



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MUHAMMAD USAMA MAZHAR
CHALLENGES FACED BY STARTUP COMPANIES IN SOFT-
WARE PROJECT MANAGEMENT

Master of Science thesis

Examiner: Prof. Pekkola, Samuli
Examiner and topic approved by the
Faculty Council of the Faculty of
Business and Built Environment
on 09th March 2016

ABSTRACT

MUHAMMAD USAMA MAZHAR: CHALLENGES FACED BY STARTUP COMPANIES IN SOFTWARE PROJECT MANAGE- MENT

Tampere University of Technology

Master of Science Thesis, 77 pages, 01 Appendix (03) pages

November 2017

Master's Degree Programme in Industrial Engineering and Management

Major: International Sales and Sourcing

Examiner: Professor Samuli Pekkola

Keywords: Software Startups, Project Management, Project management in Startups, Challenges in Project Management, Software Project Management, Challenges in Startup Companies.

The start-up methodology has become quite common and there are a lot of young entrepreneurs already experimenting different ideas with the new start up companies and its been encouraged a lot too in business industries all around the world these days. There are actually no particular set of rules and standards in terms of business or processes that start up companies are required to follow and this is one of the main reason why the start up businesses have no guarantee of sustainability or in other words profitability initially.

Most of the start-up companies around the world that follow lean start-up methodology operate in the software industry and software management is the key challenge that has been faced by the companies which can lead to long-term success or failure of the company. Project planning generally includes resource planning, time scheduling, estimation of resources and financial costs in general which are also fundamentals of software related business. Effective and successful project management is an essential requirement for any type of software related business, the company can be offering software services which are called service-based software companies or product based or project based software companies, project management is the essential tool for any of the types.

The main objective of this study is to help other software start-ups. This thesis is targeting mainly the software start-up companies therefore, the main objective of the thesis is to address and discuss the main challenges faced by software-based start-up companies related to software project management which are studied analyzed and presented by literature review and theoretical framework, and then by empirical data gathered from two case companies the challenges have been classified into 3 major groups as 1) operational, 2) internal and 3) external challenges. In the end the potential solutions and proposals that are needed to overcome these challenges based on the empirical data has been discussed and also options for future research related to this topic have been provided.

PREFACE

While working as a software developer with the software start-up companies in the last 5 years, I realized the importance of effective software project management as an essential tool to boost business and as one of the key challenges in software start-ups and this was one of my motivation to write about this particular topic. Software start-ups or technological entrepreneurship have become quite common and so-called buzzword in modern markets since the adaptation of lean start up methodologies by different companies and markets around the globe.

The following study is aimed to develop a theoretical framework after the analysis of existing literature related to software project management. The study addresses the main challenges faced by software start-up companies related to project management and provide a set of best practices and proposed solutions to cope with them after the analysis of gathered data form case companies which can be helpful for other start up companies.

I would like to say thanks to my thesis supervisor professor Samuli Pekkola, who has been extremely helpful in defining the research scope, helping in the drafting process and whose guidance and feedback has enabled me to finalize and complete the thesis successfully. Special thanks to all of my friends at Tampere University of Technology, last but not the least I would like to thank all my family members, especially my parents without their prayers it was not possible, all my achievements are dedicated to them, my siblings and particularly my wife for the support throughout the completion of my studies at TUT and my stay in Finland.

