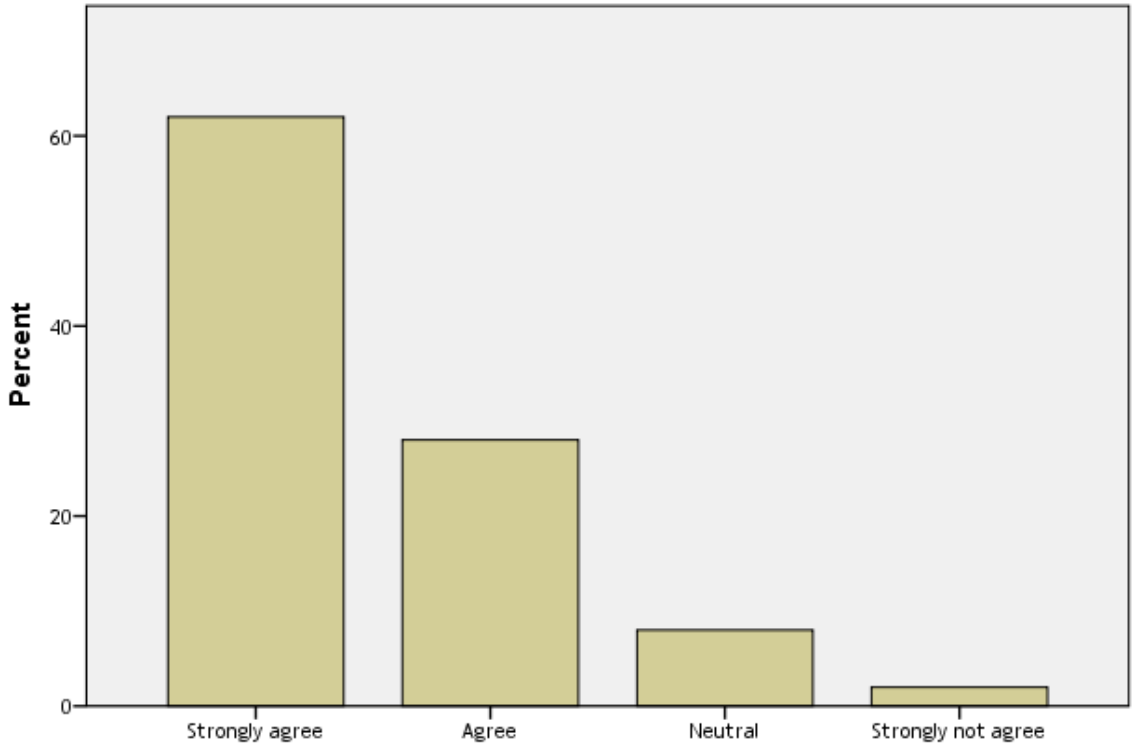
Table NO (5-11)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-11	Frequency	31	14	4	0	1	4.48	0.814	Strongly agree
	Percent	62.0	28.0	8.0	0.0	2.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-11)



Develop a methodology for improving software quality and project management based on the needs of the institutions makes it easy to apply.

Variable 1-12:

Identify and develop tools and procedures simple and inexpensive to manage process, measurement and analysis during the development process positively affect on project management.

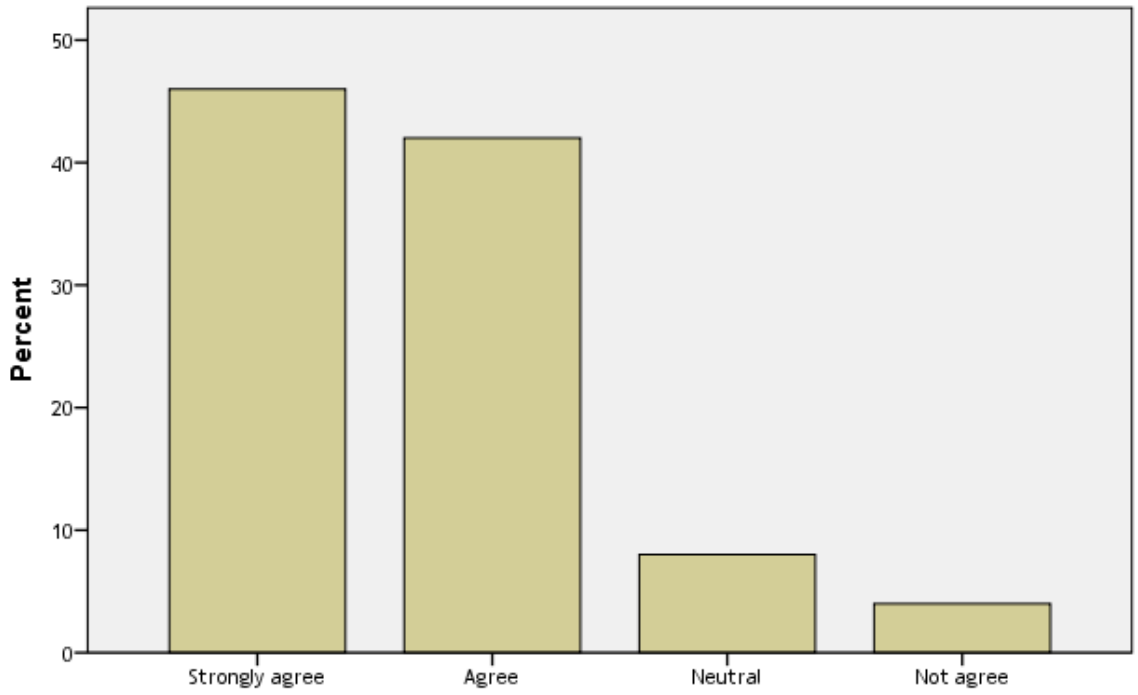
Table NO (5-12)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable# 1-12	Frequency	23	21	4	2	0	4.30	0.789	Strongly agree
	Percent	46.0	42.0	8.0	2.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-12)



Identify and develop tools and procedures simple and inexpensive to manage process, measurement and analysis during the development process positively affect on project management.

Variable 1-13:

Documents and reports that are offered through the various stages of the development process facilitate project management and ensure the quality of software product.

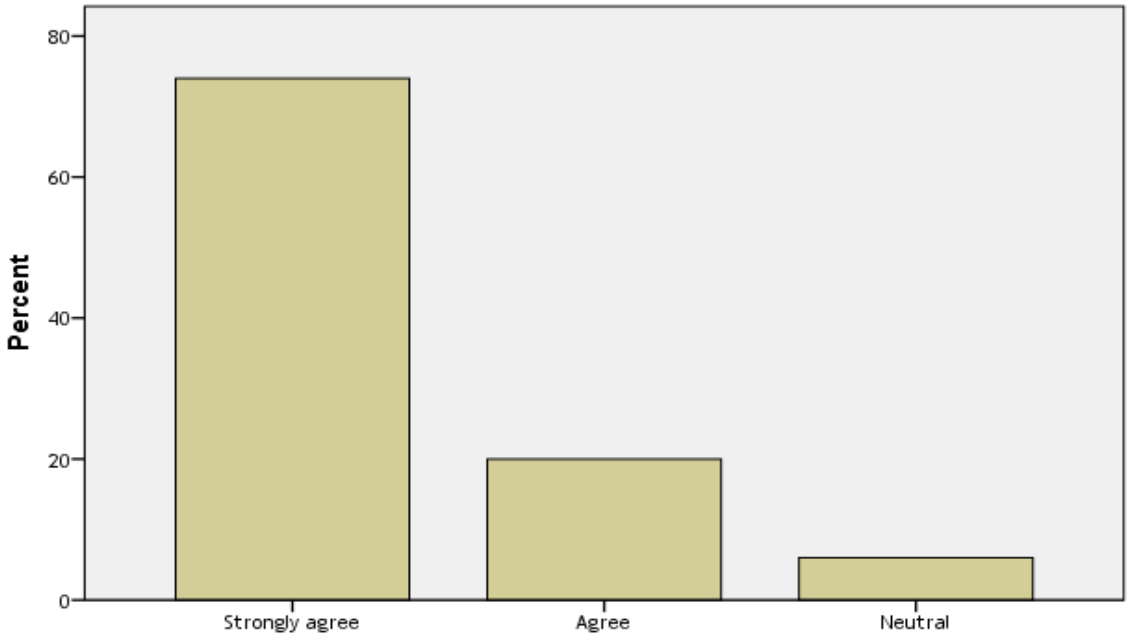
Table NO (5-13)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #13	Frequency	37	10	3	0	0	4.68	0.587	Strongly
	Percent	73.0	20.0	6.0	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-13)



Documents and reports that are offered through the various stages of the development process facilitate project management and ensure the quality of software product.

Variable 1-14:

Identify and develop simple tools, procedures and documentation to management (functional requirements and non-functional - time - cost - risk - test) facilitates project management process and ensure the quality of software product.

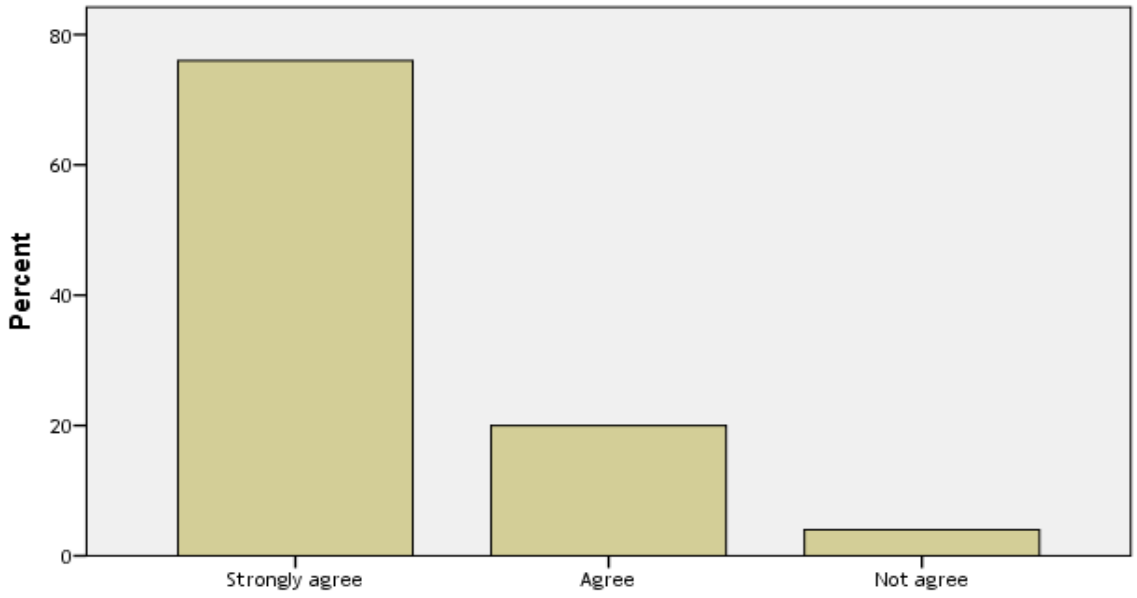
Table NO (5-14)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-14	Frequency	38	10	2	0	0	4.68	0.683	Strongly agree
	Percent	76.0	20.0	4.0	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-14)



Identify and develop simple tools, procedures and documentation to management (functional requirements and non-functional - time - cost - risk - test) facilitates project management process and ensure the quality of software product.

Variable 1-15:

It is important to sent reports and documents to customer or project owner to inform them about the level of development progressing.

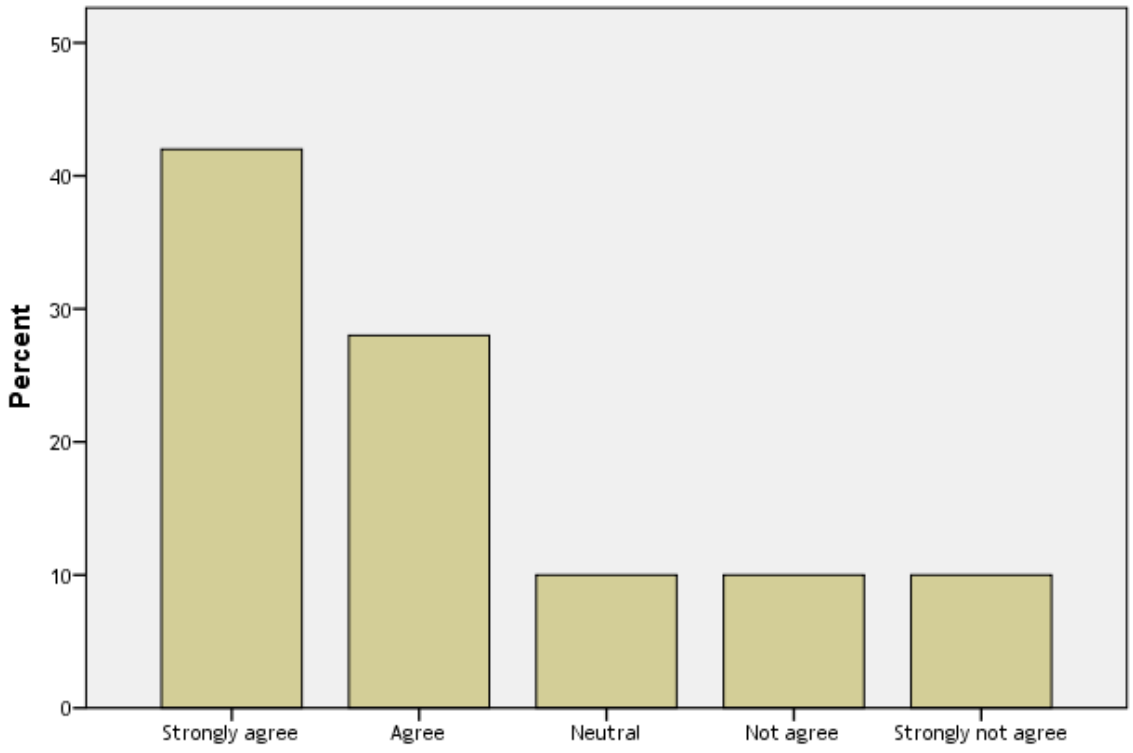
Table NO (5-15)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-15	Frequency	26	14	5	5	0	3.83	1.321	Strongly agree
	Percent	52.0	28.0	10.0	10.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-15)



It is important to sent reports and documents to customer or project owner to inform them about the level of development progressing.

Variable 1-16:

Apply the second level of the CMMI model "Project Management" enough for small businesses.

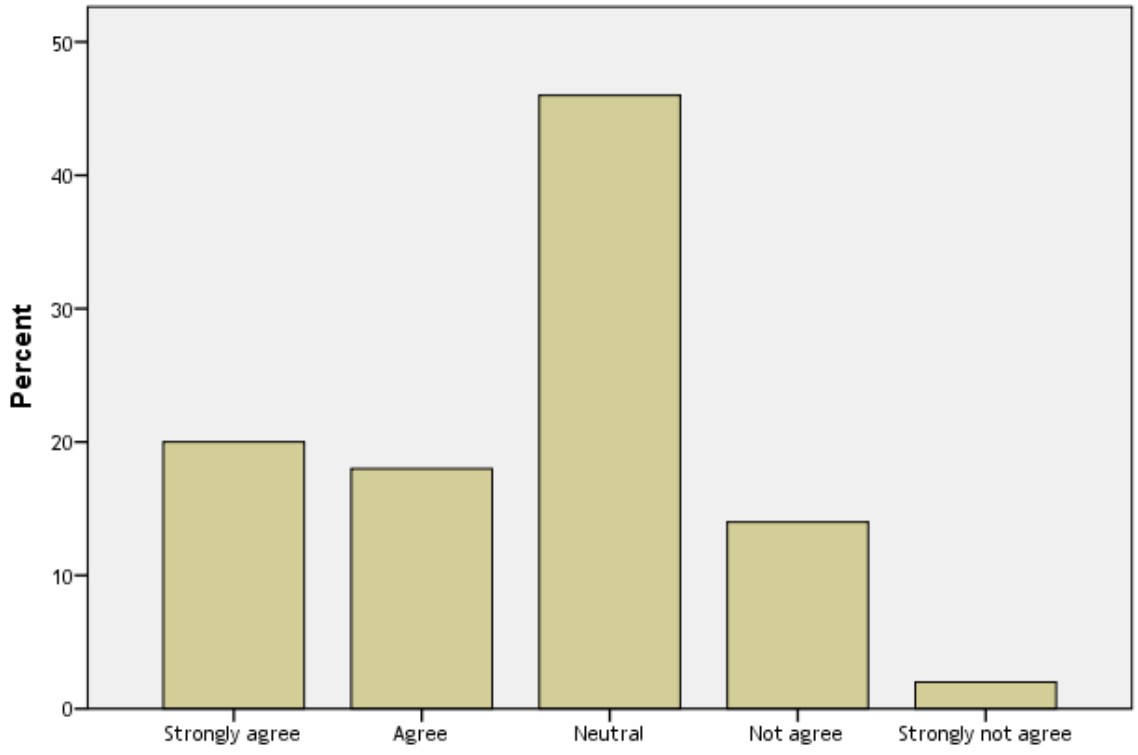
Table NO (5-16)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-16	Frequency	10	9	23	7	1	3.40	1.018	agree
	Percent	20.0	18.0	46.6	14.0	2.0			

Source: Case Study 2014

The above table Display the result of variable is: Agree.

Figure NO (5-16)



Apply the second level of the CMMI model "Project Management" enough for small businesses.

Variable 1-17:

Develop a methodology for improve the software quality and project management directed to specific development environment such as (Web - Mobile) be more effective.

