Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Management

Diploma Thesis

Management of Software Development and its Effects on Software Quality

Radwa Ibrahim Elsayed Mousa

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Radwa Ibrahim Elsayed Mousa

Informatics

Thesis title

Management of Software Development and its Effects on Quality

---

Objectives of thesis

The thesis aims to examine the various software development processes and how effective project management techniques can contribute to the quality of the developed software system by:
- Examining the process of software development and software quality metrics
- Examining various project management techniques
- Identifying suitable approaches for measuring project management effectiveness in software development
- Evaluating the meaning and applicability of the proposed software development management approach

Methodology

The theoretical part is based on the collection and analysis of the different methods and approaches of software development and their intersection with standard project management procedures with a focus on software quality metrics.

Practical part will focus on evaluating the effectiveness of management on the software development and how it reflects on the quality of the produced software by conducting survey studies on software development practitioners from various development stages (project managers, executive managers, team leads and developers).

The proposed extent of the thesis

60 – 80 pages

Keywords

Project Management, Software Project Management, Software Development, Management Effectiveness, Software Quality, Quality Metrics

---

Recommended information sources

AUSTIN, V. *Rural project management : a handbook for students and practitioners.* London: Batsford Academic and Educational, 1984. ISBN 0713444169.

CADLE, J. – YEATES, D. *Project management for information systems.* Harlow: Pearson Prentice Hall, 2008. ISBN 978-0-13-206858-1.

COTTERELL, M. – HUGHES, B. *Software project management.* London: McGraw-Hill, 2009. ISBN 978-0077122799.

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. PROVOZNĚ EKONOMICKÁ FAKULTA, – VANÍČEK, J. – STRUSKA, Z. *Measurement and rating of information systems quality : part 3: design complexity and software engineering consequences.* Praha: ČZU-PEF, 2005.

MARCHEWKA, J T. *Information technology project management : international student version/ Jack T. Marchewka.* Hoboken: John Wiley, 2010. ISBN 978-0-470-40948-0.

TAYLOR, J. *Managing information technology projects : applying project management strategies to software, hardware, and integration initiatives.* New York: Amacom, 2004. ISBN 978-0814408117.

VEERAPILLAI, G. – LEWIS, W E. *Software testing and continuous quality improvement.* Boca Raton: Auerbach Publications, 2005. ISBN 0849325242.

WILLIAMS, M. *The principles of project management.* Collingwood: Sitepoint, 2008. ISBN 9780980285864.

Expected date of thesis defence

2018/19 SS – FEM

The Diploma Thesis Supervisor

Ing. Richard Selby, Ph.D.

Supervising department

Department of Management

Electronic approval: 11. 7. 2018

prof. Ing. Ivana Tichá, Ph.D.

Head of department

Electronic approval: 19. 10. 2018

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 14. 11. 2019

Declaration

I declare that I have worked on my diploma thesis titled " Management of Software Development and its Effects on Software Quality" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break the copyrights of any person.

In Prague on 26.11.2019

Radwa Ibrahim Elsayed Mousa

Acknowledgment

I would like to thank my supervisor Selby Richard, Ph.D. for his support and advice during the work for this thesis. Additional gratitude goes to my family back at home for encouraging and supporting me every step of the way.

Management of Software Development and its Effects on Software Quality

Abstract

Planning, evaluating and optimizing the processes of project management can contribute to the successful development or deployment of software systems. Achieving high effectiveness in project management can help ensure a successful outcome and a higher quality from a software project.

The focus of the thesis is to examine the software development processes as well as various aspects of project management and how it applies in the software field. The thesis also discusses different criteria for project success and the definition of software quality to determine how they are related. Validation of this proposal is based on a survey study conducted on a number of software development practitioners spanning small to large development projects to evaluate their projects' success and confirm whether there is a connection between software project management practices and the quality of the software developed.

The motivation behind the thesis is to utilize the results and provide a number of recommendations to software practitioners on the importance of following steady and effective project management strategies in software development rather than just focusing on the deliverables and the technicalities while neglecting other elements that may be crucial to the success of software projects.

Keywords: Software development, Project Management, Project Success, Software Quality, Project Management Effectiveness.

Řízení vývoje softwaru a jeho vliv na kvalitu softwaru

Abstrakt

Plánování, vyhodnocování a optimalizace procesů při řízení projektů může přispět k úspěšnému vývoji nebo implementací softwarových systémů. Dosažení vysoké účinnosti v řízení projektů může pomoci zajistit úspěšný výsledek a vyšší kvalitu softwarového projektu.

Cílem diplomové práce je prozkoumat procesy vývoje softwaru, různé aspekty projektového managementu a jeho uplatňování v softwarové oblasti. Práce také rozebírá různá kritéria zaměřena k úspěšnosti projektu a definování kvality softwaru pro určení jejich vzájemného vztahu. Ověření tohoto návrhu je založeno na průzkumové studii provedené na řadě projektů na vývoj softwaru, které se zabývají malými až velkými vývojovými projekty, aby vyhodnotili úspěšnost svých projektů a potvrdili, zda existuje souvislost mezi postupy řízení softwarového projektu a kvalitou vyvíjeného softwaru.

Motivací této práce je využití výsledků a poskytnutí řady doporučení odborníkům v oblasti softwaru, zejména o důležitosti dodržování stabilních a efektivních strategií projektového řízení ve vývoji softwaru, spíše než jen soustředění na výstupy a technické aspekty a zanedbávání dalších prvků, které mohou být zásadní pro úspěch softwarových projektů.

Klíčová Slova: Vývoj softwaru, Projektový management, Úspěch projektu, Kvalita softwaru, Efektivita projektového řízení.

Table of content

1 Introduction 11

2 Objectives and Methodology 13

2.1 Objectives 13

2.2 Methodology 13

3 Literature Review 14

3.1 Software Projects 14

3.2 Project Management and Project Management Areas 16

3.2.1 Project Processes 19

3.2.2 Stakeholders 21

3.2.3 Output quality 22

3.2.4 Risk Management 25

3.3 The Relationship Between Project Success and Project Management 26

3.4 Software Development Models 30

3.4.1 The waterfall models 32

3.4.2 Evolutionary models 33

3.4.3 Spiral model 34

3.4.4 Agile models 34

3.5 Evaluating Project Management Effectiveness in Software Development 36

3.5.1 Subjective evaluation method 38

3.5.2 Questionnaire-based method 38

4 Practical Part 40

4.1 Survey Design 40

4.2 Data Analysis 44

4.2.1 A General Overview 44

4.2.2 Output quality 47

4.2.3 Stakeholders, Processes and Risk Management 50

5 Results and Discussion 61

6 Conclusion 64

7 References 66

8 Appendix 68

List of Figures

Figure 1 The Project Triangle 15
Figure 2 Overview of Project Management Areas and Project Management Processes 19
Figure 3 Requirement Management [From (Forsberg, Mooz, and Cotterman, 2005, P. 34)] 21
Figure 4 Project Success Model [From (Pinto and Slevin, 1988, P.69)] 27
Figure 5 Stakeholders Interested in Each Stage of The Project [From (Munns and Bjerimi, 1996, P.81)] 28
Figure 6 Five Dimensions of Project Success [From (Wiegers,1996, P.145-150)] 29
Figure 7 Software Development Life Cycle 32
Figure 8 Waterfall SDLC 33
Figure 9 Diagram of Spiral SDLC [Boehm. B. W (1988), (P.2)] 34
Figure 10 Scrum Agile Model 35
Figure 11 Four Approaches for Software Project Management Effectiveness Measurement From [Demir (2008) (P. 45)] 37
Figure 12 Overview of the Respondents' Roles in Software Development Projects 44
Figure 13 Projects Categorized Based on Whether They Exceeded Allocated Resources . 45
Figure 14 Project Resources That Exceeded Initial Estimated Efforts 45
Figure 15 Respondents Evaluation of The Projects' Success 46
Figure 16 Functional Suitability Categorized by Project Success 47
Figure 17 Software Usability Categorized by Project Success 48
Figure 18 Software Reliability Categorized by Project Success 49
Figure 19 Software Testability Categorized by Project Success 49
Figure 20 Responses to Q.2.5. You are satisfied with the Quality of the Delivered Output 50

Figure 21 Effectiveness of the Team Leader (Questions 3.1 & 3.2) 51
Figure 22 Effectiveness of Team Leader Categorized by Project Success 52
Figure 23 Responses to Q.3.4. The Project Suffered from Lack of Experienced or Qualified Team Members 52
Figure 24 Responses To Q.3.5 Task Assignments Were Fair And Based On The Qualifications of The Team Members 53
Figure 25 Communication Effectiveness Categorized by Project Success 54
Figure 26 Team Members Lacked Motivation and Are Likely to Leave The Team 55
Figure 27 All Stakeholders were Involved Different Phases of the Project Cycle 55
Figure 28 Having Regular Meetings Categorized by Project Success 56
Figure 29 Requirements were Stable and Clear Across Different Project Phases 57
Figure 30 Projects Risks were Assessed and Documented. 57
Figure 31 Risks and their Potential Impact were Communicated to Stakeholders 58
Figure 32 Potential Project Problems were Addressed Proactively and Effectively. 59
Figure 33 Well-Functioning Vs. Challenging Aspects of Project Management 59

List of tables

Table 1 Product and Process Standards 23
Table 2 Questionnaire structure 41

1 Introduction

Like most business endeavors, software companies aim to achieve success by gaining more profits while also striving to deliver valuable and reliable services to their clients. Doing so is what helps them secure their strategic position among competitors and guarantee consistent growth and sustainable development in the rapidly-evolving software market. Both the continuous technological advancement and our increasing reliability on software and electronic devices across all walks of life means the software market is still full of opportunities for these software and system development companies to cease. However, it also means that software companies nowadays are facing several challenges to stay relative in one of the most competitive fields of our age.

This constant competition for resources and market shares adds another level of complexity to the field making it a necessity for software companies to evolve and explore new grounds through a continuous process of research and innovation. Achieving this meant eliminating unnecessary obstacles or constraints that would slow down the innovation and development process. A major part of which was the administrative challenges related to the management of each project or each development cycle. Faster development processes became a must and software companies had to adopt better management techniques in order to deal effectively with the market changes while staying alert for any new opportunities that the field may present.

Attention to effective project management in the software development field is not a new concept however, but one that has been examined over the years. A report by the Standish Group (1995) found that approximately only 16% of software projects were completed on time and within budget. In addition, the projects completed contained only approximately 42% of the originally proposed features and functions.

Along the same lines, in the year 2000, a report published by the Defense Science Board (DSB) on software processes, stated that “In general, the technical issues, although difficult at times, were not the determining factor. Disciplined execution was.”

Similarly, Robertson and Robertson (2005) also mentioned in their book: “In several decades of project experience, we have never seen a project fail for technical reasons. It has always been human failures that have caused otherwise good projects to grind to a halt”.

These findings and many similar reports are often found in various resources and literature of software development and have helped bring the existence of fundamental problems in the management of software projects to attention.

The motivation for choosing to examine how using effective project management in the software development process can affect software quality for this master thesis came from a genuine interest and investment in the field and a desire to investigating the problems visible in the software development industry. Different approaches for evaluating project management effectiveness were examined to guide the selection of a suitable practical approach and research across different resources was conducted on the topic. The goal is to confirm the correlation between the application of effective management and quality improvement of the software developed and making use of the results in advocating for a better application of effective project management principles rather than just focusing on deliverables and technicalities.

2 Objectives and Methodology

2.1 Objectives

The thesis aims to examine the various software development processes and how effective project management techniques can contribute to the quality of the developed software system by:

- Examining various software project management techniques.
- Examining the process of software development and software quality metrics.
- Identifying the meaning of project management effectiveness.
- Identifying suitable approaches for measuring software project management effectiveness in software development.
- Evaluating the meaning and applicability of the proposed software development management approach.

2.2 Methodology

The theoretical part is based on the collection and analysis of the different methods and approaches to software development and their intersection with standard project management procedures with a focus on software quality metrics.
The practical part will focus on evaluating the effectiveness of management on the software development and how it reflects on the quality of the produced software by conducting survey studies on software development practitioners from various development stages (project managers, executive managers, team leads and developers).

3 Literature Review

3.1 Software Projects

Project is a widely used term across most contemporary businesses and scientific endeavors. It can describe any sequence of events or actions, executed individually or collaboratively to achieve a certain goal. There are also various other definitions of projects in the literature that goes beyond the meaning of the word itself. One example of which is an early comprehensive definition was presented by Tuman (1988) in his book Development and Implementation of Effective Project Management and Control Systems:
"A project is an organization of people dedicated to a specific purpose or objective. Projects generally involve large, expensive, unique, or high-risk undertakings that must be completed by a certain date, for a certain amount of money, within some expected level of performance. At a minimum, all projects need to have well-defined objectives and sufficient resources to carry out all the required tasks."
This definition is very helpful in distinguishing the differences between a project and other activities or tasks that may incorrectly be referred to as a "project". It defines the necessity of having a specific unique goal as well as potential risks and different constraints. From this, we can conclude that a project is a sequence of unique activities done to meet a strict goal within certain limitations such as time, money or resources.

To elaborate on this to this, we can also refer to the definitive project characteristics presented by (Pinto and Slevin, 1988, P.67-71) and later expanded upon by (Wysocki, 2006, P. 6-9) which can be summarized in the following characteristics:
- A specific, predefined goal agreed upon between the stakeholders.
- A sequence of relevant or interdependent activities each starts with input and ends with an output (either a deliverable or a component).
- A time frame defined by the specific start and end dates set to finalize each activity and for reaching the final goal.
- A limited budget or amount of resources.
- An acceptable level of quality that needs to be withheld for the activities and for the delivered end results.

These elements are often referred to as scope, process, schedule, cost, and deliverable quality. A successful project is a project that keeps these elements in balance.

The scope, time and cost constraints are often represented by what is referred to as a project triangle. In this representation, each side is a constraint that cannot be changed without affecting the other sides. Quality of the final project output is usually represented at the heart of that triangle as shown in Figure 1 below.

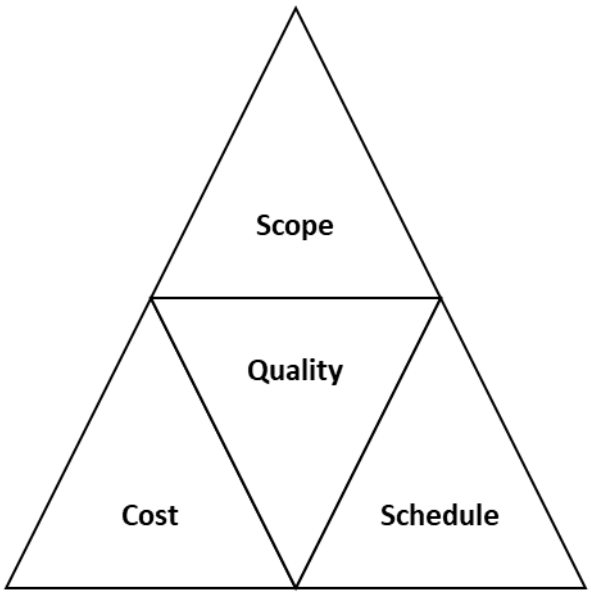

Figure 1 The Project Triangle

In his book, Effective Project Management Traditional, Agile, Extreme, Wysocki (2006) also defined the uniqueness of a project as the uniqueness of its activities as experience shows that each time a project or its activities would be repeated, something is bound to be different. The variations are usually random and can be difficult to predict which adds to the challenging nature of a project.

From these definitions, we can see that all these project characteristics also apply to software development endeavors without exception. Thus, we can consider all software development endeavors as projects with the distinction that the project emphasis, in this case, would be on software systems or a mix of software and hardware that work together and provide the functionality to end-users or to another system.

In conclusion, and for the purpose of this thesis, we can consider a software project is an undertaking that focuses on a software system and involves a series of interdependent activities to produce a specific, predefined output while adhering to the predefined limitations of cost and a defined schedule for delivering each component.

3.2 Project Management and Project Management Areas

Given the uniqueness and different constraints that can exist in each project, managing this complicated undertaking is a must. (Munns and Bjerimi,1996, P. 81-87) defined project management as "The process of controlling the achievement of the project objectives. Utilizing the existing organizational structures and resources, it seeks to manage the project by applying a collection of tools and techniques, without adversely disturbing the routine operation of the company". This can be considered an oversimplification of project management since the definition focused mainly on achieving the project goal without taking into consideration any other components or project characteristics. However, it does emphasize the importance of monitoring the project activities and using different techniques and the available resources to achieve the goal without impacting other elements outside its scope which is a core principle of project management.
In their book, Munns and Bjerimi (1996) also pointed out a significant distinction between a project and project management saying that while the scope of a project is long-term, the scope of project management is considered short-term. Project management is about planning and control. The main focus is delivering the requirements on time, not exceeding the allocated budget, and achieving the goal of each phase. All these points can be applied to each stage in the life cycle of a project independently and with variations and thus can be considered as short-term goals. In addition to that, project management tasks are usually either terminated or significantly reduced after the project's output is delivered with other aspects such as financial benefits or technical support may still carry on and these tend to be long-term in nature.

Another important distinction to make is that a project can simply be considered a reflection of its project management functions and those project management functions are unique. They each have various activities and entities as inputs to produce results while being limited with time constraints or a certain amount of resources. Activities represent the different processes such as gathering requirements or task assignments while an entity is an element that has a separate existence and can be a resource or a stakeholder.
The ideal goal of project management is to find the right and efficient combination of necessary activities and entities to reach the desired outcome. Identification of activities and entities and their arrangement in an appropriate order while dealing with different constraints is how management bring the project to life and reach the desired outcome.