Helsinki, 13.11.2017

Muhammad Usama Mazhar

TABLE OF CONTENTS

LIST OF FIGURES	VI
LIST OF TABLES	VII
1. INTRODUCTION	1
1.1 Background	1
1.2 Objective	2
1.3 Research Process	3
1.4 Data gathering methods	4
1.5 Context of the Research	5
1.6 Structure of the Thesis	5
2. RESEARCH METHODOLOGY AND SETTINGS	7
2.1 Research Design	8
2.2 Research Strategy	8
2.3 Research Execution	10
2.3.1 Presentation of Case Companies	10
2.3.2 Results and Analysis of the Research Data	10
2.3.3 Reliability and Validity of the Data	11
2.3.4 Ethical Consideration	11
2.4 Literature Review	12
2.5 Empirical Method	12
2.6 Software Project management	13
2.7 Introduction to Software Project Management	14
2.8 Normal Project Management vs. Software Project Management	15
2.9 Software Project methodologies:	18
2.10 Modern Tools for Software Project Management	20
3. TECHNOLOGICAL ENTREPRENEURSHIP	22
3.1 Introduction to Technological Start-ups	22
3.2 Start-up Lifecycle	23
3.3 Software development in technological start-ups	25
3.3.1 Process Management	26
3.3.2 Customer's involvement in Product Development	26
3.3.3 Organizational structure	27
3.3.4 Tools and training	27
3.4 Common software development practices and methodologies	27
3.5 Managerial and Organizational Practices in Software Start-ups	30
3.6 Software Quality Assurance Practices	31
3.7 Most Common challenges faced by software start-ups	32
3.7.1 Fierce Competition	32
3.7.2 Unrealistic Expectations	32
3.7.3 Building competent teams	33

3.7.4	Lack of resources and Funds	33
4.	LITERATURE REVIEW ABOUT MOST COMMON CHALLENGES IN SOFTWARE PROJECT MANAGEMENT	35
4.1	High competition from global companies.....	35
4.2	Lack of sufficient software-specific expertise	36
4.3	Testing and bug fixes leading to numerous software iterations.....	36
5.	EMPIRICAL RESEARCH	38
5.1	3 Interviews.....	38
5.2	Interview Execution	39
5.3	Case Company 1	39
5.3.1	Introduction to the company	39
5.3.2	Business Structure.....	40
5.3.3	Organizational Structure and structure of Development Team	41
5.4	Case Company 2	41
5.4.1	Introduction to the Company	41
5.4.2	Business structure	42
5.4.3	Organizational Structure and structure of Development Team	43
5.5	Comparison of Case Companies	43
6.	FINDINGS AND RESULTS.....	45
6.1	Organization structure and business domain	45
6.2	Software Development and Project management	47
6.3	Tools and platforms	48
6.4	Team management	50
6.5	Major challenges and main pitfalls that needs improvement.....	51
7.	DISCUSSION.....	53
7.1	List of main Challenges and Pitfalls with Software Project Management in Start-ups.....	53
7.1.1	Lack of necessary Tools and Infrastructure	54
7.1.2	Lack of specific Expertise for the Software Development.....	55
7.1.3	Mismanaged Resources	57
7.1.4	High competition because of Globalization.....	57
7.1.5	Vague Requirements.....	58
7.1.6	Unrealistic Time Scale.....	59
7.1.7	Poor Communication	60
7.1.8	Poor Planning.....	60
7.1.9	The problem of Poor management.....	61
7.2	Common challenges related to Software Project Management faced by both Case Companies	62
7.3	Criteria for Solution	63
7.4	Proposed Solutions.....	64
7.4.1	Setting of Well-defined and Clear Initial Requirements	64
7.4.2	Dealing with Unrealistic Customer Requirements	64

7.4.3	Managing Unrealistic Time Frame	65
7.4.4	Improving Poor Communication	66
7.4.5	High Completion and impact of Globalization.....	66
7.4.6	Tools and Infrastructure.....	68
7.4.7	Effective Management Process.....	68
7.5	Lessons Learned.....	69
8.	SUMMARY AND CONCLUSION	70
8.1	Proposals for Future Research	72
	REFERENCES	73