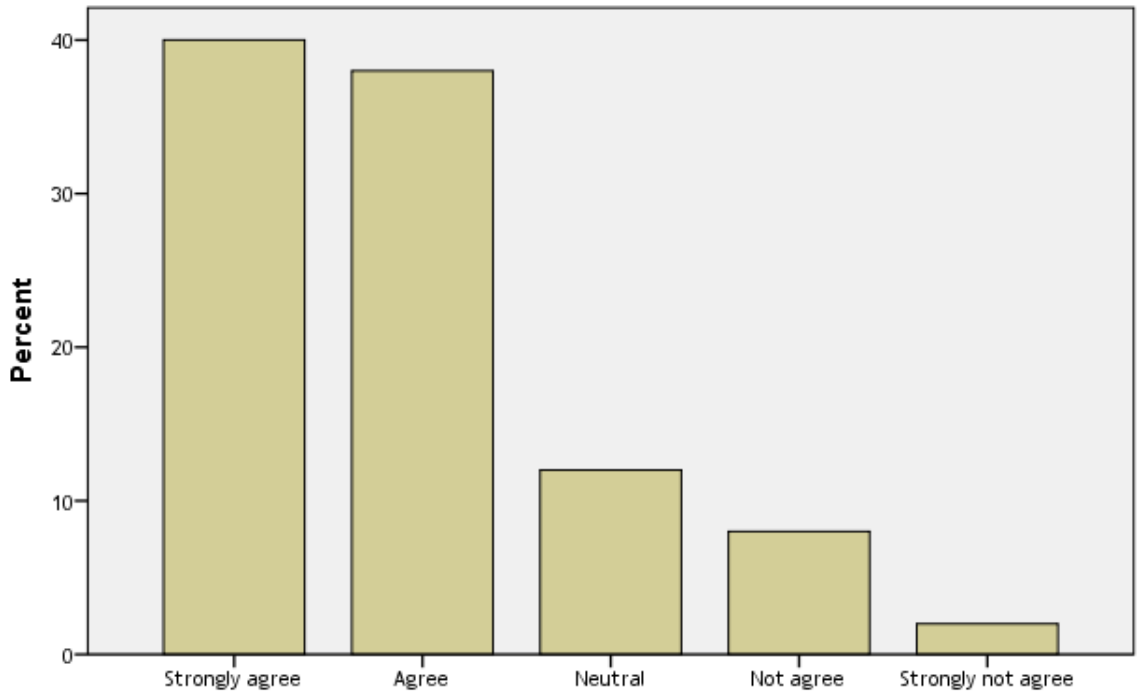
Table NO (5-17)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-17	Frequency	20	19	6	4	1	4.06	1.018	Strongly agree
	Percent	40.0	38.0	12.0	8.0	2.0			

Source: Case Study 2014

The above table Display the result of variable is: Strongly agree.

Figure NO (5-17)



Develop a methodology for improve the software quality and project management directed to specific development environment such as (Web - Mobile) be more effective.

5.2.2 Questioner number one – summary

Hypotheses number one:

Ensure success to apply software quality improvement and software quality improvement standards in the small and medium-sized enterprises must be simple, inexpensive and suit their needs.

Variables:

1. Apply software quality standards in small and medium enterprises ensure the quality of their products.
2. Financial costs and qualified staff it is the main factors that affect to apply software quality standards in small and medium enterprise.
3. Small and large companies have the same ability to apply the software quality standards.
4. Complexity of any software quality standards or project management methodology effect to applicable.
5. Each software quality standards and project management methodologies satisfy the needs of small and medium enterprises and applicable.

Table (5-18)

Hypotheses #1	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable 1	Frequency	30	17	3	0	0	4.54	.613	Strongly agree
	Percent	60.0	34.0	6.0	0	0			
Variable 2	Frequency	32	11	5	2	0	4.46	.838	Strongly agree
	Percent	64.0	22.0	10.0	4.0	0			
Variable 3	Frequency	16	9	1	20	4	3.26	1.468	Neutral
	Percent	32.0	18.0	2.0	40.0	8.0			
Variable 4	Frequency	29	12	4	4	1	4.28	1.051	Strongly agree
	Percent	58.0	24.0	8.0	8.0	2.0			

Variable 5	Frequency	22	9	6	11	2	3.76	1.333	agree
	Percent	44.0	18.0	12.0	22.0	4.0			
Hypothec result	Frequency	129	58	19	37	7	4.6	0.998	Strongly agree
	Percent	51.6	23.2	7.6	14.8	2.8			

Hypotheses number two:

Providing software quality standards that meet the needs of small enterprises must be merge more than one software quality standard and quality improvement to take advantage of their different characteristics and preferred to be a specialist in specific development environments.

Variables:

1. Merge two or more software quality standards or project management methodologies with each other to take advantage of their different characteristics in one model ensures the effectiveness of the resulting model.
2. Develop a methodology for improve the software quality and project management directed to specific development environment such as (Web - Mobile) be more effective.

Table (5-19)

Hypotheses #2	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable 1	Frequency	19	7	16	7	1	3.72	1.179	agree
	Percent	38.0	14.0	32.0	14.0	2.0			
Variable 2	Frequency	20	19	6	4	1	4.06	1.018	Strongly agree
	Percent	40.0	38.0	12.0	8.0	2.0			
Hypotheses result	Frequency	39	26	22	11	2	3.89	1.075	agree
	Percent	39	26	22	11	2			

Hypotheses number three:

To develop software product with high level of quality must be apply software quality characteristics during product development life cycle and monitor their implementation and put technical recommendations.

Variables:

1. Integrating the characteristics of the software quality standards with the software development process from initial stages (writing requirements report) ensures the quality of software product.
2. Monitor apply software quality standards characteristics during development process ensures the quality of the final product.
3. Develop technical recommendations based on the software quality standards characteristics to be implemented during development process, helps to ensure the quality of software product.
4. It is not important develop a small software project by using iterative methodology? "Divide the software project to several parts then develops and delivers each one separately".

Table (5-20)

Hypotheses #3	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable 1	Frequency	29	12	5	3	1	4.30	1.015	Strongly agree
	Percent	58.0	24.0	10.0	6.0	2.0			
Variable 2	Frequency	23	24	1	2	0	4.36	0.722	Strongly agree
	Percent	46.0	48.0	2.0	4.0	0			
Variable 3	Frequency	34	15	0	0	1	4.66	0.697	Strongly agree
	Percent	68.0	30.0	0	0	2.0			
Variable 4	Frequency	18	12	9	9	2	3.70	1.249	Strongly agree
	Percent	36.0	24.0	18.0	18.0	4.0			

Hypothec result	Frequency	104	63	15	14	4	4.26	.826	Strongly agree
	Percent	52.0	31.5	7.5	7.0	2.2			

Hypotheses number four:

Project management methodologies in small enterprises must be identifies tools for measuring, analysis, provide documents and reports during whole development process stages and care about customer opinion.

Variables:

1. Develop a methodology for improving software quality and project management based on the needs of the institutions makes it easy to apply.
2. Identify and develop tools and procedures simple and inexpensive to manage process, measurement and analysis during the development process positively affect on project management.
3. Documents and reports that are offered through the various stages of the development process facilitate project management and ensure the quality of software product.
4. Identify and develop simple tools, procedures and documentation to management (functional requirements and non-functional - time - cost - risk - test) facilitates project management process and ensure the quality of software product.
5. It is important to sent reports and documents to the customer or project owner to inform them about the level of development progressing.

Table (5-21)

Hypotheses #4	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable 1	Frequency	31	14	4	0	1	4.48	0.814	Strongly agree
	Percent	62.0	28.0	8.0	0.0	2.0			

Variable 2	Frequency	23	21	4	2	0	4.30	0.789	Strongly agree
	Percent	46.0	42.0	8.0	4.0	0.0			
Variable 3	Frequency	37	10	3	0	0	4.68	0.587	Strongly agree
	Percent	74.0	20.0	6.0	0.0	0.0			
Variable 4	Frequency	38	10	2	0	0	4.68	0.683	Strongly agree
	Percent	76.0	20.0	4.0	0.0	0.0			
Variable 5	Frequency	26	14	5	5	0	3.82	1.321	Strongly agree
	Percent	52.0	28.0	10.0	10.0	0.0			
Hypotheses result	Frequency	155	69	18	7	1	4.66	.519	Strongly agree
	Percent	62.0	27.6	7.2	2.8	0.4			

5.3 Questioner number two

5.3.1 Degree of internal consistency and reliability

Scale: all variables

Case processing summary

	Number	Percent
Valid	12	100.0
Excluded	0	0
Total	12	100.0

Reliability statistics

Alpha	N of items
.989	31

The above table display truth and the internal consistency using SPSS program for test the questions and items study, alpha value reached to (0.989) this means the degree of validity and reliability of this study is very high and this enables us to analyze data and get correct and truthful results.

Analysis based on Likert Scale

A method of ascribing quantitative value to qualitative data, to make it amenable to statistical analysis. A numerical value is assigned to each potential choice and a mean figure for all the responses is computed at the end of the evaluation or survey. Used mainly in training course evaluations and market surveys, Likert scales usually have five potential choices (strongly agree, agree, neutral, disagree, strongly disagree) but sometimes go up to ten or more as below :

Value	Weighted average
Strongly agree	From 5.00 to 4.20
agree	From 4.19 to 3.40
Neutral	From 3.39 to 2.60
Disagree	From 2.59 to 1.80
Strongly disagree	From 1.79 to 1.00

Variable 2-1:

Divide development life cycle to three stages (1- information gather, 2- Development "design, writing code, testing", 3- review and delivery) enough for small projects.

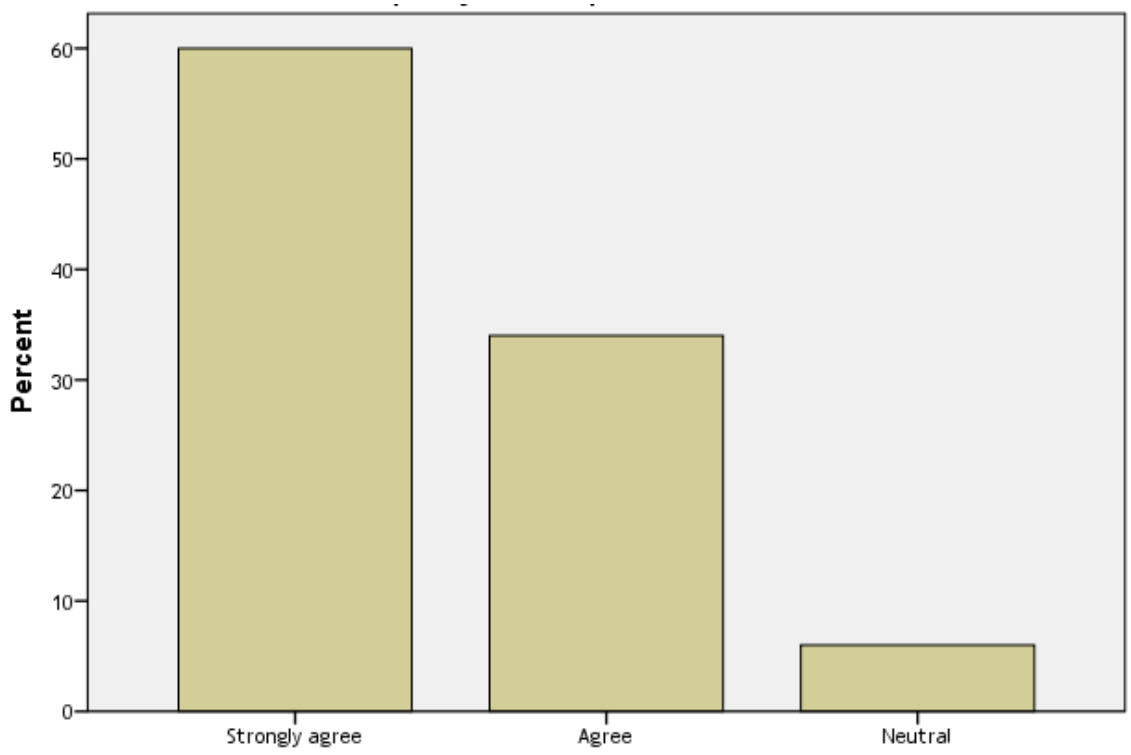
Table NO (5-22)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-1	Frequency	30	17	3	0	0	4.54	.6.13	Strongly agree
	Percent	60.0	34.0	6.0	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-18)



Divide development life cycle to three stages (1- information gather, 2- Development "design, writing code, testing", 3- review and delivery) enough for small projects.

Variable 2-2:

Integration ISO / IEC 25010 software quality standard with the requirements report ensures software quality product.

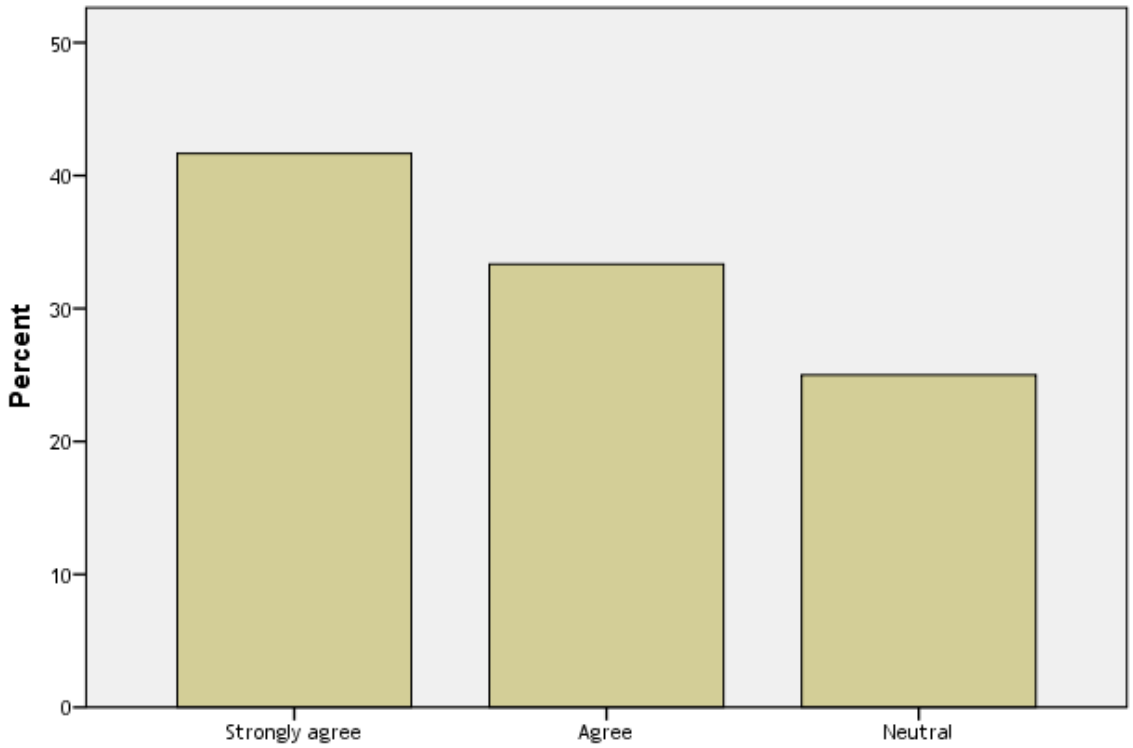
Table NO (5-23)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-2	Frequency	5	4	3	0	0	4.17	0.835	Strongly agree
	Percent	41.7	33.3	25	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-20)



Integration ISO / IEC 25010 software quality standard with the requirements report ensures software quality product.

Variable 2-3:

Merging level two for CMMI with ISO / IEC 25010 to make sure about software quality on small enterprises.

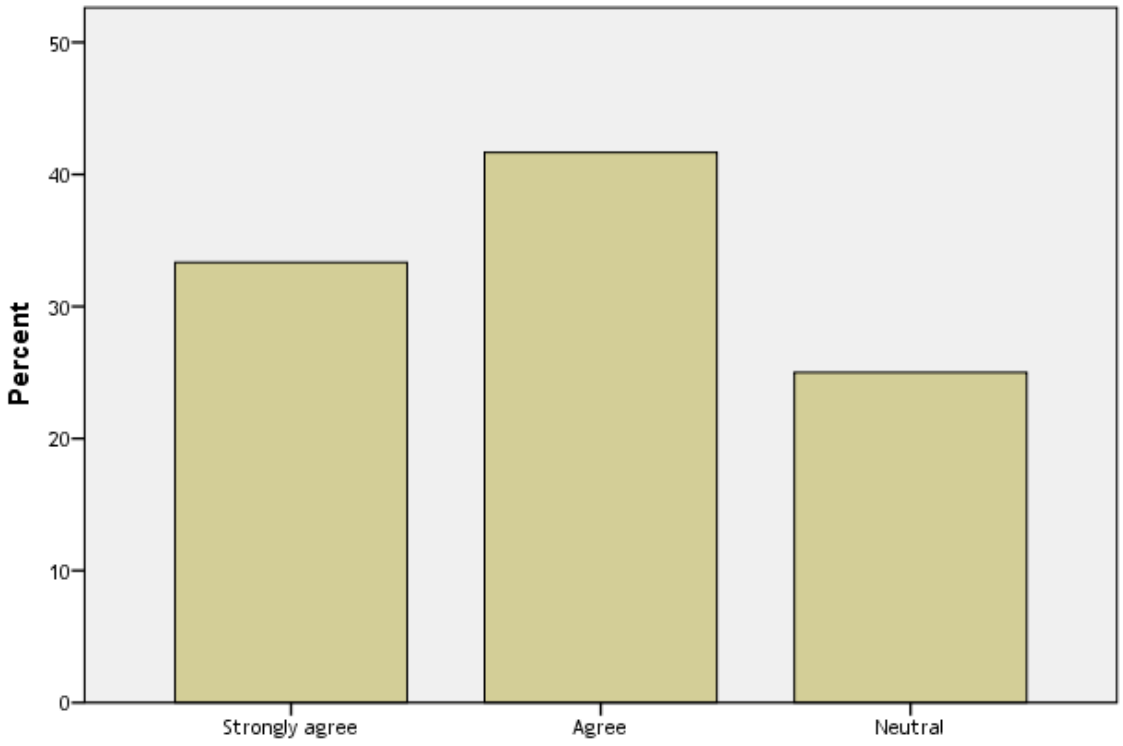
Table NO (5-24)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-3	Frequency	4	5	3	0	0	4.08	0.793	Strongly agree
	Percent	33.3	41.7	25	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-21)



Merging level two for CMMI with ISO / IEC 25010 to make sure about software quality on small enterprises.

Variable 2-4:

Submit a requirements report in triplicate (draft - proposed - review) to make sure coverage the main customer needs.