Project management processes were described by the Project Management Institute in their standardized guidelines known as the Project Management Body of Knowledge (PMBOK®) (PMI, 2004) in terms of their integrative nature and the interactions between them. The numerous project management processes were grouped into the five main groups that clearly define their functions. (Wysocki, 2006, P. 27- 30) also proposed categorizing project processes into five groups that were similar and they are:

1. **Initiating Process Group:** Defining the project goals and allocating the resources required for its execution.
2. **Planning Process Group:** Defining the project activities and establishing the constraints and boundaries for each in terms of time and other resources.
3. **Execution Process Group:** Executing the activities and implementing the work.
4. **Monitoring Process Group:** Monitoring the progress of work.
5. **Closing of a project:** Managing risk, adjusting deviations from the plan until the goal is achieved and accepted.

In addition to these project management processes group, the guide (PMI, 2004) also presented the project management knowledge areas. Each of the processes, along with its inputs and outputs fall under one of those areas. To guide the evaluation of project management effectiveness and to later evaluate its impact it is essential to define the core areas of project management and the processes that fall under each of them. The third edition of the Project Management Body of Knowledge (PMI, 2004, P.11) categorized all the aforementioned project management processes and subprocesses under nine areas of project management. A brief description of each is listed below and the processes under each area are represented in Figure 2.

1- **Project Integration Management:** An area covering the preliminary analysis of the project and defining its initial scope statement and the project plan.

2- **Project Scope Management:** This area ensures the project has a clearly defined and well-planned scope that is both achievable and controllable.

3- **Project Time Management:** An area concerned with completing the project on time as well as finding the proper sequence and schedule for their execution.

4- **Project Cost Management:** An area responsible for budgeting and keeping track of the project costs to make sure it doesn’t exceed it.

5- **Project Quality Management:** An area concerned with achieving the project goals while satisfying the standards expected from all stakeholders.
6- **Project Human resources Management:** This area describes the processes of organizing and managing the project team to ensure they are well aligned and committed to the project goal while tracking their performance.
7- **Project Communication Management:** This area ensures that all project information is collected and distributed in an efficient manner. It is concerned with exchanging updates and feedback to guide the project and its team.
8- **Project Risk Management:** This describes the processes concerned with reducing the risks included in all projects due to their uniqueness and involves subprocesses such as risk analysis and risk response planning.
9- **Project Procurement Management:** This area describes the process of acquiring the product and contractual processes.

The PMBOK went on to explain each area and its processes in details but for the purposes of this thesis, they will not be covered with such a lengthy explanation. Instead, with taking into consideration that software project management refers to project management when the project focus is on software or a mix of software and hardware, a simpler project management framework was adopted to accommodate the aforementioned nine areas of project management into fewer more generalized areas and to identify a boundary in which different project outcomes can be analyzed and evaluated to determine whether they were successful or not.

Similar simplification approaches have also been proposed as an attempt to mitigate the complexity of project management by selecting what can be considered the core areas of project management. For the purposes of this thesis, the framework chosen focused on four main areas of project management: Project Processes, Output Quality, Human Resources, and Risk management. These were selected after thorough consideration and research into the different project management knowledge areas discussed. The goal was not to omit any important areas but rather try to combine them into categories that can be measured and evaluated.

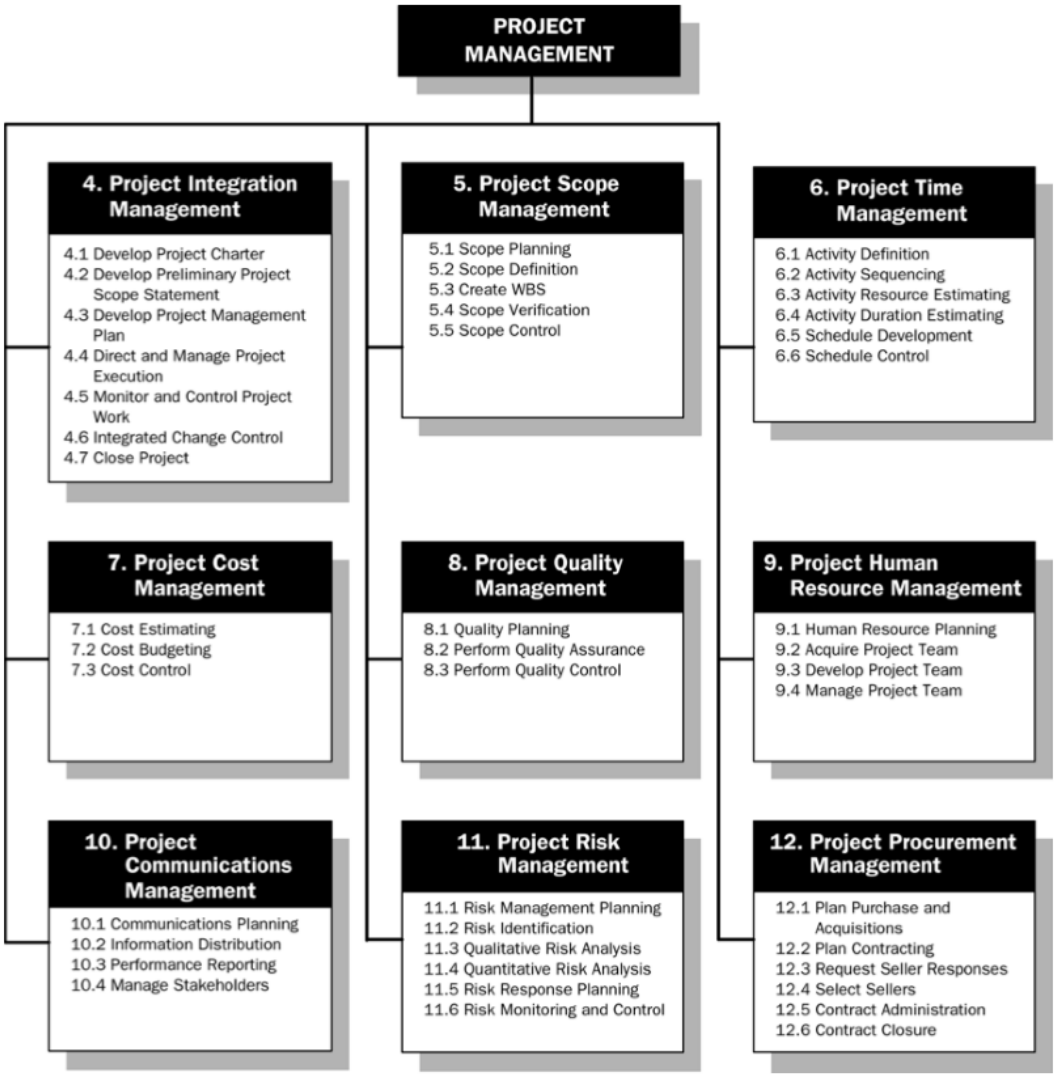

Figure 2 Overview of Project Management Areas and Project Management Processes [From (PMI, 2004 P.11)]

3.2.1 Project Processes

The project process is the first area of project management to be included in the proposed framework. “Process” is an umbrella term that covers project initiation, project planning and project controlling which are all essential parts of project management. Without defined processes, gathering a group of software practitioners and expecting them to just work effectively towards a vague goal is highly unlikely. Understanding how the project will be executed and what is the expected output is vital in any project as it provides a roadmap for all the stakeholders, increases productivity, reduces errors and helps them navigate smoothly towards achieving the goals of the project.

The process area of a software development project focuses on various key aspects of project management. It involves determining the project's goal and defining its restrictions whether it is a timeline or a budget. The process also includes planning and breaking the goal of the project down into smaller manageable components that will then be developed following the same processes and would adhere to the same limitations.
The research done in the thesis shows how having clearly defined processes in a project upfront can affect its results and covered aspects related to:

- Requirements management.
- Allocating resources and a budget.
- Scheduling project phases.
- Monitoring and control

It is important to point out here that requirement management is a critical area for software development project management to focus on. Its goal is to ensure that the stakeholder understands the project's goal on a deeper level and are fully aware of what the expected output should be. However, project requirements are not always static or crystal clear at the beginning of a project. Requirements can evolve or change as the project passes throw different phases and the stakeholders' expectations from the project may change. Gathering and monitoring the requirements of the client or end-user of a software product is a shared responsibility between both the project manager and the engineering team because while their roles are interdependent, they each have a different perspective when it comes to requirements management. The project manager, on one hand, is process-oriented as his role is more concerned with planning and controlling the project and its resources. On the other hand, the system engineer is typically more focused on the software being developed from the architectural and functional perspective. The intersection between both roles is what drives the requirements management process and later guides the processes of allocating resources and scheduling different project phases. A representation of the requirement management intersection is shown below in Figure 3

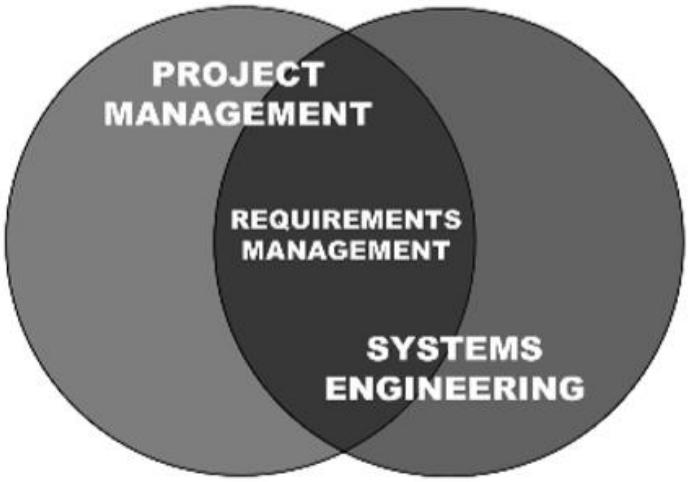

Figure 3 Requirement Management [From (Forsberg, Mooz, and Cotterman, 2005, P. 34)]

3.2.2 Stakeholders

The next area of project management the framework focused on was the stakeholders. Software projects are developed by people to help people and the development process is human-intensive compared to other industries. Hence, the importance of properly managing the stakeholders was clear from the start. The term "stakeholders" covers all parties that have an interest in the project and who are impacted by the outcome of the project. This includes the project team consisting of the project manager or project leads, the development team as well as the client and/ or end-user representatives. For the project to really flourish, all stakeholders need to be aligned towards a common goal and acknowledge their interdependency as well as their different perspectives.

Due to the temporary nature of projects, they often bring together people from different backgrounds who are not familiar with each other and who might have different objectives for the project. That is why it is important for the project leads to carefully select the project team not only based on their qualifications but also on their interpersonal skills. The project leader is also expected to monitor the team's performance and track the overall progress of the project as well as offer support to the stakeholders when needed. That is why it is important to include all activities related to the management of the stakeholders in the planning process. A project's plan must allocate time for different communication activities with both the team members and other stakeholders. The research done in this thesis investigates the impact of the stakeholders on software project management and covers aspects such as:

- Communication effectiveness.
- The impact of the project leadership.
- Team performance.
- Stakeholders' involvement.

The goal of this area of project management is to create a balanced environment for the project where feedback and improvement would be encouraged and where all stakeholders feel included and motivated. The study and analysis were done in this thesis and presented in later sections also confirmed that the stakeholders in project management are of high importance in the software development field.

3.2.3 Output quality

With software systems as a driving force in our life that are widely used on daily basis either directly or indirectly, the software community continually attempts to develop techniques that will make it easier, faster and less expensive to build software. While functional quality is one of the most important characteristics of the software as a product, there are several other non-functional attributes that also reflect the quality of software known as qualitative attributes. The software product, however, is not just lines of codes but rather includes the software and its wireframes, clear documentation, reusable components and extended services such as maintenance and other support functions. Paying attention to the quality of those elements ensures the client's satisfaction. That is why it was important to take the quality aspects of the project's output into consideration as both parts of the requirements and end results when managing the development of a software project.

As the stakeholders of a project come together to an agreement about the project's scope and processes, they should also reach an agreement on what the product quality attributes would be. It is expected that as different stakeholders have different objectives from the project, they could as well have different views on the quality metrics the project needs to adhere to. For example, to the end-user, the quality of the software would be represented by its usability and availability while for the software practitioners and developers it would be factors such as testability and efficiency. These attributes reflect the behaviors of the software during its execution and the organization of its source program. Software quality management provide independent monitoring of the software and the development process and it is divided into three main activities:

1. **Quality Assurance:**

It is the process of defining how software quality will be achieved. The main goal of quality assurance is to select the standards that will be applied to the software and the software development process. International organizations such as the IEEE have developed standards for the software projects that cover aspects of the software standards and the process standards. These standards are customizable and are agreed upon by the project's quality assurance team.

Product standards	**Process standards**
Requirements document structure	Project plan approval process
Method header format	Version release process
Java programming style	Change control process
Change request form	Test recording process

Table 1 Product and Process Standards

2. **Quality planning**

The quality planning process aims at defining the quality requirements and attributes of the software itself and describes how these requirements will be evaluated. According to Blum (1992), there are two views of quality: internal and external. While internal quality is the developer's view of the software, external quality is the stakeholders' view of the software. External quality includes usability, reliability, maintainability, etc. Internal quality includes, but is not limited to, efficiency, testability, and reusability.

The specific attributes expected from a software system depends on its application and there are different attributes listed in various literature but here are some of the essential qualitative attributes selected for the purposes of this thesis:

For the purposes of this thesis, we choose a few of the various quality metrics of software to focus on and analyze. These quality metrics will be further examined later in the thesis and in the practical part as well and they are:

- **Functional suitability:**

It defines the degree to which the developed software provides functionality that meets the users' needs and objectives.

- **Reliability and maintainability:**

Reliability expresses the degree to which the software is operational when needed and is represented often as the mean time between failures. Recoverability, on the other hand, represents how well the software system can recover from a sudden interruption and how easy it is to make changes whether to correct a failure or adjust functionality.

- **Usability:**

Usability represents the ability of the user to quickly understand and utilize the developed software. The more user-friendly and clearly organized the software functionalities the better.

- **Testability:**

This is the measure of how easy it is to test the components of developed software to validate the quality criteria and requirements are met. If the software is testable, there is a better chance of finding faults and fixing them in a timely manner.

- **Performance:**

This represents the time behavior of the developed software such as the response and processing time of its functions as well as the software's resource utilization levels. Resources may include other software components or hardware capabilities.

3. Quality control

The quality control process is the monitoring of the software development to make sure that the quality assurance standards are being followed and that the deliverables meet the requirements. In every project, it's essential that the stakeholders should come to a common understanding of what the product's quality should be and how it will be evaluated. For example, a quality attribute such as usability may mean different things for the developers and the users. Thus, it is essential to define what usable means as early as possible in the project development. The earlier this common understanding is reached the better it is. It is important to note that quality is not a feature that can be included later in the product. It should be integral to the whole software development process.

The research done in the thesis focused on the quality engineering processes to show the relationship between the areas of the project management and how they reflect on the quality of the software developed as the main indicator of a project's success.

3.2.4 Risk Management

The uniqueness of a project is one of its inherent characteristics, and it is due to this uniqueness that a certain amount of uncertainty would exist in any project. As identified in the Project Management Body of Knowledge (PMI, 2004, P.8) project risks are the uncertain events that might occur and cause a negative impact on one or more of the project objectives. Thus, every project manager and development team should take these uncertainties into consideration when planning. The project team should perform risk management activities such as anticipate potential risks, investigate these risks and figure out how to turn them into opportunities or how these risks can be mitigated if they happen. However, the reality is that often this aspect of project management often gets neglected or skipped due to lack of awareness about its importance or lack of time to carry out the risk management activities properly. (Boehm, 1991, P 32-41) points out that in most software crises, the disasters could have been avoided or significantly mitigated if potential risks were identified and resolved in the early stages of the project. Risk management activities involve two primary steps:

1. **Risk Assessment**

This means identifying the areas of potential risk, analyzing their probability and evaluating their possible impact.

2. **Risk Control**

Based on the activities of risk assessment, the various risks identified can be prioritized. They are addressed accordingly when developing countering plans, risk management planning, risk resolution, and risk monitoring.

In conclusion, project management is a large-scale process that covers different aspects of the project execution to ensure a successful outcome that satisfies all stakeholders. In software projects, the management operations are even more complex as they have to cover both technical and non-technical aspects in addition to that as well as account for technology and market changes. This section examined the various aspects of project management and briefly introduced the main areas that this thesis will focus on later in the practical part, as well as identified what project management means to the software development field.

3.3 The Relationship Between Project Success and Project Management

After examining the project management processes and how they apply to the software development process and before a further investigation into how it can affect the outcome of a project, it was important to first fully understand what it means for a project to be successful in general and what success means in the software development field.

Project success is a difficult term to pinpoint as it can be relative as well as dependent on various factors. An early attempt to identify critical project success factors was conducted by (Pinto and Slevin, 1987) where they identified ten critical factors that can be used to predict the success or failure of a project. They choose ten main factors included having a clear project mission, top management support, scheduling the project properly, involving the client as well as having adequate communication and feedback. Some of those factors were also mentioned in different theoretical researches and some were gathered based on the study they conducted on various software projects. Based on this, they formulated their results into a model which defined the relationship between project success (S) and its critical success factors (Xi):

$$S= f (X1, X2……, Xn)$$

While the approach was effective, it only identified the success factors but did not take into consideration whether they have different impact levels or the strength of the correlation between these factors and project success. In the following year, (Pinto and Slevin, 1988, P. 67-71) also presented an early developed model representing project success from their perspectives. It categorized the different success factors into two groups; one showing the perspectives of the clients indicating their interest in success factors relating to quality metrics such as the usability, end-user satisfaction and the effectiveness of the project process. While the group shows the perspective of the project team and their interest in factors such as completing the project on time and within budget and its performance levels. This model was represented as shown below in Figure 4.