LIST OF FIGURES

Figure 1.	<i>General timeline and overview of the whole research process.....</i>	<i>3</i>
Figure 2.	<i>Structure of the thesis.....</i>	<i>6</i>
Figure 3.	<i>Model of research methodology (Adapted from Saunders, Lewis).</i>	<i>7</i>
Figure 4.	<i>Components of Project management.....</i>	<i>14</i>
Figure 5.	<i>Triple Constraints for Software based Projects.....</i>	<i>15</i>
Figure 6.	<i>Traditional five-phased project management cycle</i>	<i>16</i>
Figure 7.	<i>Six phases for life cycle of a start-up (Adapted from Kane 2004).</i>	<i>24</i>
Figure 8.	<i>Basic water- fall model for software development process.....</i>	<i>26</i>
Figure 9.	<i>Lean vs Agile</i>	<i>28</i>
Figure 10.	<i>Software Quality assurance process</i>	<i>31</i>
Figure 11.	<i>Empirical Research process to gather data from Case companies</i>	<i>38</i>
Figure 12.	<i>Organizational Structure of Case Company X</i>	<i>41</i>
Figure 13.	<i>Organizational structure of case Company Y.....</i>	<i>43</i>
Figure 14.	<i>Classification of major challenges in software project management in start-ups.....</i>	<i>54</i>
Figure 15.	<i>Common challenges related to software project management in software companies</i>	<i>63</i>

LIST OF TABLES

Table 1.	<i>Data gathering methods (Gummesson 1993).....</i>	<i>4</i>
Table 2.	<i>Differences between the Interpretivism, Assumptions and the positivism approach</i>	<i>9</i>
Table 3.	<i>Comparison between normal project management and software project management</i>	<i>17</i>
Table 4.	<i>Worlds top 20 start-up ecosystem with ratings</i>	<i>22</i>
Table 5.	<i>Major differences between lean and agile methodologies</i>	<i>29</i>
Table 6.	<i>Functional themes for data gathering and target interviewees from case companies.....</i>	<i>45</i>

LIST OF SYMBOLS AND ABBREVIATIONS

B2B	Business to business
B2C	Business to Consumer
CTO	Chief Technology Officer
CEO	Chief Executive Officer
COO	Chief Operating Officer
DSDM	Dynamic Systems Development Model
FDD	Feature Driven Development
JAD	Joint Application Development
LD	Lean Development
PAAS	Platform as a Service
RAD	Rapid Application Development
RUP	Rational Unified Process
SAAS	Software as a Service
SDLC	Software Development Lifecycle
TEKES	Tekniikan edistämiskeskus (Centre for Advancement of Technology).
XP	Extreme Programming

1. INTRODUCTION

1.1 Background

A start-up is generally said to be a company which follows a particular lean start up methodology and plans to work on a certain business idea but does not know how sustainable it is going to be as the success is not always guaranteed in start-up ideology and there is always a risk involved with the start-ups in general (Blumenthal, 2013). There are actually no particular set of rules and standards in terms of business or processes that start up companies are required to follow and this is one of the main reasons why the start up businesses have no guarantee of sustainability or in other words profitability. The start-up methodology has become quiet common and there are a lot of young entrepreneurs already experimenting different ideas with the new start up companies and it is being encouraged a lot too in business industries all around the world.

Software start-ups or technological entrepreneurship have become quite common and so-called buzzword in modern markets since the adaptation of lean start up methodologies by different companies and markets around the globe. A general survey depicts that most of the start-up companies appear to start with a technological idea and therefore referred to be called as technological start-ups, this actually means that most of the start-ups appear to be using software as products or software-based services to be their core base of business ideas (Blodget, 2013).The newly formed and usually small sized enterprises are generally fast-growing business ideas which are aiming to meet a modern business and market needs need by developing or offering an innovative product, process or service. These businesses are usually involved in the design and implementation of mostly new and innovative methods, techniques, tools or services.

Most commonly software start-ups are classified into service-based, product-based or project-based companies. In today's modern global and technological oriented world, managing software based projects is one of the most crucial task and that this is mainly due to complexities and technicalities associated with the software based projects and variables associated with this sort of businesses. There are a number of variables and entities that are involved form management point of view while managing software projects.

