SOFTWARE DEVELOPMENT STANDARDS AND SOFTWARE ENGINEERING PRACTICES: A CASE STUDY OF BANGLADESH

BY

MD. JAHIDUL ISLAM

ID: 171-25-577

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Computer Science and Engineering

Supervised By

Md. Tarek Habib
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
December, 2018

APPROVAL

This thesis titled **"Software Development Standards and Software Engineering Practices a Case Study of Bangladesh"**, submitted by **Md. Jahidul Islam** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 12th December 2018.

BOARD OF EXAMINERS

____

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr. Sheak Rashed Haider Noori **Internal Examiner**
Associate Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Md. Zahid Hasan **Internal Examiner**
Assistant Professor & Coordinator of MIS
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr. Mohammad Shorif Uddin **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

I hereby declared that, this thesis has been done by me under the supervision of Mr. **Md. Tarek Habib, Assistant Professor, Department of CSE,** Daffodil International University. I also declare that neither this thesis nor any part of this thesis has been submitted elsewhere for award of any degree or diploma.

Supervised by:

____

Md. Tarek Habib
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

____

Md. Jahidul Islam
ID: 171-25-577
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

At first, I express my heartiest thanks and gratefulness to Almighty Allah for His divine blessing which allowed me possible to complete this thesis successfully.

I fell grateful to and wish my profound my indebtedness to of Mr. **Md. Tarek Habib, Assistant Professor, Department of CSE**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of wireless network influenced me to carry out this thesis. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this thesis.

Furthermore, I would like to express my heartiest gratitude to Professor Dr. Syed Akhter Hossain, Head of CSE Department, for his kind help to finish my thesis and also to other faculty member and the staff of CSE department of Daffodil International University.

I would like to thank our entire course mate at Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

There is an obvious need to research and analyze data that is used software industry in Bangladesh. In Bangladesh, there are many IT Companies and increasing day by day. There is less evidence of their usage among software practitioners in Bangladesh. While Bangladeshi software firms have not experienced much in this particular area in comparison to other countries. The main objectives of our study are: i) To understand the software development process uses by the software developer firms in Bangladesh ii) To identify the development practices based on established quality standard and iii) To establish a standardized and compatible process for the development of software for a specific project. It is revealed from this research that software industries of Bangladesh are lacking in the target set for software process and improvement, the involvement of quality control activities, and standardize business expertise practice. This paper investigates the Bangladeshi software industry in the light of the above challenges. The purpose of this study is to investigate which method are used perspectives in Bangladeshi IT companies. The ISO, IP and CMM standard has become a basic part of software development.

Table of Contents

CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLE

CHAPTER 1

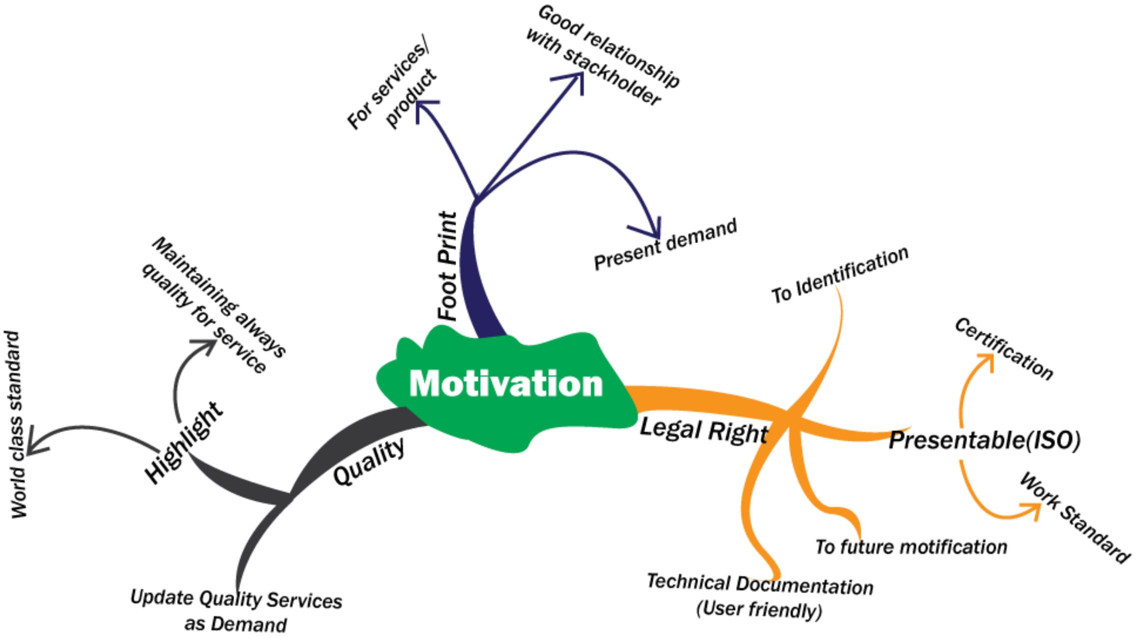
INTRODUCTION

1.1 The Context

Bangladesh has built itself on traditions of Science and Technology [1]. Today, the number of IT companies stands more than eleven hundred according to BASIS (basis.org.bd). Besides, the government set up some software technology parks in different locations. For that, a huge amount of IT Firms stands in the future. So, it is a perfect time to know appliance current software method which is followed internally our Software Industries. However, the survey system is closely related to some other aspects. The first and foremost one is surveying data. To provide legitimate solicitations, getting surveying data and analyzing it is a rudiment. This thesis keeps an effort to get a practical idea about usage of the method during developed a project.

1.2 Motivation

Our findings indicate that there is no clear understanding of the Software Engineers' job, what motivates Software Engineers, how they are motivated, or the outcome and benefits of motivating Software Engineers.



1.3 Objectives

This is the golden time to highlight our country as a fertile land for software industries or Software engineering. To do this we need to maintain ISO, CMM, IP and BASIS requirement for the Software Development Standards and Software Engineering Practices.

When most of the software firm are faced with the challenges of delivering product or services which can satisfy their customers. In the effort to deliver effective services or product, the software development sector having the problem to representable due to maintaining the world standard of the development.

1.4 The Problem Statement

In Bangladesh, there are many IT Companies and increasing day by day. There is less evidence of their usage among software practitioners in Bangladesh. That's for, to know regarding appliance software method in Bangladeshi IT companies. There are some key problems has been identified.

Firstly, the first and foremost one is data. So effective ways of communicating with the users and getting an authentic response from them to get a thorough idea about their using method are the main focus.

Secondly, after understanding respondent data, the necessary criterion is done. Based on the criterion, the possibility of maintaining ISO, IP, CMM and BASIC membership when developing any software. If it is possible, then necessary data which users are willing to share is collected.

Finally, Different information filtering algorithms which can be applied to user-given data to provide personalized recommendation are studied. To come up with the best techniques of data mining and relevant researches are the focuses of this phase.

1.5 Structure of the Thesis:

The following description is provided to understand which chapter covers which topics and their relevant discussions.

- ✓ In the chapter "Introduction", the basic purpose of this thesis is already described.
- ✓ In the chapter "State of The Art", relevant researches and works which have been studied will be briefly described.

- ✓ In the chapter "Background", a brief discussion of methods.
- ✓ In the chapter "Research Methodology", a brief discussion will be made on how the thesis has been conducted.
- ✓ In the chapter "The Survey", various aspects of surveys and how the survey has been set up to understand the scenario will be elaborately discussed.
- ✓ In the chapter "Analysis", a brief summary of how the thesis can be used and continued to solve the remaining problems in the statement are discussed.
- ✓ In the chapter "Conclusion", shows the goals of this thesis.

CHAPTER 2

STATE OF THE ART

To a lot of people in the business, it will come as a great surprise most of the people are not aware of the fact that a state of the art software development process actually exists. But it is a great feature that not only exit but also important when anyone trying to start career in software development. Start of the Art is very basic step that can help to get feature idea of life (whatever any project or Services).

2. Software Development Standard

To be a software developer or software development firm, should be maintain the life cycle data required to fulfill the Software Development Standard and guidelines. Software development standard help a software firm to get the unique platform as the comparative world. The life cycle data required to fulfill the Software Development Standard consists of the following:

- Getting Software Aspects of Certification
- Maintain Software Development Standards
- Maintain Software Requirements Document
- Maintain Software Design Description
- Maintain Software Source Code
- Maintain Executable Object Code
- Maintain Software Verification Plan
- Maintain Software Verification Results
- Maintain Software Quality Assurance Plan
- Maintain Software Quality Assurance Records
- Maintain Problem and Action Reports
- Maintain Software Configuration Management Plan
- Maintain Software Configuration Management Records
- Maintain Software Configuration Index
- Maintain Software Accomplishment Summary

This very important to develop a software or taking a project of software development, we should maintain those stage [4].

2.1 Software Requirements Standards

According to Software Development Standard and Software Engineering Practices, the software requirements process uses the system requirements and system architecture to develop high to level requirements for the desired software. The objectives of this process are to ensure the clarity, consistency, and completeness of those requirements allocated to the software. Hence, there also is no system safety assessment which is an important aspect of any development process that needs to comply with the Software Development Standard and Software Engineering Practices. The Software development project started with the definition of software requirements for a specific component of a guidance and control system. Without system requirements, certain assumptions must be made in the development of the software requirements. Lack of system requirements also impacts the extent to which the project will comply with the Software development since no traces can be made from the software requirements back to the system requirements and safety assessment.

The following section describes the development of the original specification for the software, including the methods, rules, and tools used in the development of the high-level requirements.

2.1.1 Requirements engineering (RE):

Requirements engineering (RE) is the process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed. The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process. Requirements may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification [5].

Mainly it classifies in two type:

User requirements:

User Requirement is the high-level summary requirements written as statements, in a natural language plus diagrams, of the services or the system is expected to provide to system users.

System requirements:

System requirements is the detailed description of what the system should do including the software system's functions, services, and operational constraints. Sometimes it's called a functional specification, it should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.

Three classes of requirements:

- Functional requirements
- Non-functional requirements
- Domain requirements

2.1.2 Functional requirements

Functional requirements describe functionality or system services. They depend on the type of software, expected users and the type of system where the software is used. Functional user requirements may be high-level statements of what the system should do. Functional system requirements should describe the system services in detail. In principle, requirements should include descriptions of all facilities required, and should be no conflicts or contradictions in the descriptions of the system facilities.

2.1.3 Non-functional requirements

The Non-functional requirements is the good practices for Requirement Engineering for software engineer or software developer. It's define system properties and constraints like as reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc. Process requirements may also be specified mandating a particular IDE, programming language or development method. Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless. Figure 2.1 is showing details about Non-Functional Requirements.

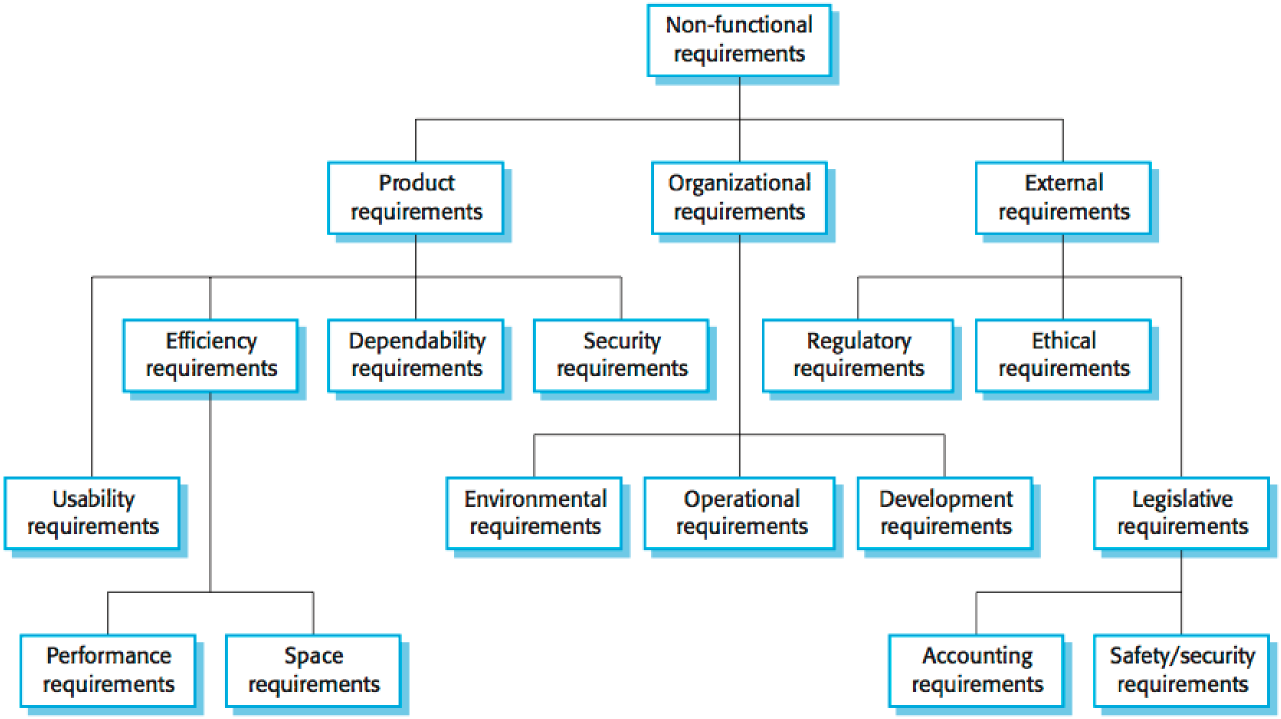


Figure 2.1: Non-functional requirements

Three classes of non-functional requirements:

- Product requirements
- Organizational requirements
- External requirements

2.1.4 Domain requirements

The system's operational domain imposes requirements on the system. The domain requirements are the new functional requirements, constraints on existing requirements or define specific computations. When the domain requirements are not satisfied, the system should be unworkable. To maintain the domain requirement, we should maintain two main problems with domain requirements:

Understandability:

> The requirements are expressed in the language of the application domain, which is not always understood by software engineers developing the system.