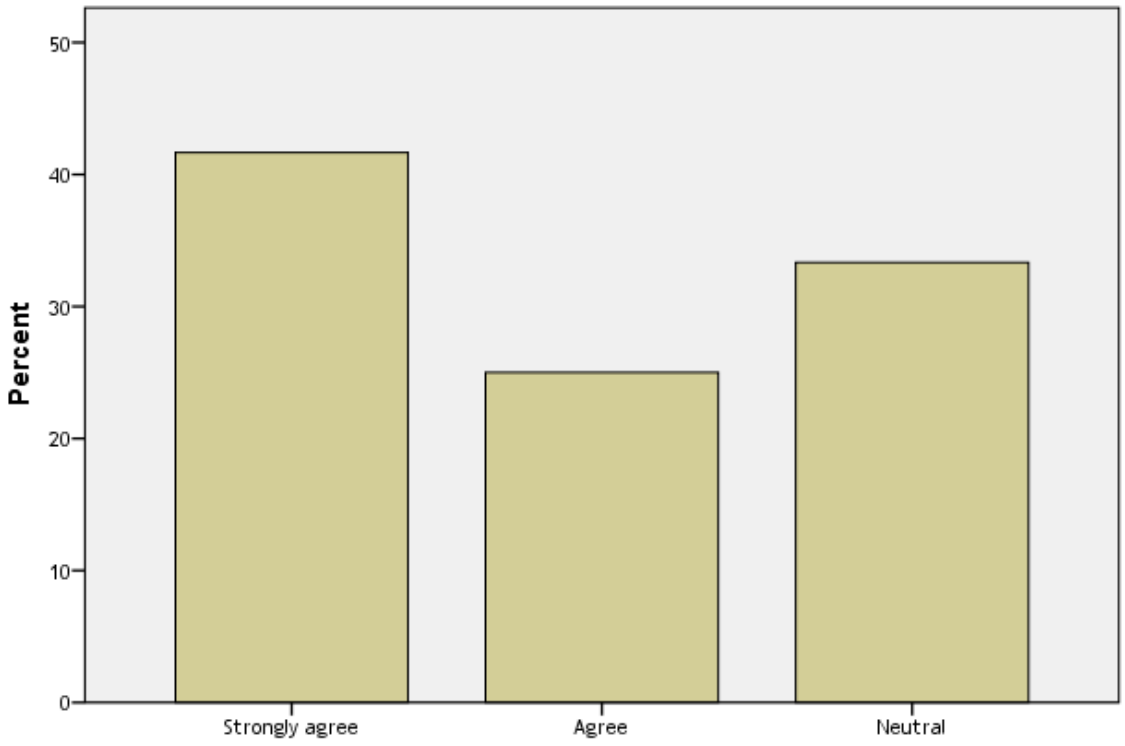
variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-4	Frequency	5	3	4	0	0	4.08	0.900	Strongly agree
	Percent	41.7	25	33.3	0.0	0.0			

Table NO (5-25)

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-22)



Submit a requirements report in triplicate (draft - proposed - review) to make sure coverage the main customer needs.

Variable 2-5:

The initial meeting between customer and project manager ensures to cover the main functional and non-functional requirements and then negotiated.

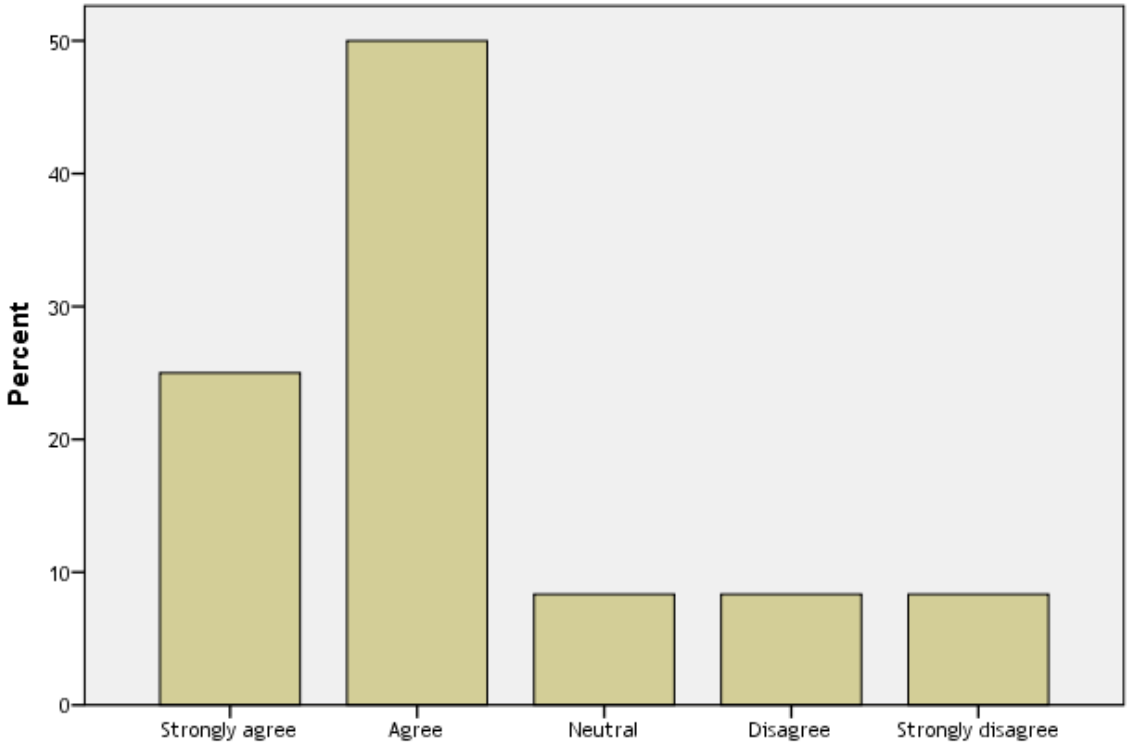
Table NO (5-26)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-5	Frequency	3	6	1	1	1	3.75	1.215	Strongly agree
	Percent	25	60	8.3	8.3	8.3			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-23)



The initial meeting between customer and project manager ensures to cover the main functional and non-functional requirements and then negotiated.

Variable 2-6:

Review the initial copy of requirements (draft) report by project manager makes it Uncluttered and orderly for readers.

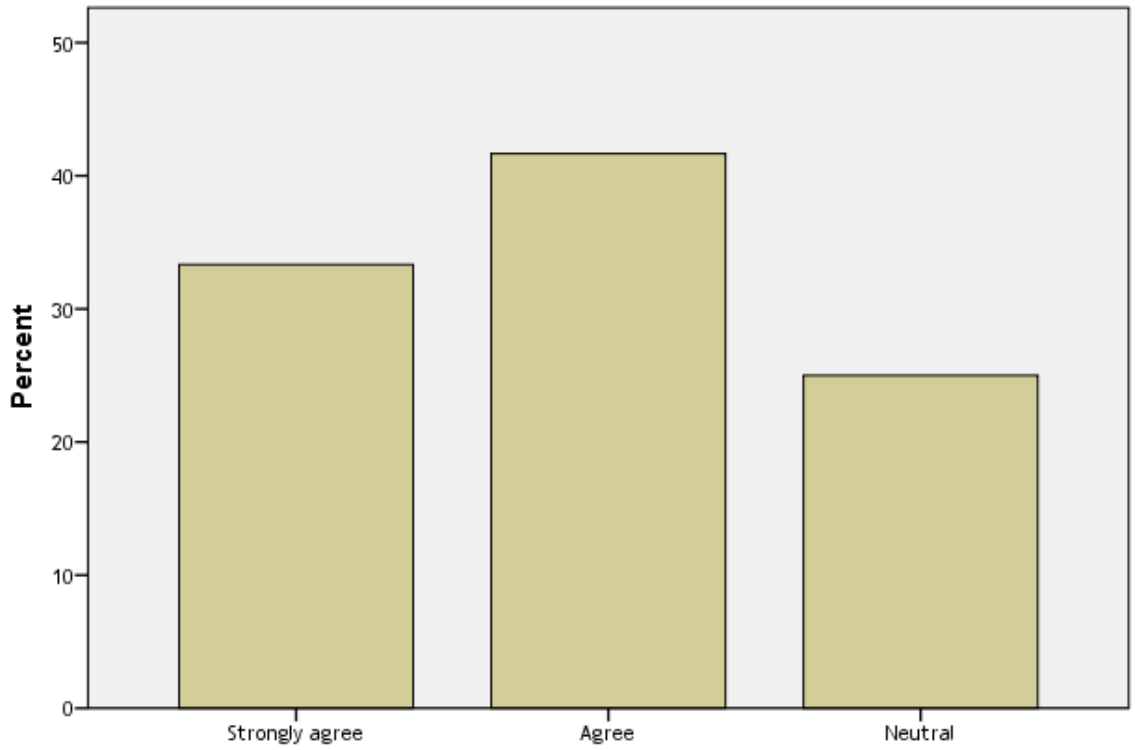
Table NO (5-27)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-6	Frequency	4	5	3	0	0	4.08	0.793	Strongly agree
	Percent	33.3	41.7	25	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-24)



Review the initial copy of requirements (draft) report by project manager makes it Uncluttered and orderly for readers.

Variable 2-7:

It is very importance to send a requirements report to customer for final approval.

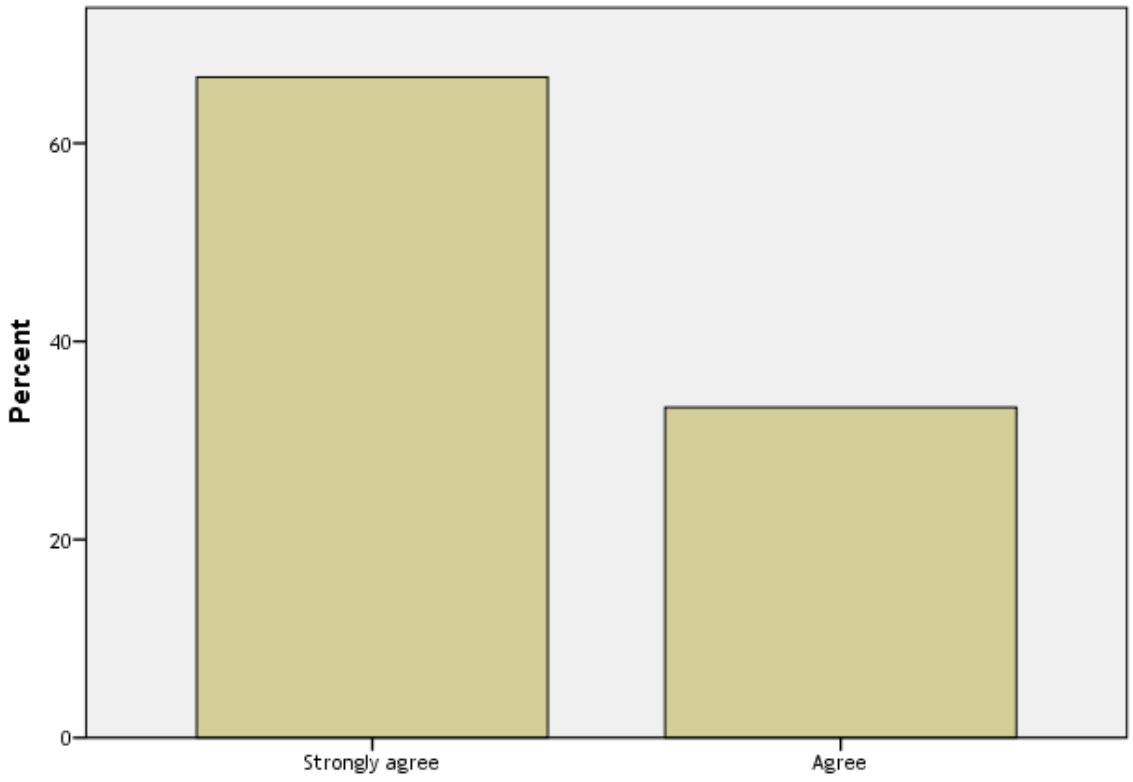
Table NO (5-28)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-7	Frequency	8	4	0	0	0	4.67	0.492	Strongly agree
	Percent	66.7	33.3	0	0	0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-25)



It is very importance to send a requirements report to customer for final approval.

Variable 2-8:

When customer approval the final copy of a requirements report reflects the level of understanding and agreement between the customer and the project manager about project requirements.

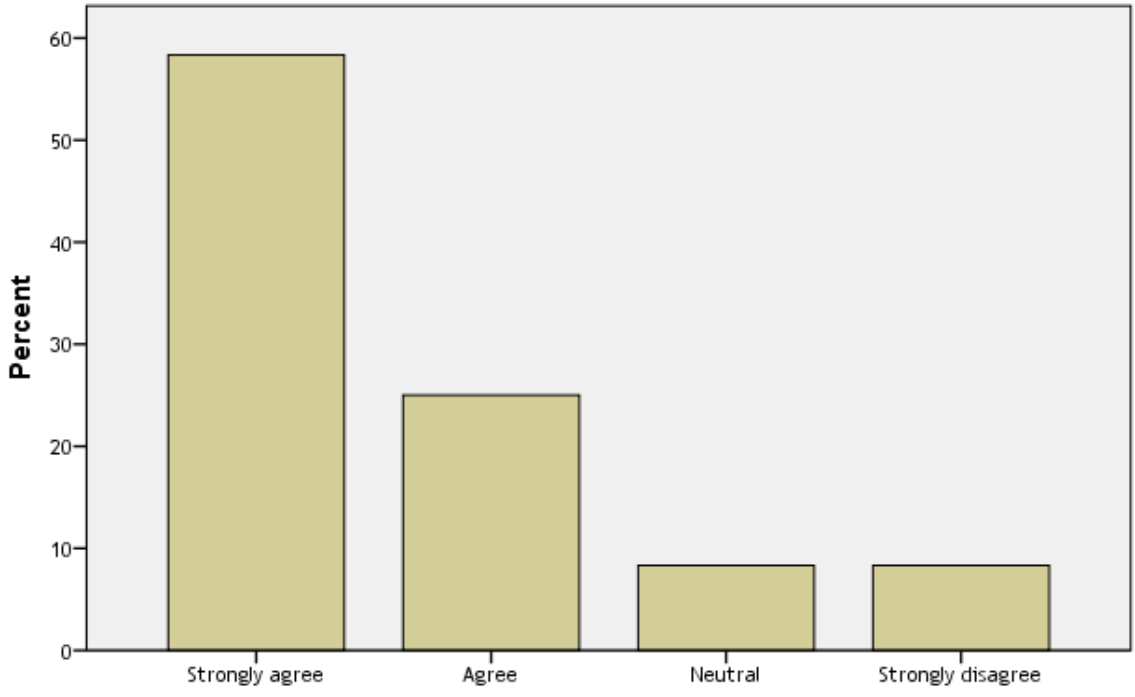
Table NO (5-29)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-8	Frequency	7	3	1	0	1	4.25	1.125	Strongly agree
	Percent	58.3	25	8.3	0	8.3			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-26)



When customer approval the final copy of a requirements report reflects the level of understanding and agreement between the customer and the project manager about project requirements.

Variable 2-9:

It is very importance to develop separate report for technical requirements. "Project limits, cost estimation, time estimation, software quality and other requirements".

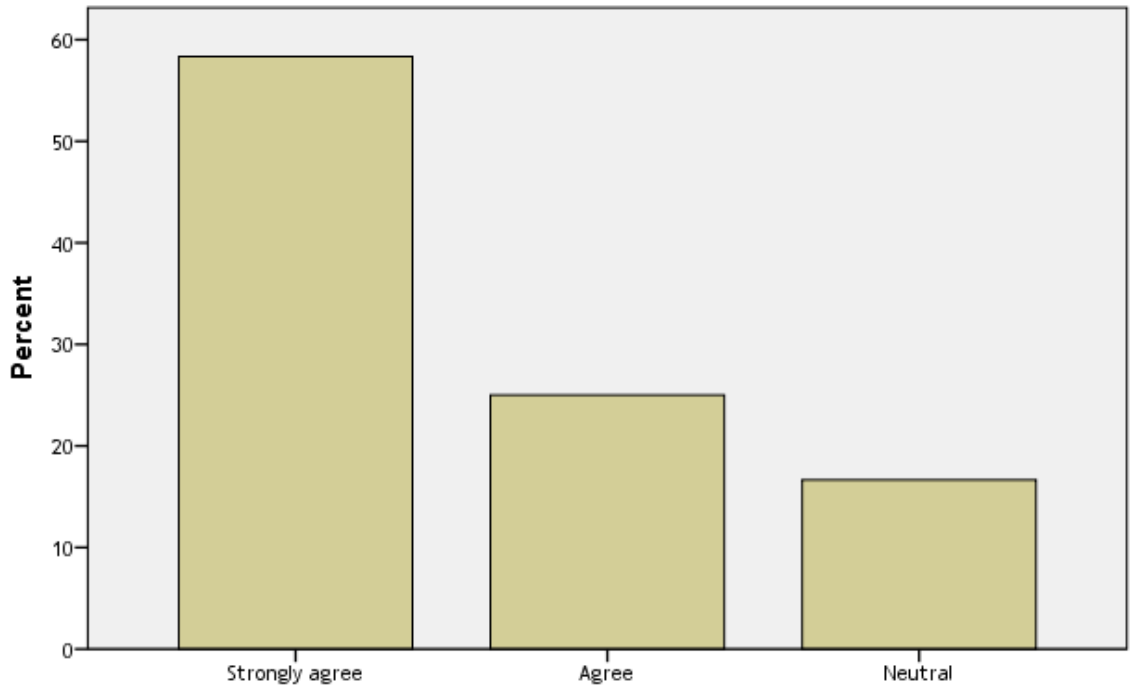
Table NO (5-30)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-9	Frequency	7	3	2	0	0	4.42	0.793	Strongly agree
	Percent	58.3	25	16.7	0	0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-27)



It is very importance to develop separate report for technical requirements. "Project limits, cost estimation, time estimation, software quality and other requirements".

Variable 2-10:

Provide the technical report requirements in three copies (draft - proposed – approval) ensures to coverage of all the technical requirements and the effectiveness of the resulting report.