Planning is essential for any sort of business to grow and become sustainable, in case of software or technological start-ups project management is the essential component that is required for the long term sustainable planning of the firm. Nowadays and with the advancement of technology by time, tools and methodologies, in general, project man-

agement has become an essential pillar of any type of company. Software project management in general, revolves around the process building and planning of the activities involved in performing a particular project, this includes scheduling, creating set of activities needed for project completion and assigning the activities and roles to the resources also known as persons who are responsible for particular tasks or activities, the main person responsible for all these activities is the Project manager but since the start-ups have limited number of resources and budget, this has become a bit more complex. The literature related to project planning activities or the project management activities in software start-ups is bit scarce and it depends on the domain of particular business since there is no standard set of best practices that companies follow but the main idea of the thesis is to address the pitfalls that usually most software start-ups have related to project management activities since the advancement of software start-ups is increasing day by day in the software industry (Crow, 2003).

1.2 Objective

Project planning which generally includes resource planning, time scheduling, estimation of resources, costs etc. is an essential requirement for any type of business, particularly for software related business be it any type of company offering software services for instance service-based software companies, product based project based software companies project management is the essential tool , this thesis is targeting mainly the software start-up companies therefore, the main objective of the thesis is ...

... to address and discuss the main challenges faced by software based start-up companies related to software project management and to come up with potential solutions and proposals.

Based on the objective described above the thesis will address the following research questions:

- 1) What are the major challenges that start-up companies face related to software project management?
- 2) What are the common causes of major pitfalls and challenges for software project management in start-up companies?
- 3) What are the proposed potential solutions or measure needed to overcome major challenges and pitfalls in software project management for start-up companies?

The following study is aimed to develop a theoretical framework for the analysis of existing literature related to software project management. The study addresses the main challenges faced by software start-up companies related to project management and provide a set of best practices and proposed solutions to cope with them after the analysis of gathered data form case companies.

1.3 Research Process

In order to achieve the main objective of the thesis study for the masters degree it needed to start with some research topic, a research question or potential research idea, it all started with the idea generation in 2015 when the author of this thesis was working as a software developer in a start-up company and at the same time completing his master's studies, he was actually observing the working practices and processes that he thought were major challenges in software project management, these challenges particularly relate to methods and practices that the companies where the author worked were using. It basically started with a rough idea in mind according to his previous experience in software companies so he started initial discussions with the thesis supervisor during the beginning of 2016 and now he had to come up with a thesis topic or research question to start the study in formal manner. The main motivation was the idea that he always wanted to study more about project management and like to continue working with it as a part of his career further so it was a good opportunity to study and utilize the existing knowledge in the field of project management. The main phases of the whole research process and timeline is described below in Figure 1.

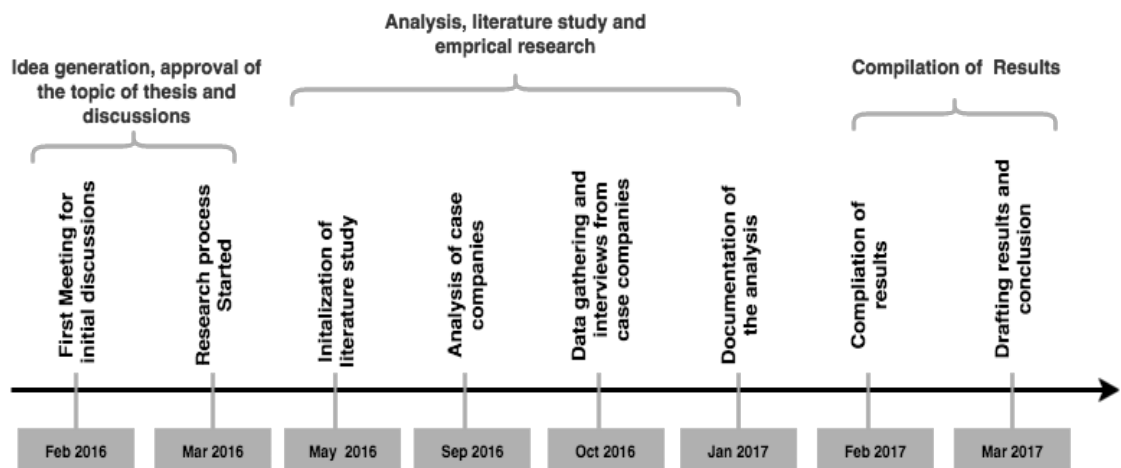


Figure 1. General timeline and overview of the whole research process.