Implicitness:

Domain specialists understand the area so well that they do not think of making the domain requirements explicit.

2.1.5 Requirements engineering process

Processes vary widely depending on the application domain, the people involved and the organization developing the requirements. In practice, requirements engineering is an iterative process, in which the following generic activities are interleaved:

- Requirements elicitation;
- Requirements analysis;
- Requirements validation;
- Requirements management.

Requirements elicitation and analysis:

Software engineers work with a range of system stakeholders to find out about the application domain, the services that the system should provide, the required system performance, hardware constraints, other systems, etc.

Requirements specification:

Requirements are documented and input into the next round of the spiral. Closed (based on a pre-determined list of questions) and open interviews with stakeholders are a part of the RE process. User stories and scenarios are real-life examples of how a system can be used, which are usually easy for stakeholders to understand. Scenarios should include descriptions of the starting situation, the normal flow of events, what can go wrong, other running activities, the state of the system when the scenario judgment.

Use-cases is the scenario-based technique in the UML which identify the actors in an interaction and which describe the interaction itself. The set of use cases should describe all possible interactions with the system.

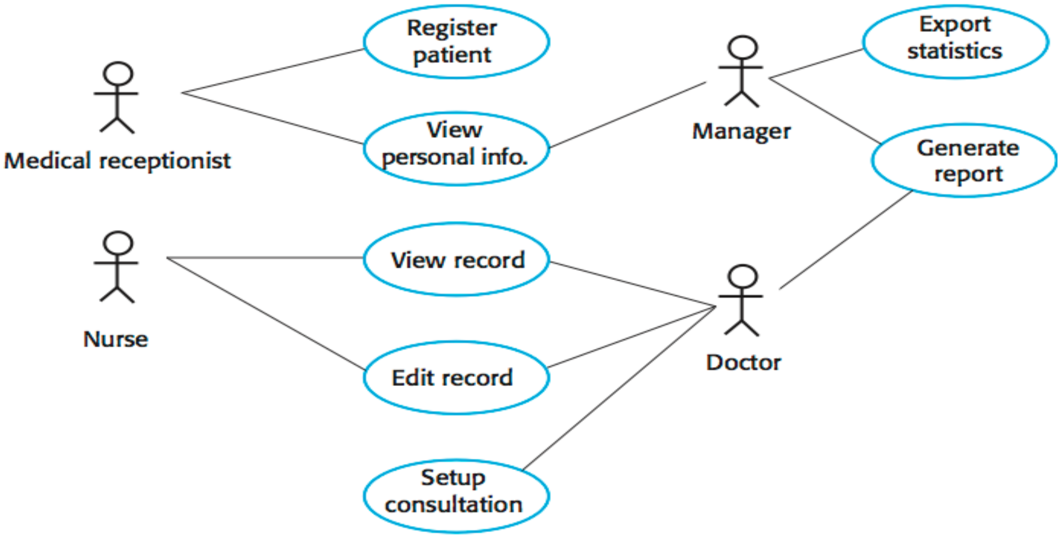


Figure 2.2: Use-Case of Requirements specification

Requirements specification is the process of writing down the user and system requirements in a requirements document. System requirements are more detailed requirements and may include more technical information. The requirements may be part of a contract for the system development and it is important that these areas complete as possible.

In principle, requirements should state what the system should do and the design should describe how it does this. In practice, requirements and design are inseparable.

User requirements are almost always written in natural language supplemented by appropriate diagrams and tables in the requirements document. The system requirements should be written in natural language (expressive, intuitive and universal) but other notations based on forms, graphical system models, or mathematical system models can also be used. Because it should be understood by users and customers.

Structured natural language is a way of writing system requirements where the freedom of the requirements writer is limited and all requirements are written in a standard way. This approach maintains most of the expressiveness and understand-ability of natural language but ensures that some uniformity is imposed on the specification.

2.1.6 Requirements validation

Requirements validation is concerned with demonstrating that the requirements define the system that the customer really wants. Requirements error costs are high so validation is very important.

What problems to look for:

- Validity: does the system provide the functions which best support the customer's needs?
- Consistency: are there any requirements conflicts?
- Completeness: are all functions required by the customer included?
- Realism: can the requirements be implemented given available budget and technology?
- Verifiability: can the requirements be checked?

Requirements validation techniques:

Requirements reviews:

Systematic manual analysis of the requirements. Regular reviews should be held while the requirements definition is being formulated. What to look for:

- Verifiability: is the requirement realistically testable?
- Comprehensibility: is the requirement properly understood?
- Traceability: is the source of the requirement clearly stated?
- Adaptability: Should the requirement be changed without a large impact on other requirements?

2.1.7 Requirements change

Requirements management is the process of managing changing requirements during the requirements engineering process and system development. New requirements emerge as a system is being developed and after it has gone into use. The reasons why requirements change after the system's deployment:

- The business and technical environment of the system always changes after installation.
- The people who pay for a system and the users of that system are rarely the same people.
- Large systems usually have a diverse user community, with many users having different requirements and priorities that may be conflicting or contradictory.

2.2 Derived Requirements and Modifications:

According to Software Development Standard and Software Engineering Practices, the Software Development Standard specification is classified under control category which means that the project must provide a formal system of problem reporting, change control, and change review for that data. All changes to the Software Development Standard specification, along with the other project support documentation, are made through a system of Support Documentation Change Reports. All questions raised by any member of the development team regarding the Software Development Standard specification are brought to the system analyst. The system analyst reviews all questions and determines if changes to the specification are required. When changes are deemed necessary, the system analyst submits a description of the necessary modification to the SQA representative and project leader for review. The chapter "Problem and Change Reporting" gives a more detailed description of the procedures and forms used for tracking, reviewing and approving changes to the Software Development Standard specification.

Once the modification is approved, a copy of the modification description is distributed to all project participants. The programmers are required to consider the impact of each modification to the software requirements on their implementation and make any appropriate changes to their software design and code. Similarly, the verification analysts should determine the impact of any modifications on the verification activities, especially test cases and requirements in the traceability data, and make any necessary corrections to the appropriate artifacts.

The following chapter describes the software design standards defined for the Software Development Standard project.

2.3 Software Design Standards

The purpose of the software design process is to refine the software high-level requirements into a software architecture and the low-level requirements that can be used to implement the source code. The software design standards are provided to define the methods, rules, and tools to be used in the development of the software architecture and low-level requirements, as described in Subsection Software Development Standard and Software Engineering Practices. These standards should enable the software implementations to be uniformly designed.

During the transitional design process of the Software Development Standard and Software Engineering Practices, the programmers are required to develop detailed software designs from existing Software Development Standard designs, as delivered from ISO and CMM. A detailed design should be a complete statement of the software low-level requirements that addresses exactly what needs to be accomplished in order to fulfill the objectives stated in the ISO and CMM specification; that is, the detailed design should contain an algorithmic solution. The low-level requirements should be directly translatable into source code, with no further decomposition required.

2.4 Design Methods, Rules, and Tools

For the Software Development, the design of an ISO implementation should be developed using the structured analysis and design.

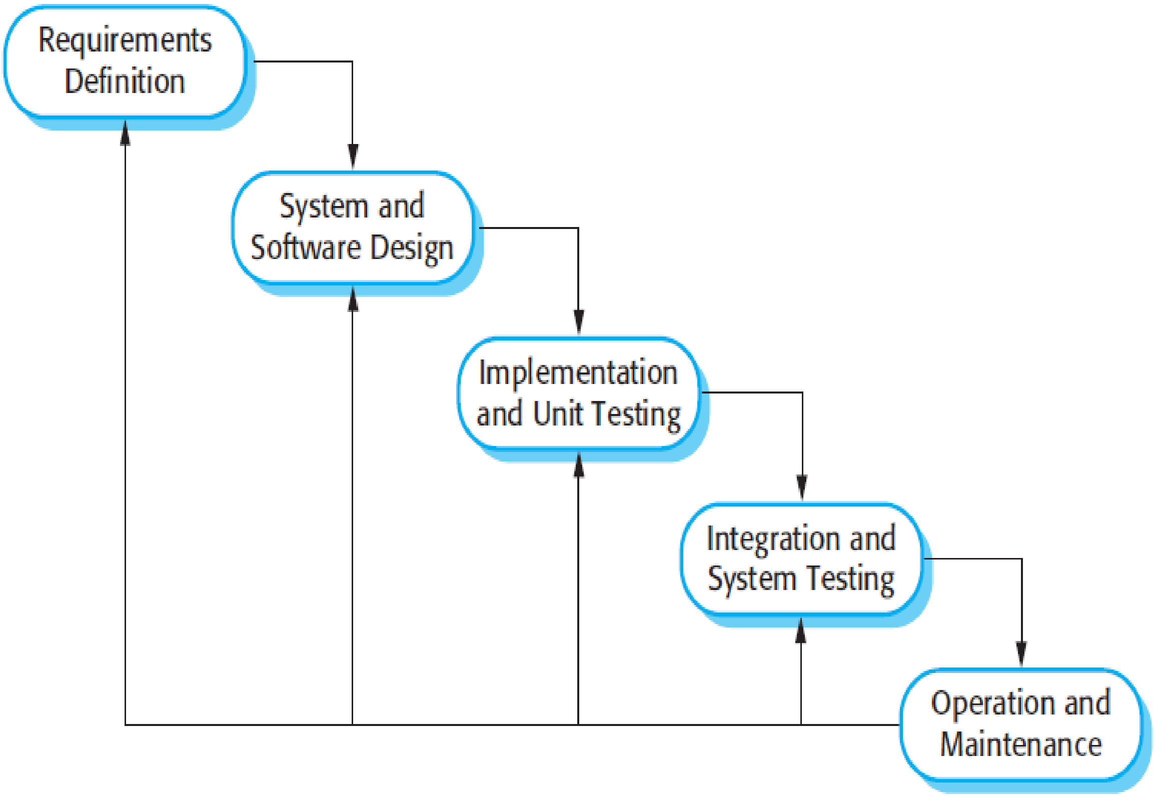

Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure 2.3: Basic ISO implementation

2.5 Design and Development Documentation

As discussed in Subsection ISO, the design description defines the software architecture and the low-level requirements that satisfied the software high-level requirements. The design document configuration shown below describes the required contents of the detailed design description for each Software design and development implementation. This documentation includes introductory and overviews commentary on the design generated with the teamwork tool. The document produced from the configuration will be analyzed during the design review and will also be used to trace changes in the design to the code. The software code was developing and modify, the design and the code will be modified to be kept consistent. Thus, it is important to have a carefully documented description of the software design [6].