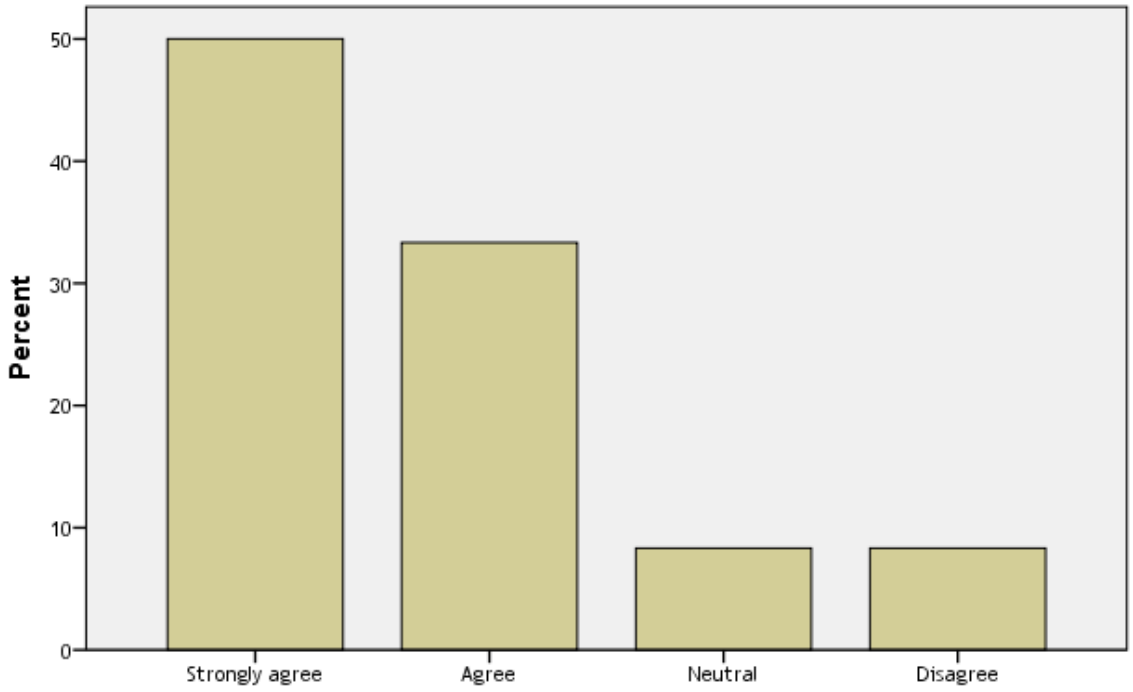
Table NO (5-31)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-10	Frequency	6	4	1	1	0	4.25	0.965	Strongly agree
	Percent	50	33.3	8.3	8.3	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-28)



Provide the technical report requirements in three copies (draft - proposed – approval) ensures to coverage of all the technical requirements and the effectiveness of the resulting report.

Variable 2-11:

Documents and reports that are offered through various stages of development process facilitate project management and quality assurance.

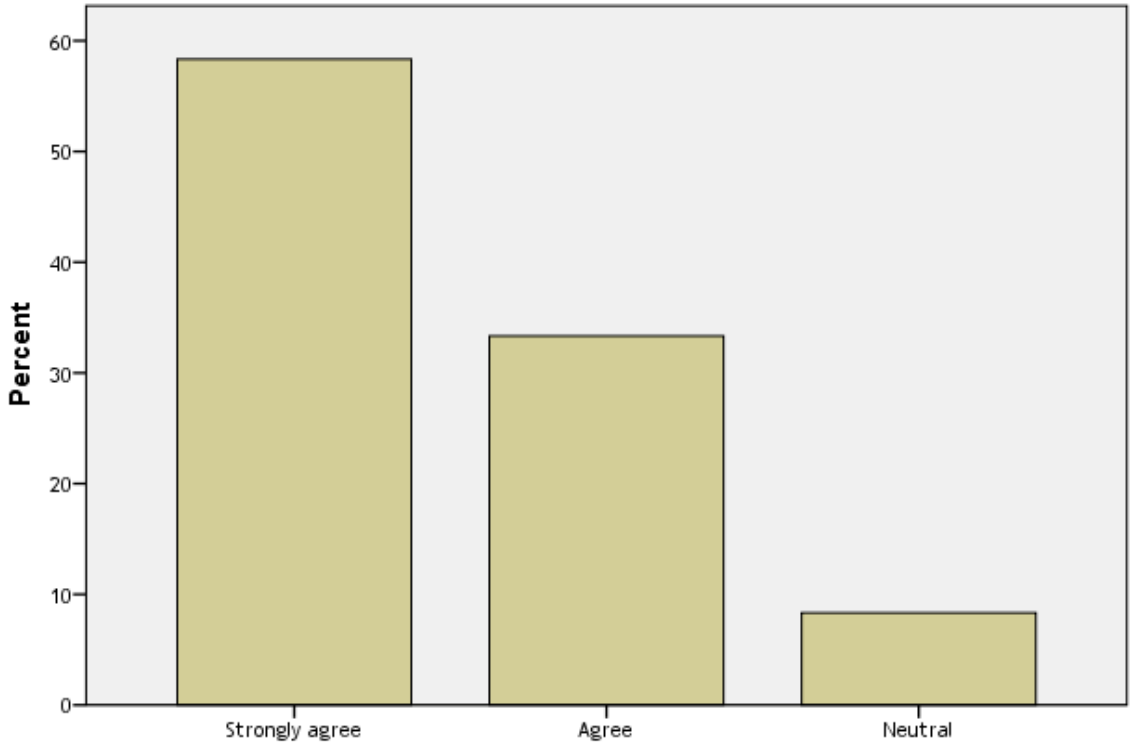
Table NO (5-32)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-11	Frequency	7	4	1	0	0	4.50	0.674	Strongly agree
	Percent	58.3	33.3	8.3	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-29)



Documents and reports that are offered through various stages of development process facilitate project management and quality assurance.

Variable 2-12:

Pay a percentage of the total cost after writing the basic requirements reports grantees the customer right and suppliers.

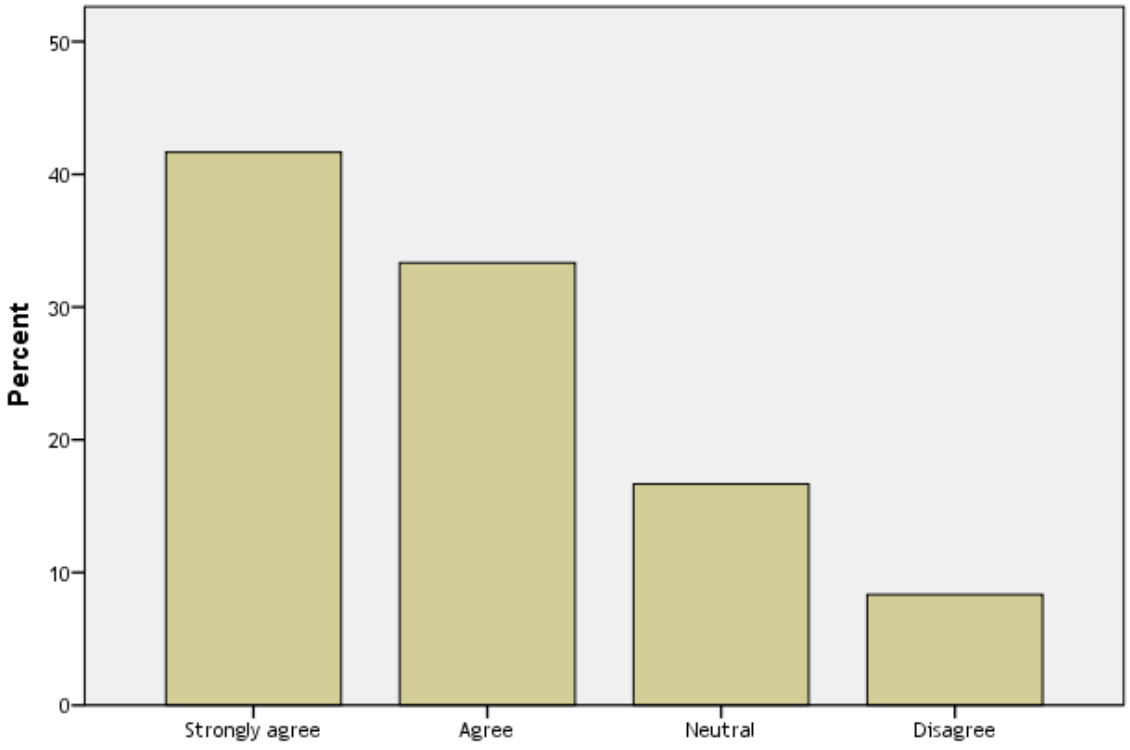
Table NO (5-33)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-12	Frequency	5	4	2	1	0	4.08	0.996	Strongly agree
	Percent	41.7	33.3	16.7	8.3	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-30)



Pay a percentage of the total cost after writing the basic requirements reports grantees the customer right and suppliers.

Variable number 2-13:

Deliver a model or sketch of the proposed design to customer gives him a clear idea about how the future system will be like.

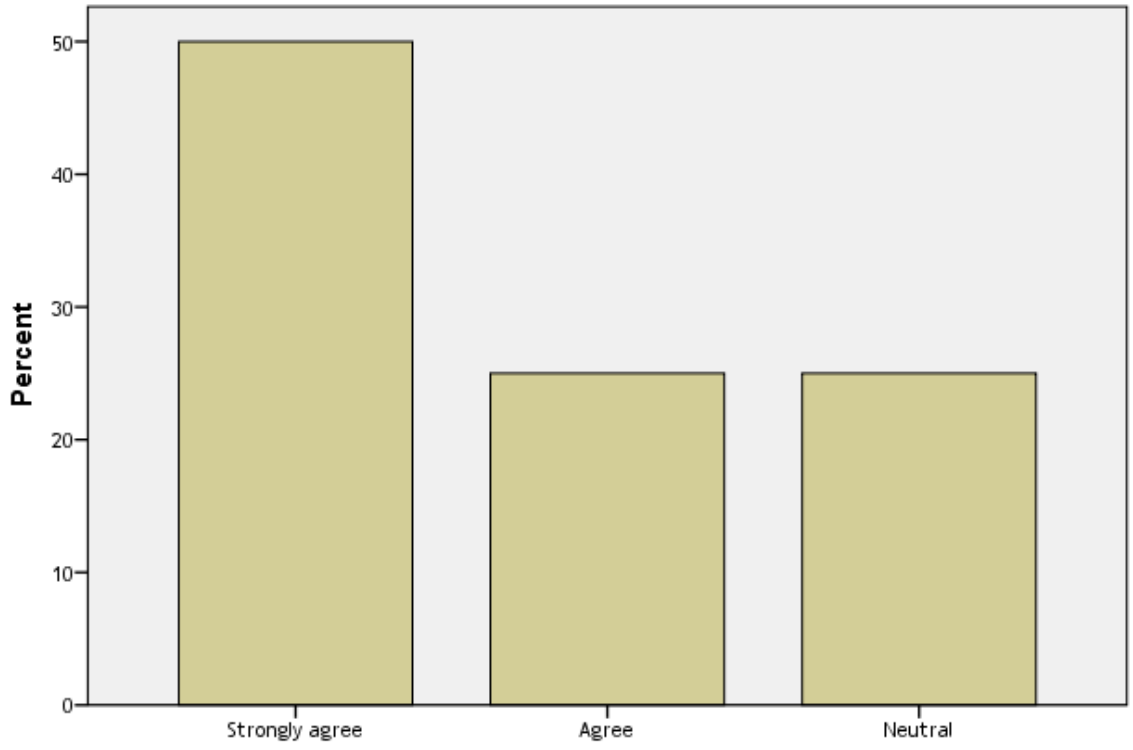
Table NO (5-34)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-13	Frequency	6	3	3	0	0	4.25	0.866	Strongly agree
	Percent	50.0	25	25	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-31)



Deliver a model or sketch of the proposed design to customer gives him a clear idea about how the future system will be like.

Variable 2-14:

It is very importance accept the final design by customer and must be care about his opinion.

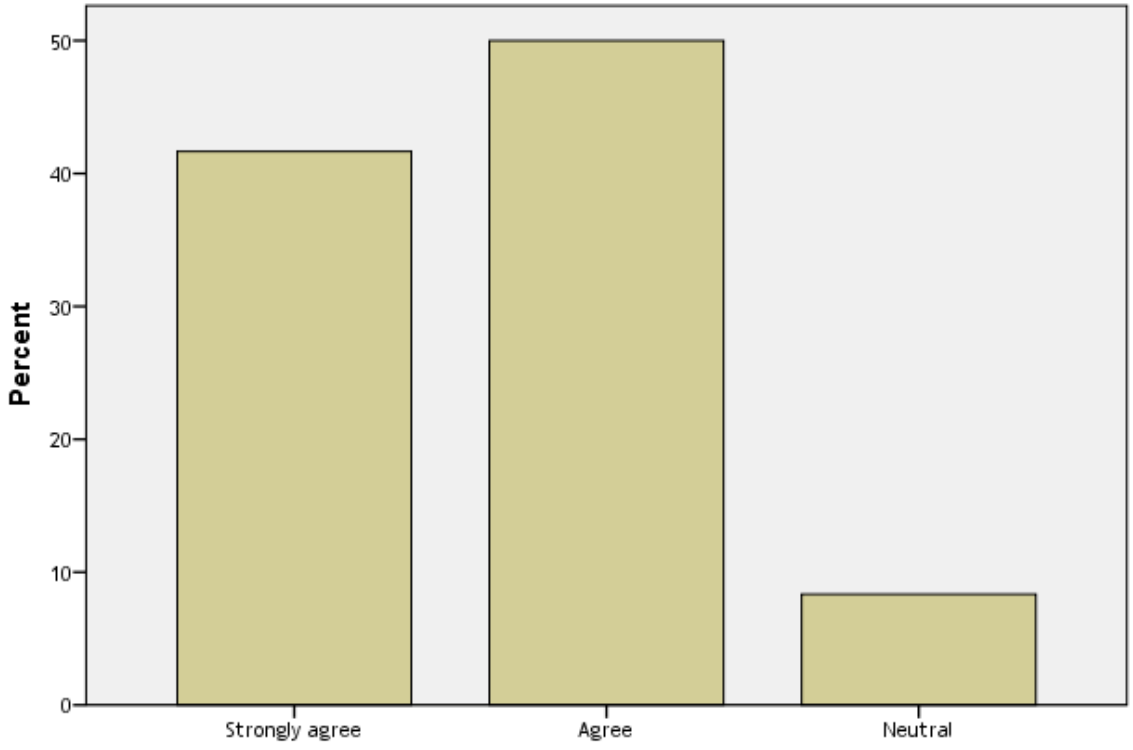
Table NO (5-35)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-14	Frequency	5	6	1	0	0	4.33	0.651	Strongly agree
	Percent	41.7	50	8.3	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-32)



It is very importance accept the final design by customer and must be care about his opinion.

Variable 2-15:

Implementation the final design must be based on the final approval by customer.

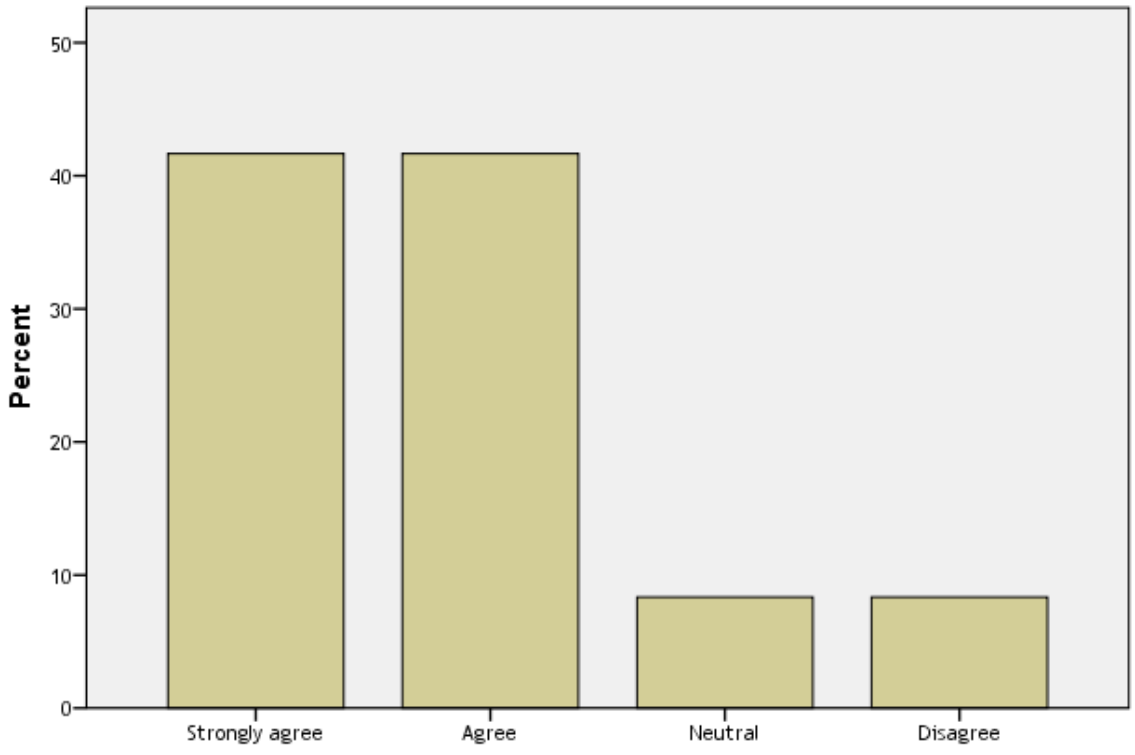
Table NO (5-36)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-15	Frequency	5	5	1	1	0	4.17	0.937	Strongly agree
	Percent	41.7	41.7	8.3	8.3	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-33)



Implementation the final design must be based on the final approval by customer.

Variable 2-16:

Attention to (standards to write code - defensive programming) in small enterprises guarantees write good code line and achieve software quality standards.