The research process mainly consists of 3 major phases, these phases included idea generation, literature study and empirical research that includes interviews and empirical data gathering and the final phase is compiling the results and drafting the conclusion. The first meeting with the thesis supervisor was held in Feb 2016, where the initial idea of the research was discussed and the research questions and the initial topic of the thesis was decided. The final topic was approved by faculty council in March 2016 and the research process was started. The drafting process was actually started in summers 2016 from May till August where initial 3 chapters were completed, the empirical data gathering and interviews were conducted in September and October 2016 and the results and conclusions were drafted by the starting of this year. Since the author was also working

as a software developer in one of the case companies on the fulltime basis so the whole process took a bit more time than planned first draft was submitted for the review in summers 2017 and the final draft was completed and submitted by the end of the year.

1.4 Data gathering methods

Research is basically a process of finding out new and useful information after the analysis and study of existing material in logical ways, this generally happens by utilizing the facts and figures that are already present related to the particular field. Studying and analysis of the existing literature and gathering empirical data through the current practices or studies is always involved in the research process (Rajasekar et al. 2006). Various forms of data gathering methods can be used in order to conduct a research for a particular field, theme or a topic. According to Gummesson (1993) there are five main research methods that can be used to conduct research for any theme topic or field they are described below in Table 1.

Table 1. Data gathering methods (Gummesson 1993).

Research Method	Description
Existing literature	It is referred to the existing data or any secondary source of information that has been published by other parties for example books, articles, publications journals etc.
Questionnaire surveys	This is referred to data that is gathered from different types of feedback channels and surveys
Interviews	This is referred to a structured set of open-ended questions or surveys intended for a particular research topic. Most commonly used in academic writings.
Observation	The research data that has been gathered after the observation through a particular subject of study.
Action research	The type of research that can contain all other types of research methods and requires the active involvement of the person or author doing the research.

According to Gummesson (1993) there are five different research methods that can be applied to a particular field of research, for this thesis the existing literature, questionnaire, interviews and observation methods are used.

1.5 Context of the Research

The main context of this research is to address the practices and standards that are required for project management as a main part of strategical plans for a particular company, identify the most common techniques that most start up companies are using for their business and to compare it with the standard practices and methodologies according to the literature study carried out for software project management practices and tools.

This research is carried out to study and analyze the main challenges that early stage software start up companies face, the study in result can beneficial be helpful for other start up companies operating in the same domain. The main context also involved the fact that the major experience and expertise of the author of this thesis is also in the related field that is mostly working with the start up companies in Finland as a software developer, this research was also carried out to utilize the previous experiences and some knowledge in the field, so finally to document the challenges that software companies face in everyday situations related to software project management.

1.6 Structure of the Thesis

The thesis is mainly divided into 8 main chapters, chapter 1,2,3 and 4 are related to the literature study where chapter 1 describes the background of the thesis topic, initial motivation for research and main objective of the thesis, chapter 2 is about research methodology and settings which includes information about the research methodologies used and introduction to software project management and also creates an initial theoretical framework. Chapter 3 is about the introduction to technological entrepreneurship and technology start-ups, chapter 4 explains the existing literature about software project management and common challenges already addressed through various researches in the existing literature. Chapter 5 and 6 and 7 are related to the empirical study and include the introduction to the case companies, the comparison for the two case companies is also presented, it also consists of results from the empirical data and discussion. In the last summary and reflection of the study are concluded in chapter 8.