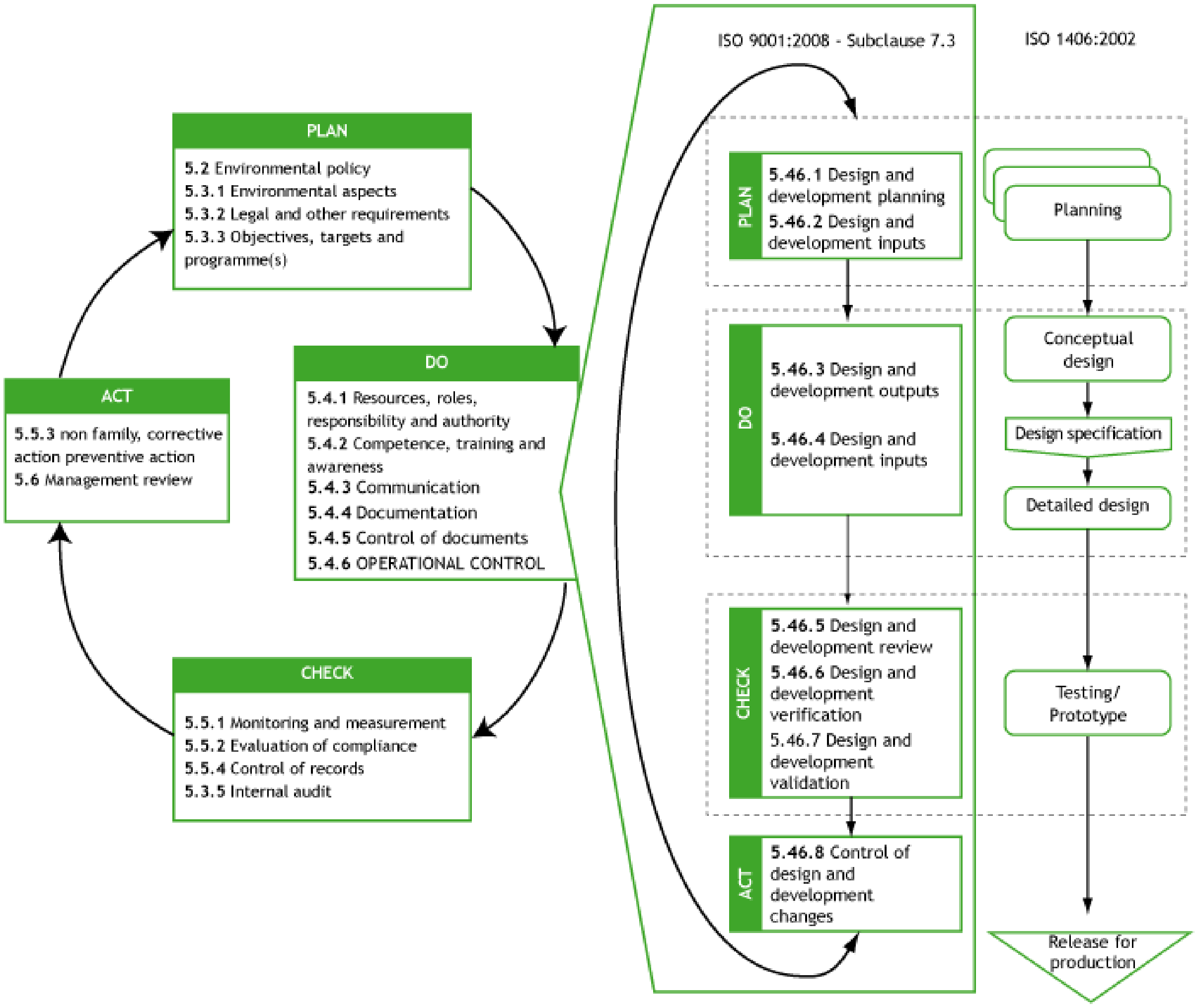

ISO 9001:2008 - Subclause 7.3
ISO 1406:2002
PLAN
5.2 Environmental policy
5.3.1 Environmental aspects
5.3.2 Legal and other requirements
5.3.3 Objectives, targets and programme(s)
DO
5.4.1 Resources, roles, responsibility and authority
5.4.2 Competence, training and awareness
5.4.3 Communication
5.4.4 Documentation
5.4.5 Control of documents
5.4.6 OPERATIONAL CONTROL
ACT
5.5.3 non family, corrective action preventive action
5.6 Management review
CHECK
5.5.1 Monitoring and measurement
5.5.2 Evaluation of compliance
5.5.4 Control of records
5.3.5 Internal audit
PLAN
5.46.1 Design and development planning
5.46.2 Design and development inputs
DO
5.46.3 Design and development outputs
5.46.4 Design and development inputs
CHECK
5.46.5 Design and development review
5.46.6 Design and development verification
5.46.7 Design and development validation
ACT
5.46.8 Control of design and development changes
Planning
Conceptual design
Design specification
Detailed design
Testing/ Prototype
Release for production

Figure 2.4: Design and Development Documentation [6]

2.5.1 Planning:

- ➢ Determine all necessary design and development phases
- ➢ Determine the appropriate design review, verification, and validation techniques
- ➢ Determine who has responsibility for design and development
- ➢ Determine the necessary authorities for design and development
- ➢ Maintain records

2.5.2 Input:

- ➢ Determine functional and performance related characteristics
- ➢ Determine relevant statutory and regulatory requirements
- ➢ Determine relevant information from previous designs
- ➢ Determine other requirements that are essential to the product

- Maintain records

2.5.3 Output:

The outputs must reconcile with the inputs:

- Determine whether the outputs meet the input requirements for design and development
- Determine whether the outputs provide suitable information for purchasing
- Ensure the outputs provide a reference to product acceptance criteria
- Determine whether the outputs accurately specify essential characteristics
- Maintain records

The final drawing or specification should meet the input requirements. The outputs would also normally be used as the basis for purchasing and process control.

2.5.4 Review:

At key stages of development, the design should be reviewed to ensure that the inputs are met, and any technical problems identified and resolved.

- Determine whether the results conform to the specified requirements
- Identify any gaps in the design
- Propose and implement solutions to close gaps in the design
- Maintain records

2.5.5 Verification:

- Determine whether the outputs meet the input requirements for the design
- Maintain records

2.5.6 Validation:

- Ensure the product meets the specified requirements
- Undertake prototype testing
- Maintain records

2.5.7 Changes:

Consider the impact of proposed changes, especially in terms of backward compatibility and ensure control over design and development changes. Design changes must be:

- Identified
- Recorded
- Reviewed

- Verified
- Validated
- Approved

Any design changes must be fed back through the entire design and development process.

2.6 Important of Software Standards:

Now a day the modeling methods is the very formal affairs. Definitions were left to intuition, and there were many gaps. For adaption software standard people will be more formal to methods background criticized these methods for software projects [2].

By adapting the software standard many of the software organizations today are endeavoring to improve their software development processes to improve product quality, project team productivity and reduce development cycle times, thereby increasing their competitiveness and profitability. Software Engineering Institute (SEI) sparked awareness regarding software process improvement, with the release of its original software process maturity model. Following the advice of the SEI, many software organizations initiated software process improvement efforts to improve the quality of their products by improving the processes that produce those products [2].

According to Kevin Hyde and David Wilson, CMM-based software process improvement may deliver both tangible and intangible benefits to an organization. The survey-based research was conducted in the development organization during the early 2000s to find out from software professionals if they believed the intangible benefits were being realized. The survey results showed that the organization had experienced improvements in the quality of work life, organization communications, organization learning and efficiencies, the ability to attract, retain and develop software professionals and the coherency of its organization culture [3].

All these results indicate the prominent need of a standard for development and measurement of the software for its size, complexity, costing, and quality attributes. A formally defined standard is also required to control the development process so that if a group violates the standard, a release could be 'prepared' by another group.

CHAPTER 3

BACKGROUND

3.1 The Software Development Methodologies

As a modern technology and computer technology offers proficient and high-performance information processing, it has got popularity over the home and office users in the whole world. By the decade of 1990, in Bangladesh, it has also taken an important role. Since during this time PCs become more user-friendly and attractive, the number of users had been increased. Besides the general users, in Bangladesh, the number of software developers have been increased as well. Many of Computer Science and Engineering graduates from public and private universities, as well as computer diplomas from training institutions, are getting employed to the local software companies. As time goes, the overall development of the skill of software developers has been increased with respect to Bangladesh.

Bangladesh stands out distinctly as a potential software-exporting nation, considering the analytical and technological ability of its people. Bangladesh is one of the potential countries where software development is to be grown as a software industry. According to Bangladesh Association of Software and Information Service (BASIS), there are around 300 plus companies, are working closely with the development of software for the local and international market for different information and communication technology services [1]. Bangladesh is a country, where the only surplus property is the human resource. Take into account the earning of foreign exchanges and removing of the unemployment problem, the is software industry a very prospective field. To make this field more advantageous, several plans have been done by the government and private organizations in the last several years. The Government of Bangladesh made an in-depth study on how the software sector of the country could be designed to suit the needs of the global market. To follow up on the conclusion of the study and to monitor the issues associated with the sector's growth and development, a high powered National Standing Committee (NEC) on software export has been formed. This standing committee has brought together the concerned government offices organizations and leaders of the software trade to work in unison to study the problems and prospects of the sector 2. Table 1 shows the business application nature of software service of the software industry at Bangladesh**.** It is notable that each software company in the software industry develops multiple categories of software service.

Table 3.1. Products/Service Range of Local Software Industry

Products/Services Category	**% of Companies Offering Services**
Android Base Application	82%
Accounting & Financial Management	69%
Inventory Management	59%
Human Resource Software	58%
Web Site/Web Application Development	57%
ERP (Enterprise Resource Planning)	48%
Software Implementation & Integration	46%
Billing	43%
Asset Management	38%
POS (Point of Sales)	37%
E-Commerce	36%
Data Entry/Data Conversion	34%
CRM (Customer Relationship Management)	32%
E-Governance Application	29%
SCM (Supply Chain Management)	27%
Data Warehousing	23%
Access Control	22%
Mobile/Wireless Application Development	18%
E-Learning	17%
Data Security	14%
Gaming Software	12%
AI Health Consultant (E- Consultant)	10%
E-Travel and Transport System	8%

3.2 Methodology of the study

The methodology of the study discusses the way it has been approached, how data have been collected and then analyzed. This is an exploratory study trying to find out the current software development standard of Bangladesh. Secondary studies are conducted to acquire the preliminary knowledge to explain the primary data. Data have been collected through a questionnaire survey and interviews. A structured questionnaire was developed to collect relative information. In conformity with BASIS survey, there are over 800 registered software and ITES (IT Enabled Service) companies in Bangladesh. There are 420 BASIS listed software firms who are directly or indirectly developing software for the local market as well as international market. To find out the existing process in place in different companies, we selected only those companies who have been doing only software development i.e. software development and sell are their main business. Convenience sampling method has been employed to select 50 software companies from Dhaka City. Chief Executive Officer (CEO) of each selected company has been asked to fill out the questionnaires. The questionnaire, which is mentioned before to be used for data collection of this study, will be distributed among the CEO through the Post Graduate Diploma in Information Technology (PGDIT), students of session 2008-2009, Institute of Information Technology, University of Dhaka, who were selected and trained them up for conducting the interview through the structured questionnaires. All the relevant information was collected from May 2008 to July 2008. At first, descriptive statistics (frequency & percentage) were presented to show the overall condition of the sample. For the above analyses, Statistical Package for Social Science (SPSS), Version 10.0.1 and MS Excel was used. Therefore, all results of this paper are developed from the primary source if not otherwise mentioned.

3.3 Software Development Process:

A software process is a framework for the tasks that are required to build high-quality software 3. Therefore, software process defines the approach that is taken as software is engineered. It may be an ad-hoc process devised by the team for one project. But the team often refers to a standardized documented methodology which has been used before on similar projects or one which is used habitually within an organization. Some managers who are held accountable for software development may seek to find the commonalities in the efforts of their organizations. If those managers are process oriented, then they may seek methodologies or other proxies which can serve as templates for the software development process. Another reason why software development process is important is that a process provides organizational stability and more control to its activity.

3.4 Software Development Life Cycle (SDLC)

The process of software development is often modeled as a series of stages that define the software life cycle. Software Development Life Cycle (SDLC) is an approach to develop a software product that is characterized by a linear sequence of steps that progress from start to finish. The SDLC model is one of the oldest systems development models and is still probably the most commonly used. The six general steps are: 1) evaluate existing system/software; 2) define new system requirements; 3) design system; 4) develop new system; 5) implementation; and, 6) maintain. SDLC is explained in Figure 1.

Software development life cycle models include prototyping, evolutionary prototyping, incremental development, spiral model, and V model. The use of these models is, for the most part, confined to the overall management of the project. However, projects are now considered better controlled if the model best suited to them.

Some experienced and highly respected project leaders and programmers consider the rigid application of lifecycle plans to be a theory that does not work well in practice. Linux Torvalds, the very highly regarded project leader of the Linux Kernel, made the following statement on the Linux kernel mailing list: “No major software project that has been successful in a general marketplace (as opposed to niches) has ever gone through those nice lifecycles they tell you about in Computer Science classes” [7].