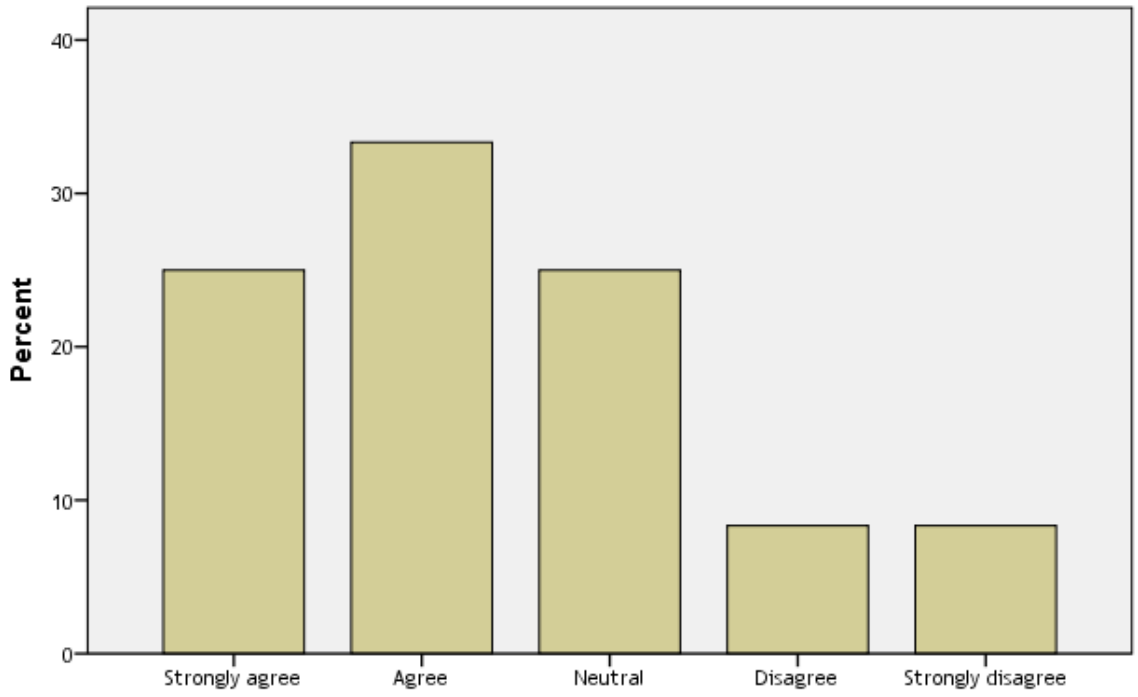
Table NO (5-37)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-16	Frequency	3	4	3	1	1	3.58	1.240	Agree
	Percent	25	33.3	25	8.3	8.3			

Source: Case Study 2014

The above table Display the result of variable is Agree.

Figure NO (5-34)



Attention to (standards to write code - defensive programming) in small enterprises guarantees write good code line and achieve software quality standards.

Variable 2-17:

Identify tools and weaknesses must be tested as acceptable level of protection for software product.

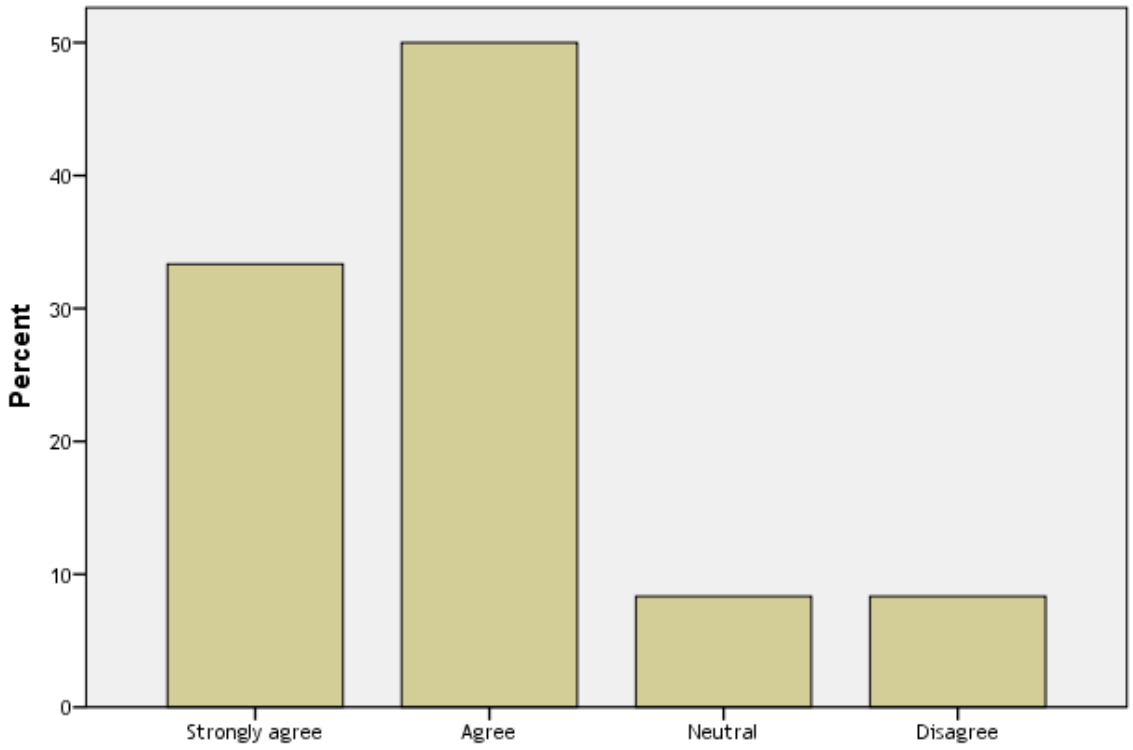
Table NO (5-38)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-17	Frequency	4	6	1	1	0	4.08	0.900	Strongly agree
	Percent	33.3	50	8.3	8.3	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-35)



Identify tools and weaknesses must be tested as acceptable level of protection for software product.

Variable number 2-18:

When the first version of product is delivered, customer must be establishment checklist about his observations to ensure value (customer satisfaction).

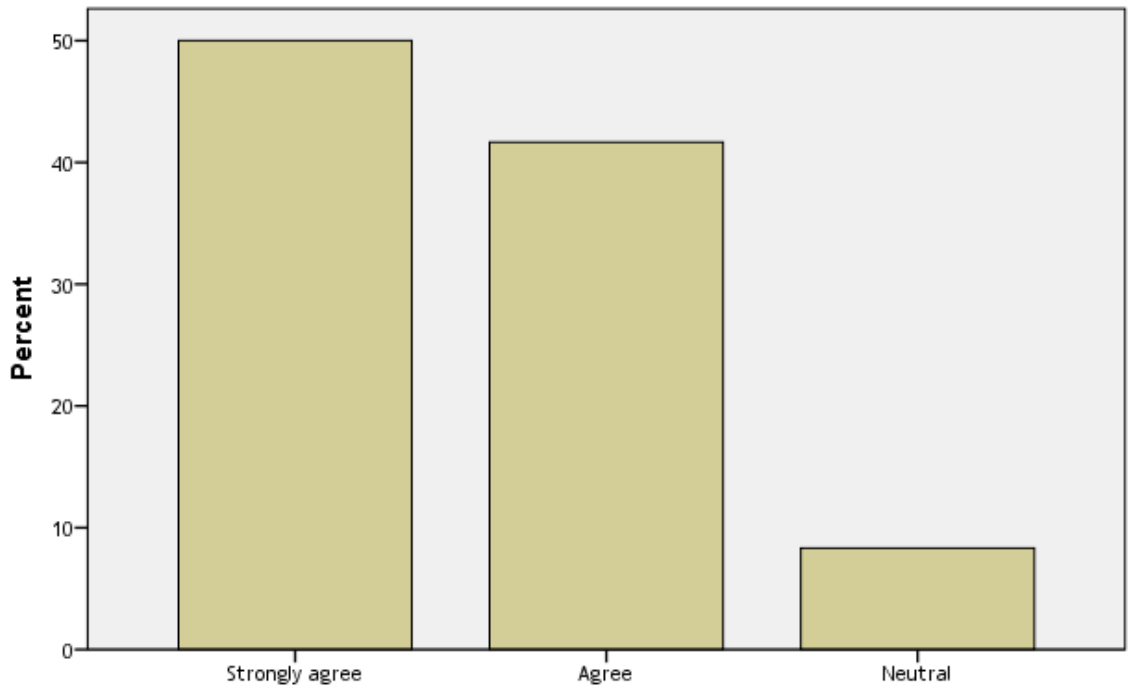
Table NO (5-39)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-18	Frequency	6	5	1	0	0	4.42	0.669	Strongly agree
	Percent	50	41.7	8.3	0	0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-36)



When the first version of product is delivered, customer must be establishment checklist about his observations to ensure value (customer satisfaction).

Variable 2-19:

Meeting between customer and project manager to discuss checklist leads to understanding customer observations clearly.

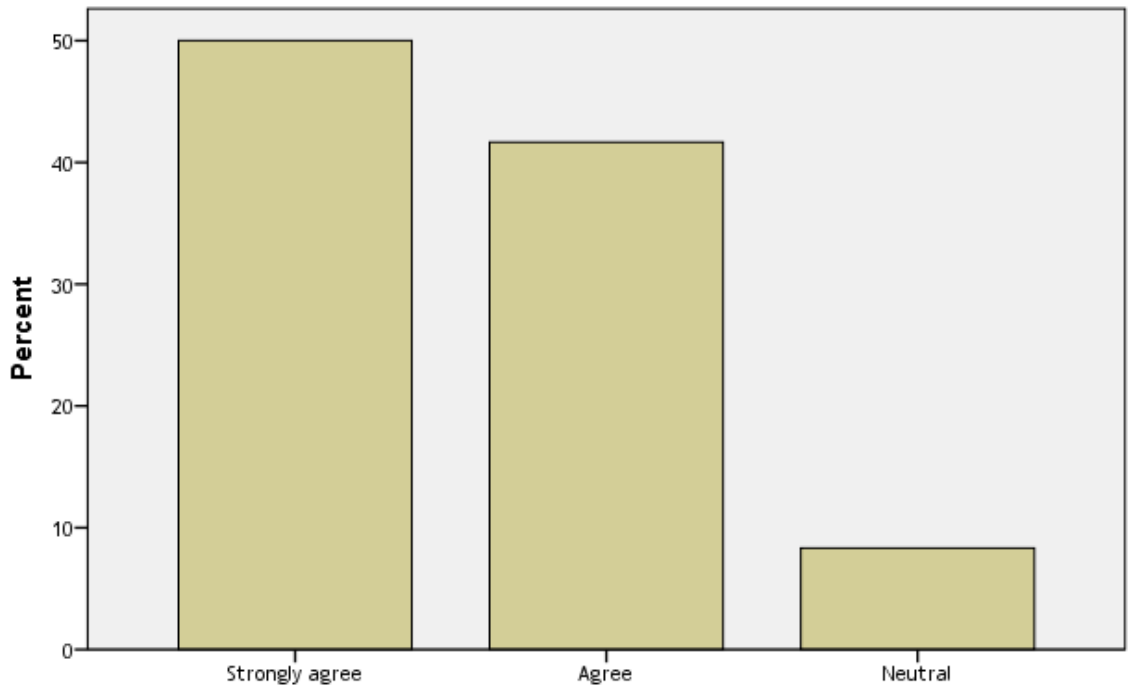
Table NO (5-40)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-19	Frequency	6	5	1	0	0	4.42	0.669	Strongly agree
	Percent	50	41.7	8.3	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-37)



Meeting between customer and project manager to discuss checklist leads to understanding customer observations clearly.

Variable 2-20:

After development stage are completed, customer must sign a document acknowledges the system are work successfully and based on his expectations “sign-off project”.

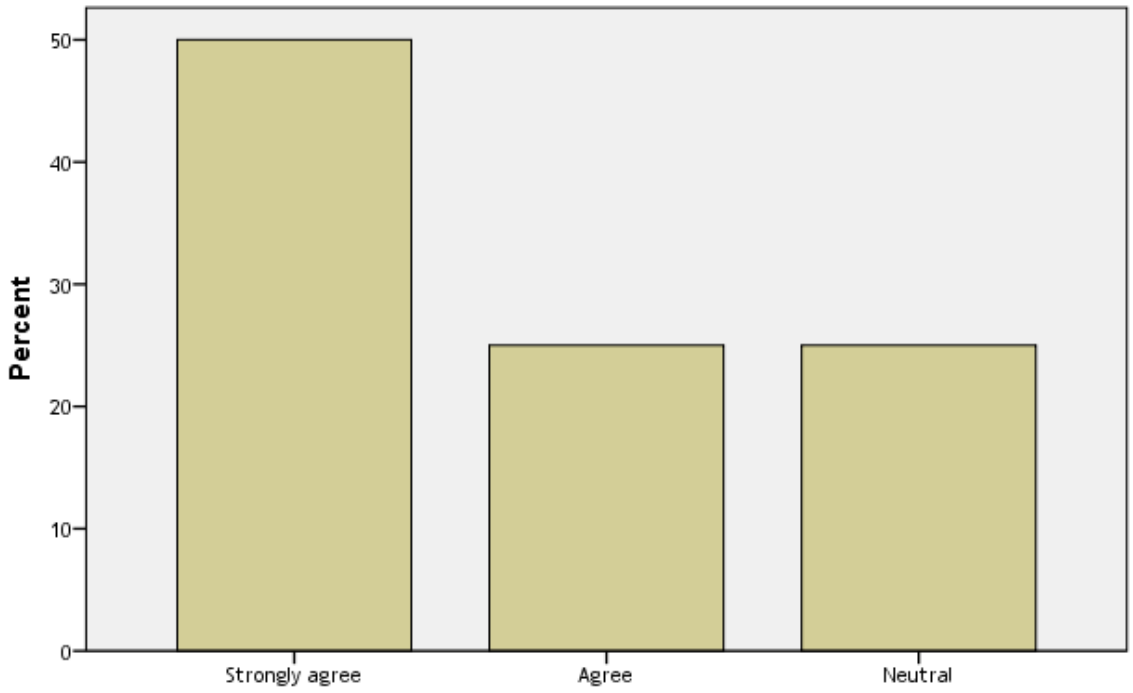
Table NO (5-41)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-20	Frequency	6	3	3	0	0	4.25	0.866	Strongly agree
	Percent	50	25	25	0.0	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-38)



After development stage are completed, customer must sign a document acknowledges the system are work successfully and based on his expectations “sign-off project”.

Variable 2-21:

The proposed methodology meets the needs of small and medium enterprises for project management and software quality assurance.

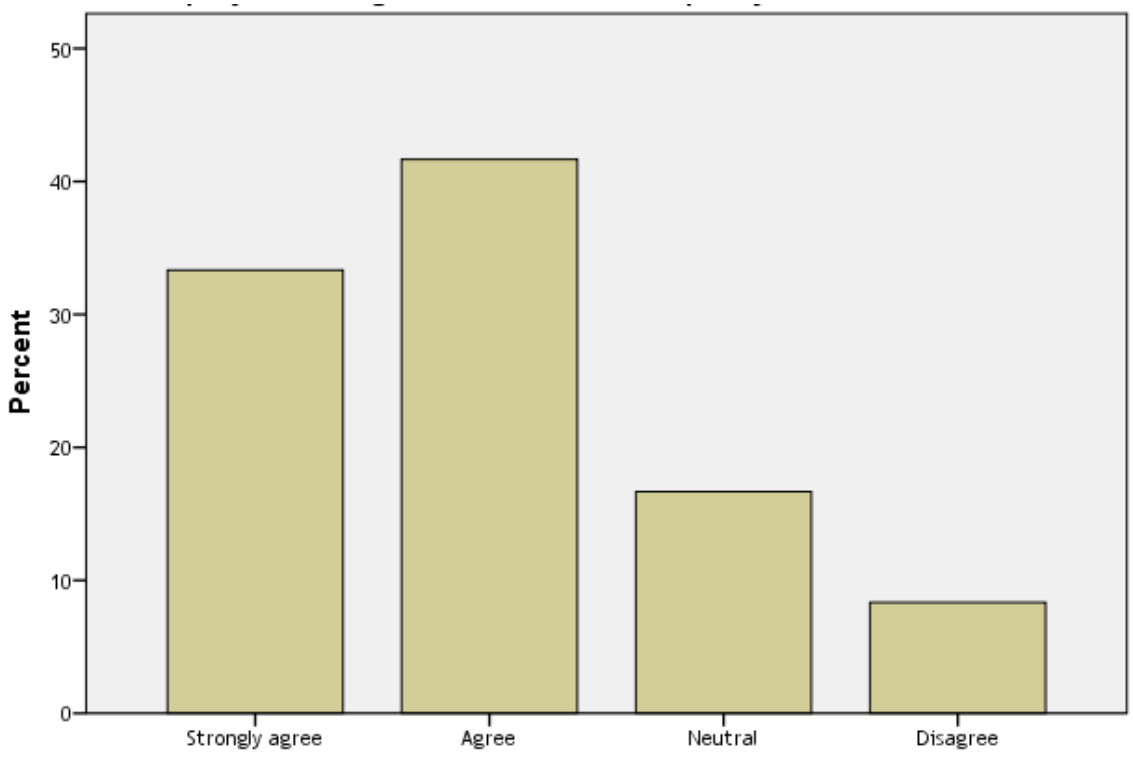
Table NO (5-42)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-21	Frequency	4	5	2	1	0	4.00	0.953	Strongly agree
	Percent	33.3	41.7	16.7	8.3	0.0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees

Figure NO (5-39)



The proposed methodology meets the needs of small and medium enterprises for project management and software quality assurance.

Variable number 2-22:

I will use the proposed methodology in future.