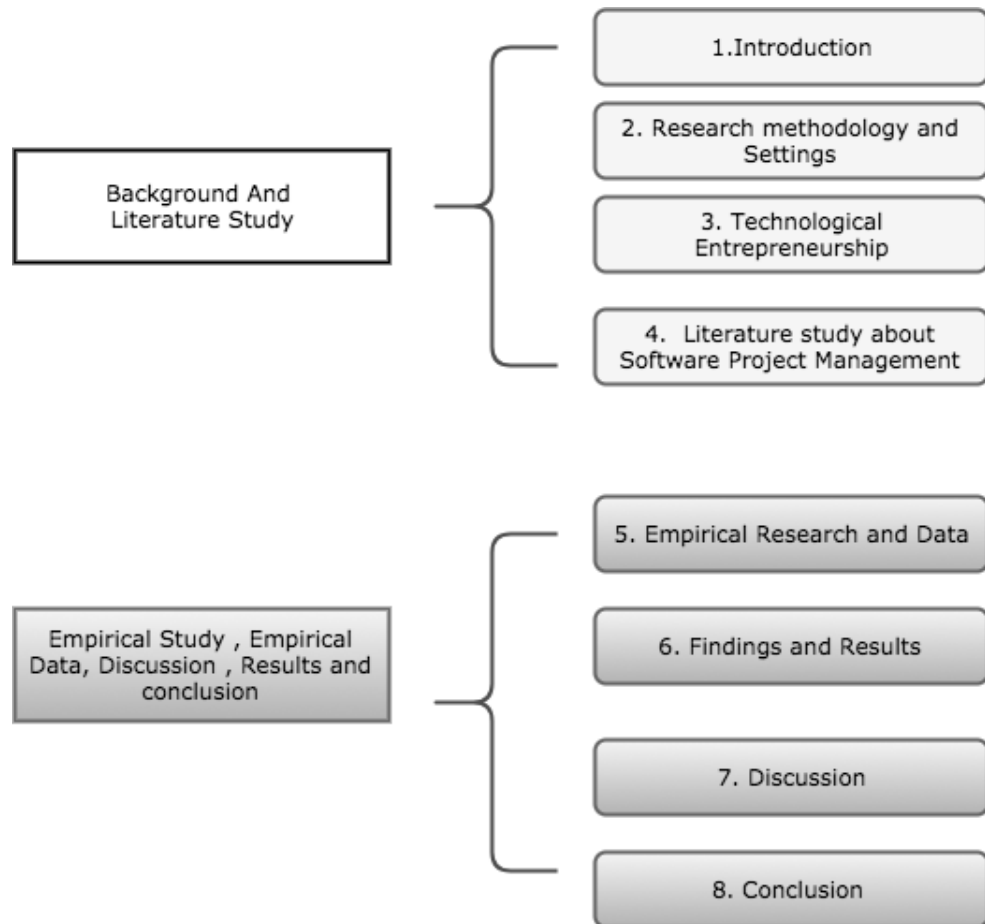


Figure 2. *Main Structure of the thesis*

Figure 2 explains the main structure of this study and the thesis that is divided into 2 major parts the background and literature part and the empirical part, the background part consists of background of the study, literature review and settings and the empirical majorly consists of the data gathered during the study, introduction to case companies followed by discussion and proposed solutions.

2. RESEARCH METHODOLOGY AND SETTINGS

This chapter contains the description of all the procedures and methods that have been applied throughout the research process to collect data and fulfill the research questions and objectives. Furthermore, later in the chapter, the general literature about Software start-ups is presented.

The methodological approach as described by the authors underpins the philosophical approach as also chosen for this research topic. The research methodology will generally follow some aspects of the onion model as developed by (Saunders, Lewis & Thornhill, 2009). The onion structure is a representation of a model which guides the researcher in answering the research questions in methodology and in philosophy (Saunders, Lewis & Thornhill, 2009). Figure 8 general representation of the Onion model.