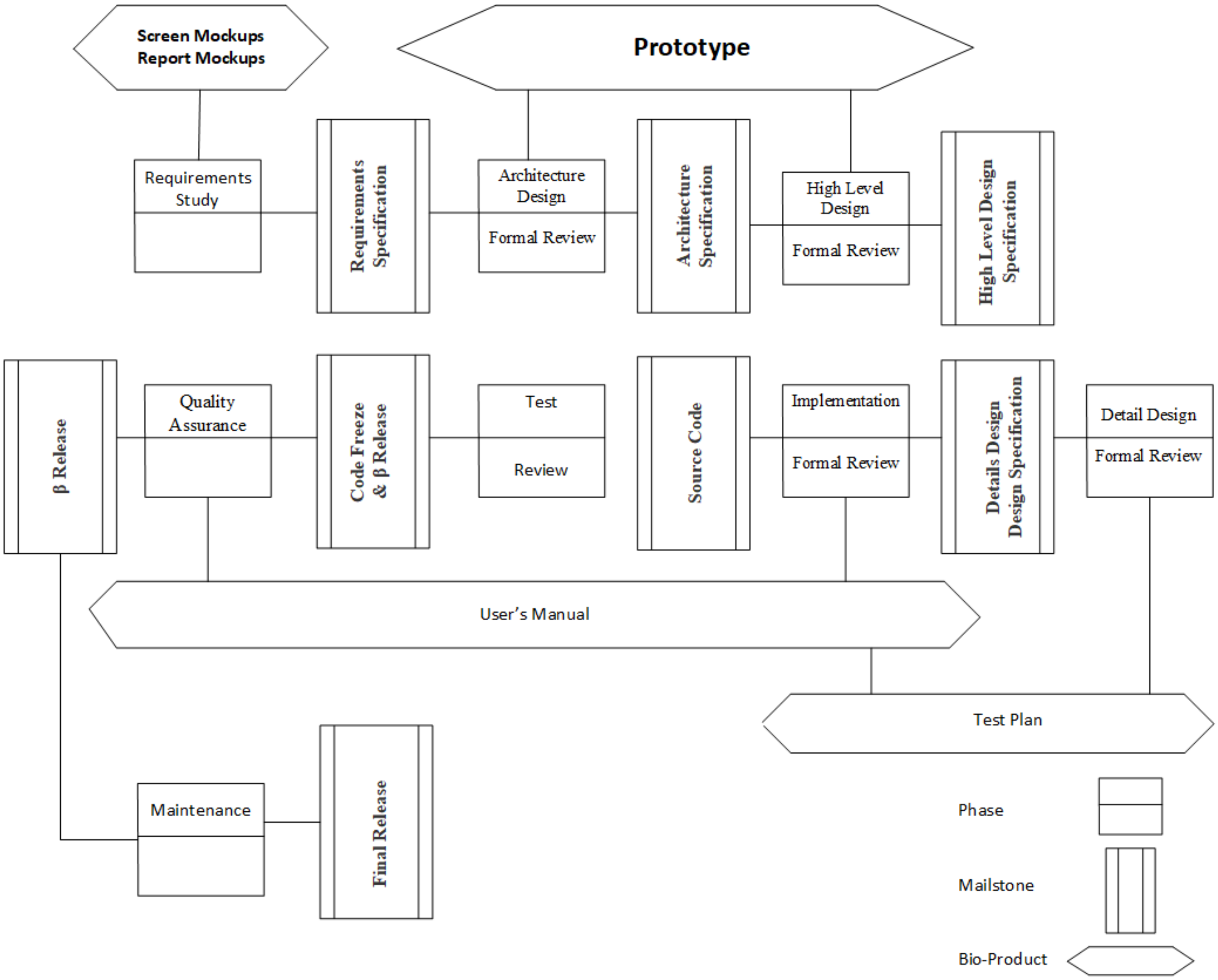


Figure 3.1. Software Development Life Cycle

3.5 ISO

ISO Certification is the declaration of a company that he maintains quality in his management system. It is the attestation of any third party Certification body that the company is following the international guideline for managing the specific field.

ISO Certification is requirement for submitting tender, extend international market, it increases company's revenue and increases customer satisfaction. It helps to continual development of any organization. It increases trust among the buyers that the company will keep its commitment to supply products or services on time [8].

International Organization for Standardization (ISO) is an international standard setting body. The organization promulgates worldwide proprietary, industrial and commercial standards. ISO was established on February 23, 1947. The headquarter of ISO is in Geneva, Switzerland [9].

ISO is the world's leading developer and publisher of standards. In addition to quality standards, ISO has issued two sets of standards. These are ISO 9000 to govern manufacturing process and their quality management and ISO 14000 to govern environmental protection.

3.5.1 Overview

The ISO 9000 standard concerning us is ISO 9001, since it applies to "quality assurance in design, development, production, installation and servicing". As we have said, this standard is written for manufacturing industry, and this poses some problems when applying it to development and maintenance of software.

Figure 3.2 illustrates manufacturing and software development from this perspective. The rectangles symbolize cost or effort.

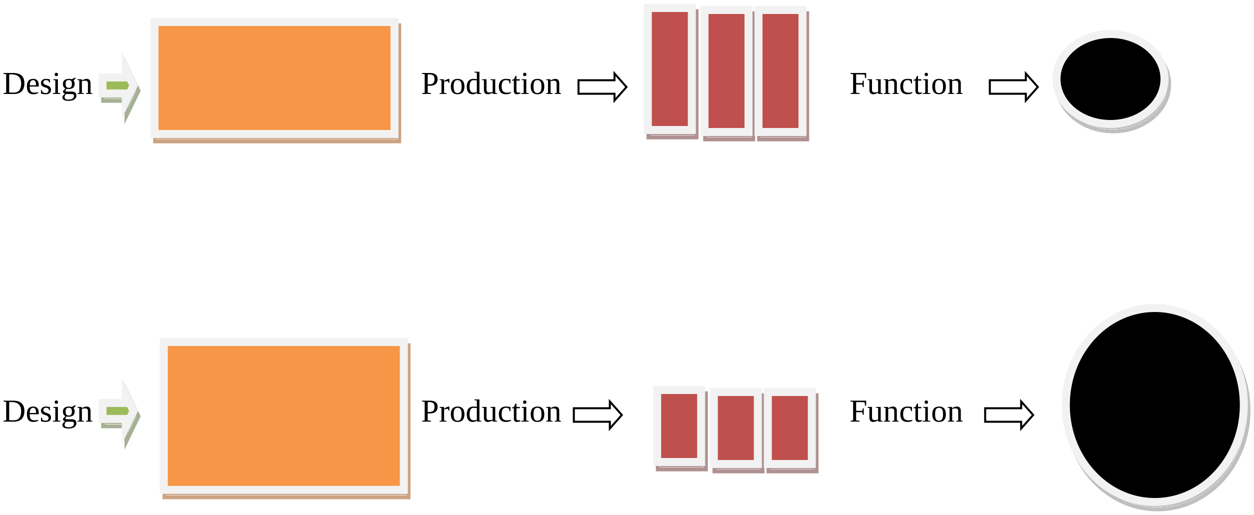


Figure 3.2: ISO illustrates manufacturing and software development [9]

If we first look at manufacturing, e.g. of kettles, we see that design is a relatively small activity. Instead, the cost for each manufactured item is notable, so that when a few items have been produced, production is by far the major part of the activity. Therefore, when we talk about quality or productivity problems and improvements in manufacturing, we tend to focus on production.

Software development, however, is nearly 100% design. Production means to copy executable code to diskettes, tapes or ROM: s, and is performed and checked automatically. So, when talking about quality and productivity we focus on design.

3.5.2 Background

ISO 9000 is a set of international standards on quality and quality assurance developed to help companies effectively document the quality control system elements to be implemented to maintain an efficient quality control system. They are not specific to any industry and can be applied to organization of any size.

ISO 9001 is one of the standards within the range of ISO 9000 standards. ISO 9001 is a set of internationally agreed standards that provide guidelines for quality management systems (QMS). ISO 9001 is a document which outlines the criteria or requirements of QMS. ISO 9001 sets out the preconditions that an organization must maintain in its quality control system. ISO 9001 was first published in 1987 by ISO. Its QMS gradually evolved up to present time.

The following are the ISO 9001 standards: -

ISO 9001:1987, ISO 9001:1994, ISO 9001:2000, ISO 9001:2008 and ISO 9001:2015

Functions of ISO 9001:

ISO 9001:2015 is the current and latest version of ISO 9001. The main functions of ISO 9001 are to set out the criteria for QMS and also to provide the basic standards for certification of any organization. It can be used by all kinds of organizations, whether

Large or small, regardless of them activities. In fact, there are over one million companies and organizations in over 170 countries certified.

The main objective of ISO 9001 is to provide an organization the quality control system that will provide the foundation to better customer satisfaction, staff motivation and continual improvement.

ISO 9001 certification provide maximum benefit to an organization if the organization implements the QMS in a practical way. By adopting QMS of ISO 9001, an organization can start to implement more efficient working practices and can focus more on its business objectives.

Certification:

ISO does not certify organizations itself. Numerous certification bodies exist, which audit organizations and upon success, issue ISO compliance certificates. ISO 9001 certified means an organization has met the requirements of ISO 9001.

The Following Procedure Must Be Followed to Obtain ISO 9001 Certification

- Commit to get ISO 9001 Certification
- An Appointment of internal Project Manager
- Allocate Resources
- Establishment of Baseline Status
- Development of the System
- Auditing the System
- Selecting an external Auditor
- To have the External Audit
- Get the Certificate and celebrate
- Maintaining the Certificate

ISO 9000 and Bangladesh

In Bangladesh, ISO is represented by BSTI and functions through BAS (Bureau of Assessment Services). The headquarter of BAS is in London, UK. BAS has its office in Bangladesh. BAS in Bangladesh is working to issue ISO 9001 certificates for various industries. BAS in Bangladesh is an authorized agent of BAS of London, UK [9].

What is ISO Certification Process?

ISO Certification process is much easier than before. Documentation many requirements have been demolished. So, it does not require to wait for long time to get certification. You can get certificate by using less documents than previous period. ISO 9001:2015 has less documentation then ISO 9001:2008 Standard. However, see the steps from below:

- ✓ Select certification body & know what are necessary to get certification
- ✓ Prepare documents
- ✓ Implement those documents
- ✓ Participate in audit
- ✓ Get certificate

ISO 9001-Quality Management:

Among all the modern Management Systems ISO 9001- Quality Management System stands foremost in the world. This system enables the companies to maintain excellent procedures in order to meet customer needs at the global level in terms of Quality and Standards [10].

Customers are increasingly becoming quality-conscious shoppers. They want to know up front that your business will meet their needs. A certified Quality Management System demonstrates your commitment to quality and customer satisfaction. Implementing a Quality Management System will help you enhance customer satisfaction, achieve consistency, and improve internal processes. It can minimize the risk that customer expectations are not met.

Eight Quality Management Principles

- ✓ Customer focus
- ✓ Leadership
- ✓ Involvement of people
- ✓ Process approach
- ✓ System approach
- ✓ Continual Improvement
- ✓ Fact based decision making
- ✓ Mutually beneficial supplier relationship

ISO 9001– It's Key Benefits

- ✓ Boosts your image
- ✓ Increases sales
  - ✓ Improves your earnings
  - ✓ Lessens operational cost
  - ✓ Increases customer satisfaction
  - ✓ Enhances better management skills
  - ✓ Maximizes efficiency

3.5.3 Process

An overview of the Software Process: ISO is provided in Figure-3.1:

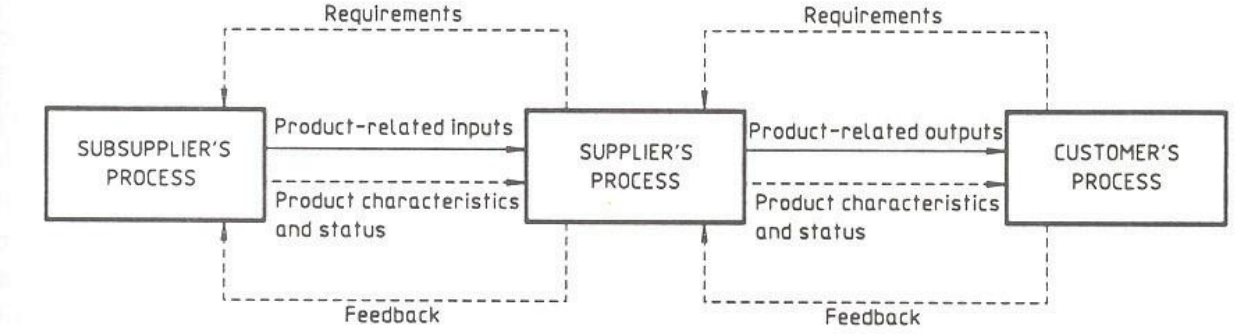


Figure 3.3: Software Process: ISO

3.5.4 Scope of ISO 9000

ISO 9000 can help a company satisfy its customers meet regulatory requirements and achieve continual improvement. However, it should be considered to be a first step, the base level of a quality control system and not a complete guarantee of quality control. ISO 9000 is most widely known and has perhaps had the most impact among 13,000 standards published by ISO. It serves many different industries and organizations as a guide to quality of products, services and managements. An organization can be ISO 9000 certified if it successfully follows the ISO 9000 standards for its industry.