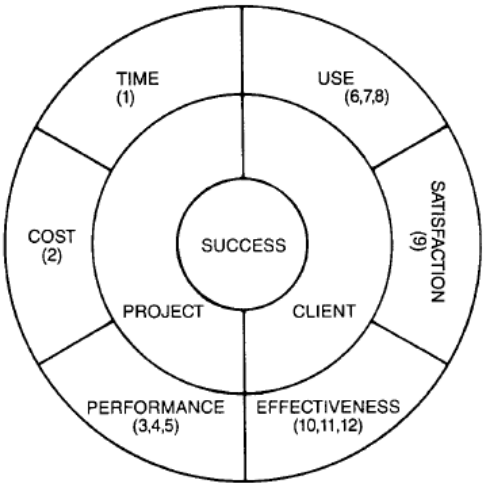

Figure 4 Project Success Model [From (Pinto and Slevin, 1988, P.69)]

There were many other attempts over the years in the literature on project management to identify critical factors of project success or successful project implementation. Each of them presented their own framework and produced a list of critical success factors, but this principle was once again emphasized (Linberg, 1999, P.177-192) in a study on software practitioners' perception of software project success where he pointed out that the definition of project success is different between stakeholders. Often software practitioners would consider a project successful even when it exceeded the predetermined timeframe or budget because they are more concerned with the quality and functionality they deliver. That is why among the key challenges of project management is identifying the project goals and aligning the stakeholders towards them while making sure to address their different concerns. If achieved correctly, this could be the key to project success.

It is also important here to point out that while a project's success is often considered as a reflection of its project management, it is necessary to understand that project management success is different than project success. Even though the terms project and project management often generally overlap and the goal of both is achieving project success;
there's an important distinction between them that we mentioned earlier which is their scope. The scope of a project is long-term benefits that go beyond the handover of a project while the scope of project management is generally more short-term and focused on the planning and development phases. Project management is more concerned with factors such as on-time delivery, meeting performance standards, project development within budget expectations, etc. The variation in what constitutes success in the perspective

of different stakeholders and the distinction between the long-term scope of project success versus the relatively short-term scope of project management was best represented (Munns and Bjeirmi, 1996, P.81) as shown in Figure 5 which shows how the scope of project management success is like the scope of project management itself and it is short term and focused on the early stages of the project life cycle. It also shows that the scope of project management success is a part of the entire project's success scope. The figure also represents the different stakeholders interested in each stage of the project life cycle. Considering this, we can say that project management success is a part of project success, yet it is possible to achieve project success despite project management failure.

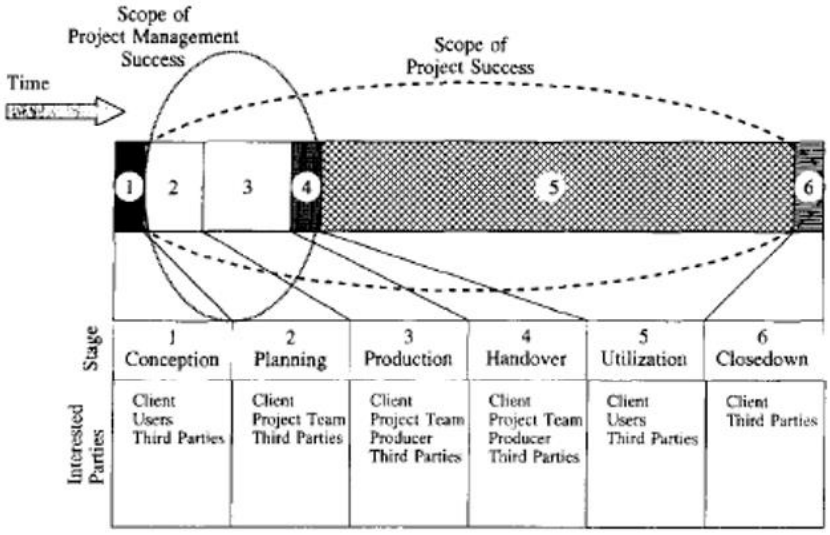

Figure 5 Stakeholders Interested in Each Stage of The Project [From (Munns and Bjerimi, 1996, P.81)]

For the purposes of this thesis, the definition of project success was initially built on the basis of the project triangle (Figure 1) which means that for a project to be considered successful, it needs to keep the balance between its schedule, budget, and desired scope; but it is important not to sacrifice quality to achieve this. However, with the rise in competitiveness in the software field, came the need for a deeper definition of project success as the three conventional criteria of project management (time, cost, scope) were no longer enough to maintain a competitive edge in the market. Maintaining the stakeholders' satisfaction and supporting the team members as the human resources of the project became the key to the success of software development companies as a business. A similar approach to project success being reliant on five factors rather than just three was also presented by (Wiegers, 1996) and the balance between them is shown in Figure 6.

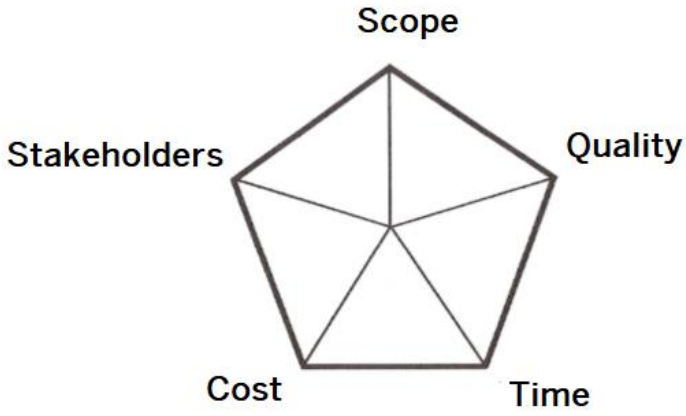

Figure 6 Five Dimensions of Project Success [From (Wiegers,1996, P.145-150)]

The five dimensions of project success he chooses were:

1- **Scope:** Did the project achieve its intended result? To have a real measure of the project's success, we must determine if the objectives of the project agreed upon between the stakeholders were achieved.
2- **Quality:** As discussed earlier, quality, in general, is a critical factor and the most discussed aspect of any product especially comes to software development where the goal of a project is not to just deliver an output but to also deliver consistent value to its users and quality assurance processes often extend beyond project delivery.
3- **Time:** Was the project delivered on time? This is an obvious criterion for success, however, not all projects manage to achieve it either due to poor planning or unexpected problems. Completing the project's milestones within their scheduled timeframe is critical for calling a project a success.
4- **Cost:** Was the project delivered within budget? Just like the time criterion, this is an essential indicator of success.
5- **Stakeholders satisfaction:** Both team and client satisfaction are essential to survive the tremendous competition of the modern world. To evaluate if a project was a success or failure, you must evaluate the team's motivation and commitment as well as the client's satisfaction with the overall process and progress.

We can notice from this model that there are similarities between Wiegers's five dimensions of success and the four main project management areas that were proposed earlier. However, in the framework proposed for this thesis, the project's scope, time and

cost can be included under the umbrella of Project processes. Quality and the stakeholders' satisfaction were represented as well. Along with these factors, the framework included the project risks as a contributor to the success of the project as discussed previously.

Ultimately, to judge a project as success or failure, we need to evaluate if the project met its goal within the predefined limitations, but we also should not neglect the produced quality and the satisfaction of all parties involved if we want to ensure sustainable success.

One last concept that is important to highlight in the discussion of project success is that it should not be considered strictly as a binary measure. There are always opportunities for learning and improving even when a project is considered a failure. In the software development field especially, a failed development experience should not be discarded in its entirety. Parts or modules of the software that was being developed can be repurposed or built upon in later projects. The failed project processes themselves should be studied. The closer a project is evaluated, the more the team would learn from it thus increasing their chances of success in subsequent endeavors.

3.4 Software Development Models

"Software project management is considered both an art and science of planning and leading software projects." (Stellman, 2005, P. ix). As discussed in previous chapters of this thesis, software project management requires knowledge of project management processes itself as well as knowledge of software development techniques and technologies. The Software development process often referred to as the software development life-cycle is a framework composed of a number of clearly distinct phases for designing, creating and maintaining software. The development phases in a development life cycle shown in Figure 7 represent the five fundamental activities that are common to all software endeavors:

1- **Requirement gathering**

   During this phase, the development team along with the rest of the stakeholders define the functional and nonfunctional requirements of the software that will be developed. This could include activities such as interviews and requirement workshops with the end-users as well as thorough documentation.

2- **Software design**

In this phase the project features and functionalities are detailed, and the system modules are described. This is also known as the system architecture as it defines how the system will be built using different components. It includes activities such as creating a work breakdown structure of the system and assigning resources to each of them. The software architecture phase focuses heavily on the specifications of the subsystems in terms of functionality and interface design.

3- **Software development and implementation**

The phase where the development activities and implementation of the software are carried out. In this phase, the requirement and design are converted by software developers into data structures, algorithms and software components.

4- **Software validation**

This phase is intended to test the software to make sure it conforms to its specifications and meets the requirements of the client and/ or end-users. In this phase, the developed components are tested independently before they are integrated together and tested again as a whole. The software validation confirms that the system meets both the functional and non-functional requirements.

5- **Software evaluation**

The aim of this phase is examining the system and the development process overall to ensure it was successful and maintain its reliability. The need to implement changes or perform maintenance may arise during this phase to help the software system stay relevant given the changes in the user or market requirements. This can be referred to as system evolution.

It is important here to notice the similarity between these phases of the software development life- cycle and five project management process groups discussed earlier. These fundamental activities may be labeled or described at different levels of detail for different software projects.

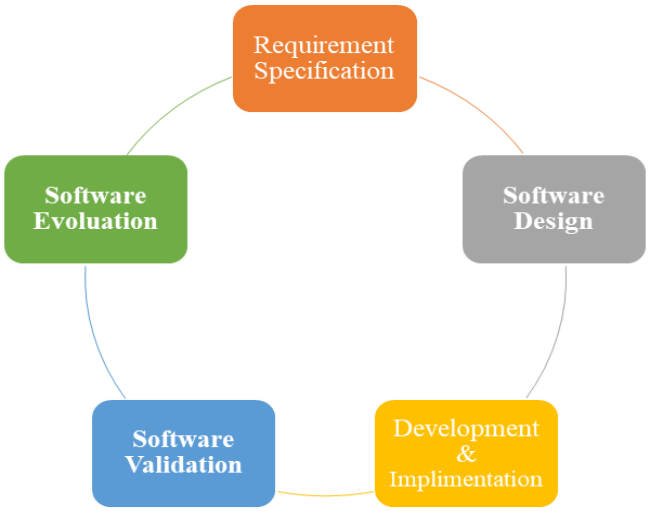

Figure 7 Software Development Life Cycle

The execution sequence and arrangement of these activities also can and often vary between projects based on the type of software being developed. This is known as the Software Process Model or Software Development Model. The software development model is a simplified framework representing the software development processes and how they are executed. Choosing an inappropriate software model may reduce the quality of the software produced and may increase the development costs. In this chapter of the thesis, different software development models were examined, and the traditional models were compared against the newer agile methods. Most software process models are based on one of these general models or a mixture between them. There are several models that explain different software development approaches, and each has its own project management methodology but, in this section, we examine briefly the most common models:

3.4.1 The waterfall models

The waterfall approach is one of the sequential processes in which progress is seen as flowing steadily downwards through the phases of software development. It was one of the earliest models introduced where development only goes on to the following phase when the preceding phase is finished, reviewed and verified. It is a traditional model that follows strict, well-defined steps to complete a project where each phase results are passed on to the next and the following phase can't start until the previous is completed. In practice, these stages can overlap and there's often a feedback stream between the phases, however, the method is rigid and requirement-focused which means you must have a clear idea of the software project requirements as it would be difficult to go back and make changes.

However, this also can count as an advantage as it could prompt the team to produce excellent output most of the time and ensure the steady progress of the project. An example of the waterfall model is shown in Figure 8.

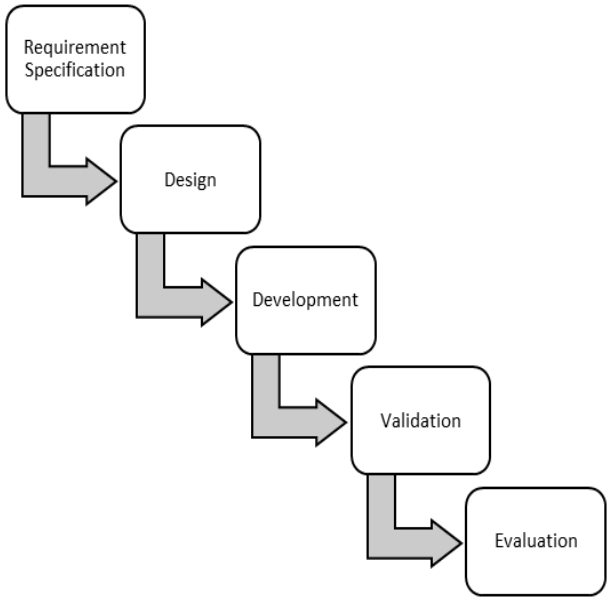

Figure 8 Waterfall SDLC

3.4.2 Evolutionary models

The evolutionary approach is an iterative development process. The basic idea of this model is to develop a system through repeated cycles in smaller portions at a time. The initial software is developed quickly based on the abstract specifications provided to create a product to which the user can react. Then during each iteration, the system is modified and improved further based on the feedback of the client or the end-user. The design modifications are made by creating a new prototype until satisfying software if fully developed or adding new functional capabilities. This approach interleaves the activities of specification, development, and validation.

The initial system is then refined with customer input to produce a system that satisfies the customer's needs. This method is often faster and more effective than the waterfall approach, especially if the requirements are not clear or are subject to changes since the software requirements can be developed incrementally but if not applied carefully, the continuous changes could corrupt the software design and structure.

3.4.3 Spiral model

It is an approach that combines the advantages of the waterfall and the evolutionary models as it allows for requirement changes but does not sacrifice the software structure or maintainability. The spiral model depends on multiple process iterations where each loop of the model represents a phase of the software development process. The phases are executed in sequence with feedback from one activity to the next. The inner loop could, in this case, be the core software requirements and the next would be software design and so on. Figure 9 shows the original diagram of the spiral development as presented in (Boehm, 1988, P. 62-72).

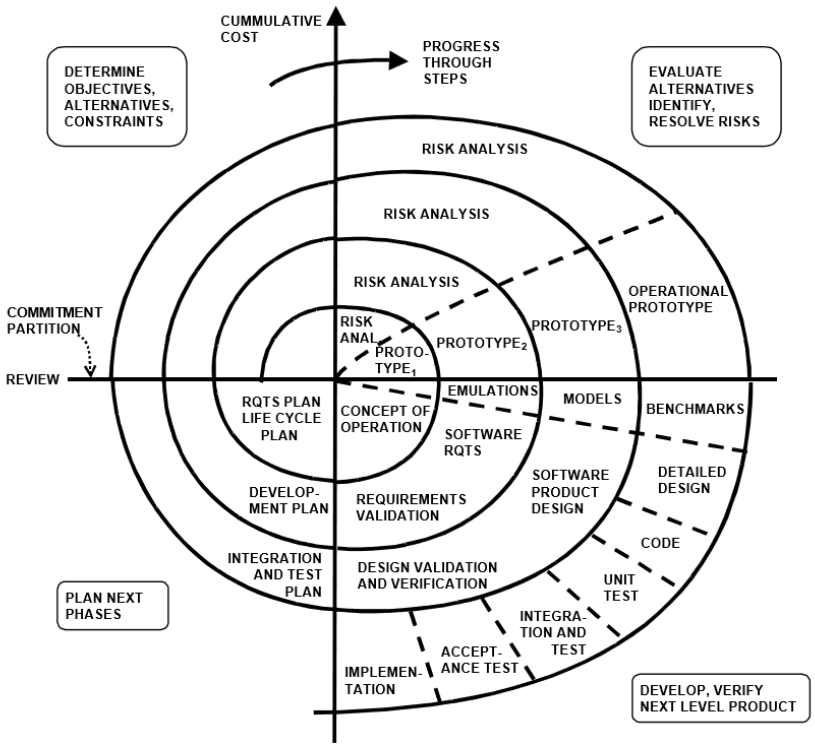

Figure 9 Diagram of Spiral SDLC [Boehm. B. W (1988), (P.2)]

3.4.4 Agile models

Since now most software companies must adapt to the rapidly changing needs of the market, a swift development process is essential. The agile model is a combination of the iterative and incremental process that was presented to allow the development team to focus on the software itself as well as customer satisfaction. It breaks the software into small incremental builds that evolve through collaboration between the team and the stakeholders to deliver useful software to the customer quickly.

There are many development methods that fall under the agile category such as:

1- Extreme Programing

Extreme programming is a methodology aimed at improving software quality and responsiveness to meet the client's requirements and include doing extensive code reviews and unit testing of all codes which provides simplicity and clarity. This model aims to reduce the cost of system evolution as well as the cost of changes in client requirements. It is based on five basic values which are, Robust Communication, Simplicity, Continuous feedback, Courage, and Respect amongst stakeholders.