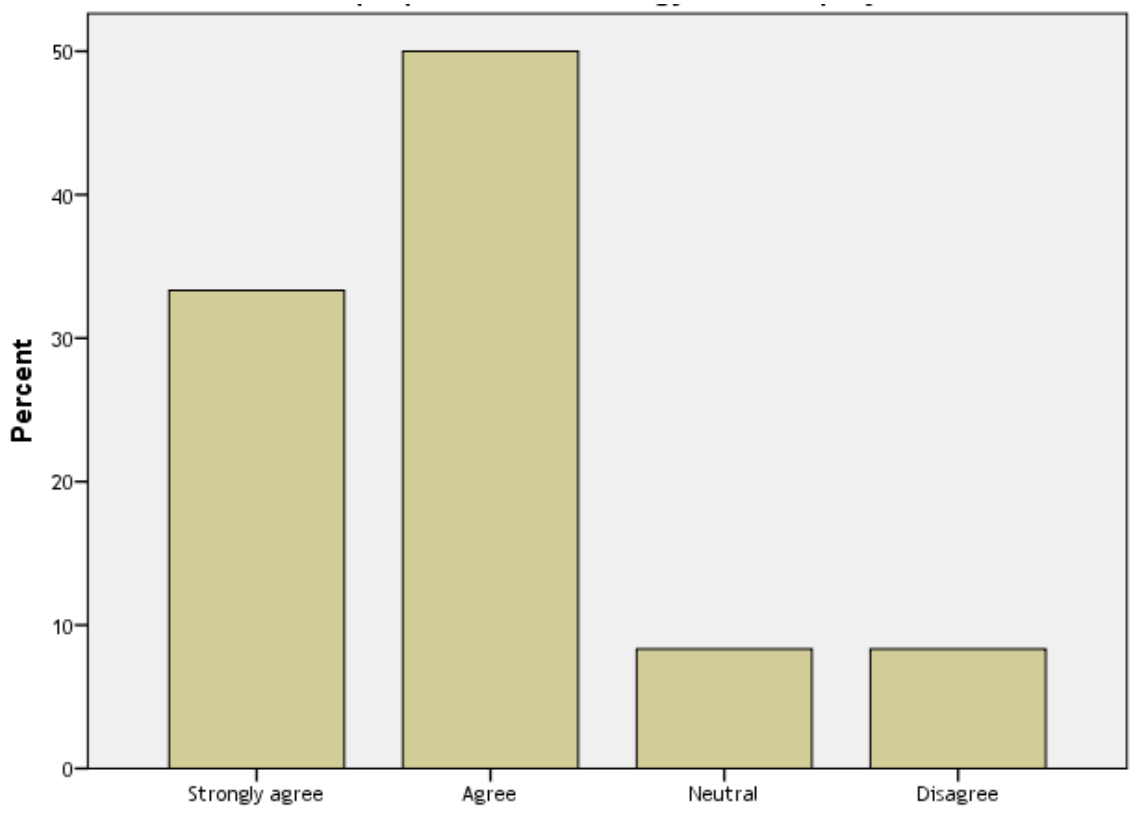
Table NO (5-43)

variable	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-22	Frequency	4	6	1	1	0	4.08	0.900	Strongly agree
	Percent	33.3	50	8.3	8.3	0			

Source: Case Study 2014

The above table Display the result of variable is strongly agrees.

Figure NO (5-40)



I will use the proposed methodology in future.

5.3.2 Questioner number two – summary

Hypotheses number one:

The most important characteristic of the proposed methodology, shortening development life in three basic stages (**initial stage** "information gathering" **development** "design – code writing – testing", **review and delivery**), merge CMMI model Level II with ISO / IEC 25010 standard and identify an effective mechanism for manage payment process. Variable:

1. Divide development life cycle for three stages (1- information gather, 2- Development "design, writing code, testing", 3- review and delivery) enough for small projects.
2. Integration ISO / IEC 25010 software quality standard with the requirements report ensures software quality product.
3. Merging CMMI level II with ISO / IEC 25010 in one methodology enough for small enterprises.
4. Submit a requirements report in triplicate (draft - proposed - review) ensures to coverage the main customer needs.

Figure NO (5-41)

Hypotheses #1	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #1-1	Frequency	5	6	1	0	0	4.33	0.651	Strongly agree
	Percent	41.7	50	8.3	0.0	0.0			
Variable #1-2	Frequency	5	4	3	0	0	4.14	0.835	Strongly agree
	Percent	41.7	33.3	25	0.0	0.0			
Variable #1-3	Frequency	4	5	3	0	0	4.08	0.793	Strongly agree
	Percent	33.3	41.7	25	0.0	0.0			
Variable #1-4	Frequency	5	3	4	0	0	4.08	0.900	Strongly agree
	Percent	41.7	25	33.3	0.0	0.0			

Hypotheses result	Frequency	19	18	11	0	0	4.19	0.724	Strongly agree
	Percent	39.59	37.5	22.91	0.0	0.0			

Hypotheses number two:

Effective management for information gathering stage must be submit two reports, the first one (functional requirements - non-functional requirements), the second report for technical requirements, each report submitted in three copies (draft - a proposal - approval). Variable:

1. The initial meeting between the customer and project manager ensures to cover all functional and non-functional requirements and then negotiated.
2. Review the project manager the initial copy of requirements (draft) report makes it Uncluttered and orderly for readers.
3. It is very importance to send a requirements report to customer for final approval.
4. If the customer approval the final copy of a requirements report reflects the level of understanding and agreement between the customer and the project manager on the project requirements.
5. It is very importance to develop separate report for technical requirements? "Project limits, cost estimation, time estimation, software quality and other requirements".
6. Provide the technical report requirements in three copies (draft - proposed – approval) ensures to coverage of all the technical requirements and the effectiveness of the resulting report.
7. Documents and reports that are offered through the various stages of the development process facilitate project management and quality assurance.
8. Pay a percentage of the project total cost after writing the basic requirements reports grantees the customer right and suppliers.

Figure NO (5-42)

Hypotheses #2	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #2-1	Frequency	3	6	1	1	1	3.75	1.215	Strongly agree
	Percent	25	50	8.3	8.3	8.3			
Variable #2-2	Frequency	4	5	3	0	0	4.08	0.793	Strongly agree
	Percent	33.3	41.7	25	0.0	0.0			
Variable #2-3	Frequency	8	4	0	0	0	4.67	0.492	Strongly agree
	Percent	66.7	33.3	0.0	0.0	0.0			
Variable #2-4	Frequency	7	3	1	0	1	4.25	1.215	Strongly agree
	Percent	58.3	25	8.3	0.0	8.3			
Variable #2-5	Frequency	7	3	2	0	0	4.42	0.793	Strongly agree
	Percent	58.3	25	16.7	0.0	0.0			
Variable #2-6	Frequency	6	4	1	1	0	4.25	0.965	Strongly agree
	Percent	50	33.3	8.3	8.3	0.0			
Variable #2-7	Frequency	7	4	1	0	0	4.50	0.675	Strongly agree
	Percent	58.3	33.3	8.3	0.0	0.0			
Variable #2-8	Frequency	5	4	2	1	0	4.08	0.996	Strongly agree
	Percent	41.7	33.3	16.7	8.3	0.0			
Hypotheses result	**Frequency**	47	33	11	3	2	4.26	0.807	Strongly agree
	Percent	48.95	34.37	11.45	3.13	2.1			

Hypotheses number three:

Effective management development stage should be divided into three phases under schedule already specified (design - writing code - test), **design** – deliver preliminary design to customer for review and take his opinion to modify or adopt implementation, **code writing** - should be based on clear criteria and defensive programming, **testing** - should be based on specific and clear criteria and tools. Questions:

1. Deliver a model or sketch of the proposed design to customer gives him a clear idea about how the future system will be like.
2. It is very importance the customer accepts the final design and must be care about his opinion.
3. Implementation the final design must be based on the final approval of the customer.
4. Attention to (standards in write code - defensive programming) in small enterprises guarantees write good code line and achieve software quality standards.
5. Identify tools and weaknesses must be tested as acceptable level of protection for software product.

Figure NO (5-43)

Hypotheses #3	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #3-1	Frequency	6	3	3	0	0	4.25	0.866	Strongly agree
	Percent	50	25	25	0.0	0.0			
Variable #3-2	Frequency	5	6	1	0	0	4.33	0.651	Strongly agree
	Percent	41.7	50	8.3	0.0	0.0			
Variable #3-3	Frequency	5	5	1	1	0	4.17	0.937	Strongly agree
	Percent	41.7	41.7	8.3	8.3	0.0			

Variable #3-4	Frequency	3	4	3	1	1	3.58	1.240	Agree
	Percent	25	33.3	25	8.3	8.3			
Variable #3-5	Frequency	4	6	1	1	0	4.08	0.900	Strongly agree
	Percent	33.3	50	8.3	8.3	0.0			
Hypotheses result	Frequency	23	24	9	3	1	4.06	0.899	Strongly agree
	Percent	38.4	40	15	5	1.6			

Hypotheses number four:

Effective management Review and delivered stage -customer must be delivered a initial version of project to create a list “checklist” for his observations to achieve customer satisfaction value, holding meeting with the project manager to discuss customer observations, rework on customer observations and then customer pay final paid before customer signs a document acknowledging receipt of the software product correctly and without errors according to their needs and expectations, according to report agreed requirements. Variable:

1. When the first version of product is delivered the customer must be establishment checklist about his observations to ensure value (customer satisfaction).
2. Meeting between customer and project manager to discuss the checklist leads to understanding the customer’s observations clearly.
3. After development stage are completed, customer must sign a document acknowledges the system are work successfully and based on his expectations “sign-off project”.

Figure NO (5-44)

Hypotheses #5	scale	Strongly agree	agree	Neutral	Disagree	Strongly disagree	Mean	Std. Deviation	Result
Variable #4-1	Frequency	6	5	1	0	0	4.42	0.669	Strongly agree
	Percent	50	41.7	8.3	0.0	0.0			
Variable #4-2	Frequency	6	5	1	0	0	4.33	0.888	Strongly agree
	Percent	50	41.7	8.3	0.0	0.0			
Variable #4-3	Frequency	6	3	3	0	0	4.25	0.866	Strongly agree
	Percent	50	25	25	0.0	0.0			
Hypotheses result	Frequency	18	13	5	0	0	4.36	0.758	Strongly agree
	Percent	50	36.2	13.8	0.0	0.0			

5.4 conclusions

By the end of this chapter, I was covered and discuss data analytics and result for questionnaires (questionnaire number one – questionnaire number two, questionnaire number one collecting information about proposed methodology from viewpoint of researchers, developers, project managers and each one related to software development process about time and cost estimation, design, coding and testing until deliver the software product and find out what are the difficulties they face in each of these steps and then analyze the data for giving clear outlines about the proposed solutions.

Questionnaire number two, Send proposal methodology for testing in the real development environment and then collecting data about its efficiency and ability to resolve problems. Include all the ideas that relate to the assessment proposal activity, can also obtain notation or sufficient information when we needed to make some modified for methodology.

After completion analyzing data using SPSS and Likert Scale, we have got clear and accuracy results represent s a summary for study allow us to determine recommendations and how to applying software quality standards on small and medium enterprises.

CHAPTER (6)
CONCLUSION AND FUTURE WORK

6.1 Introduction

The biggest challenges facing software industry in Sudan and developing countries in particular and generally startup and small company on the software development field is how to follow software process improvement and software process improvement standard, definitely that need additional resources and a lot number of experienced and specialists employees in the field of software quality, often these resources are not available in these institutions and there is no sufficient time to train existing staff for software quality standards, because these institutions rely on rapid development and profit for manage their work, on the other hand, customers lack the knowledge of the software quality standards and the importance to apply in their applications, also developers and project managers on small and medium enterprises lacks a culture for using and applying software quality standards, top management believes software quality standards are developed and published for large organizations that have considerable resources and a sufficient number of employees and make customer satisfaction the main goal for development team, ignoring long list of the software quality characteristics and project management activities that must be adhered to develop a product meets customers needs work correctly, efficacy and without errors reflected negatively on the software performance and shows weakness in the basic functional tasks as a result of random planning or poor implementation these challenge the biggest challenge facing small and startup enterprises as well as lack of enough testing related to functional requirement, usability, protection and other sensitive features before deliver project, these results are a summary of previous studies in the field of the software quality standards, I mentioned in detail on previous chapters.

6.2 General - effort of study

The current study aims to find solutions for problems that faced small and medium-sized companies in the field of software development on the Sudanese market particular and generally developments country because whole target companies faced the challenges and problems [6].

The proposal solution is to development methodology for project management and software quality, the main characteristics is simple, efficient and inexpensive use to merge ISO/IEC 25010 standard with a CMMI model level II.

Apply the proposal methodologies on tow track first one called “Customer responsibility” the second track called “Supplier responsibility” which one summarizes customer and supplier responsibilities during whole development life cycle of product from the initial stage (writing functional and non-functional requirements - technical project scope) until deliver project. Abbreviated development cycle in three main stages to simplify as follows:

1. The initial stage, at this stage submission two main reports the first one software requirements specification (functional requirements and non-functional requirements) the second report is technical project scope (cost and time estimation, quality, risk management, etc.) Each report written in three copies (draft - proposed - approval) and there are several meetings between customer and project manager to discus and covered all customer and project needs, at the same stage after knows the main functional, nonfunctional requirements and feature for proposal project customer must be paid a percentage of the total cost and specify the payment mechanism.

2. Development stage, divided into three main sub-stages (design, code writing, testing “system evolution”) as follows:
   i. Design, at this phase must be delivered model or sketch to customer about proposal design shows shape of the future system, discussed between customer and project manager to reach a final agreement and then implement the final design.
   ii. Code writing, at this phase start to implement project architecture and components, code must be written on clear standards and defensive programming concepts to avoid a long list of problems and challenges that will be facing product during operation.

iii. Testing, at this phase are preformed simple and quick tests before delivery project, protection and security must be the most importance test at this phase and then hold a meeting between development team members to make sure the project has been developed correctly, without errors according to customer expectations knows Beta Presentation.

3. Review and delivery, at this stage must be delivered an initial version of the project to customer to be reviewed and express an opinion about the system performance using simple form called “Check-list” and then hold a meeting between customer and project manager to discuss the checklist and then re-work after complete work must be inform customer to re-check the list again and makes sure the system work correctly, without errors according to his expectations, finally sign a document known “Sign-off project” as acknowledgment about that and pay the remaining cost before receipt of the final version.

Proposed methodology has been developed based on assumptions have been substantiated by questionnaire, respondents who were targeted researchers, developers and project managers, follow complete the development solution, methodology was send to a number of institutions to be tested in a realistic working environment then collect data using another questionnaire to be measure, analyze result and know the level of the study's success to achieve its objectives, chapter five Percents the results detail.

6.3 limitations

A great effort was made in this study aiming to develop effective and applicable solutions to overcome the problems, challenges faced by small and medium-sized enterprises meet their needs under their material and human potential based on the recommendations of previous studies that research in the problems faced small and medium enterprises about how to apply software standards and on the other hand studies tried to develop solutions for the seam problems, views of a large segment of specialists and researchers in the field of software engineering, and finally prove some hypotheses used to develop methodology has proved a great success

according to data collected and results from institutions that have tested proposed solutions, but it faces some limitation can be identified as follows:

1. Proposed methodology target a small- and medium-sized companies that have a few number of employees, members of development team ranging from 5 to 20 members for each project that means it is ineffective when the development team is too big.

2. Proposed methodology, target small and medium-sized projects that can be completed in a short period of time about 4 to 8 weeks in the sense that it is not effective with large projects that require months or years to developed.

3. The most technical recommendations and suggestions on the proposed methodology for web projects that thing probably represent some shortcomings but at advanced level of abstraction will suitable for other development environment.

4. Proposed methodology was not subject to enough tests in the large scale of institutions to ensure effectiveness results, number of the institutions that test proposal solution about 12 companies.

The limitation of the current study may be new keys questions and hypotheses can be overcome and resolved in subsequent studies or later versions of the proposed methodology meets the needs and ambitious project managers and developers in the field of software development.

6.4 recommendations

The current study was conducted in a vitality area and high degree of importance in the field of software development, discuss problems that facing software industry in the Sudanese market, developed solutions to avoid these problems and reduce the failure of products produced by national institutions and raise the quality of their products to become at the same level big companies that has enough resource to follow software process improvements and software process improvements standards.

Attention to products quality of the national companies working in the software development field reflected positively on progress and success and on the other hand supports and bushes the national economy. The current study concluded many recommendations that must be adhered to bypass to apply software process improvement and software process standards in small and medium-sized companies in Sudan in particular and the Middle East generally, can be summarized as follows:

1. Ensure success to apply software quality standards in the small and medium-sized enterprises, software standards or project management methodology must be simple, inexpensive and suit their needs.

2. Providing software quality standards meet small enterprises needs must be merge more than one software quality standard and quality improvement to take advantage of their different characteristics and preferred to be a specialist in a specific development environments.

3. Applying ISO/IEC 25010:2011 for software quality assessment and improvement on small and medium enterprise enough to improve their software quality.

4. Applying level II from CMMI model on small and medium enterprise meets their needs and enough to improve their software development process maturity.

5. To develop software product with high level of quality must be apply software quality characteristics during product development life cycle and monitor their implementation and put technical recommendations.

6. Project management methodologies in small enterprises must be identifies tools for measuring, analysis, provide documents and reports during whole development process stages and care about client opinion.

7. Provide simple and clear training program during developing projects to qualify staff and improve their skills about how to apply quality standards.

8. Must be specified relationship between customer and project Manager and roles of each it at the whole software development stages.

9. Must be finding easy way to convince top management to integrate software process improvement and software process improvement standards within their institutions routine.

10. Proposed methodology able to solve most of the problems and challenges facing small and medium-sized companies, so strongly recommend using in target companies.

Recommendations of the current study added to the large number of studies in the software development field, I hope contribute a positive and effective to solve one of the biggest challenges facing software industry in Sudan and raise the level of software products quality produced by national institutions.

6.5 future work

The current study is a simple and modest contribution according to challenges and problems that faced software development failed to apply software process improvement and software process improvement standards, I hope to contribute to find a new research questions that can recommend as following:

1. Study the level of the software quality before and after apply the proposal methodology for companies that work in the software development field.

2. Compare the proposed methodology with the other project management methodologies to determine the effectiveness of each one.

3. Conducting studies during different periods of time regarding level of committed companies to apply software quality standards and determine the level of improvement on their products.

4. Conducting studies about level of the effect companies that committed to apply software quality standards with the level of produce software.

The above recommendations on future research areas are considered as effective contribute and push a scientific research in the field of software development, software process improvement, software process improvement standards and develop solutions to raised problems.

6.6conclusions

By the end of this chapter, represents as the last chapter in this study discusses several topics about software process improvement, software process improvement standards, challenge faced the national companies to apply software quality standards and propose solution as follows:

Chapter one – introduction, discuss research problem, research objectives, proposed methodology, problem scope and main research question.