ISO 9000 deals with the fundamentals of quality management systems, including the eight management principles, upon which the family of standards is based. ISO 9000 was first published in 1987. It was based on the BS 5750 series of standards from BSI that was proposed to ISO in 1979.

Eight Management Principles of ISO 9000 Family of Standards:

1) Customer Focus Organization:

Organizations depend on their customers and therefore should understand current and future customers' needs, should meet customers' requirements and strive to exceed customers' expectations.

2) Leadership:

Leadership is very important in commercial transactions. Without clear and strong leadership, a business flops. Proper leadership is essential for the direction of the organization.

3) Involvement of People:

People's involvement is very crucial. An organization is nothing without its staff whether part time, full time, in house or outsourced. It is their abilities that help maximize business success.

4) Process Approach:

The process approach is all about efficiency and effectiveness. It is also about consistency and understanding that good processes also speed up activities. Great processes reduce cost, improve consistency, eliminate waste and promote continuous improvement.

5) Systematic Approach to Management:

To increase organizations' effectiveness and efficiency in achieving its objectives, it should identify, understand and manage interrelated processes as a system.

The organization should establish quality policy and quality objectives to provide a focus and direction to the organization.

6) Continuous Improvement:

Organization should maintain current levels of performance. It ought to respond to the changing conditions and circumstances. An organization must identify, create and exploit new opportunities when they establish and sustain ongoing focus on improvement.

7) Factual Approach to Decision Making:

Efficient decisions are based on the analysis of data and information. Appropriate decisions, based on experience and entrepreneurial intuition can only be reached when these data and information are reviewed and verified continuously.

8) Mutually Beneficial Organization and Supplier Relationship:

Everything is interdependent. Organizations particularly depend on good business relationship with their suppliers. This is the only way for parties to make the maximum contribution to the creation of mutual value and mutual trust. Indispensable to that end is transparent communication, agreement on common goals while taking account of the customers’ interests and customers’ cooperation in the development and improvement of products.

3.6 Capability Maturity Model (CMM)

Capability Maturity Model (CMM) broadly refers to a process improvement approach that is based on a process model. CMM also refers specifically to the first such model, developed by the Software Engineering Institute (SEI) in the mid-1980s, as well as the family of process models that followed. A process model is a structured collection of practices that describe the characteristics of effective processes; the practices included are those proven by experience to be effective [11].

3.6.1 Overview

The Capability Maturity Model was initially funded by military research. The United States Air Force funded a study at the Carnegie-Mellon Software Engineering Institute to create a model (abstract) for the military to use as an objective evaluation of software subcontractors [11]. The result was the Capability Maturity Model, published as Managing the Software Process in 1989. The CMM is no longer supported by the SEI and has been superseded by the more comprehensive Capability Maturity Model Integration (CMMI).

What is Capability Maturity Model (CMM) Levels?

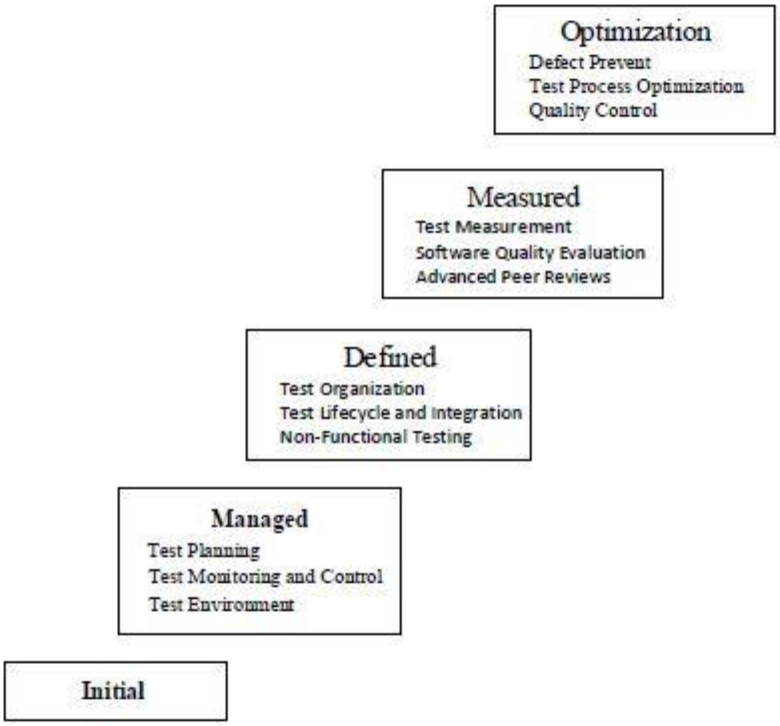


Figure 3.4: Capability Maturity Model (CMM) Levels [12]

Level One: Initial - The software process is characterized as inconsistent, and occasionally even chaotic. Defined processes and standard practices that exist are abandoned during a crisis. Success of the organization majorly depends on an individual effort, talent, and heroics. The heroes eventually move on to other organizations taking their wealth of knowledge or lessons learnt with them.

Benifit: None. A project is Total confusion

Level Two: Repeatable - This level of Software Development Organization has a basic and consistent project management processes to track cost, schedule, and functionality. The process is in place to repeat the earlier successes on projects with similar applications. Program management is a key characteristic of a level two organization.

Benifit:

- Processes become easier to comprehend
- Managers and team members spend less time explaining how things are done and more time in executing it
- Projects are better estimated, better planned and more flexible
- Quality is integrated into projects
- Costing might be high initially but goes down over time
- Ask more paperwork and documentation

Level Three: Defined - The software process for both management and engineering activities are documented, standardized, and integrated into a standard software process for the entire organization and all projects across the organization use an approved, tailored version of the organization's standard software process for developing, testing and maintaining the application.

Benefit:

- Process Improvement becomes the standard
- Solution progresses from being "coded" to being "engineered"
- Quality gates appear throughout the project effort with the entire team involved in the process
- Risks are mitigated and don't take the team by surprise

Level Four: Managed - Management can effectively control the software development effort using precise measurements. At this level, the organization set a quantitative quality goal for both software process and software maintenance. At this maturity level, the performance of processes is controlled using statistical and other quantitative techniques and is quantitatively predictable.

Benefit:

- Optimizes Process Performance across the organization
- Fosters Quantitative Project Management in an organization.

Level Five: Optimizing - The Key characteristic of this level is focusing on continually improving process performance through both incremental and innovative technological improvements. At this level, changes to the process are to improve the process performance and at the same time maintaining statistical probability to achieve the established quantitative process-improvement objectives.

Benefit:

- Fosters Organizational Innovation and Deployment
- Gives impetus to Causal Analysis and Resolution

3.6.2 Background

CMM was developed by the SEI at Carnegie Mellon University in Pittsburgh. It has been used extensively for avionics software and government projects, in North America, Europe, Asia, Australia, South America, and Africa. Currently, some government departments require software development contract organization to achieve and operate at a level 3 standard.

How long does it take to Implement CMM?

CMM is the most desirable process to maintain the quality of the product for any software development company, but its implementation takes a little longer than what is expected [13].

- CMM implementation does not occur overnight

- It's just not merely a "paperwork."
- Typical times for implementation is
    - ✓ 3-6 months -> for preparation
    - ✓ 6-12 months -> for implementation
    - ✓ 3 months -> for assessment preparation
    - ✓ 12 months ->for each new level

3.6.3 Scope of CMM

Each level in CMM is defined into key process area or KPA, except for level-1. Each KPA defines a cluster of related activities, which when performed collectively achieves a set of goals considered vital for improving software capability

For different CMM levels, there are set of KPA's, for instance for CMM model-2, KPA is

- ➢ REQM- Requirement Management
- ➢ PP- Project Planning
- ➢ PMC- Project Monitoring and Control
- ➢ SAM- Supplier Agreement Management
- ➢ PPQA-Process and Quality Assurance
- ➢ CM-Configuration Management

Likewise, for other CMM models, you have specific KPA's. To know whether implementation of a KPA is effective, lasting and repeatable, it is mapped on the following basis

1. Commitment to perform
2. Ability to perform
3. Activities perform
4. Measurement and Analysis
5. Verifying implementation

Limitations of CMM Models:

- ➢ CMM determines what a process should address instead of how it should be implemented

- It does not explain every possibility of software process improvement
- It concentrates on software issues but does not consider strategic business planning, adopting technologies, establishing product line and managing human resources
- It does not tell on what kind of business an organization should be in
- CMM will not be useful in the project having a crisis right now

The Important of CMM:

Today CMM act as a "seal of approval" in the software industry. It helps in various ways to improve software quality [13].

- It guides towards repeatable standard process and hence reduce the learning time on how to get things done
- Practicing CMM means practicing standard protocol for development, which means it not only helps the team to save time but also gives a clear view of what to do and what to expect
- The quality activities gel well with the project rather than thought of as a separate event

It acts as a commuter between the project and the team

3.6.4 Context

In the 1970s, technological improvements made computers more widespread, flexible, and inexpensive. Organizations began to adopt more and more computerized information systems and the field of software development grew significantly. This led to an increased demand for developers—and managers—which was satisfied with less experienced professionals. Unfortunately, the influx of growth caused growing pains; project failure became more commonplace not only because the field of computer science was still in its infancy, but also because projects became more ambitious in scale and complexity. In response, individuals such as Edward Yourdon, Larry Constantine, Gerald Weinberg, Tom DeMarco, and David Parnas published articles and books with research results in an attempt to professionalize the software development process.

Watts Humphrey's Capability Maturity Model (CMM) was described in the book Managing the Software Process (1989). The CMM as conceived by Watts Humphrey was based on the earlier work of Phil Crosby. Active development of the model by the SEI began in 1986.

The CMM was originally intended as a tool to evaluate the ability of government contractors to perform a contracted software project. Though it comes from the area of software development, it can be, has been, and continues to be widely applied as a general model of the maturity of processes in IS/IT (and other) organizations.

The model identifies five levels of process maturity for an organization. Within each of these maturity levels are KPAs (Key Process Areas) which characterize that level, and for each KPA there are five definitions identified:

1. Goals
2. Commitment
3. Ability
4. Measurement
5. Verification

The KPAs are not necessarily unique to CMM, representing - as they do - the stages that organizations must go through on the way to becoming mature.

The assessment is supposed to be led by an authorized lead assessor. One way in which companies are supposed to use the model is first to assess their maturity level and then form a specific plan to get to the next level. Skipping levels is not allowed.

3.6.5 Current state:

Although these models have proved useful to many organizations, the use of multiple models has been problematic. Further, applying multiple models that are not integrated within and across an organization is costly in terms of training, appraisals, and improvement activities. The CMM Integration project was formed to sort out the problem of using multiple CMMs. The CMMI Product Team's mission was to combine three source models:

- The Capability Maturity Model for Software (SW-CMM) v2.0 draft C

- The Systems Engineering Capability Model (SECM)
- The Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98
- Supplier sourcing

CMMI is the designated successor of the three source models. The SEI has released a policy to sunset the Software CMM and previous versions of the CMMI. The same can be said for the SECM and the IPD-CMM; these models were superseded by CMMI.

Future direction:

With the release of the CMMI Version 1.2 Product Suite, the existing CMMI has been renamed the CMMI for Development (CMMI-DEV), V1.2. Two other versions are being developed, one for Services, and the other for Acquisitions [13].

In some cases, CMM can be combined with other methodologies. It is commonly used in conjunction with the ISO 9001 standard, as well as with the computer programming methodologies of Extreme Programming (XP), and Six Sigma.

3.7 Intellectual Property (IP)

Many people underestimate one of the most important aspects in the software development process – Intellectual Property (IP) issues. Failure to pay the necessary attention to these issues may result in serious consequences, such as lawsuits and huge sums for compensation.

3.7.1 Overview

Intellectual property for software is computer code or software protected by law under either a copyright, trademark, trade secret or software patent.