2- Scrum

Scrum is an agile software development framework that is based on the iterative evolutionary approach. It focuses on maximizing the team's abilities and enables them to self-organize by encouraging daily efficient face to face communication to deliver and quickly respond to new requirements. The scrum framework consists of a scrum team that each has a specific role, scrum events, and scrum artifacts. Each component serves a purpose and is essential to the success of each sprint or phase. Figure 10 shows an example of the Scrum Agile Model. It shows the scrum artifacts and events where the requirements are the product backlogs, the sprint is the time-defined cycle to complete a sprint backlog (an increment).

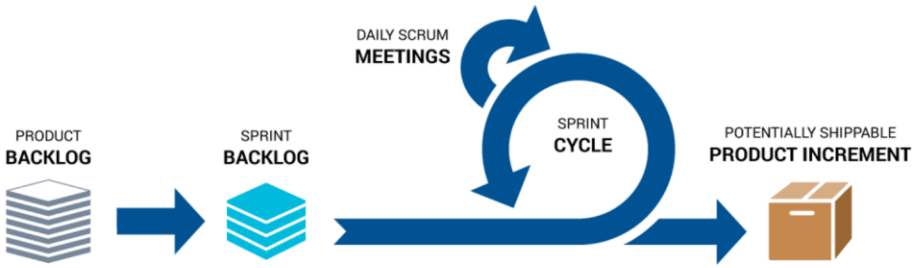

Figure 10 Scrum Agile Model

As one of the most trending models nowadays, the scrum focuses on enhancing the team's ability to deliver the software faster by removing any obstacles and maximizing feedback opportunities. It is worth noting that in applying these methods, producing system documentation is not cost-effective and is usually completed during later stages so it becomes relatively difficult for management to assess the development progress and it also limits the usability of the components. Scrum models also usually require special skills for the team and a higher level of knowledge.

3.5 Evaluating Project Management Effectiveness in Software Development

Evaluating software quality and the project management activities carried out during its life cycle is key for improving both the development process and the outcome in future projects. Defining software quality metrics help with the evaluation process as it allows us to compare these values to previous projects or to the predefined standards to know whether a project was a success or failure. Multiple approaches to measuring the effectiveness of software project management were developed over the years.

The Quality Management Metric (QMM) developed by Machniak (1999) is considered one of the most notable approaches to doing so. The Quality Management Metric focuses on evaluating four important areas of project development which were requirements management, planning management, people management, and risk management. In the software development field, those areas are investigated through conducting surveys on various software practitioners then performing a quantitative analysis of the project. The results can then be used for improving future software developments. This approach achieved remarkable success in measuring the quality of project management in the software development field when used; mainly on projects that were commissioned by the Department of Defense in the United States of America.

For the purposes of this thesis, a similar approach was chosen and adjusted to be more focused on the software development models used in business and commercial software environments. In this approach, measuring the success of a software project can be considered a part of the quality control process or as an independent process based on the model being used for development. The evaluation is carried out according to the following steps:

1- Selecting the attributes to be evaluated based on their relevance to the quality assurance standards.
2- Selecting the tools used for the assessment.
3- Measure the selected attributes and analyzing the measurements.
4- Examine the project results to define the overall quality of the outcome.
5- Evaluating the contribution of the selected attributes to the project’s outcome.

The attributes chosen to evaluate were driven from the project management aspects that were discussed earlier which are the project processes, the stakeholders, the quality of the output and the Risk management activities. Generally speaking, there are two main methods of measurement: direct or indirect and the key difference between these methods is whether we obtain the measurement with or without the necessity of measuring other qualities. Examples of direct measurement in software engineering are source lines of code, duration of an activity such as testing, number of defects, and effort in a number of man-hours or man-months. These are all clear direct measurements with specific units that can be determined quantitively. On the other hand, indirect measurement in software engineering is measurements such as productivity defined by the number of lines of code over the effort in manhours, requirements stability defined as the number of requirements at the start over the total number of requirements at the time of measurement, etc. These all measurements that require two different direct measurements to calculate. Since our goal is to measure the effectiveness of project management in a software project, the complexity of the concept required a combination of both. This helps us measure certain aspects of activities and entities working on the project.

Demir (2008) has identified four different approaches for developing methods to measure the effectiveness of software project management which are shown in Figure 11 along with their corresponding metric types.

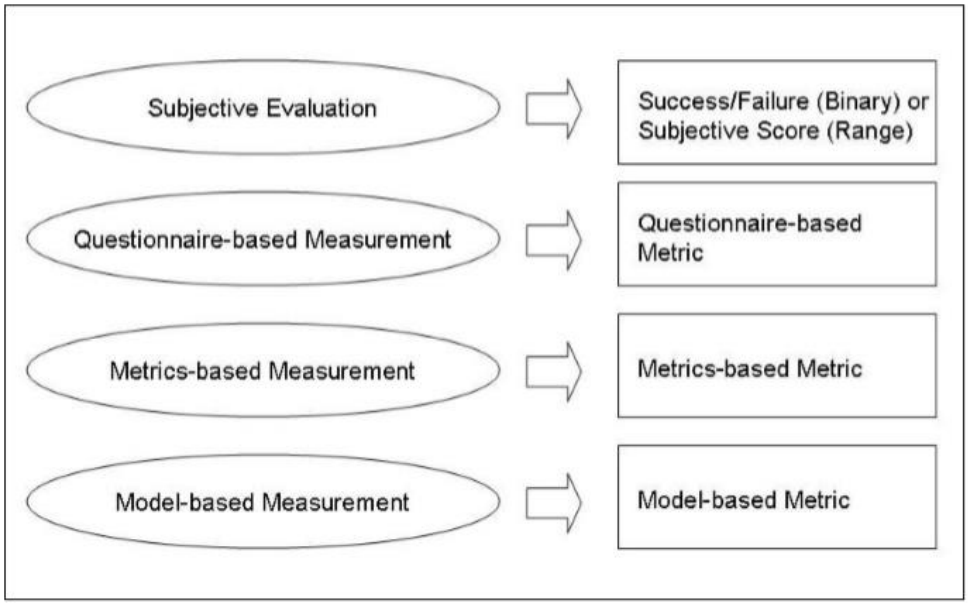

Figure 11 Four Approaches for Software Project Management Effectiveness Measurement From [Demir (2008) (P. 45)]

To choose the appropriate method for evaluating projects and validating my hypothesis about the relationship between project management and the quality of the software produced, all four approaches were examined. Two of which were found suitable for the purposes of this thesis and they are the subjective evaluation method and the questionnaire-based method.

The questionnaire-based approach was chosen in this thesis for gathering data from a number of software development practitioners spanning small to large development projects to evaluate their projects' success and the project management processes applied. While the subjective evaluation approach was chosen to evaluate the results of each project based on the responses given by the software practitioners. The results from both measurements were then compared and analyzed.

3.5.1 Subjective evaluation method

In the subjective evaluation method, the stakeholder's perception of the project's success or failure is used to evaluate the project's success and the effectiveness of the project management. However, since this method is based on the perception of the stakeholder, it is very subjective to their opinions and their objectives from the project. In this method, the person is asked to categorize the project on a binary scale as either a success or a failure. It is a very simple approach, but it is sufficient since each of the stakeholder's opinions of the project is a factor that we cannot disregards. However, we must be careful of the significant risk of mislabeling a project as a success or failure if we do not first define a clear set of success criteria as Pinto and Slevin (1998, p. 357) pointed out.

3.5.2 Questionnaire-based method

In the questionnaire-based method, the evaluation is based on weighing responses to a set of coherent questions related to the topic. This is a common method that has been efficiently used before in studies related to the development of the quality management metric itself since it is suitable for measuring abstract concepts such as communication effectiveness, teamwork, commitment, etc. which are otherwise hard to quantify and evaluate.

In this thesis, the questionnaire-based measurement was implemented using a mix of question types; all of which have been utilized by other researchers in previous literature on the topic and have many examples of its applications. The project success evaluation

will also be judged based on the participant's subjective evaluation which was covered by my questionnaire. For the execution of this method, a member of the project team who has broad knowledge on all aspects of the project management fills out this questionnaire where they are asked about their experience with the project and to assess its success based on their perspectives. They will rate statements on different project management areas using a Likert linear scale (0 to 5 scale, with 0 being the lowest and 5 being the highest value). This type of question is one of the most common scales for weighing individual opinions. The Likert linear scale is easy to understand, and it produces clear and distributed results for analysis and evaluation later. The data gathered is then used to measure the effectiveness of the management. Responses to questions in the instrument are assigned with specific scores. The evaluation model simply combines these scores in a systematic way as it is hypothesized.

4 Practical Part

4.1 Survey Design

A project's success depends on various elements, but all projects need people to work on it. That's why I decided to conduct a survey study as a method of gathering information from the people who work on software projects. Various team members such as software developers, project managers as well as other stakeholders in major software development companies in Egypt as well as few companies here in Prague were asked to look back on their recent projects and evaluate their overall experiences. The purpose of the questionnaire was to validate the theory of my thesis about the correlation between a software project's success and its management processes by measuring the effect of the project management processes. It consisted of 26 questions and was developed using the web-based surveying tool Google Forms as a simple and quick way of creating and managing questionnaires. The survey was conducted in the English language and the questions which are presented in the table below were divided into 3 major sections which are:

1. **General questions about the participants' recent project and his role**

In this section, the goal was to have a general overview of the project and the participant's role in it. The first question was about their role in the project. The questions then went on to ask about the overall project's outcome from their perspective. The section included five questions that aimed to help me understand the participants' evaluation of the project and whether the project manager and developer would have a different judgment on a project's outcome. Four of the questions were multiple choices while the last question of this section was open-ended. This gives the participants a chance to provide their opinions and additional commentary on their answers regarding the project or its work environment in an open text format. This, in turn, will help me better understand their answers and get an initial idea about which of the various areas of project management they were more concerned with.

2. Output Quality

This section covered another area of software project management that was included in the framework which was the quality of the delivered software as the main output of software development projects. It included 5 linear scale questions about various quality metrics. When combined with the questions from the first section, they should provide a clear definition of which projects were successful and which were not.

3. Stakeholders, Processes, Risk management

This section is structured in the same way section two was and covered the other three main areas of project management discussed earlier which were: the stakeholders' impact, the project's processes, and the risk management activities. The results were then compared to the subjective evaluation results that were obtained from the previous sections of the survey to deduce the impact of each area on the project's outcome.

A full overview of the questionnaire structure and questions is presented in Table 2 below. It was sent to colleagues from different nationalities who are working in the software development field in companies based in Cairo, Egypt and other selected companies here Prague, Czech Republic. There was great cooperation from both sides and the responses were received in a timely manner. The questionnaire takes about 6 minutes to complete and yield 97 responses in total.

Table 2 Questionnaire structure

Section	Question	Type	Answer(S)
Section 1: General questions about the participants' recent project and his role			
1.1	What was your role in your past projects?	Multiple choices	• Project Manager • Requirement Engineer • Software Designer • Software Developer • Configuration Engineer • Software Tester • Software Maintenance
1.2	Did the project exceed its allocated resources (Time, Budget, manpower, etc.)?	Multiple choices	• Yes • No
1.3	If yes, please specify which aspect exceeded the initial estimated effort	Checkbox	• Duration • Number of team members • Functional requirements • budget

1.4	How would you rate the overall success of the project and the quality of the delivered output?	Likert linear scale	• Very successful • Successful • Neutral • Could've been better • Failure
1.5	Please elaborate on your answers and the variations where possible	Open-ended	
Section 2: Statements on the "Output quality" area of software project management			
2.1	The requirements were met in the delivered output and it worked as intended.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
2.2	Users found the delivered system easy to use and understood its functionality.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
2.3	The delivered system is stable and would be able to recover fully from any downtime.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
2.4	The delivered system consisted of solid, thoroughly tested modules.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
2.5	You are satisfied with the quality of the delivered output.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
Section 3. Statements on the "Stakeholders", "Project processes", "Risk management" areas of software project management			
3.1	The project manager was supportive and accessible to the project team and stakeholders in a timely manner.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.2	The project manager understands the project risks and was a proactive problem solver.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.4	The project suffers from a lack of experienced or qualified team members	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.5	Task assignments were fair and based on the qualifications of the team members.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.6	All communications activities were streamlined and accounted for in the planning.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.7	The project team members lack motivation due to various reasons including external factors.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.8	You would move to another team because yours didn't provide support or guidance during the project.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree

3.9	All stakeholders were involved in planning, requirement gathering and other stages of the project cycle.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.10	There were regular and impactful meetings to assess the team's performance and project status.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.11	Requirements are stable and clear during the project development	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.12	The projects risks were assessed and documented during the planning phase	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree
3.13	Risks and their potential impact on the project were communicated to the stakeholders	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree • NA
3.14	Potential project problems were addressed proactively and in a timely manner.	Likert linear scale	• 1- Strongly Disagree • 5- Strongly Agree • NA
3.15	Which of the following areas were the most challenging during the project's life cycle? (Check all that apply.)	Checkbox	• Team Leadership • Team communication, • Team commitment • Task assignment • Client Involvement • Requirements Management • Project Planning • Quality management • Risk Assessment • Other (Support Activities, Tools, Training, etc.)
3.16	Which of the following were clearly communicated and went smoothly during the project's life cycle? (Check all that apply.)	Checkbox	• Team Leadership • Team communication, • Team commitment • Task assignment • Client Involvement • Requirements Management • Project Planning • Quality management • Risk Assessment • Other (Support Activities, Tools, Training, etc.)

4.2 Data Analysis

4.2.1 A General Overview

From the first question of the section, we can better understand their perspectives and how their role makes an impact on the project. All the 97 respondents provided an answer to this question. Figure 12 shows a representation of their responses.

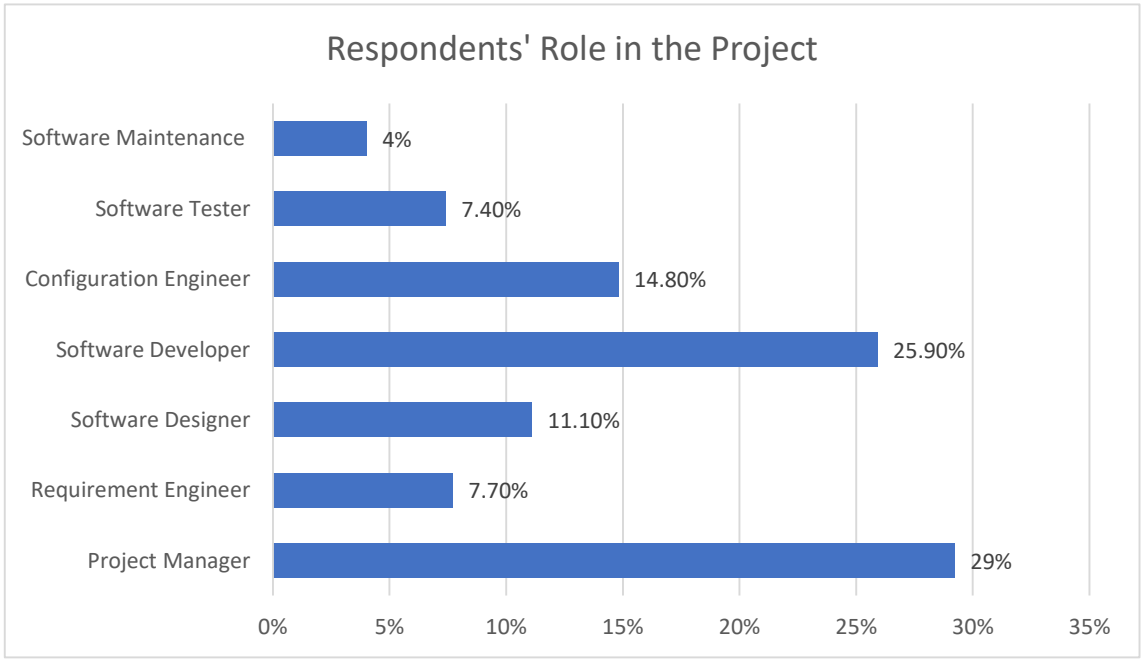

Figure 12 Overview of the Respondents' Roles in Software Development Projects

From the figure, we can see most of the respondents (70%) are developers, testers, maintenance, and engineers while approximately 30% are project managers or team leaders. The aim of the question was to distinguish if they had a management role in the project or not since the impact of the leadership on the project's outcome is one of the aspects being evaluated. This would be important to detect if there is any bias in the responses that gave a positive evaluation of the project's leadership in later questions.

With questions 1.2 and 1.3, the goal was to compare the difference between estimated and actual results of their projects. The respondents were asked whether their project exceeded its allocated resources and which resource it was. With a 66% majority, the responses to my second question confirmed their project did exceed one or more of its intended resources.