Chapter two – literature review , overview about what software quality from different view or perspectives, the basic concepts of software engineering related to the software process improvement (SPI) and software process improvement standard (SPI stander), the purpose and benefit for using SPI and SPI standard, the most important institutions and organization responsible to develop quality standards for software such as IEEE, ISO and IEC, the most important quality models and overview about CMMI, study about the ability to apply SPI with in Sudanese organization and the problem faced and finally studies that try to solve the problem and propose quality model appropriate with small and medium enterprise.

Chapter three – research methodology , overview about main concept and types of research, theoretical and main types of research methodology, research methods, main hypothesis of this study, develop proposal methodology, questionnaire number one collecting information about the components of proposed methodology from viewpoint of researchers, developers, project managers and analyze data and then make any modification on the proposed work, prove the hypothesis of the study by sending the proposal methodology to a number of target companies for testing it in the real live with real development process , questionnaire

number tow collecting information and give feedback from peoples who deal with the proposal methodology and their assessment of the level of successful proposal solutions for their problems about how to apply software process improvement and software improvement standards and then find out the result after analysis and the last point in this chapter the concludes discuses how to determine recommendations and future works.

Chapter four – proposal methodology, overview about study main idea, over view about CMMI level II, most importance project management methodologies (Agile, Scrum, XP), overview about proposal work (initial stage, development stage, Review and delivered stage), main feature about proposal methodology, initial stage (software requirements specification – SRS, SRS section, technical project scope), development stage (design stage, tips to design a prefect web form layout, search engine optimizations), code writing (defensive programming, rules for writing code, software project security, beta Presentation), Review and delivery stage (review and checklist created, checklist desiccation meeting, checklist rework, review checklist , sign-off the project, final ratio paid, publish project or system).

Chapter five - result, discussion and validation, at this chapter discuss the result of data analysis and validation about questioner number one and questionnaire number two in details.

Chapter six – conclusion and future work, discuss the general effort of study, limitation of the study, recommendation of the study and future works.

REFERENCES

Reference:

[1] JEFF TIAN - Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement - 2005.

[2] Dr. Holger Giese - Software Quality Assurance

[3] Prof. Motoei AZUMA - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE

[4] ISO/IEC 25010 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Software and quality in use models - 2011.

[5] A. M. Adb-algager "*Problems and Future Trends of Software Process Improvement in Some Sudanese Software Organizations*" in international conference on computer electrical and electronic engineering (ICCEEE) - 2013.

[6] Dr. G.K.Viju **"***The Impact of Software Process Improvements in Small and Medium Scale Enterprises***"** in International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-4, September 2013.

[7] S. Rajasekar - School of Physics, Bharathidasan University, Tiruchirapalli – 620 024, Tamilnadu, India - Software Quality Assurance - Research methodology.

[8] C.R. KOTHARI – Research methodology methods and techniques

[9] OWASP TOP 10 – 2013 the ten most critical security risk.

[10] John Doe - Recommended Practice for Software Requirements Specifications (IEEE).

[11] Karl E. Wiegers – software requirement specification.

[12] Gerard J. Holzmann - Rules for Developing Safety Critical Code.

[13] Manahil H. Hassan – Software industry in sudan : challenges and possible solutions.

[14] Len Estrin, Bill Anderson - Working with Small Manufacturing Enterprises: An Analysis of TIDE Demonstration Projects.

[15] Mark C. Paulk - Practices of High Maturity Organizations, 1999 SEPG Conference, Atlanta, Georgia, 8-11 March 1999.

[16] Amr Noaman Abdel-Hamid - Process Increments: An Agile Approach to Software Process Improvement - 2011 Agile Conference.

[17] O. Tolga Pusatli - Software Measurement Activities in Small and Medium Enterprises: an Empirical Assessment - Acta Polytechnic Hungarica Vol. 8, No. 5, 2011.

[18] Deepti Mishra and Alok Mishra - Software Process Improvement in SMEs: A Comparative View.

[19] Mark C. Paulk - Using the Software CMM in Small Organizations.

[20] Z. Abbassi - CMM for Small Organizations in Iran - The Second International Conference on Innovations in Information Technology (IIT'05).

[21] Ashish Agrawal, Sadhana Singh, Malay Tripathi, L.S. Maurya - Implementing AGILE in Small Organizations - International Journal of Information & Computation Technology.ISSN 0974-2239 Volume 4, Number 7 (2014).

[22] Kitchenham and Pfleeger - Pfleeger et al., 2002.

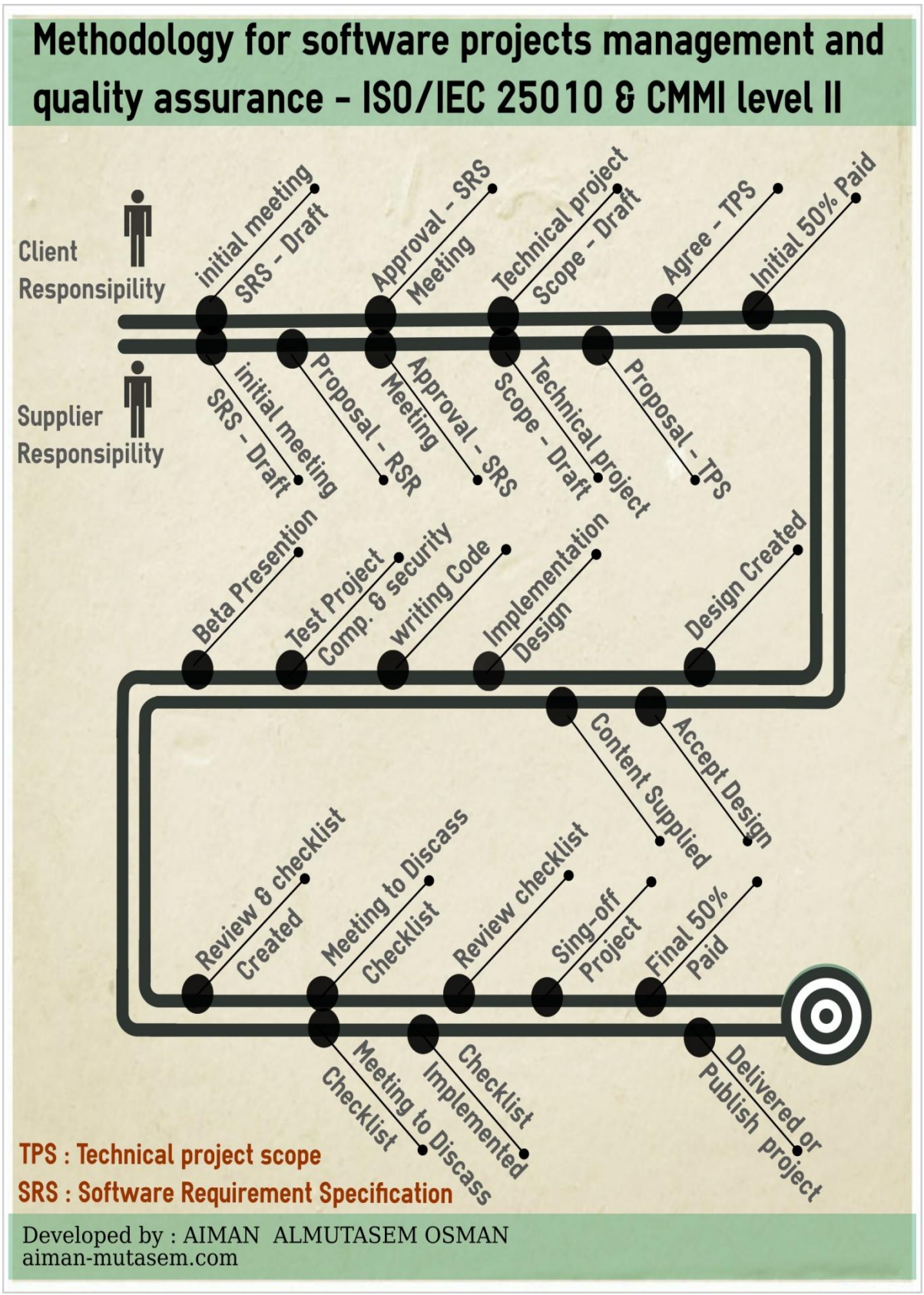
[23] Abbott, J.J, "Software Process Improvement in a Small Commercial Software Company" Software Engineering Process Group Conference – 1997.

[24] D. Johnson and J Brodman, "Tailoring the CMM for Small Businesses, Small Organizations, and Small Projects", *Software Process Newsletter* No 8, 1997.

[25] K., Hansen, H.W., These, K.:" *Applying and Adjusting a Software Process Improvement Model in Practice: The use of the IDEAL Model in a Small Software Enterprise*". Proceedings of ICSE 2000.
[26] Michael James, Scrum Reference Card, 2012 – 2013.

APPENDICES

Appendix 1: Proposal methodology



Appendix 2: Software requirements specification (SRS).

The proposal software requirements specification (SRS) content seven sections the first one ***Introduction*** content project overview and project scope, ***General description*** content project state, project user characteristics, project assumption ***project requirements*** content functional requirements, user interface requirements, performance efficiency, operability, security, compatibility, Maintainability and transferability ***product user senior and use case*** - ***Appendix*** - ***References*** and the last one ***index***.

Project Name: Requirements Document (version 1.0)

Project:
Date(s):
Prepared by:
Document status: *Draft* __ *Proposed* __ *Approved*

1. Introduction

1.1 Purpose of This Document

This document is intended to guide development of ***project name***. It will go through several stages during the course of the project:

1. **Draft:** The first version, or draft version, is compiled after requirements have been discovered, recorded, classified, and prioritized.
2. **Proposed:** The draft document is then proposed as a potential requirements specification for the project. The proposed document should be reviewed by several parties, who may comment on any requirements and any priorities, either to agree, to disagree, or to identify missing requirements.
3. **Validated and Approved:** Once the various stakeholders have agreed to the requirements in the document and then accepted by represent atives of each party of stakeholders as an appropriate statement of requirements for the project. The developers then use the requirements document as a guide to implementation and to check the progress of the project as it develops.

1.2 projects over view

Describe this project or product and its intended audience, or provide a link or reference to the project.

1.3 scope this Specification

Describe the purpose of this specification and its intended audience. Include a description of what is within the scope what is outside of the scope of these specifications. For example:

In scope

This document addresses requirements related to stage 2 of Project X:

- ……..
- ……..

Out of Scope

The following items in stage 3 of Project X are out of scope:

- ……..
- ……..

2. General Description

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

2.1 Project State

How does this product affected to other products? *(If there is product family), is it independent and self-contained? Does it interface with a variety of related systems? Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.*

2.2 Project User Characteristics

Create general user categories for each type of user who will be using the product such as follow:

- category name (manager/visitor/registered users)
- experience
- technical background
- Skills or other characteristics needed in the product/service.

2.3 Project Assumptions

List any assumptions made about the project as follow:

Resource assumptions, can be any one item utilized that is required to complete the project. Resources can be human resource and materials. An

assumption made on projects for human resources is that individuals will work 40 hours per week on the day shift. The materials to complete the project will be readily available to be utilized.

Scope assumptions, are related to scope change, and managing the scope change; a scoped assumption would be following a scope change control so that if the scope changes that other dependent factors are taken into consideration if additional scope is being added or removed.

Delivery assumption, focus on the end, in which the project will provide the end result of why it was being done in the first place. Projects are started to enhance and or create something new, which will add value. The delivery assumes that what was planned for within the project will in fact be delivered.

Budget assumptions, not limited to, margin of error, percentage allocation for resources versus materials, the overall cost of day-to-day activity will not increase, personnel costs will not change and overall economical conditions will stay the same.

2.4 Project Constrain

Describe items that will constrain on the project options for example: as follow

- Interact with an old system.
- Operation system environment.
- Accessibility, management and security.
- Other constraints such as programming language or framework.

2.5 project Dependencies

List dependencies that affect on the requirements or project implementation, for examples:

- The new product will require a software product X.
- The new product will require a specific version of Framework X.
- Task X needs to be completed before this Task X can be built.

3. Project Requirements

Describe all system requirements in enough detail for development team and testers to verify that the system satisfies requirements.

Organize these requirements in a way that works best for development process.

Describe every input, every output from the system, and every function performed by the system in response to an input or in support of an output.

Each requirement must be numbered to identify.

Each requirement must be proirized as (hight, mediam, low).

Take on account the good characteristic of requirement as correct, unambiguous, complete, ranked, verifiable, modifiable.

3.1 Functional Requirements

Describe whole functional requirements project on details, clearly, numbered.

NO #	Requirement	Comments	Priority	Date	Reviewed / Approved
1	Descript requirement #1	More information about requirement #1	high	mm/dd/ yy	who reviewed and approved
2	Descript requirement #2	More information about requirement #2	high	mm/dd/ yy	who reviewed and approved
3	Descript requirement #3	More information about requirement #3	medium	mm/dd/ yy	who reviewed and approved
4	Descript requirement #4	More information about requirement #4	low	mm/dd/yy	who reviewed and approved

Note: Requirements and tasks must be implemented in order.

3.2 User Interface Requirements

Describe the importance characteristics of the main interface between the product and its users and prefer use sketch and prototype.

3.3 Reliability

Specify software characteristics maintain a specified level of performance when used under specified conditions. Such as follow:

3.3.1 **Availability**

such as hours of operation and Level of availability required.

3.3.2 **Fault tolerance**

including specific characteristics in cases of software faults or of infringement of its specified interface.

3.3.3 **Recoverability**

Including specific characteristics re-establishes a specified level of performance and recovers the data directly affected in the case of a failure.

3.4 Performance efficiency

Specify software characteristics inflicted to appropriate performance, relative to the amount of resources used, under stated conditions. Such as follow:

3.4.1 Time behavior: Including specific and measurable requirement for:

- Rate of the response time.
- Rate of the processing time.

3.4.2 Resource utilization

Rate of the project resource utilization like RAM utilization, HDD space needed.

3.5 Operability

Specify a set of the software characteristics and sub characteristics to understand, learned, used and attractive to the user, when used under specified conditions as follow:

3.5.1 Recognizability

Identify features allow users to find out whether the product is suitable for their functional needs or not for example, initial impressions or any other relevant documentation.

3.5.2 Learnability

Identify features allow users how to implement and execute the program.

3.5.3 Easy to use

Identify features make product easy to use and control it such as rich text format tools or tools for reordering items.

3.5.4 Helpfulness

Identify features allow program or application to provide assistance when users needed as easy to access information in a comprehensive and effective.

3.5.5 Attractiveness

Identify attributes of the software that increase the pleasure and satisfaction of the user, such as the use of color and the nature of the graphical design.

3.5.6 Technical accessibility

Identify feature to allow users disabilities associated with ageing to interact with system. Such as font size and colors contrast.

3.6 Security

Specify the protection of system items from accidental or malicious access, use, modification, destruction, or disclosure.

3.6.1 Confidentiality

Identify features allow the product protection from unauthorized disclosure of data or information, whether accidental or deliberate.

3.6.2 Integrity

Identify features allow product to ensure the data is correct and complete during processing and transport.

3.6.3 Non-repudiation

Identify features allow product to ensure to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.

3.6.4 Accountability

Identify features allow product ensure to which the actions of an entity can be traced uniquely to the entity.

3.6.5 Authenticity

Identify features allow product ensure to which the identity of a subject or resource can be proved to be the one claimed.

3.7 Compatibility

Specify software characteristics allow two or more software components to exchange information and/or to perform their required functions while sharing the same hardware or software environment.

3.7.1 Replaceability

Identify feature allow product to which the software product can be used in place of another specified software product for the same purpose in the same environment, such as component for software update.

3.7.2 Co-existence

Make sure software product can co-exist with other independent software in a common environment sharing common resources without any detrimental impacts.

3.7.3 Interoperability

Identify feature to allow software product to cooperatively operable with one or more other software products.

3.8 Maintainability

Specify feature allow software product can be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

3.8.1 Modularity

Identify feature allow a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.

3.8.2 Reusability

Make sure a system or computer program an asset can be used in more than one software system, or in building other assets.

3.8.3 Analyzability

Include providing mechanisms for the software product to analyze its own faults and report on the conditions prior to a failure or other event.

3.8.4 Changeability

Identify feature allow software product enables a specified modification to be implemented. The ease with which a software product can be modified. Implementation includes coding, designing and documenting changes.

3.8.5 Modification stability

Identify feature allow a software product to avoid unexpected effects from modifications of the software.

3.8.6 Testability

Identify feature allow a software product enables modified software to be validated.

3.9 Transferability

Specify features allow a system or component can be transferred from one environment to another.

3.9.1 Adaptability

Identify feature allow a software product can be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered. Includes the scalability of internal capacity (e.g. screen fields, tables, transaction volumes, report formats, etc.)

3.9.2 Portability

Specify features allow a system or component can be transferred from one hardware or software environment to another.

3.9.3 Installability

Identify feature allow a software product to be successfully installed and uninstalled in a specified environment.

Note : *some of the above characteristics embedded automatically with in the development environment and not need any effort to develop it again, must be write note about that, any characteristic has been ignored must be write note Why was ignored*.

4. Product User Scenario and Use Case

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document. A business scenario:

- Describes a significant business need.
- Identifies, documents, and ranks the problem that is driving the scenario.
- Describes the business and technical environment that will resolve the problem.
- States the desired objectives.
- Shows the “Actors” and where they fit in the business model.
- Is specific, and measurable, and uses clear metrics for success.

5. Appendix

If you wish to append any documents, do so here. You may wish to include some or all of the following:

- Personas and scenarios developed for this project
- Transcripts of user interviews, observations, or focus groups
- Copies of messages which contain user requirements
- Original project proposals or other historical documents
- Lists of similar projects or products, with notes about how they differ from yours
- A list of requirements which were "wish-listed" or marked unfeasible at Percent
- Original screen mockups, if they are relevant

6. References

List references and source documents, if any, in this section.

7. Index

If your document is very large, consider compiling an index to help readers find specific items.

Appendix 3: simple software requirements specification

Simple copy of (SRS) used when the project is small, clear requirement and project manager is expert by Describe all system requirements in enough detail for development team and testers to verify that the system satisfies requirements.

- Describe every input, every output from the system, and every function performed by the system in response to an input or in support of an output.
- Each requirement must be numbered to identify.
- Each requirement must be proirized as (hight, mediam, low).
- Take on account the good characteristic of requirement as correct, unambiguous, complete, ranked, verifiable, modifiable.
- Describe whole functional requirements project on details, clearly, numbered.