Software innovation is valuable to individuals, start-ups, and businesses. The law is the best way to protect material such as software. To use the law as protection, programmers and businesses treat software as intellectual property.

When you treat your software as intellectual property, you have more control over who gets to use it and how it gets to the public. Otherwise, people might use it without permission, and you'll lose the chance to get paid when people use your software. In extreme cases, you might lose the right to use the software you created [14].

3.7.2 Background

A software license is a legal means of exploiting intellectual property rights (IPR). It constitutes a contract between a licensor (a software publisher) and a licensee granting the latter to use this software under certain conditions. In addition to this, a software license normally includes liability and responsibility terms.
Which one will be suitable to protect your product? Let’s see what are the options.

Software licenses are usually divided into two categories: proprietary licenses and free and open source licenses [15].

Proprietary licenses:

Under a proprietary license, the licensor grants the use of one or more copies of software to the licensee, but ownership of those copies remains with the licensor (the software publisher). Software modifications and access to the source code are usually prohibited by in this type of license. A proprietary license involves strict control of software use conditions which are normally reflected in the end-user license agreement (EULA).

This license category involves different types of contracts, most of which assume a financial contribution from the licensee in order to be granted the use of the product (although shareware and freeware licenses are different) [16].

Free and open source licenses

Free and Open source licenses assume granting the licensee the rights to run the software, to study how it works, to modify it, and to redistribute both unmodified and modified copies [2]. This license category can be further divided into the following two sub-categories:

- permissive licenses (such as BSD license and MIT license) allowing a licensee to use, study, privately modify the software and imposing just minimal restrictions on further software redistribution;

- copyleft licenses (such as GNU General Public License (GPL)) allowing a licensee to use, study, privately modify the software and redistribute it (provided it is consistent with GPL terms and conditions) [1].

The image below gives a general understanding of how these two license categories differ in terms of the rights reserved.

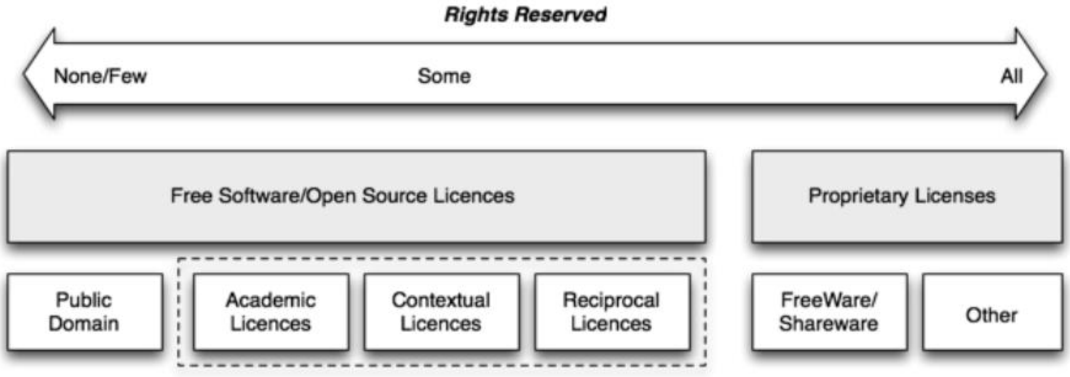


Photo Source iprhelpdesk.eu

Which option to choose will depend on the type of product you are developing.

3.7.3 Context

You have already decided how you will protect your product once it is ready. But have you ever thought of the challenges you might face during the software development process? Let's see which IP issues can arise while developing your product.

Program code reskinning:

If you came up with a decision not to develop the whole project from scratch, but to purchase the source code of an already existing app and reskin it the way that it will constitute a new unique project, it is important to know how to avoid possible legal troubles.

- First, the code should be purchased under completely legal circumstances and from a reliable vendor.
- Second, you need to check what the purchasing agreement actually allows you to do with the code – modify it and use it for your own needs, or distribute the modified version further.
- Third, you have to make sure that you customize the purchased material enough for it to be considered a new product [17].

Keeping these things in mind, you might prefer creating the project from the ground up to reskinning the existing code. Read more about the pros and cons of both options here and here.

Graphic and UI design:

Most of the software projects require not only clean programming code creation, but also a beautiful design. It is important to initially make sure the designer is not going to use other designers' works. In case the designer is going to purchase images, ask him to show you payment documents. If the designer uses free images, you should make sure their usage does not violate ownership rights. Make sure that while making mockups Provider will not use similar existing interfaces as a basis as well.

Parsing third-party sources:

Some projects involve scanning information from the third-party services – flight tickets/hotels booking services, price comparison websites, etc. You will need to check the Terms of Use of the third-party resources and make sure that you can use their materials, and on which conditions. Most of the resources will require giving credit to them.

Source code ownership rights:

The deployment stage of the project assumes placing the source code on the customer's server (or on customer's hosting provider's server), and project documents handing over to the customer (for example user manual) if such has ever been created.

Together with that, you should consider whether the software will be competitively sensitive. If you feel that your Provider can potentially develop something similar for others in future, you might want to obtain IP ownership [18] For example, in our company the exclusive ownership rights are always handed over to the customer by means of signing an IP rights transfer agreement (an Agreement for compensatory transfer of exclusive intellectual property rights).

3.7.4 Scope of IP

There is clearly a benefit to your company owning the IP in the software as it means we have complete control over how it is used, including how it is commercialized, who can use it and for what purposes. If the developer agrees to this, your contract should contain a clear assignment of the IP in the software to your company. Types of the IP are:

Copyrights:

Copyright law defines copyright as: "original works of authorship fixed in any tangible medium of expression." we can find this quotation and a longer definition in Copyright Act, 2000 (Act No. 28 of 2000, as amended up to 2005) (2005) BD. Copyright Law. So the way we express an idea, like a work of fiction or software in code, falls under Copyright law.

What Copyright Protects:

- Against word-for-word copying
- Internationally, as soon as you create it

Registering your copyright with the Copyright Office is a good idea for legal purposes.

Rights Granted by a Copyright:

A copyright grants you specific rights in terms of your software. When you hold the copyright to software code, you can:

- Make copies of your software code
- Sell or give away the code
- Make a "derivative work," which is a second software that uses a lot of the original code
- Post the code somewhere, or otherwise, display it

An official copyright registration is easy to get and doesn't cost we much. It can cost as little as 1300tk and only takes about four months.

Patent:

To protect a process, like the function of software, you need a patent. A patent will protect things like:

- Systems
- Functions
- Solutions to computer problems

we can use two types of patents to protect software: utility and design. Utility protects what the software does. Design protects any decorative part of your software.

Unlike copyright law, the patent law protects the invention itself. That way, someone can't create a software program with different code that does the exact same thing your software does. But the patent doesn't protect your specific lines of code against plagiarism the way copyright does.

Things to Consider Before Applying for a Patent:

- Patents last for 20 years after the day you receive the patent. Then the work goes into Public Domain.
- If you patent solving a specific problem, you might block other programmers from solving the problem a different way.
- Getting a patent can take as long as two years.

Trade Secret:

A trade secret is a unique identification or information of a company, that other people don't have. Company uses this information in business, and it gives you a leg-up over your competition. we don't file any documents or apply with an office to get a trade secret. Instead, the way we treat your software can make it a trade secret. we have to take "reasonable measures" to keep the software a secret:

- Keep the software away from the public.
- Have employees sign non-disclosure agreements.
- Have employees sign non-compete agreements.
- Do exit interviews with employees who are leaving to make sure they aren't bringing IP with them.
- As soon as an employee quits or is fired, take away all their file and data access.
- Investigate any suspicious employee activity.
- Keep IP data stored in compartments, and only give access to employees who need it.

we can maintain a trade secret for as long as you want. Unless someone discovers your secret by what the law calls "fair means," your trade secret will last forever. If someone else discovers, on their own, a trade secret similar to yours, you can't take legal action. Sometimes companies and individuals don't see trade secrets as secure enough protection for valuable software inventions.

3.7.5 Intellectual Property Rights in Software:

People talk a lot in the information technology business about “intellectual property rights.” But what are they? How do they apply to software technology? Why should you protect them? How do you protect them?

Intellectual property rights are at the foundation of the software industry. The term refers to a range of intangible rights of ownership in an asset such as a software program. Each intellectual property “right” is itself an asset, a slice of the overall ownership pie. The law provides different methods for protecting these rights of ownership based on their type.

There are essentially four types of intellectual property rights relevant to software: patents, copyrights, trade secrets and trademarks. Each affords a different type of legal protection. Patents, copyrights and trade secrets can be used to protect the technology itself. Trademarks do not protect technology, but the names or symbols used to distinguish a product in the marketplace. We’ll save a discussion of trademarks for a later issue [18].

3.7.6 Conclusion

To ensure that everything goes smoothly, be sure to check that all these points are stated in your software development agreement. Ask your lawyer for help if needed.

Share this information with your colleagues and relatives to be sure they will not encounter these IP issues [19].

3.7.4 Roles and responsibilities

1. Project Manager
2. Team Leader
3. Developer
4. Tester
5. User

CHAPTER 4
RESEARCH METHODOLOGY

This section provides the roadmap to achieve the research objectives [20]. The research methodology was based on a comprehensive survey conducted through structured interviews and an online survey in Software development industry [21]. Since the main objectives of the study and the research are to understand usage software methodology of the software industry in the Bangladeshi context. To successfully conduct the thesis below steps were taken.

1. Various papers on Software Development Standard and Software Engineering Practices were studied, which were published.
2. Various papers and books on survey methods were studied, more than 30 studies were identified and 20 studies were selected for the final review.
3. The key factors which are needed to understand were identified, which are: Paper Title, Abstract, Keywords list, Literature Review, Survey and so on.
4. The questions relevant to those facts were identified.
5. Desired answers were divided into quantitative and qualitative data.
6. Based on the ISO, CMM, IP and BASIS rule on Software Development Standard and Software Engineering Practices questionnaires were developed.
7. The survey setup and survey conduction plan were developed.
8. Finally, a complete set of questionnaires and survey conduction plan is proposed

which ultimately will be used to understand usage software methodology of the software industry in Bangladeshi context.

CHAPTER 5

THE SURVEY

5.1 A Brief History of Survey Methodology

- The guideline for survey design was first developed early in the twentieth century.
- Neyman‘s paper on the representative method was published in 1934 which was a landmark on survey methodology [6].
- The method of the interpenetration of interviewer assignments to estimate errors made by survey field-workers was developed by Mahalanobis in the 1940s [From Wikipedia].

5.2 The concept of Survey

“A prerequisite to designing a good survey instrument is deciding what is to be measured” [22]. So, a research project has some survey objectives and concepts**.** They are given below [22]:

- A survey concerns a set of objects comprising a population. The population under study has one or more measurable properties.
- The goal of the project is to describe the population by one or more parameters defined in terms of the measurable properties. This may require observing a sample of the population.
- A sample of objects is selected from the frame in accordance with a sampling design that specifies a probability mechanism and a sample size.
- Observations are made on the sample in accordance with a measurement process (i.e. a measurement method and a prescription as to its use).
- Based on the measurements, an estimation process is applied to compute estimates of the parameters when making an inference from the sample to the population.
- The general objective of the survey, as well as the main design characteristics, i.e. the target stated.
- Close contacts (informal and formal) with the key users are indispensable. Informal contacts and tests, as well as expert meetings, are of help.
- During conceptualization, the ultimate goal of operationalization (transfer into indicators and observable variables) should be considered.

5.3 The Survey Process

The word survey is often used to describe a method of gathering information from a sample of units, a fraction of the persons, households, agencies, and so on, in the population, the sampling design, the preferable data collection mode, should be clearly a population that is to be studied. To understand the present and to plan for the future, data are needed on the preferences, needs, and behaviors of people in society as well as other entities, such as business establishments and social institutions. For many researchers and planners, just like my case, sample surveys and censuses are major sources of this information. As shown in Figure 5.1, the survey process is composed of a number of steps that are executed more or less sequentially, from determining the research objectives to analyzing the data [6].