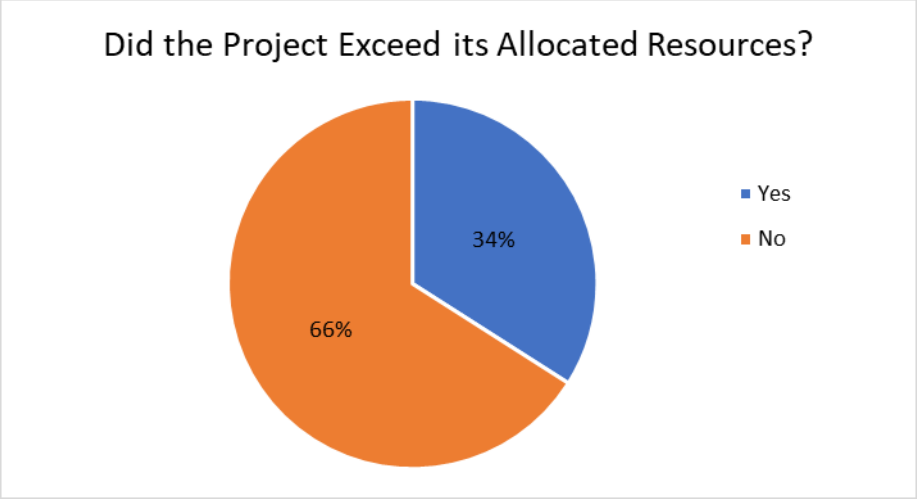

Figure 13 Projects Categorized Based on Whether They Exceeded Allocated Resources

When identifying which of the main resources did software projects exceed, the check-box question type was selected here to enable the respondents to choose more than one aspect as projects can exceed one or more at the same time.

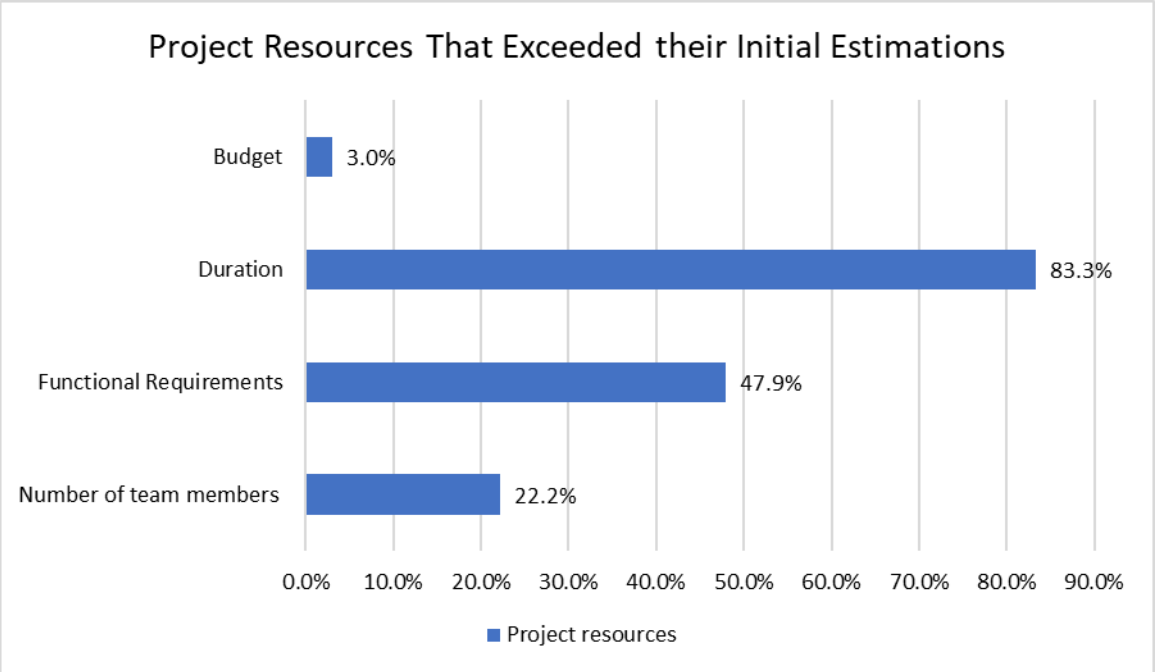

Figure 14 Project Resources That Exceeded Initial Estimated Efforts

As we can see from Figure 14 a project's duration tends to exceed the expected timeframe with 83.3% responses. The functional requirements were the second aspect that tended to exceed its initial estimation with approximately 48% of the responses.

Question 1.4 addressed the goal of this section in a more direct way by asking the respondents for their evaluation of the overall success of the project. We can see from Figure 15 below that despite projects always exceeding their estimated efforts, the majority was still considered a success (70%).

Figure 15 Respondents Evaluation of The Projects' Success

Based on answers to these three questions, I've divided the responses into 2 categories; successful projects and unsuccessful projects. The success threshold was determined as a score of 4 or higher on the linear scale and this was used for further analysis in later sections.

When asked to elaborate on their answers in the last question of the section, three of the respondents who indicated their project exceeded the allocated time explained that missing the deadline was due to external issues such as delays with third parties or User Acceptance testing (UTAs). One respondent who indicated their projects exceeded the expected number of functional requirements mentioned in his explanation that they ended up delivering more requirements as the client became more demanding and requested further configuration changes during implementation phases. On the other hand, one respondent commented on the project's success instead and indicated that even though the overall quality could've been improved, the output was approved and accepted by the client which was satisfying for them. Another respondent indicated that the project was completed according to the best practices and delivered the intended levels of quality and that was enough to rate it as a success. Both the respondents were amongst the majority who indicated their projects exceeded the estimated timeframe of the project.

4.2.2 Output quality

In the second section of the questionnaire, the goal was to identify what constituted success and quality for the respondents in a software project, hence the respondents were asked about their evaluation for a select few quality attributes.

As one of the areas of software project management chosen in the framework proposed for this thesis and as an indication of project success, evaluating the quality of the produced software in each of their projects was key. To further examine the outcome of the project and why they'd described it as a success or failure in question 1.4. so, the respondents were asked about the relevance of the delivered output to the original requirement and whether it worked as intended. The data showed that 81.5% of the respondents agree that they delivered a functional and relevant output (scores of 4 and 5 on the linear scale). On the other hand, only 18.5% think their output's functionality was sufficient or not satisfying.

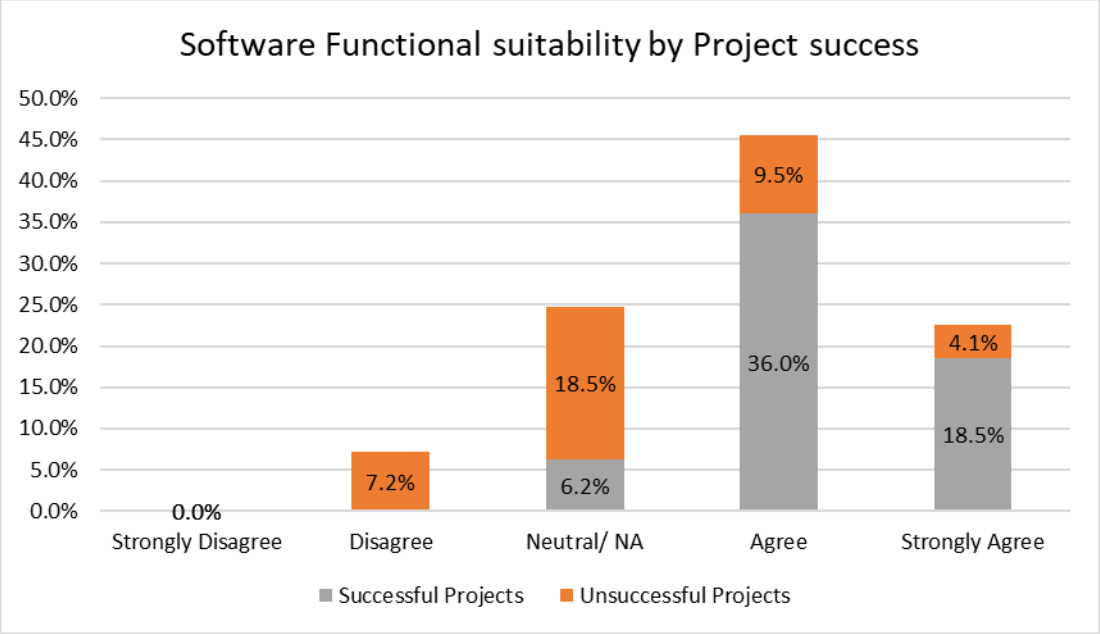

Figure 16 Functional Suitability Categorized by Project Success

From the figure above, we can see that the majority of respondents who think their project was successful, indicated high levels of functional suitability. This indicates a high focus on the functionality aspect of the delivered software as a main quality measurement.

When the respondents were then asked to evaluate the usability of their delivered output and whether or not the end users found it easy to understand. The answers were mostly positive for this question with 88% of the respondents indicating high usability levels for their delivered system (a score of 4 or 5 on the linear rating scale).

Figure 17 below shows that the majority of respondents who think their project was successful, indicated high levels of usability as well. This can be considered as a reflection of the trend to focus on simplifying the user interface and creating a more appealing user experience as a competitive edge in the software market.

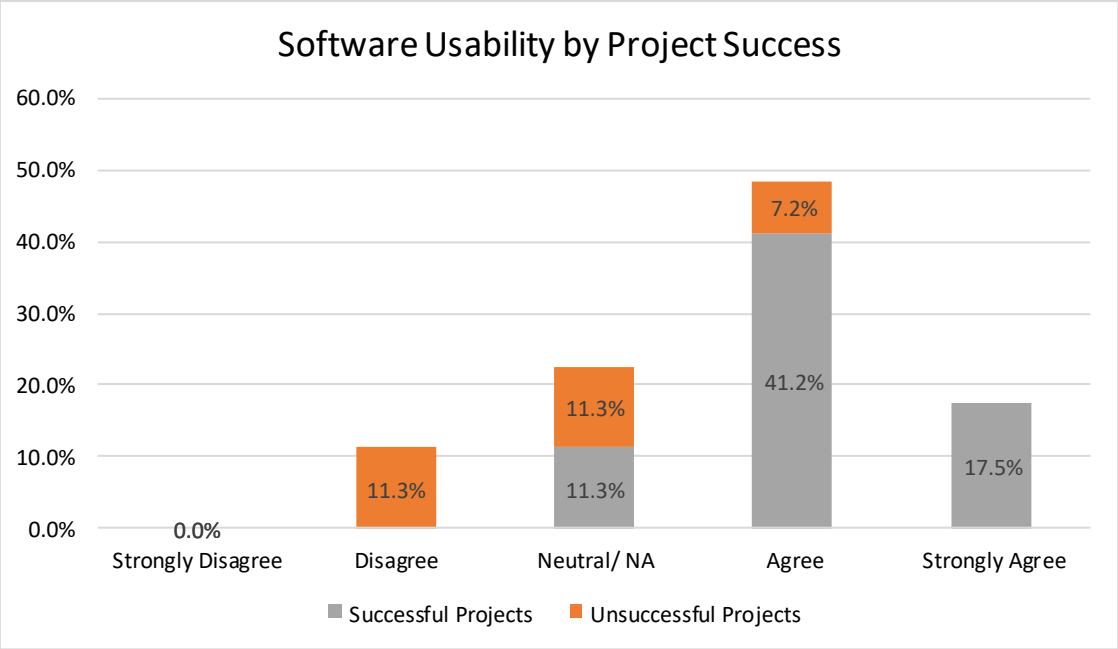

Figure 17 Software Usability Categorized by Project Success

In my next question, the participants were asked about the reliability of the delivered output represented in its stability and its ability to recover from any unexpected failure that might cause downtime. These are all measurements of availability and reliability of the software product which is one of the valuable quality metrics of software is.

Figure 18 shows that the distribution respondents' evaluation of the software reliability categorized based on whether their project was a success or a failure. A majority of the reviews were mostly positive once again for the reliability aspect of the software quality with this question with a majority of the respondents indicating high-reliability levels for their delivered system (a score of 4 or 5 on the linear rating scale).

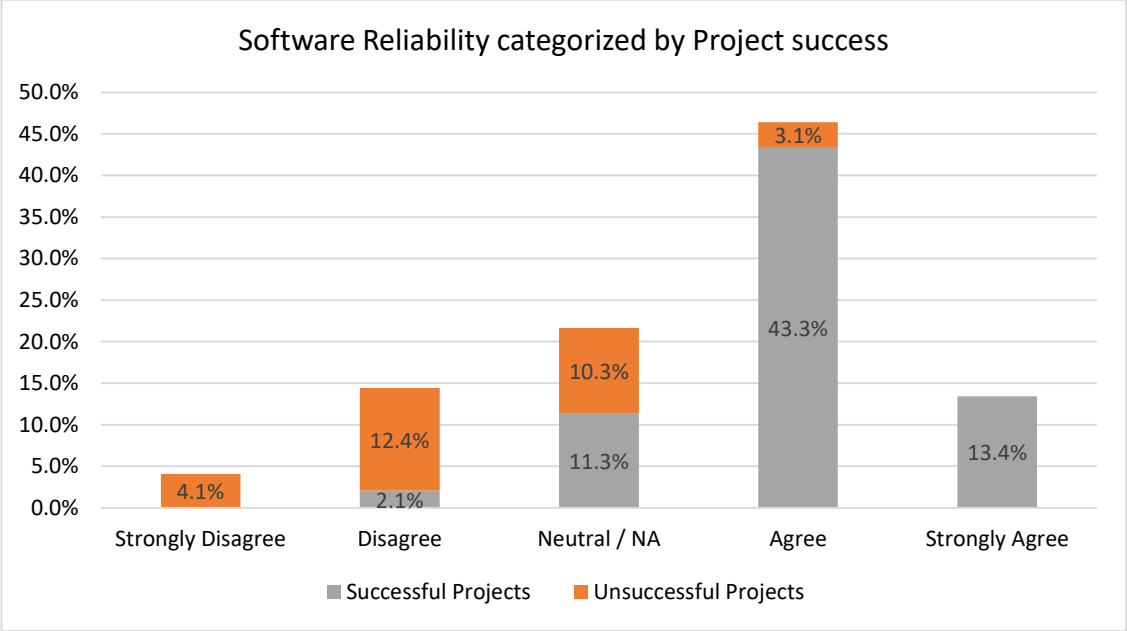

Figure 18 Software Reliability Categorized by Project Success

In question 2.4 of the section, the respondents were asked whether or not the delivered project consisted of thoroughly tested modules. The results are represented in Figure 19 below and we can notice from the figures that the percentages of positive evaluation (levels 4 and 5 on the linear scale) here are a bit lower than the previous quality metrics evaluated.

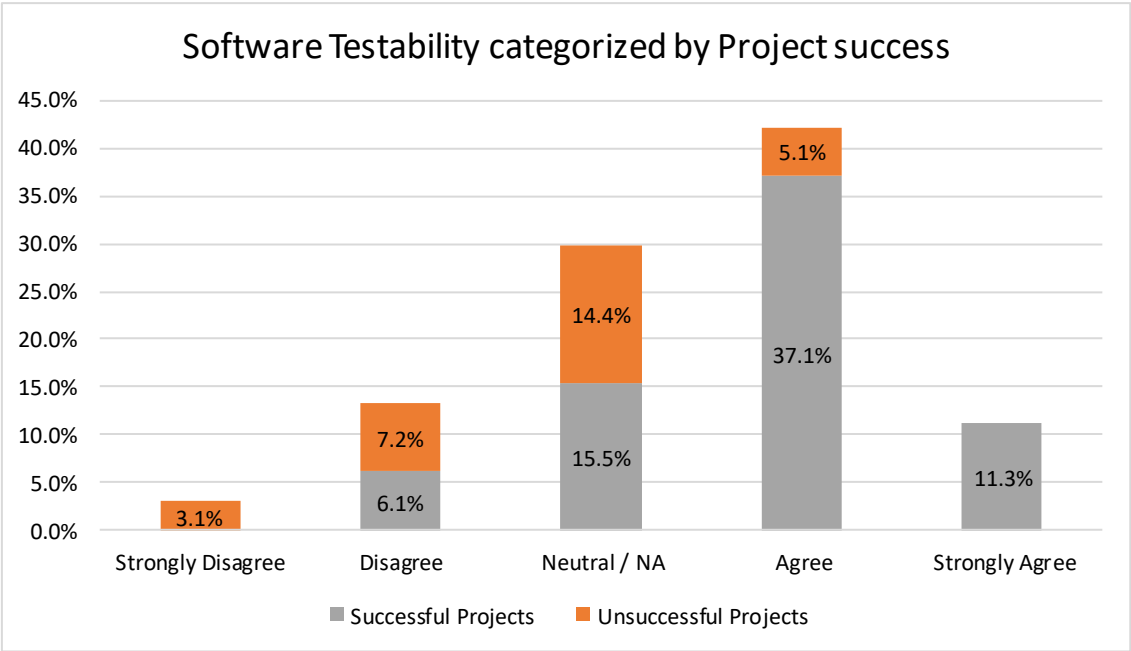

Figure 19 Software Testability Categorized by Project Success

For the last question of the section, the respondents were again about their overall satisfaction with the quality of their delivered system. The aim was to evaluate how they felt about the quality of the project's output after evaluating some of the key quality metrics and whether it correlates with project success in their perspective.

We can see from the graph in Figure 20 that when it came to the quality of the output the majority of the responses were high ratings (approximately 66%) This indicates that the majority of the respondents were not only considering their projects a success, but they were also satisfied with the quality they delivered despite some quality metrics that were less than perfect and despite exceeding their allocated resources in most projects.

This also confirmed to me that the initial categorization of the projects into successful and unsuccessful based on question 1.4 holds true as the projects categorized as successful did eventually rank higher in all the quality aspects examined.