NO #	**Requirement**	**Comments**	**Priority**	**Date**	**Reviewed / Approved**
1	Descript requirement #1	More information about requirement #1	high	mm/dd/yy	who reviewed and approved
2	Descript requirement #2	More information about requirement #2	high	mm/dd/yy	who reviewed and approved
3	Descript requirement #3	More information about requirement #3	medium	mm/dd/yy	who reviewed and approved

Note: Requirements and tasks must be implemented in order.

Appendix 4: technical project scope form

Technical project scope form

Project name

Version <Type Version #>

(Draft, Proposal, Approval Document)

Project Manager:

.....

.....

Project Owner:

.....

.....

My signature indicates approval of this Project Scope Statement.

Approved by:

……

……

……

……

The technical project scope (TPS) identifies the preliminary scope of a system. It expands on the earlier work done in the software requirements specification (SRS). This statement is a required deliverable of the Concept Development Phase of the System Development Life Cycle (SDLC) and replaces the System Boundary Document deliverable.

PURPOSE/JUSTIFICATION

Copy over project purpose and justification from the software requirements specification. Elaborate with any additional information as it relates to scope definition.

Any work not explicitly included in the technical project scope is implicitly excluded from the project.

OBJECTIVES

Define project objectives. Describe each objective using measurable success criteria, such as anticipated productivity improvements, cost reduction, improvements in business processes, citizen support, revenue enhancements, technical efficiencies, etc. The stated objectives will become the basis for the acceptance criteria.

SCOPE DESCRIPTION

List the applications, business processes, organizations, and technical components that will be affected/changed by the completion of the project. By inventorying these items, the scope of the project can be derived. The scope description may be developed using the following table.

Hardware • **< Item 1 >**
Legacy Applications and Databases • **< Item 1 >**
Interfaces and Networks • **< Item 1 >**

Locations • **< Item 1 >**
Organizations • **< Item 1 >**
Security • **< Item 1 >**
Business Processes • **< Item 1 >**

BOUNDARIES

Describe the limitations of the project scope, typically expressed as items outside of the project jurisdiction.

Hardware • **< Item 1 >**
Applications • **< Item 1 >**
Interfaces • **< Item 1 >**
Locations • **< Item 1 >**
Organizations • **< Item 1 >**
Security • **< Item 1 >**
Business Processes • **< Item 1 >**

DATA MIGRATION STRATEGY

Identify the number and type(s) of legacy data sources that will need to be converted to the new system. Estimate the volume of data to be converted from each source. Provide a description of each of the data source characteristics (legacy database type, hardware platform, etc.) Determine time period of historical data to be converted, and identify any special requirements that may be necessary to convert this data. Also, describe preliminary strategy for data conversion and any special controls required.

DELIVERABLES

Begin to describe the deliverables that will be required from this project, including project management plans, systems documentation, etc. Describe deliverables in tabular form, and add lines and/or categories as required: Deliverables can be organized either by domain area or by SDLC phase.

<Project Management> • < Item 1 >	
<Applications> • < Item 1 >	
<System Documentation> • < Item 1 >	
<Hardware> • < Item 1 >	

ACCEPTANCE CRITERIA

Acceptance criteria are the metrics that must be met before the project services and proposed deliverables will be accepted. The acceptance criteria should describe unambiguous and measurable processes and conditions.

Project Management • Project Management Plan (PMP)	*Project Management Body of Knowledge* (*PMBOK*) compliant PMP, containing all relevant subsidiary plans, including scope, quality, staffing, communication, and integration plans. Plans shall include procedures, templates, and authority levels for all project management processes to be used on the project also specify.
Applications • Item 1	
System Documentation • Item 1	
<Hardware> • Item 1	

CONSTRAINTS

Describe any known project limitations or constraints, including resource limitations, schedule deadlines, support deadlines, regulatory issues, funding deadlines, etc., that are internal or external to the project. Include perceived or stated restrictions or limitations that may affect the performance of a project.

- First Constraint

ASSUMPTIONS

List and describe any perceived or stated project assumptions and the potential impacts of those assumptions if they prove to be false. Include any assumptions made to prepare the cost estimates, schedule, deliverables, and milestones. Also, include or address assumptions listed in the requirement specification.

- First Assumption

COST ESTIMATES

Detail the forecasted budget for the project, and identify all sources of funding. Estimate the total project cost, and provide a statement of the general accuracy of the estimate "rough order of magnitude". Ensure that costs are categorized based on expected expenditures per SDLC phase. The table provided is a sample; add more lines and change content as necessary.

The forecasted budget for the <Project Name> project is $<amount>. It is to be funded through <funding source/budget>.

Software Development	$
Internal Resources	$
Hardware Purchases	$
Operation and Maintenance Support Expenses	$
Independent Verification and Validation	$
Total ROM Budget Estimate	$

RISKS

Identify and describe any early known risks. Consider the project constraints and assumptions to help identify potential risks. Be sure to include risks identified in the Preliminary Risk Statement in the Project Charter, and add any new risk information. If known, include identified risk mitigation strategies for each risk.

- First Risk
- Second Risk
- Third Risk

FUND LIMITATIONS

Describe any sources of project funding that may have explicit expiration/sunset dates, and describe the impact of any funding limitations to the project.

STANDARDS

Describe known standards that will apply to this project, including SDLC methods, coding standards and conventions, platform preferences, project management standards, security, disaster recovery, and others.

This project will be executed in accordance with the standards specified in the following documents:

- Department of Information Technology SDLC.
- ISO/IEC 25010 software quality assessment and improvement.

Appendix 5: Security report form

Project name: ……

Security incident / occurrence report

To: …… incident NO: ……..

File Ref(s):

……

……

……..

……

Date/Time Received:

Report by:

Contact:

DETAILS OF INCIDENT/OCCURRENCE REPORT

Neater:

Date/time:

Location:

Description:

Action taken:

Submitted by: Date:

Appendix 5: Checklist review form

Checklist review form		
Project Title:	Date:	
Project manager:	Project owner:	
Sections	Issues	Comments
Section name - x	Describe issue number 1# on section x	
	Describe issue number 2# on section x	
	Describe issue number 3# on section x	
	Describe issue number 4# on section x	
	Describe issue number 5# on section x	
	Describe issue number 6# on section x	
Section name - y	Describe issue number 1# on section y	
	Describe issue number 2# on section y	
	Describe issue number 3# on section y	
	Describe issue number 4# on section y	
	Describe issue number 5# on section y	
	Describe issue number 6# on section y	
	Describe issue number 1# on section y	

Review checklist Form				
Project Title:	Date:			
Project manager:	Project owner:			
Sections	Issues	Y	N	Comments
Section Name - x	Describe issue number 1# on section x	☐	☐	
	Describe issue number 2# on section x	☐	☐	
	Describe issue number 3# on section x	☐	☐	
	Describe issue number 4# on section x	☐	☐	
	Describe issue number 5# on section x	☐	☐	
	Describe issue number 1# on section x	☐	☐	
Section Name - y	Describe issue number 1# on section y	☐	☐	
	Describe issue number 2# on section y	☐	☐	
	Describe issue number 3# on section y	☐	☐	
	Describe issue number 4# on section y	☐	☐	
	Describe issue number 5# on section y	☐	☐	
	Describe issue number 1# on section y	☐	☐	
	Describe issue number 1# on section y	☐	☐	

Appendix 6: project sign-off sheet

Project Name:	**Project Manager:**
Start Date:	**Completion Date:**
Project Duration:	**Sponsor:**
Project Goal:	
Project Deliverables:	
Clients:	
By signing this document, I acknowledge that I have delivered all the stated deliverables at the agreed to quality levels.	By signing this document, I acknowledge that I have received all the stated deliverables at the agreed to quality levels.
Project Manager Name and Signature:	**Sponsor Name and Signature:**
Date:	**Date:**

Remarks

Appendix 7: Questionnaire number one

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

<u>إستبيـــــــــــــــــان</u>

الأخ الكريم / الأخت الكريمـــــــــــــة
السلام عليكم و رحمة الله تعالي و بركاته
أدعوكم للمشاركة في دراسة أكاديمية لنيل درجة الماجستير في هندسة البرمجيات بعنوان "تطوير منهجية لتحسين جودة البرمجيات وإدارة المشاريع في المؤسسات الصغيرة والمتوسطة". هدف الدراسة تطوير منهجيه بسيطة فعالة وغير مكلفه تساعد في تحسين مستوي جودة البرمجيات وإدارة المشاريع في المؤسسات الصغيرة والمتوسطة وتلبي احتياجاتها . أرجو شاكراً الإجابة علي الأسئلة أدناه.

البيانات الشخصية

١. الدرجة العلمية ؟
بكلاريوس ☐ ماجستير ☐ دكتوراه ☐ غير ذلك ☐

٢. طبيعة العمل؟
باحث ☐ مطور ☐ مدير مشروع ☐ غير ذلك ☐

٣. سنين الخبرة؟
٢- ٥ ☐ ٥ – ١٠ ☐ ١٠ - ١٥ ☐ ١٥ فأكثر ☐

٤. العمر؟
أقل من ٣٠ ☐ ٣٠ – ٤٠ ☐ ٤٠ - ٥٠ ☐ ٥٠ فأكثر ☐

٥. الجنس؟
ذكر ☐ أنثي ☐

جودة البرمجيات

٦. تطبيق معايير جودة البرمجيات في المؤسسات الصغيرة والمتوسطة يضمن جودة منتجاتها؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٧. التكلفة المادية تؤثر في تطبيق معايير جودة البرمجيات؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٨. الشركات الصغيرة والكبيرة لها نفس القدرة علي تطبيق معايير جودة البرمجيات؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٩. تعقيد أي معيار لجودة البرمجيات أو منهجية لإدارة المشاريع يؤثر في عملية التطبيق؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٠. كل معايير جودة البرمجيات تلبي احتياجات المؤسسات الصغيرة وقابله للتطبيق؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١١. دمج أي من معايير جودة البرمجيات مع بعضها البعض للاستفادة من خصائصها المختلفة في نموذج واحد يضمن فعاليه النموذج الناتج؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٢. دمج خصائص معايير جودة البرمجيات في عملية التطوير ومن المراحل الأولية (كتابة تقرير المتطلبات) يضمن جودة المنتج البرمجي؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٣. مراقبة تطبيق خصائص معايير جودة البرمجيات خلال مراحل تطوير المنتج البرمجي يضمن جودة المنتج النهائي؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٤. وضع توصيات فنية بناء علي خصائص معايير جودة البرمجيات لتنفيذها أثناء عملية التطوير يساعد في ضمان جودة المنتج البرمجي؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

إدارة المشاريع

١٥. تطوير منهجية لتحسين جودة البرمجيات وإدارة المشاريع بناء علي احتياجات المؤسسات يجعل من السهل تطبيقها؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٦. تحديد وتطوير أدوات وإجراءات بسيطة وغير مكلفة لإدارة عملية القياس والتحليل أثناء عملية التطوير تؤثر بإيجابية في إدارة المشروع؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٧. الوثائق والتقارير التي يتم تقديمها خلال مراحل عملية التطوير المختلفة تساعد في سهولة إدارة المشروع وضمان جودة المنتج البرمجي؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٨. تحديد وتطوير أدوات وإجراءات ووثائق بسيطة وواضحة لإدارة (المتطلبات الوظيفية وغير الوظيفية - الزمن – التكلفة – المخاطر – الاختبار) يسهل عملية إدارة المشروع ويضمن جودة المنتج البرمجي؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٩. تطبيق المستوي الثاني من نموذج CMMI "إدارة المشاريع" كافي للمؤسسات الصغيرة؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢٠. تطوير منهجية لتحسين جودة البرمجيات وإدارة المشاريع موجه لمنصات تطوير محدده مثل (الويب – الموبايل) تكون أكثر فعالية؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢١. ضرورة إرسال تقارير ووثائق لاطلاع العميل أو صاحب المشروع بمستوي تقدم إنجاز المشروع خلال كل مراحل التطوير؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

إضافة إذا أمكن

.....

.....

.....

.....

شكراً جزيلاً علي اهتمامك بالإجابة علي الأسئلة أعلاه والمشاركة برأيكم الكريم. أملي كبير في أن تسهم كل الجهود المبذولة بصورة إيجابية في تحقيق أهداف الدراسة وتقديم حلول فعاله وبسيطة للمشاكل المطروحة.

الباحث : أيمن المعتصم بالله

Appendix 8 : Questionnaire number two

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

إستبيـــــــــــــــــان

الأخ الكريم / الأخت الكريمـــــــــــــة

السلام عليكم و رحمة الله تعالي و بركاته

أدعوكم للمشاركة في دراسة أكاديمية لنيل درجة الماجستير في هندسة البرمجيات بعنوان "تطوير منهجية لتحسين جودة البرمجيات وإدارة المشاريع في المؤسسات الصغيرة والمتوسطة". هدف الدراسة تطوير منهجيه بسيطة فعالة وغير مكلفه تساعد في تحسين مستوي جودة البرمجيات وإدارة المشاريع في المؤسسات الصغيرة والمتوسطة وتلبي احتياجاتها . أرجو شاكراً تحميل المرفقات والإطلاع عليها ومن ثم الإجابة علي الأسئلة أدناه.

البيانات الشخصية

١. الدرجة العلمية ؟

بكلاريوس ☐ ماجستير ☐ دكتوراه ☐ غير ذلك ☐

٢. طبيعة العمل؟

باحث ☐ مطور ☐ مدير مشروع ☐ غير ذلك ☐

٣. سنين الخبرة؟

٢-٥ ☐ ٥ – ١٠ ☐ ١٠ - ١٥ ☐ ١٥ فأكثر ☐

٤. العمر؟

أقل من ٣٠ ☐ ٣٠ – ٤٠ ☐ ٤٠ - ٥٠ ☐ ٥٠ فأكثر ☐

٥. الجنس؟

ذكر ☐ أنثي ☐

المرحلة الاولية – جمع المعلومات

٦. تقسيم دورة تطوير التطبيقات لثلاث مراحل (١- جمع المعلومات ٢- التطوير"تصميم، كتابة الكود ألاختبار – ٣- المراجعه والتسليم) كافي للمشاريع الصغيره؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٧. دمج معيار جودة البرمجيات ISO/IEC 25010 في تقرير المتطلبات يضمن جودة المنتج البرمجي؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٨. دمج المستوي الثاني من CMMI مع ISO/IEC 25010 في منهجية واحدة كافي للمؤسسات الصغيره؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٩. تقديم تقرير المتطلبات في ثلاث نسخ (مسودة – مقترحة - مراجعة) يضمن تغطية كل احتياجات العميل؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٠. الاجتماع الاولي بين العميل ومدير المشروع يضمن تغطية كل المتطلبات الوظيفية وغير الوظيفية ومن ثم التفاوض عليها؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١١. مراجعة مدير المشروع لنسخة المتطلبات الاولية (مسودة) يجعل التقرير مرتب ومنظم لقارئه؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٢. ضرورة إرسال تقرير المتطلبات للعميل للموافقة النهائية؟

أوافق ☐ أوافق بشدة ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٣. موافقة العميل علي النسخة النهائية لتقرير المتطلبات يعكس مستوي التفاهم والاتفاق بين العميل ومدير المشروع حول متطلبات المشروع؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٤. ضرورة وضع متطلبات المشروع التقنية في تقرير منفصل؟ "حدود المشروع، تقدير التكلفة، تقدير الزمن، مستوي الجودة، وغيرها من المتطلبات"

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٥. تقديم تقرير المتطلبات التقنية في ثلاث نسخ (مسودة – مقترحة - مراجعة) يضمن تغطية كل المتطلبات التقنية وفعالية التقرير الناتج؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٦. الاهتمام بوضع آلية محددة لعملية الدفع يضمن الاستمرار بفعالية في إدارة المشروع ؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٧. دفع نسبة من تكلفة المشروع بعد كتابة تقرير المتطلبات الاساسية وتقرير متطلبات المشروع التقنية يضمن حق العميل والجهة المسئولة من تنفيذ المشروع؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

المرحلة الثانية – التطوير (التصميم – كتابة الكود - الاختبار).

١٨. تسليم العميل نموذج للتصميم المقترح يمنح العميل فكرة واضحة عن شكل النظام المستقبلي؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

١٩. ضرورة قبول العميل للتصميم النهائي والاهتمام بملاحظاته إن وجدت؟
أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢٠. تنفيذ التصميم النهائي لابد أن يكون بنائاً علي الموافقة النهائية للعميل؟
أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢١. الاهتمام (معايير متفق عليها في كتابة الشفرة – البرمجة الدفاعية) في المؤسسات الصغيرة يضمن كتابة الكود بصورة سليمة وداعمة لتحقيق معايير جودة البرمجيات؟
أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢٢. تحديد أدوات ونقاط ضعف يجب اختبارها يضمن تحقيق مستوي حماية مقبول للمنتج البرمجي؟
أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

المرحلة الثالثة – المراجعة والتسليم

٢٣. عند تسليم أول نسخة للعميل يكون مطالب بإنشاء قائمة بملاحظاته يضمن تحقيق قيمة (إرضاء العميل)؟
أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢٤. عقد اجتماع بين العميل ومدير المشروع لمناقشة ملاحظات العميل يقود لفهم ملاحظات العميل بصورة واضحة؟
أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢٥. بعد الانتهاء من تطوير المنتج البرمجي بناء علي احتياجات وتوقعات العميل يجب علي العميل التوقيع علي وثيقة إكمال المشروع بنجاح؟
أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

٢٦. يمكن أن أستخدم المنهجية المقترحة في المشاريع المستقبلية؟

أوافق بشدة ☐ أوافق ☐ محايد ☐ لا أوافق ☐ لا أوافق بشدة ☐

إضافة إذا أمكن

.....

.....

.....

.....

.....

شكراً جزيلاً علي اهتمامك بالإجابة علي الأسئلة أعلاه والمشاركة برأيكم الكريم. أملي كبير في أن تسهم كل الجهود المبذولة بصورة إيجابية في تحقيق أهداف الدراسة وتقديم حلول فعاله وبسيطة للمشاكل المطروحة.

الباحث : أيمن المعتصم بالله