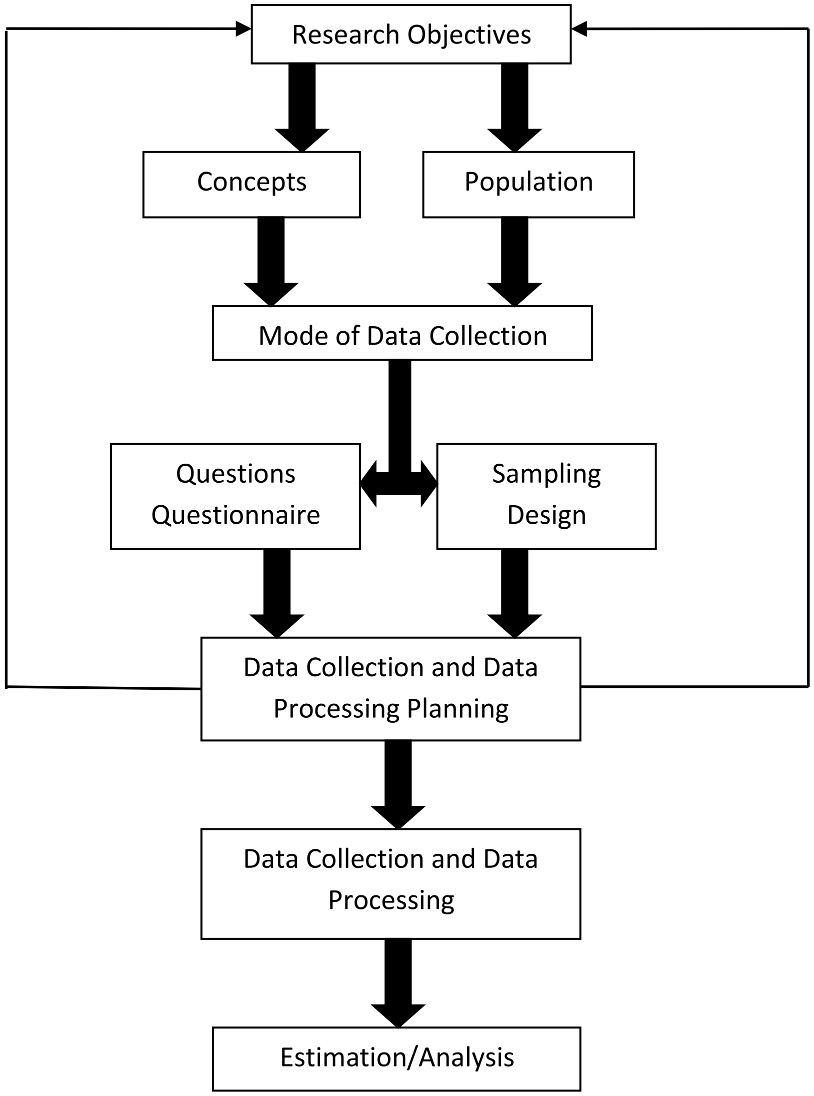


Figure 5.1: Survey process

The planning stages of the survey process are largely iterative. At each stage of planning, new information may be revealed regarding the feasibility of the design. The research objectives, questionnaire, target population, sampling design, and implementation strategy may be revised several times prior to implementing the survey.

5.3.1 The Survey Objective

The first step in the survey process is to determine the research objective [23]. A well-specified set of research objectives is a critical component of the survey process and will facilitate many of the decisions involved in survey design. Defining the research objectives is often accomplished best by identifying a small set of key research questions to be answered by the survey. As we will see later in this chapter, it will be shown that such a key research question was developed. Eventually, each question posed for the research can be linked to one or more data elements to be collected in the data collection phase of the process. These data elements or items are in turn linked to one or more questions on the survey questionnaire or form. In fact, it is good practice to ensure that every question on the questionnaire corresponds to at least one research question, to avoid the situation where questions that are superfluous and really not needed for the purposes of the survey somehow find their way onto the questionnaire. This process of linking research objectives and survey questions also ensures that all survey questions necessary to address the research objectives fully are included in the questionnaire [23].

Table 5.1: Correspondence between Research Questions and Survey Questions.

Part of the questionnaire	1st	2nd	3rd	4th	5th
Survey Question	7	4	6	3	3
Research Question	5	3	6	2	3

A table such as this is useful for identifying redundant or unnecessary questions in the questionnaire or unaddressed research questions.

5.3.2 The Target Population:

The next step in the survey process is to define the population to be studied or the target population for the survey. The target population is the company or the group of persons or other units for whom the study results will apply [6]. In Bangladesh, maximum IT companies are established after 2000-2017. Besides, a lower amount IT companies are established before 2000. A tiny amount of IT companies is not certified from BASIS, remaining all companies are certified. Not only certified BASIS but also they follow CMMI maturity level. Table 5.2 shows “Profile” of Software Industry in Bangladesh.

Table 5.2: “Profile” of Software Industry in Bangladesh.

Serial No.	Company Name*	Address	Established Year		CMMI Level	ISO Certified	IP Maintain	BASIS Certified
			Before 2000	2000-2017				
1.	RAIT Ltd.	Dhaka		✓	Level-1	✓	Copyrights, Trade Secret Patent	✓
2.	IPCP Services	Dhaka	✓		Level-2	✓		✓
3.	MKB Technologies	Rangpur		✓	Level-3	✓	Copyrights	✓
4.	ZSI Bd.	Rangpur		✓	Level-4	✓		✓
5.	Banglalink	Dhaka	✓		Level-5	✓	Copyrights, Trade Secret	✓
6.	PCN (Pvt) Ltd.	Dhaka	✓		Level-5	✓		✓

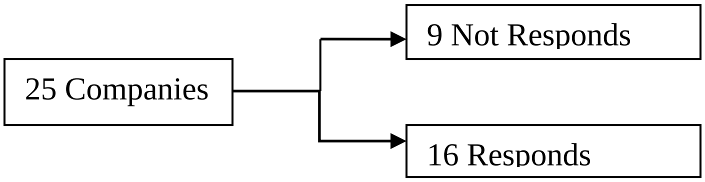
*real names camouflaged [24]

5.3.3 The Mode of Administration:

Having specified the research objectives and defined the target population, the next step in the process is to determine the mode of administration for the survey. To develop the survey, two types of interviews were conducted.

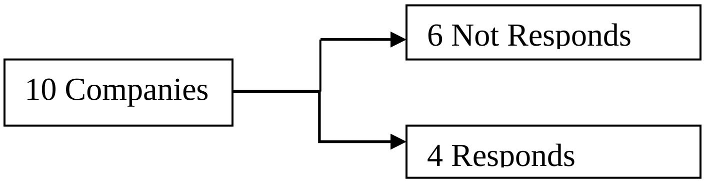
5.3.3.1 Face-to-face interview survey:

This technique implies the paper questionnaire and the presence of the interviewer [23]. Totally, 25 software companies were initially selected for the survey. There were 16 companies that responded. Only 8 of them didn't respond to the survey.



5.3.3.2 Online Survey (Mail & Telephone):

In mail surveys paper questionnaires are sent to the respondents by mail, totally 10 software companies were initially selected for the survey. There were 4 companies that responded. Only 6 of them didn't respond to the survey.



5.3.4 Developing the Questionnaire:

The next step of the survey process is the development of the questionnaire or instrument. A questionnaire was designed, with 26 questions on SDMs and 8 questions on Company Demography [4]. There are two blocks of total 34 questions.

Block 1. On the SDM (ISO, CMM, IP) Methods:

The questionnaire contained 26 questions divided into five parts. It includes 23 multiple-choice questions and 3 open question. The titles of the questionnaire are given below [25]:

1. Requirements Analysis
2. Planning/Design
3. Development/Coding
4. Testing
5. Miscellaneous

All questions were based on Roles and Responsibilities, Artifacts of ISO, CMM and IP Method Specially Software Development Standard and Software Engineering Practices and DSDM.

Block 2. On the Company:

The questionnaire contained 8 questions. It includes 6 multiple-choice questions and 2 open questions. All questions were based on [12]:

1. Information of the company/organization
2. City where the company is located
3. Some information (Membership of BASIS, CMM level)

5.3.5 Designing the Sampling Approach

Having defined the target population and the research objectives and determined the mode of administration, the next stage of the survey process can begin, that of specifying the sampling design. The sampling design specification describes the sampling frame (i.e., the list of population members) to be used for the survey, the methods used for randomly selecting the sample from the frame, and the sample sizes that are required. The sampling frame is simply the list of target population members from which the sample will be drawn. It may also be a combination of several lists, a map, or any other device that can be used to select the sample. As mentioned previously, the frame chosen for sampling depends to a large extent on the mode of administration for the survey [6]. For this survey, a statistical frame and theoretical frame is a company who has followed to SDM.

5.3.6 Developing Data Collection and Data Processing Plans

Once the initial, basic design decisions are made, the data collection and data processing plans are developed [6]. Totally 55 software companies were selected for the survey. Only 20 of them didn't respond to the survey. Only 35 software companies were responded that has given the previous part (The Mode of Administration). Our sample consisted of 13 large firms with over 100 plus employees,15 medium firms with over 21-50 employees and 7 small firms with below 20 employees. All the firms had a business focus on IT and IT consulting. When Face-to-face interview and Online Survey (Mail & Telephone) were combined, the period was 4 months from 1st January 2017 to 30th April 2018 [11].

5.3.7 Collecting and Processing the Data

The next step of the survey process involves implementing the data collection and data processing plans developed in the previous steps [6]. For this survey, a statistical frame is people who have access to the method. Even in a well-planned survey, unforeseen problems can develop which require deviations from the plans. Here it is important for the project staff to monitor carefully the progress of the data collection operations via measurements on key process variables to identify potential problems before they develop into real problems. Thus, an important aspect of the data collection plan is a process for routine monitoring of data collection and obtaining feedback from the supervisory staff. Once the data are in computer-readable form, it can be edited, cleaned, and prepared for estimation and analysis. Editing the data involves correcting out-of-range or inconsistent responses, possibly recontacting respondents to obtain additional information, and generally, cleaning the data of many discernible errors. Information obtained from an open-ended question—that is, a question that elicits an unstructured response is often converted into code numbers that summarize the verbal information provided by the respondent [23].

5.3.8 Estimation and Data Analysis

Finally, the data are weighted to compensate for unequal probabilities of selection, missing data, and frame problems, and the estimates are computed following the plans previously developed for estimation and analysis [6]. The estimation and analysis plan lists the major research questions that should be addressed in the analysis, the estimates that will be computed, and the statistical analyses that will be performed. Thus, our guiding philosophy is that it is more useful to learn a few basic techniques for dealing with the underlying causes of survey error and the general theories leading to their development rather than to learn numerous ad hoc methods that essentially treat the same causes of survey error but under a variety of special circumstances [23]. The data were transcribed and they were compared with the notes that were taken during the interview. It was ensured that only the relevant data were transcribed. The results were entered into spreadsheet and the calculations were done. Data were presented using pie charts with the help of Microsoft Excel [5]. Results were categorized as follows:

- The percentage of respondents who follow software development method.
- Adoption of ISO
- Adoption of CMM.
- Adoption of IP.
- The percentages of different Software Development Standard and Software Engineering Practices used by the respondents.

CHAPTER 6

ANALYSIS

6.1 Adoption of SDM

The respondents were asked whether or not they were following the Software Development Method.

Answer	Number	Percent
Yes	25	50%
No	5	10%
We use techniques and tools, but no method.	20	40%

Table 6.1: Proportion of SDM.

As displayed in Figure-6.2, 50% of the respondents are currently using and involved with SDM methodologies in their developing process. These shows that almost 3/5 of the companies are adopting SDM methodologies as a working practice [20]. However, 10% has responded that SDM methodologies are not being used in their development process. And 40% of companies are used their own techniques and tools.

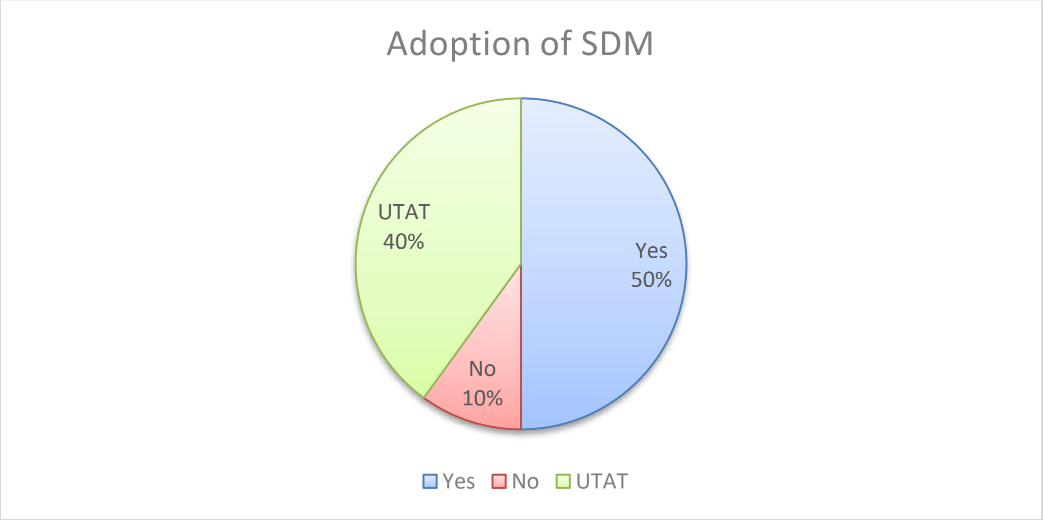


Figure 6.1: Percentage of SDM (UTAT=Use Techniques and Tools).