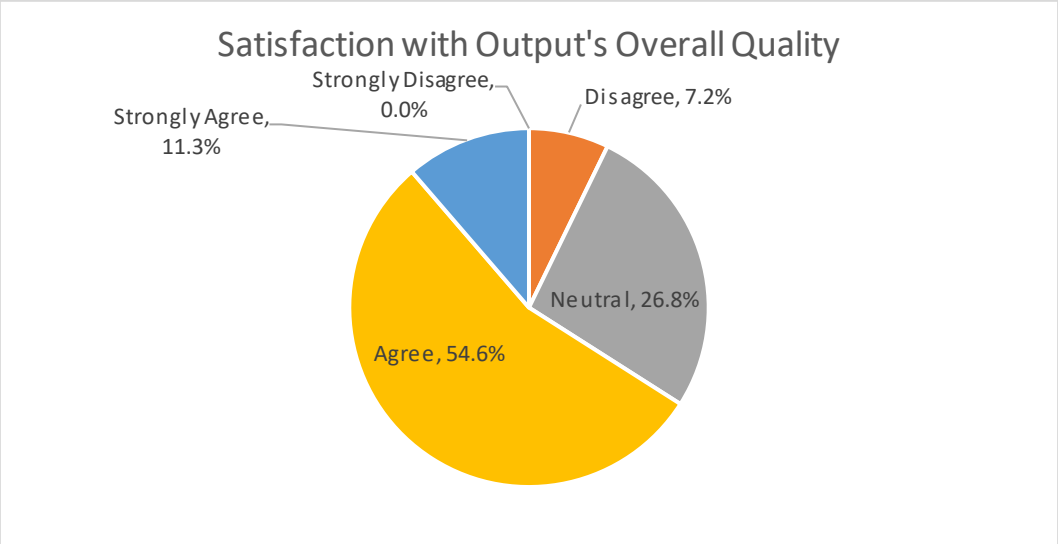

Figure 20 Responses to Q.2.5. You are satisfied with the Quality of the Delivered Output

4.2.3 Stakeholders, Processes and Risk Management

The third and final section of the survey covered the remaining aspects of the software development life cycle, starting first with the stakeholders then moving on to the development process and their involvement in it. I choose not to divide these aspects into three separate sections as some of the questions used overlap between them.

This overlap is natural considering managing the stakeholders is ideally considered a part of the project's processes and we can also argue the same for risk management activities. As we discussed earlier in this thesis, the software development life cycle is a series of interdependent activities regardless of the model you follow.

To start my examination of these aspects, the questionnaire started by first asking about the team manager as part of the stakeholders and one of the people working on the project. The aim was to evaluate the team manager's effectiveness from the perspective of the respondents. This was measured in two questions. First, question 3.1 asked whether the project manager was supportive and accessible to the stakeholders during the project's cycle. Then in question 3.2, the respondents were asked whether the project manager was able to foresee the project risks and was a proactive problem solver. As shown in Figure 21, 74% of the respondents indicated high availability from the team manager and 60% confirmed he was a proactive problem-solver who can foresee the project's risk.

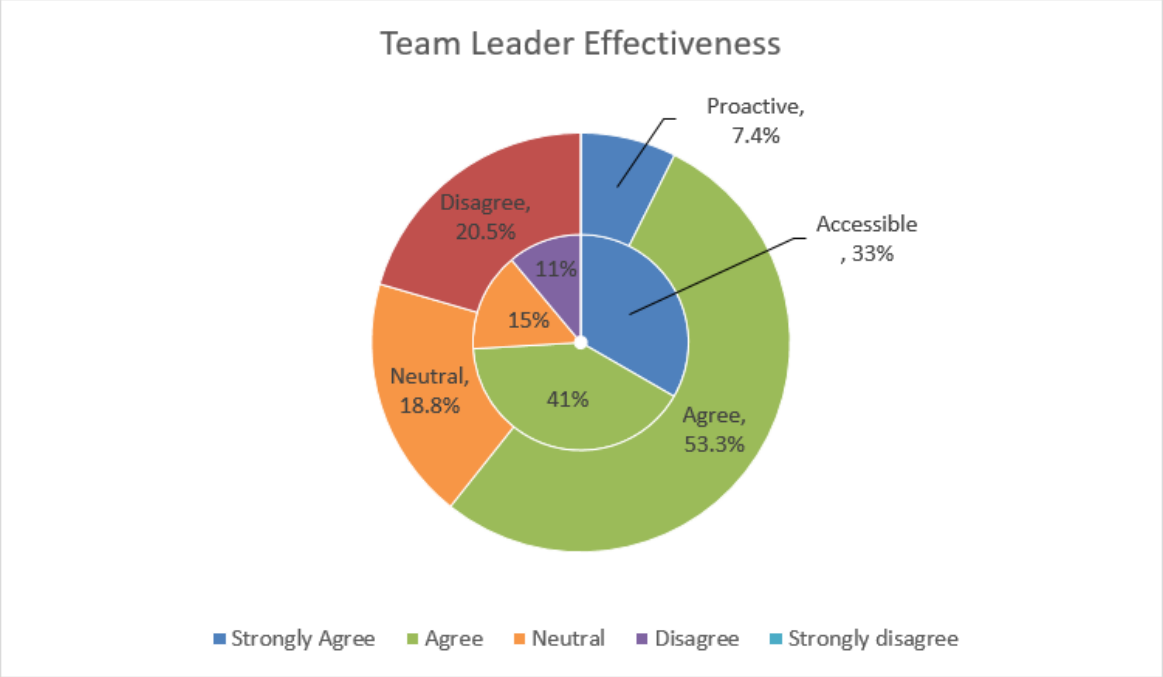

Figure 21 Effectiveness of the Team Leader (Questions 3.1 & 3.2)

To have a better understanding of the responses' distribution between successful and unsuccessful projects, the responses were categorized and used in clustered stacked bars to compare the results as shown below in Figure 22. The distribution in the figure shows that a majority of positive responses (Rating of 4 and 5 on the linear scale) in both questions is indeed coming from successful projects. This shows how important of an impact the project manager or the team's leadership can have on the outcome of a project.

The project manager has the responsibility to both be supportive of his team and the stakeholders. He also has to be proactive and resourceful when it comes to dealing with problems and foreseeing the risks of the project.

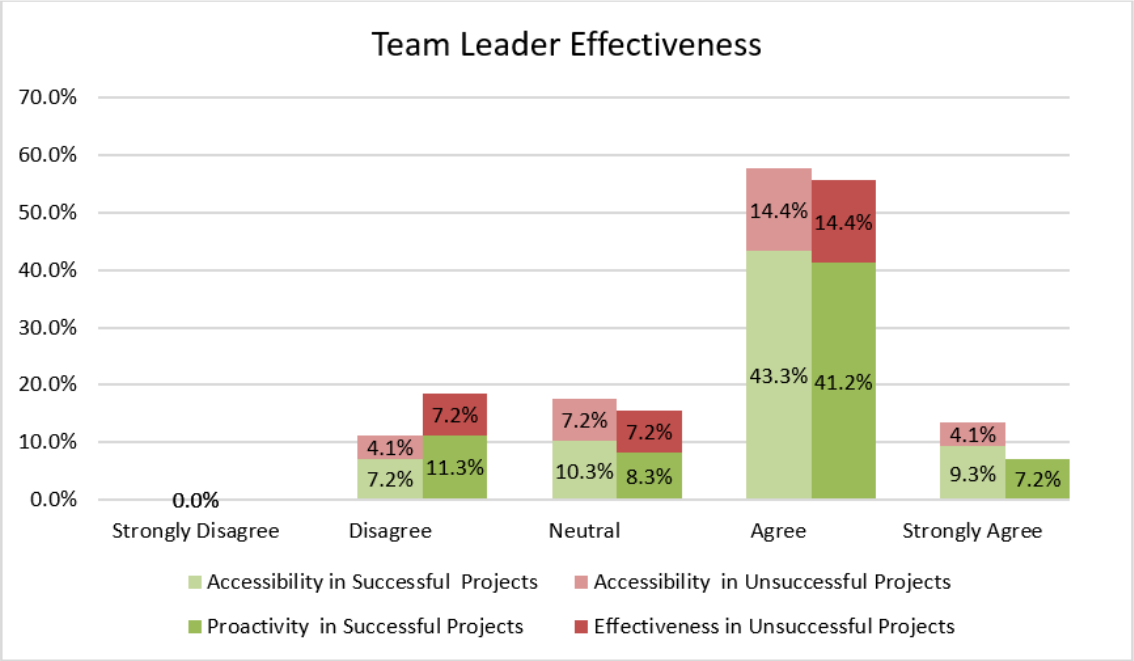

Figure 22 Effectiveness of Team Leader Categorized by Project Success

When asked if their projects were negatively impacted by not having enough qualified or experienced team members, the majority of the respondents gave a rating of 3 or lower to the statement.

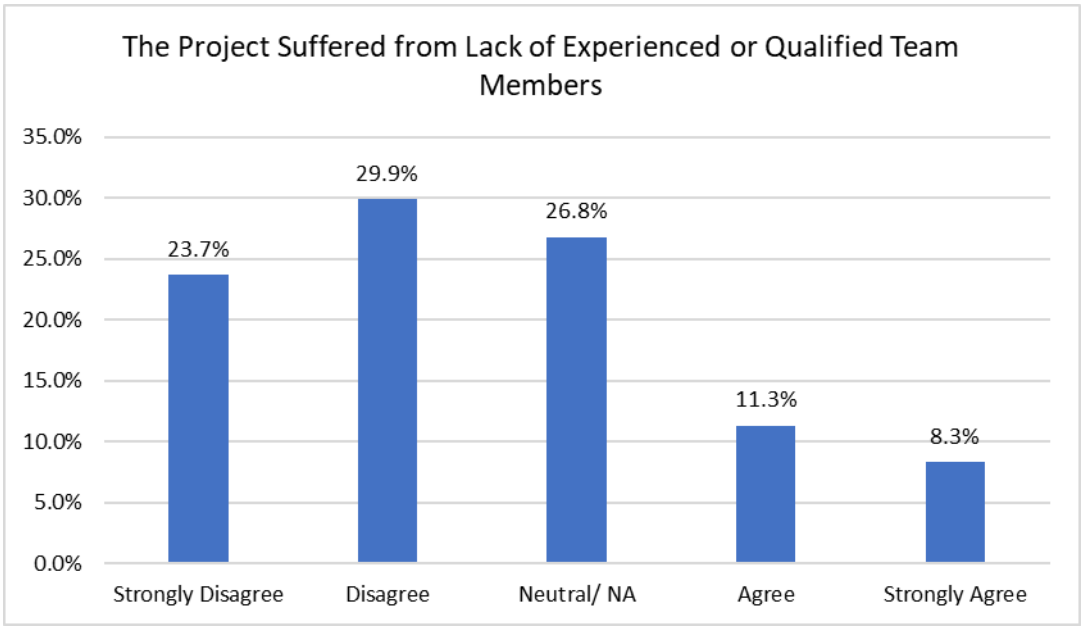

Figure 23 Responses to Q.3.4. The Project Suffered from Lack of Experienced or Qualified Team Members

We can see from the distribution in Figure 23 that approximately 80% of the respondents disagree regardless of the outcome of their project. This is an indication that the team

members selection was successful in most cases or that the lack of skillful software practitioners was not really a significant issue.

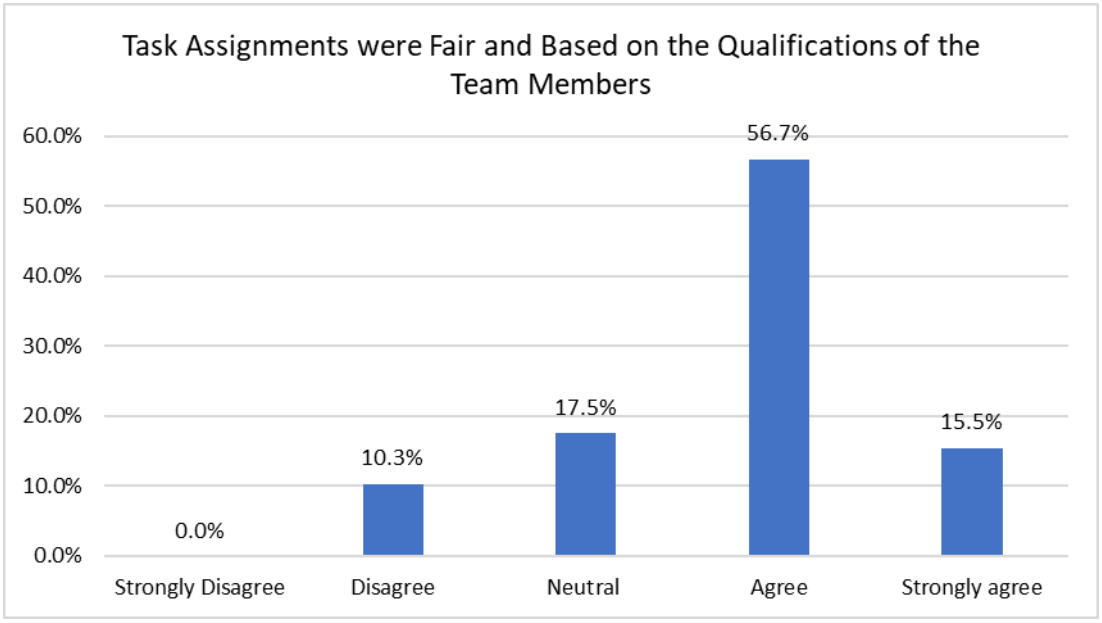

Figure 24 Responses To Q.3.5 Task Assignments Were Fair And Based On The Qualifications of The Team Members

When asked whether or not task assignment was based on qualifications of the team members, it was noted that task assignment in approximately 72% of the projects was indeed dependent on their skill levels as shown in Figure 24. Examining the responses of the remaining 28% who disagreed or were neutral about the statement (scores of 2 or 3 on the scale) it was noted that the majority of the respondents who disagreed with the statements were indeed from unsuccessful projects or projects that exceeded the timeline defined for the project.

The communication activities between team members and other stakeholders was another aspect I focused on. In question 3.6, I asked the respondents whether communication activities in their projects were streamlined and accounted for in the planning phases. Responses were almost evenly distributed with 35% agreeing, 33% being neutral and 32% disagreeing. To get a better understanding of the results, the responses were categorized based on whether or not their project was successful and were represented in Figure 25.

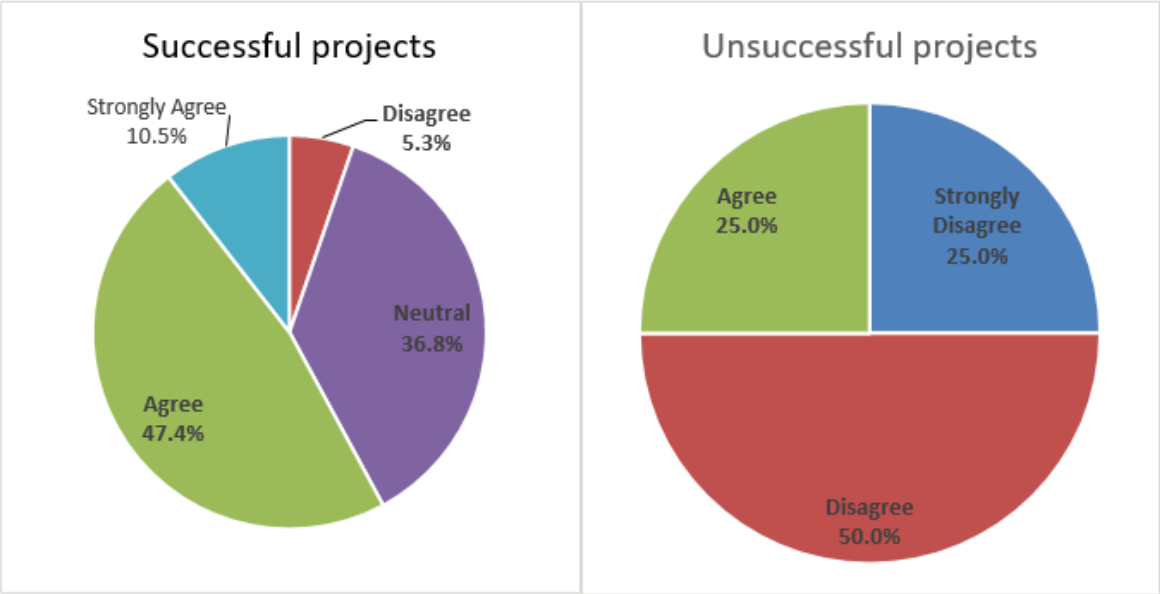

Figure 25 Communication Effectiveness Categorized by Project Success

From the pie chart of successful projects, we can see that approximately 58% of the respondents confirmed having good practices when it comes to communication. This means communication activities were accounted for in the planning and went smoothly during the development life cycle. On the other hand, in the pie chart of unsuccessful projects, we can see the opposite. 75% of the respondents disagreed with the statement which shows they potentially encountered problems during the project due to miscommunication.

The questionnaire then shifted attention towards the overall morale of the team members. I asked whether they lacked motivation and support in the project and if they had a good sense of orientation while working with their teams.

Lack of motivation amongst team members can happen due to various factors like a confusion of responsibilities or conflict of opinions and would have a negative impact on their commitment to both the project and the team. While the results below are considered a good indication of their satisfaction with the team, we must not overlook the team members who indicated they lacked motivation and would like to change the team because of it. Losing any of the human resources can cost the team and the company good skills, time and money.

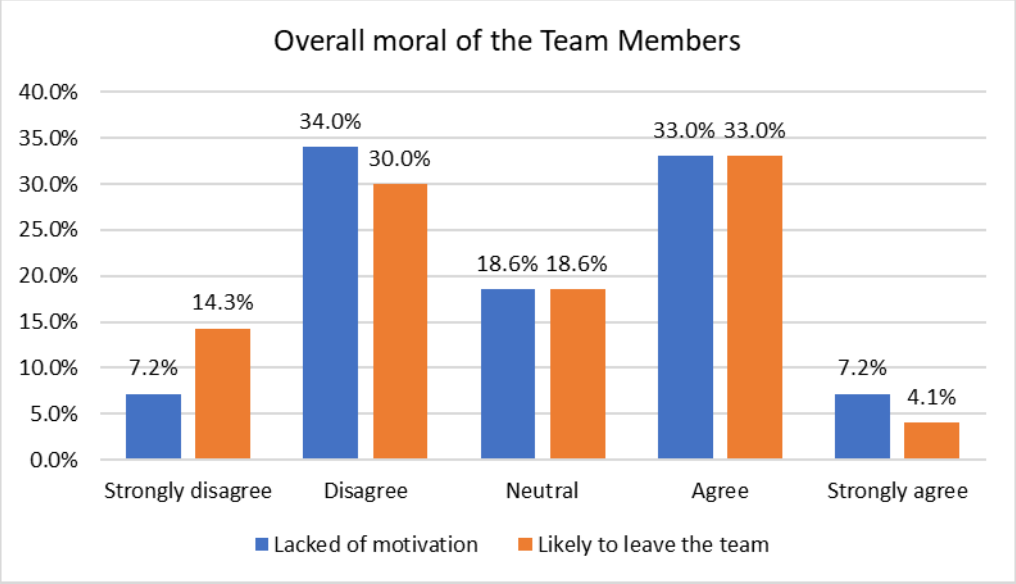

Figure 26 Team Members Lacked Motivation and Are Likely to Leave The Team

We can see in Figure 26 that the lack of motivation and sense of achievement amongst the team members highly correlates with them being willing to move out of the team if possible.

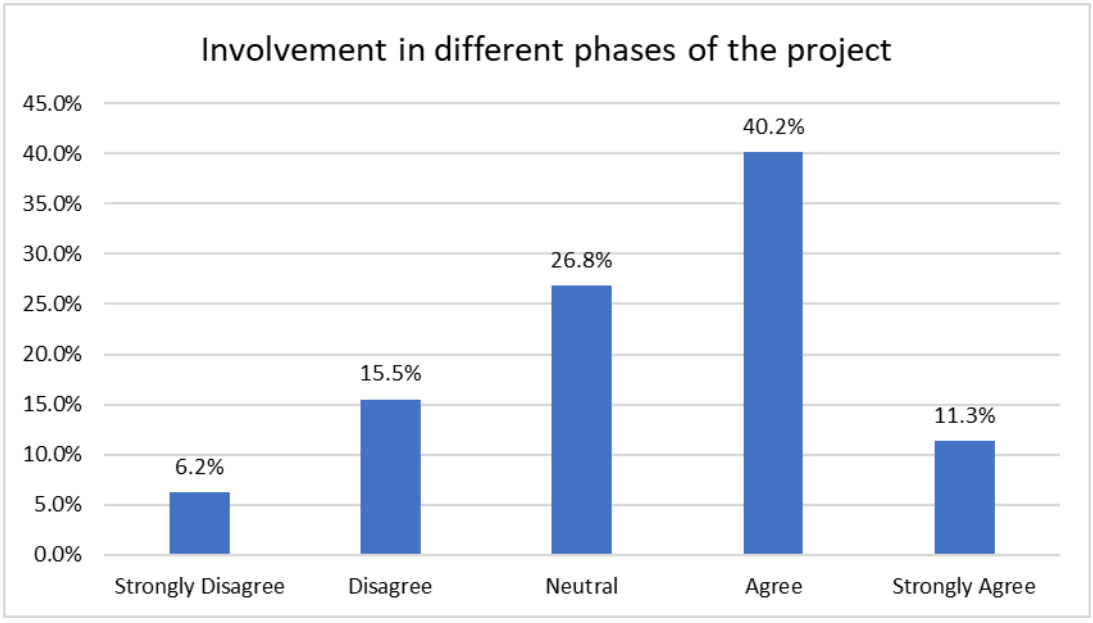

Figure 27 All Stakeholders were Involved Different Phases of the Project Cycle

Figure 27 shows the respondents' evaluation of the stakeholders' level of involvement in the project phases. The majority of the responses indicated high levels of involvement in general with only 21.7% disagreeing as we can see from the chart. This, however, did not prevent the projects from encountering other problems down the road and not meeting the expected or planned restrictions.

In question 3.10, when the respondents were asked if they had regular and impactful status meetings during the project's cycle to assess the team's performance and monitor project risks. 55% of the respondents indicated they had meetings regularly but since follow-up meetings and team huddles are part of the communication activities, I wanted once again to divide the responses based on their outcome to confirm if the same observation from Figure 25 would hold. As we can see from Figure 28, 87% of the respondents who indicated their projects were successful were among those who confirmed having regular and impactful meetings where they tracked their performance and the progress of the project. On the other hand, in unsuccessful projects, the majority said they did not have those follow-up meetings.

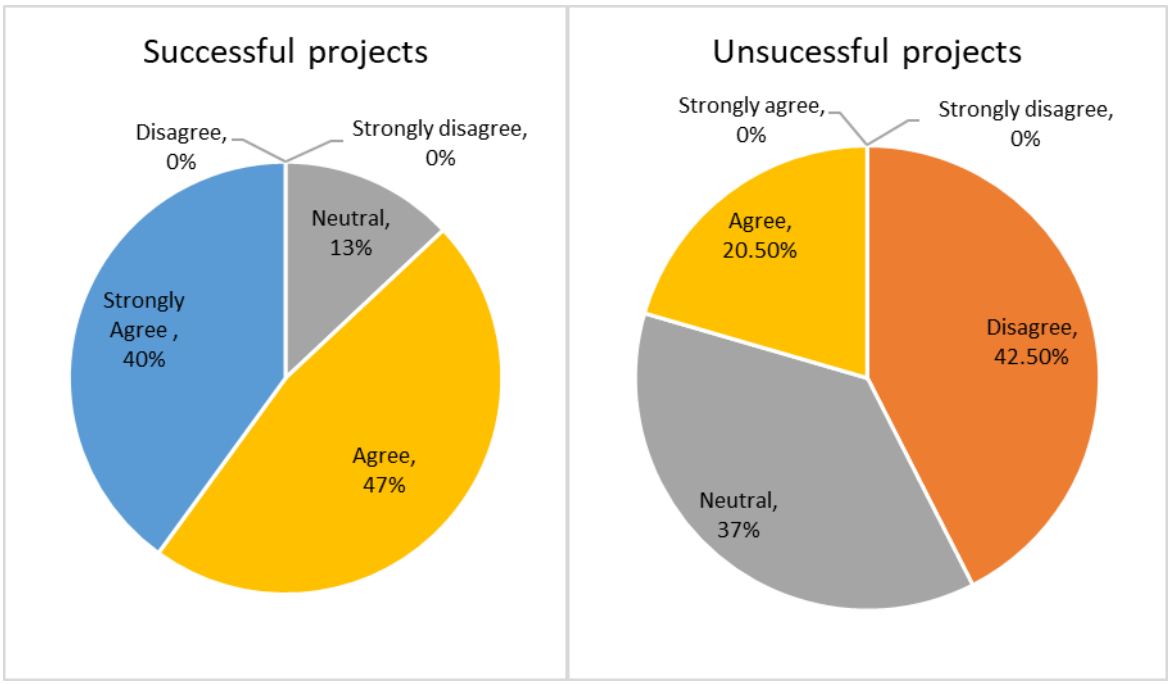

Figure 28 Having Regular Meetings Categorized by Project Success

When the participants were asked if their project's requirements were stable and clear e across the project development phases, approximately only 30% of them agreed they were. Figure 29 shows the distribution of the responses and we can see that the majority indicated the client's requirements were not stable or were somewhat unclear at different phases of the project development (rating of 1, 2 or 3 on the linear scale).

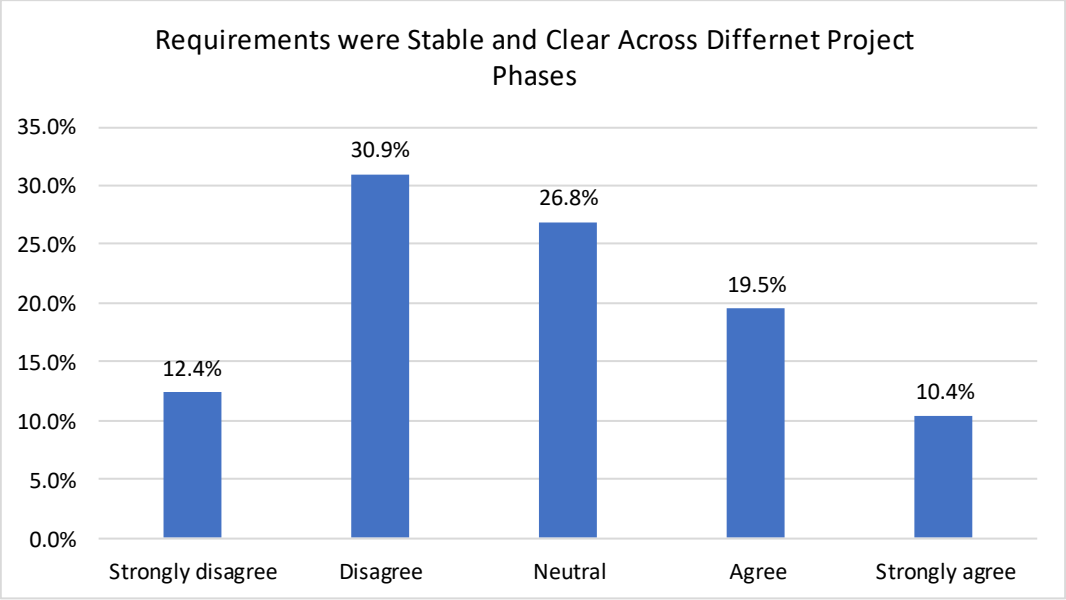

Figure 29 Requirements were Stable and Clear Across Different Project Phases

The next group questions were focused on covering the risk management activities carried out during the project cycle like risk assessment and risk control. The goal was to evaluate whether the team took the time to analyze any uncertainties they had and whether they prepared for potential risks that might have a negative impact on the project's outcome.

I also asked them if they were transparent in sharing this information with other stakeholders who might be affected. The third question of the group was focused on risk control and whether the team could proactively and effectively address those risks that were identified.

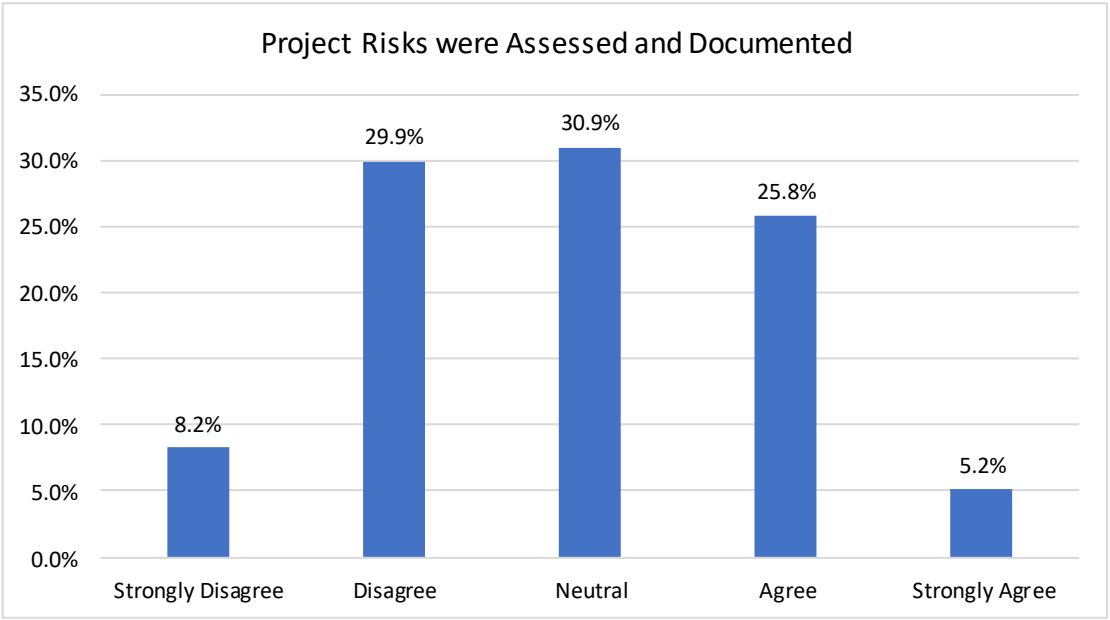

Figure 30 Projects Risks were Assessed and Documented.

Question 3.12 asked the participants if risk management activities performed and whether the risks of the project were assessed and documented. As shown in Figure 30 only 31% of the respondents confirmed performing risk management activities. I separated the answers of those participants who confirmed performing risk management activities to analyze how they treated those identified risks based on their answers in the following two questions.

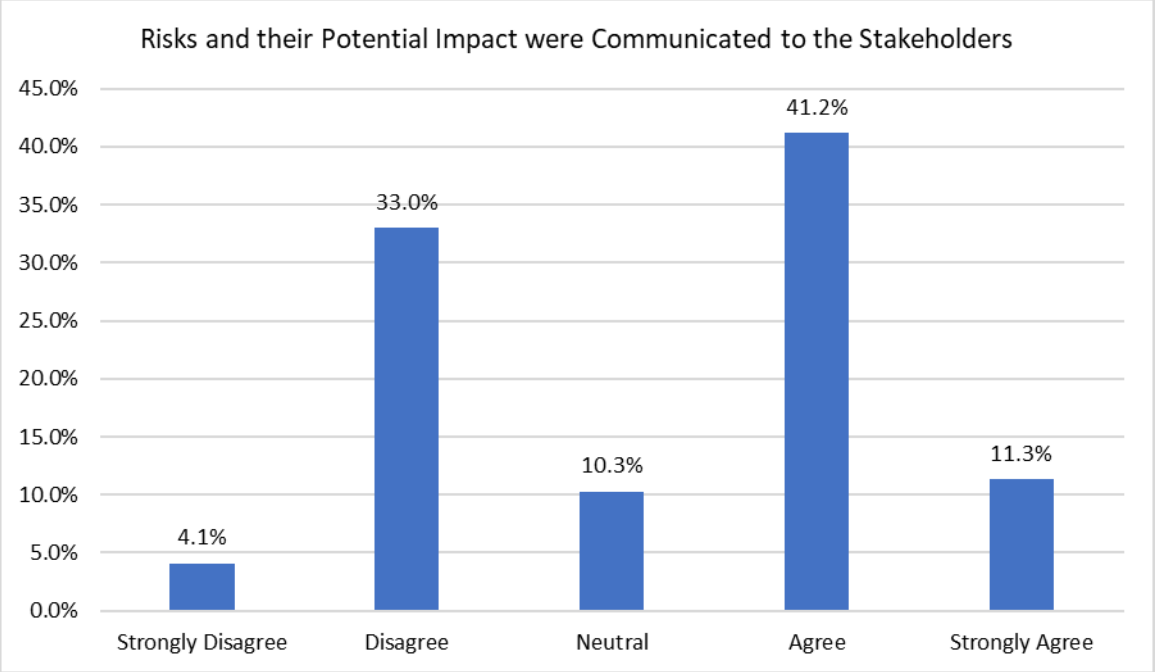

Figure 31 Risks and their Potential Impact were Communicated to Stakeholders

When the respondents were asked in question 3.13 if the potential risks identified and their impacts on the project were shared with all stakeholders, it was noted that 60% of the respondents shared these results with the stakeholders as shown in Figure 31.

Lastly, I asked about risk control to evaluate whether project problems and risks were being addressed effectively and in a timely manner. As we can see from Figure 32, out of the participants who performed risk management activities, 60% of them agreed with the statement (a rating of 4 or 5 on the linear scale). This means they were able to effectively address and resolve these uncertainties.