6.2 Adoption of ISO

The respondents that answered ‘yes ‘were then asked the questionnaire. The result is shown in table-6.2. When viewing the data of the individual practices of ISO Requirements, the appreciated ISO practices among the respondents that had actually applied the practices were [26]: 1. Role of Have Certification 2. Analyses the Requirements 3. Role of Design Team 4. Team Size of SW Developers.

The most appreciated Scrum practice seemed to be the daily ISO meeting (80%) as shown in Table-6.2. They also used the Burndown Chart for completing the project and divided into the small group if the team size is large.

ISO Activities	No. of Responses	Percent
Have Certification	40	80%
Analyses the Requirements	45	14%
Role of Design Team	38	6%
Total	50	100%

Table 6.2: Adoption of ISO

6.3 Adoption of CMM

Compared with ISO requirements for development, the appreciated CMM practices among the respondents that had actually applied the practices were: 1. Role of Tracker 2. Customer Involvement 3. Role of Programmer 4. Team Size of SW Developers 5. The significance of Method as shown in Table-6.3. The most used practices of CMM are the 40 h week (65%) and pair programming (67%) as shown in Table-7.1, both of which can be applied in any process model of software development whether agile or traditional [26].

CMM Activities	No. of Responses	Percent
Role of developing plan	45	90%
Testing and monitoring	40	85%
Software qualities Evaluation	42	88%
Test Measurements	35	75%
Advance Peer Review	40	85%
Total	50	84%

Table 6.3: Adoption of CMM.

6.4 Adoption of IP

Table-6.4 shows, the results on fundamental activities of the IP for the Software development process. The appreciated IP practices among the respondents that had actually applied the practices were: 1. Having Copyrights 2. Having Patent 3. Having Trade Secret

IP Activities	No. of Responses	Percent
Having Copyrights	10	34%
Having Patent	7	14%
Having Trade Secret	14	40%
NO IP	6	12%
Total	20	88%

Table 6.4: Adoption of IP

The survey has come up with a decision that different types of Software Development Standard and Software Engineering Practice a Case Study of Bangladesh methods is used in Bangladeshi IT Companies. It was found that “ISO” was leading the way at 80%, followed by “CMM” at 84% and “IP” at 88% as shown in Figures.

CHAPTER 7

CONCLUSIONS AND LIMITATIONS OF THE STUDY

The purpose of this study was to develop an understanding of the appliance of Software Development Standard and Software Engineering Practices development methodologies in software industry perspective in Bangladesh. Early discussed, Software Development Standard and Software Engineering Practices is considered one of the most popular software design and development Methodologies [28]. Especially, ISO (58%), CMM (24%) and IP (18%) are one of the most popular software design and development Method in Bangladesh. However, generally speaking it could be argued on the basis of the results of this study that embedded software development organizations seem to be able to apply the three Software Development Standard and Software Engineering Practices methods, namely-ISO, CMM and IP and their individual practices in their projects and report fairly positive results of their application [26].

As explained previously, the result shows that the software industry in Bangladesh is methodological and technological [9]. This study adds evidence not only IT Industry in Bangladesh but also the knowledge of software engineering and software process [20].

We acknowledge that there are limitations to this research. Because of company policy, some companies did not agree to deliver their information. In hindsight, we acknowledge that the questionnaire could have been improved [27]. More general question on the method activities could have generated more information on this issue. However, although our survey respondents represent a sizeable proportion of the Bangladeshi software development sector.

7.1 Future work and Recommendations

The outcome of this research, there are some possible future studies relevant to this research, which can be considered as further extensions. Now that we know what methodologies are used, we can conduct interviews and study groups using ethnographic techniques to learn how to identify and alleviate some of the more specific problems they face deploying Software Development Standard and Software Engineering Practices. The results of our study contribute to our understanding of how Software Development Standard and Software Engineering Practices methodologies are being implemented in the workplace.

Our findings show that although there is a positive bias towards the appliance software development method, the importance of appliance software method is perceived to develop the project. it would be interesting to explore the future.

Our survey revealed that companies perceive software development methods and that's for the need to know about the usability of software development methodologies. it also would be interesting to explore the future.

The recommendations are:

1. There need to be developed their skills in software development methodology.
2. Need to establish viable training facilities.
3. Need to improve their working environment.

There need to know about the usability of software development methodologies.

References

[1] Rishi Dave," Patterns of Success in the Indian Software Industry"

Available at: https://web.stanford.edu/group/scip/avsgt/Dave_thesis.pdf

[2] Importance of Processes and Standards in Software Development

Available at: https://www.codeproject.com/Articles/17121/Importance-of-Processes-and-Standards-in-Software

[3] GNU Coding Standards

Available at: http://www.gnu.org/prep/standards/standards.html

[4] DO-178B, "Software Considerations in Airborne Systems and Equipment Certification"

Available at: http://www.dcs.gla.ac.uk/~johnson/teaching/safety/reports/schad.html ; August 2018

[5] CS 410/530 - Software Engineering Requirements Engineering; Sommerville, Software Engineering, 10 ed., Chapter 4:

Available at: http://www.cs.ccsu.edu/~stan/classes/CS410/notes16/04-Requirements.html

[6] ISO 9001 Checklist

Available at: https://www.iso-9001-checklist.co.uk/tutorial/7.3-design-and-development.htm

[7] Andrew Begel , Nachiappan Nagappan,"Usage and Perceptions of Agile Software Development in an Industrial Context An Exploratory Study", Conference Paper, pg.1-11,October 2007.

[8] ISO Certification

Available at: https://www.aas-bd.com/iso-certification/

[9] In Bangladesh, ISO is represented by BSTI and functions through Bureau of Assessment Services

Available at: http://www.theindependentbd.com/printversion/details/108867

[10] ISO 9001 - quality management system

Available at: http://www.bascertification.com/isocertification-iso9001.php

[11] Tutorials point "Testing and testing technique"

Available at: https://www.tutorialspoint.com/software_testing_dictionary/age_testing.htm

[12] Tutorials point "Capability Maturity Model"

Available at:
https://www.tutorialspoint.com/software_testing_dictionary/capability_maturity_model.htm

[13] Capability Maturity Model (CMM) & it's Levels in Software Engineering

Available at: https://www.guru99.com/capability-maturity-model-cmm-cmm-levels-a-fool-s-guide.html

[14] Intellectual Property Protection for Software: What to Know

Available at: https://www.upcounsel.com/intellectual-property-software

[15] Software license (WiKi)

Available at: https://en.wikipedia.org/wiki/Software_license

[16] Fact Sheet IPR management in software development

Available at:

https://www.iprhelpdesk.eu/sites/default/files/newsdocuments/Intellectual%20Property%20Rights%20Management%20in%20Software%20Developments_updated.pdf

[17] Five Key IP Issues in Software Development Agreements -From the Experts -"Daniel C. Glazer and Arianne de Govia"

Available at:
http://www.friedfrank.com/siteFiles/Publications/CC_5KeyIPIssuesSoftwareDevelopmentAgreements_022713.pdf

[18] Intellectual Property Issues in Software Development

Available at: https://issart.com/blog/intellectual-property-issues-software-development/

[19] How to avoid copyright infringement when reskinning your app

Available at: https://www.appsfresh.com/blog/how-to-avoid-copyright-infringement-when-reskinning-your-app/

[20] M.M.M.Safwan, Thavarajah G., Vijayarajah N., K. Senduran, C.D. Manawadu "An Empirical Study of Agile Software Development Methodologies: A Sri Lankan Perspective", International Journal of Computer Applications,pg.1-7, December 2013

Available at:

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.401.9304&rep=rep1&type=pdf

[21] S.Nithila, K. Priyadharshani, Y. S. G. Attanayake, T. Arani and C.D. Manawadu,

Emergence of Agile Methodologies: "Perceptions from Software Practitioners in Sri Lanka", International Journal of Scientific and Research Publications, Volume 3, pg.1-6,November 2013

Available at: http://www.ijsrp.org/research-paper-1113/ijsrp-p2381.pdf

[22]G.Brancato,S.Maccia,M.Murzia,M.Signore,G.Simeoni,K.Blanke,T.Korner,A.Nimmergut,P.Lima,R.Paolino,Hoffmeyer-Zlotnic,"Handbook of Recommended Practices for Questionnaire Development and Testing in the European Statistics System",ECGA-200410300002,pg.1-162.

Available at:

http://ec.europa.eu/eurostat/documents/64157/4374310/13-Handbook-recommended-practices-questionnaire-development-and-testing-methods-2005.pdf/52bd85c2-2dc5-44ad-8f5d-0c6ccb2c55a0

[23] PAUL P. BIEMER, LARS E. LYBERG," Introduction to Survey Quality",

Available at: http://www.books.mec.biz/tmp/books/7YC3VBEVW3EENB22K1LW.pdf

[24] Sriram Rajagopalan, Saji K Mathew, "Choice of Agile Methodologies in Software Development: A Vendor Perspective", Journal of International Technology and Information Management, Vol. 25,pg.1-17,2016.

Available at: http://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1251&context=jitim

[25] Bernhard Rumpe, Astrid Schröder," Quantitative Survey on Extreme Programming Projects", Third International Conference on Extreme Programming and Flexible Processes in Software Engineering,pg.1-6, May 26-30, 2002.

Available at: https://arxiv.org/ftp/arxiv/papers/1409/1409.6599.pdf

[26] O. Salo and P. Abrahamsson," Agile methods in European embedded software

development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum",pg.1-8,2008.

Available at: http://digital-library.theiet.org/content/journals/10.1049/iet-sen_20070038

[27] "Software Development Methods and Usability. Perspectives from a Survey in the Software Industry in Norway"

Available at: http://bura.brunel.ac.uk/bitstream/2438/3424/3/Software%20development%20methods%20and%20usability.pdf

[28] Al-Zewairi1, M.,Biltawi, M., Etaiwi, W. and Shaout, "Agile Software Development Methodologies: Survey of Surveys.", Journal of Computer and Communications,pg.1-24, March 31, 2017.

Available at : http://file.scirp.org/pdf/JCC_2017033115471602.pdf

APPENDIX

Table 7.1: Questionnaire with dataset.

Question	Dataset

Do you follow the Software Development Method?	Yes=50%
	No=10%
	We use techniques and tools, but no method.= 40%

Who gathers user stories?	a) Product owner=71% b) Tracker=19% c)Others=10%
How about your customer involvement?	a) once and again=70% b) on-site customer=18% c)Others=12%
Who makes the planning/design part?	a) Design team=50% b) Programmer=44% c) Others:=6%
What is the number of software developers in the development department?	a) 1-10 =68% b)11-20=16% c) 21-50=5% d) 50+=11%
Which is significant according to development/coding?	a) method=6% b) framework=18% c) all of the above=76%
Do you hold a daily Scrum meeting?	a) Yes=65% b) No=35%
Do you use the Burndown Chart for completing the project?	a) Yes=44% b) No=56%
If the team size is large, then what you do?	a) divided into small group=75% b) unchangeable=25%

Approximate working hour per-week.	a) 30 hours=6% b) 40” =65% c) 50 ”=12% d) others:6%
Do you allow pair-programming?	a) Yes=67% b) No=33%
According to your project, which is changeable/fixed?	a)Time(Changeable=57% Fixed=43%
	b) Cost (Changeable=57% Fixed= 43%
	c) Functionality/scope (Changeable=93% Fixed =7%

Table 7.2: Questionnaire with dataset for company.

Question	Dataset
Established Date	a) Before 2000 =15% b)2000-2017 =85%
What is your company/organization's main industrial section?	a) Authority b) e-commerce c) Usability, HCI d) Health and medicine sector e) IT Consultancy =80% f) Financial sector =5% g) Telecommunication h) Game =10% i) Others =5%
Please select the total number of employees in your company/organization.	a) 1-20 =45% b) 21-50=40% c) 51+ =15%
CMMI stages maturity level:	a) Level-1(Initial) =20% b) Level-2(Managed) =25% c)Level-3(Defined) =30% d)Level-4(Quantitatively) =20% e)Level-5(Optimizing) =5%
Membership of BASIS:	a) Yes =60% b)No =40%