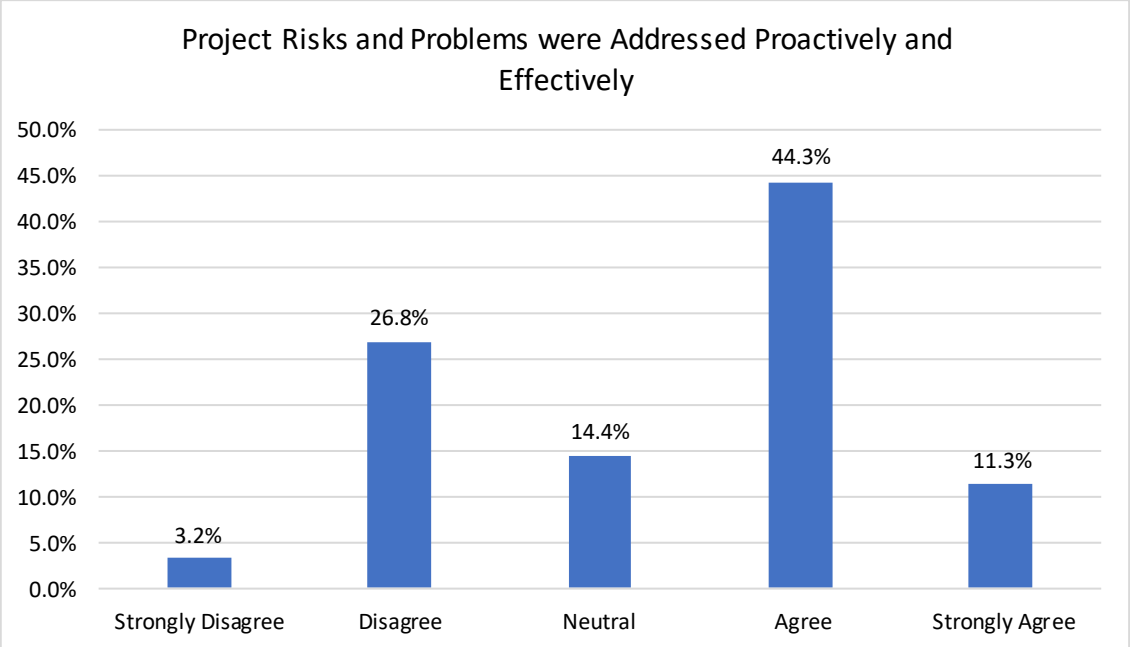

Figure 32 Potential Project Problems were Addressed Proactively and Effectively.

To round up my questionnaire, I added two multiple-choice questions where I asked the respondents to choose which aspects of their project's life cycle was challenging in their opinion and which were well-functioning. The questions were checkbox type so respondents could choose more than one.

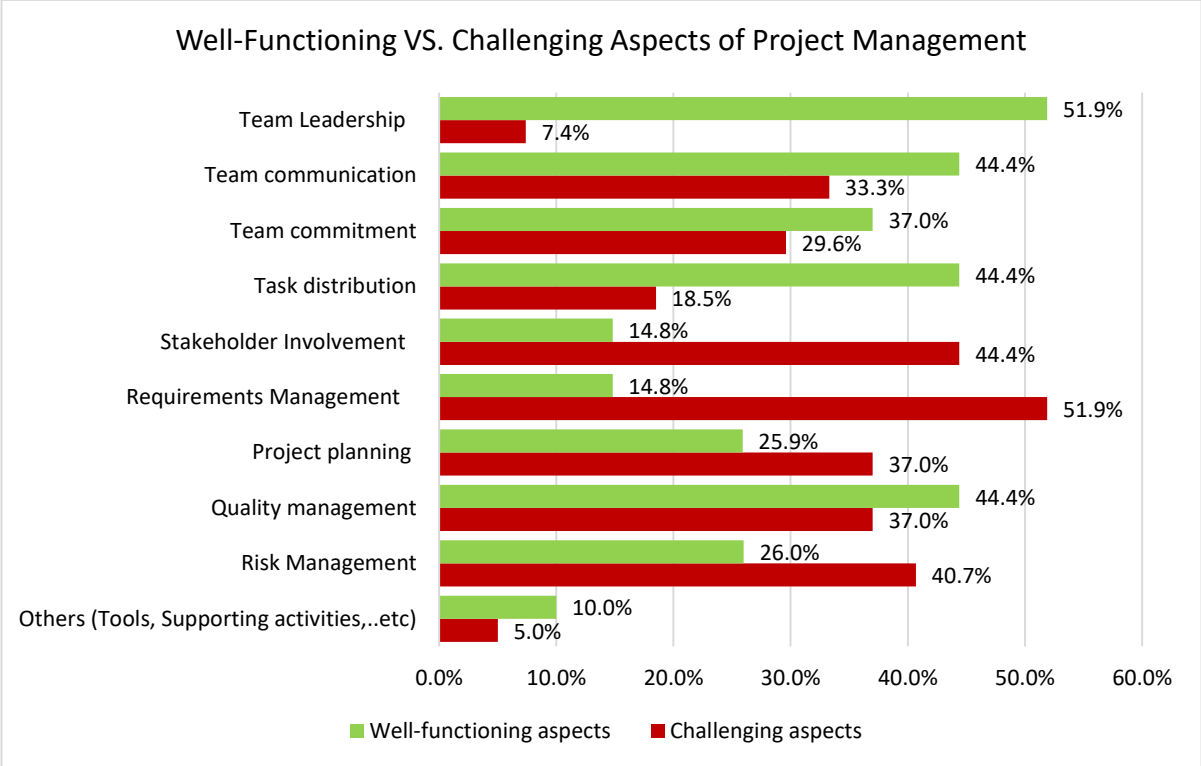

Figure 33 Well-Functioning Vs. Challenging Aspects of Project Management

Responses to both questions were mapped against each other as represented in Figure 33 to compare them. The questions were checkbox type so respondents could choose more than one. The responses show that team leadership, team communication along with task distribution and quality management were the most voted as well-functioning. Most of these areas are related to stakeholder's management and fall under the project manager's responsibility.

On the other hand, requirement management and the stakeholders' involvement during the project were both among the most challenging aspects according to the respondents along with risk management. In addition to that, respondents who choose "Other" listed a few other elements that were not given as a choice such as supporting activities and trainings. The list of answers is included in the appendix section.

5 Results and Discussion

The results of the questionnaire were interesting to analyze because while some questions were in line with my expectations, others were not. However, upon further examination, the overall findings proved to be consistent with my observations of the software development field. In general, the responses from different participants and the results were found to be consistent and coherent. They have helped me confirm the validity of my approach and the theory proposed in the premise of this thesis.

For example, as discussed in earlier chapters of the thesis, staying within the timeframe of the project is a core factor in judging its success. However, from Figure 14, it was noted that the majority of software projects (83.3% of responses) are not delivered on time. The figure also showed that the functional requirements were the second most exceeded aspect compared to its initial estimates (43% of responses). The fact that this happens so frequently in the software development field could be attributed to a number of factors.

The first reason that comes to mind would be the lack of stability and clarity in requirements which was observed in Figure 29 where 70% of the respondents said the requirements were not stable in their projects. Requirements are usually dynamic. They can change during the development life cycle and evolve with each phase whether due to changes in the client's needs or even changes in the market itself. As the change in requirements would usually mean more work and more testing, it leads to an increase in a project’s duration or the manhours needed to finish it. This would explain why many companies are switching towards using agile methods and rapid development to quickly produce and deploy software systems.

Figure 15 showed that even when projects exceeded one or more of their estimated resources, 70% of the respondents still considered them a success if the practitioners feel they delivered sufficient quality as shown in Figure 20 or if the client was satisfied with the output. By looking at their evaluation of the different software quality metrics I asked

about, we saw from Figure 16, Figure 17, Figure 18, Figure 19 that they were are highly rated even occasionally in failed projects.
Figure 17 was especially interesting as it shows a majority of respondents who think their project was successful, indicated high levels of usability as well. This can be considered a reflection of the trend to focus on simplifying the user interface and creating a more appealing user experience as a competitive edge in the software market.
The results from Figure 22 was interesting for me as it showed what significant impact the project manager has on the project success. Having an effective project manager who is both experienced and supportive of his team can help them stay aligned and motivated, thus achieving a better outcome that would satisfy everyone involved.

Both Figure 23 and Figure 24 highlighted the fact that lack of skillful software practitioners was not really a significant issue in the software field as the majority of the participants were satisfied with the team's level and the task assignment. We can also give credit here to the project manager as this shows that the team members' selection was successful, and the task distribution was fair in most projects.

Figure 25 and Figure 28 both showed the importance of communication to project success. 87% of the respondents who indicated their projects were successful were among those who confirmed having regular and impactful meetings where they tracked their performance and the progress of the project and approximately 58% confirmed having streamlined communication in general.
Having streamlined communication channels and dedicating time for communication activities and regular meetings are essential. Those activities help stakeholders follow up with each other and share information or feedback regularly. Good communication also ensures that all stakeholders are aligned and have a clear understanding of where the project is headed as well as track the progress made. Whether it is via emails, a project management tool or follow up meetings, communication activities are a common practice in software development. They may vary significantly in how they are conducted but they are always going to be a part of the process.
That is why they should be optimized; because if they are not executed carefully, they could have a negative impact on the project. For example, if meetings or emails become too frequents without any added value or if they do not have a well-defined agenda, they

would not be serving a purpose or helping the project. Instead, they just become a waste of the stakeholders' valuable time.

We can see in Figure 29 from the closeness of the percentages of votes on both questions that the lack of motivation and sense of achievement amongst the team members highly correlates with them being willing to move out of the team if possible. That's why now we see a great focus on taking care of the human resources in software and technology companies. Most of them are now trying to offer a more supportive and relaxing environment for their employees to reduce the pressure of a strict project schedule and encourage creativity.

In Figure 30 where the risk management activities were analyzed, I came to interesting results. Most teams were not paying much attention to risk assessment in their projects and were not documenting those risks. Only 30% confirmed conducting risk analysis in their projects. That was somewhat shocking if not totally surprising.

Not only that but Figure 31 showed that 47.5% of practitioners who conduct risk assessment activities do not always communicate their findings. Even though this is not a majority, it is still significant. Transparency with the stakeholders is important as it helps set the level of their expectations regarding the possible outcome of the project.

Figure 32 highlighted another issue related to risk management which is controlling the risks themselves. Understanding what can potentially go wrong in any situation is important and is not an exclusive recommendation in software development projects. Knowing the potential risks of a software project should help the team either avoid it altogether or prepare solutions to mitigate its impact. However, the figure showed that approximately 56% of the practitioners who perform risk assessment activities actually address them proactively which came as a surprise to me.

In Figure 33 it was noticed that elements related to the area of stakeholder management such as team leadership, team communication, and task distribution were voted among the well-functioning aspects of the project. So were the quality management area due to the heavy focus on the technical aspects of the software development projects. On the other hand, tracking the project requirement and dealing with the client's involvement proved to be trickier. Risk management was highlighted as a major challenge as well. Not only was

this in line with earlier findings and analysis conducted in the thesis but it also shows that the participants are capable of critiquing their projects and aware of their positives and negatives.

6 Conclusion

The software field has been growing and evolving rapidly in recent years. Both software and smart devices are now an integral part of our daily lives. However, despite this, developing a software system is still considered a challenging endeavor. Developing a successful software can at times be an overwhelming undertaking with all of its interdependent elements. Keeping track of the process, the technicalities, the people involved, and their requirement is not an easy task as things can get out of control quickly. That is why the need for good project management practices is essential to the success of a software development project.

The analysis done in this thesis, in both the literature review and the questionnaire, explored the relationship between applying good project management practices and achieving a successful outcome in the software field and the findings confirmed it. While the importance of having enough knowledge and experience in the software engineering field is undeniable, it is not a guarantee for project success by itself.

The thesis showed that while it is hard to pin down one definition of project success due to the different perspectives of the different stakeholders in a software project, we can identify its key elements. If those elements were monitored and controlled properly, the success of a software development project would become more likely. The elements identified were categorized into different areas which are; Output quality, Stakeholders, Processes and Risk management. The questionnaire examined each of these areas across different projects based on the responses of different software practitioners to validate its relevance.

The thesis also showed how goal-oriented the majority of software practitioners are; focusing all their attention on the developed system and considering it the main standard for their success. However, this has proven not to be a good approach towards achieving sustainable success as projects often exceed their intended time and budget or take a toll on the people involved. Amongst other things, the research done in this thesis confirmed the importance of paying attention to the different needs of the various stakeholder as well as

the significant variations in their perspectives. What constitutes quality for the developing team may not be the main concern for the end-user of the system. Taking the time at the start to identify the objectives and expectations of the stakeholders is what gives the project its goal and a clear sense of orientation.

Not only that but establishing good communication practices and transparency amongst the stakeholders also proved vital to the success of the project. Streamlined communication would help the flow of information throughout the development project and would make sure everyone feels included and their inputs valued. Offering support to the development team and showing empathy for the stakeholders' needs is a key responsibility of anyone in a leadership position and would have a positive impact on the project.

In addition to that, defining clear processes and procedures for the execution constitutes a roadmap toward achieving the project's goal and breaking down the project and its complicated processes into smaller phases and milestones helps both the team and the stakeholder have clarity; not only on what will be achieved but how and when it will be achieved as well.

Nevertheless, projects are unique and software development projects are even more so due to their complexity as well as the constant booming development in the field and things will not always work exactly as planned. Identifying the potential risks of a project and preparing to proactively counter those risks or mitigate their effects should not be neglected. In fact, taking them into consideration can increase the project's chances of success.

In conclusion, while technical knowledge and functional quality are of utter importance in the software development field, they are not enough to guarantee success in this highly driven and competitive field. Paying attention to the needs of those involved, having a well-defined process, preparing for the unexpected and honest transparent communication are the keys to securing a sustainable presence in the software development field.

7 References

AUSTIN, V. Rural, 1984, Project Management: A Handbook for Students and Practitioners. London: Botsford Academic and Educational, 1984, ISBN 07-13444169.

BOEHM, B. W. 1991, Software Risk Management: Principles and Practices. IEEE Software, 8(1). P32-31, ISBN:0-8186-8906-4

CLANCY, T. 1995. The Standish Group Report: Chaos. Retrieved Feb 20, 2019, from https://www.standishgroup.com/sample_research_files/chaos_report.pdf.

DEFENSE SCIENCE BOARD (DSB), 2000, Report of the Defense Science Board Task Force on Defense Software. Washington, D.C: 20301-3140

DEMIR, K. A. 2008. Measurement of Software Project Management Effectiveness. (Dissertation, Naval Postgraduate School, CA, 2008). NSN 7540-01-280-5500

FORSBERG, K., Mooz, H. and Cotterman, H. 2005. Visualizing Project Management. Hoboken, NJ: John Wiley & Sons, Inc.

FORTUNE, J. and White, D. 2006. "Framing of Project Critical Success Factors by a Systems Model." International Journal of Project Management.

HUGHES, B. and COTTERELL, M. 2009. Software Project Management. London: McGraw-Hill, ISBN 978-0077122799.

LEWIS, W E. and VEERAPILLAI, G. Software testing and continuous quality improvement. Boca Raton: Auerbach Publications, 2005, ISBN 0849325242.

MACHNIAK, M. J. 1999. Development of a Quality Management Metric (QMM) Measuring Software Program Management Quality (Master's thesis, Naval Postgraduate School, Monterey, CA, 1999).

MARCHEWKA, J T. 2010. Information Technology Project Management: International Student Version/ Jack T. Marchewka. Hoboken: John Wiley, 2010, ISBN 978-0-470-40948-0.

MUNNS, A. K., & Bjeirmi, B. F. 1986. The Role of Project Management in Achieving Project Success. International Journal of Project Management, 14(2). (P.81-87).

PINTO, J. K., & Slevin, D. P. 1988. "Project Success: Definitions and Measurement Techniques." Project Management Journal, 19 (1), P. xix, 67-71

PROJECT MANAGEMENT INSTITUTE, 2004, A Guide to the Project Management Body of Knowledge (PMBOK ®Guide), Third Edition., P.41, USA: PA 19073-3299

RITTINGHOUSE, John. 2003. Managing Software Deliverables: A Software Development Management Methodology. Digital Press. ISBN 155558313X.

ROBERTSON, S., and Robertson, J., 2005. Requirements-Led Project Management. Boston, MA: Pearson Education, Inc. Media. ISBN 978-0-596-00948-9

ROYCE W. W., 1970. Managing the Development of Large Software Systems: Concepts and Techniques USA: IEEE WESCON, P. 1-9, ISBN:0-89791-216-0

STELLMAN, Andrew; and Greene Jennifer., 2005. Applied Software Project Management. O'Reilly. P. ix, ISBN: 978-0-596-00948-9

TAYLOR, J., 2004. Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware, and Integration Initiatives. New York: Amacom, ISBN 978-0814408117.

TUMAN, G. J., 1988. Development and Implementation of Effective Project Management Information and Control Systems. In D. I. Cleland and W. R. King (Eds.), 1983, Project Management Handbook. New York: Van Nostrand Reinhold Co. P.459-532

WIEGERS, K. E., 1996. Creating a Software Engineering Culture. New York: Dorset House Publishing. P.145-150, ISBN-13: 978-0932633330

WILLIAMS, M., 2008, The Principles of Project Management. Collingwood: Sitepoint, ISBN 9780980285864.

WYSOCKI, Robert K.,2006, Effective Software Project Management: Traditional, Agile, Extreme. 5th ed., Wiley Publishing, Inc., P.6-9, ISBN 978-0-470-42367-7.

YEATES, D. and CADLE, J., 2008, Project Management for Information Systems, Harlow: Pearson Prentice Hall, ISBN 978-0-13-206858-1

8 Appendix

Answers to question 1.5: Please elaborate on your answers and the variations in project efforts where possible.

- The project was completed according to the best practices and intended quality but took longer than originally planned due to problems in communication with the client
- The client requested multiple configuration changes.
- The vendor missing some points and even delayed finishing the UAT of the users after the deployment and go-live date.
- The project was delayed because the client was not very responsive to emails and requests from our side.
- The customer was taking a long time to make decisions about the requirements of the information asked from them.
- The quality could have been improved (by having functional experts or a better know-how/documentation) and the duration aspect was a lot longer than expected but the client approved the system.
- The quality is not that good cause corporate in software development culture is depending on an individual evaluation, not the whole project evaluation .. the project evaluation is project management responsibility.

Other challenging or well-functioning aspects of the project that was listed by respondents in questions 3.15, 3.16:

- Admin tasks
- Supporting teams
- Tools
- 3rd parties
- Training
- Soft skills