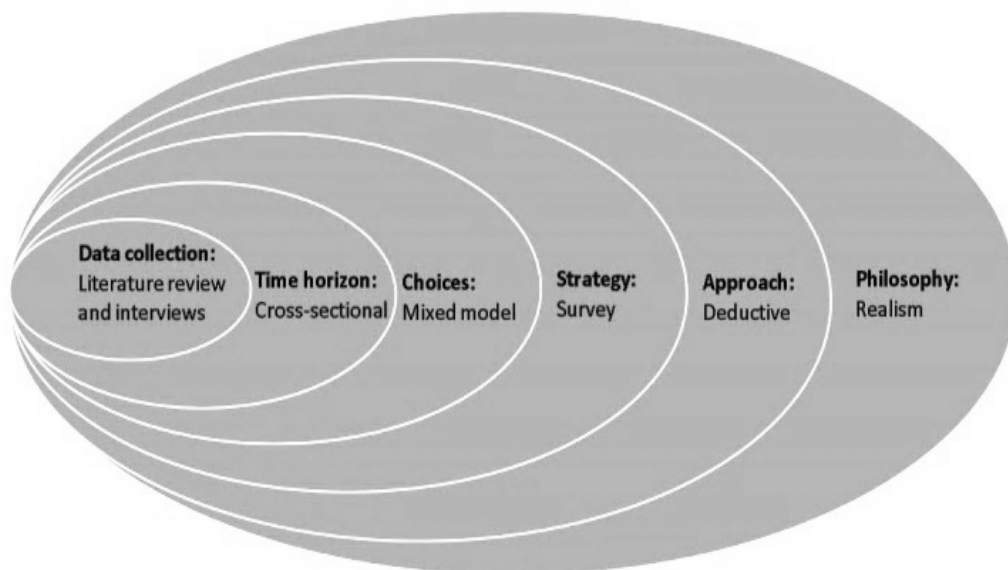


Figure 3. Model of research methodology (Adapted from Saunders, Lewis & Thornhill, 2009).

As explained in the figure 3 the outer most section is the philosophical approach used in the research process. It shows the different approaches to the research philosophies. Basically, two main research philosophies are discussed. The ontology and the epistemology; Epistemology contains the discussion of the acceptable knowledge and the most viable means of accessing information regarding it. Ontology emphasis on reality

(Hirsjarvi et al. 2007). In regard to this thesis, the main focus is the challenges of software start-up on the project management. It is evident that the research does not need to unravel the reality concerning the challenges software start-up face in regard to project management. The reality of such information does not have any productive impact. Rather, the best strategy is to acquire information concerning real challenges faced by software start-up with project management in practice, combine the patterns and come up with evaluable themes and results. The results are then in this way justified and impactful, in this regard, the epistemological philosophical approach was the most suitable for the thesis.

2.1 Research Design

The research follows cross-sectional research design. The main objective of the study to pinpoint the challenges faced by the software start-up in the project management; It will be met by comparing two start-ups software companies in Finland operating in Helsinki. The companies have been re-named “**Company X**” and “**Company Y**”. Company X is purely product based offering software products as a service whereas company Y is project based and offers software solutions like web development, digital services etc. More detailed information about the case companies is provided later in chapter 5.

Cross-sectional research design provides analysis basing on the existing differences within the groups of interest (Menard, 2008). The main focus of this particular research design is also to identify main differences in challenges within the groups. The design however, does not have a particular dimension. The cross-sectional design does not an emphasis on change rather it measures differences in phenomena, subjects, and the variety of people (Menard, 2008). The researcher findings are based on inferences accumulated through the passive approach. The interview questions were designed based on 5 themes related to software project management, the designed interview themes identified through action research and literature review mainly. The main objective was to interview the main players in the company who are directly associated with software project management practices and methodologies in the company, for example, CTO, project manager, team lead or software developer.

2.2 Research Strategy

Qualitative research strategy was applied generally in this research process. The strategy was both interpretive and inductive. Howe, (2003) argues that Interpretivism contains diverse approaches such as hermeneutics, phenomenology, and social constructivism. The approach undertakes the idealism position in philosophical studies (Buchanan, 2011). The Interpretive approach is applied by a researcher who intends to appreciate the differences of the subjects under study (Williamson, Bow & Charles Sturt University, 2002). The main focus according to the approach is the meaning and it is allowed to

employ different approaches to arrive at the meaning (Williamson, Bow & Charles Sturt University, 2002). In this regard, Helsinki region was selected and two start-ups were chosen for the study these are also the companies researcher has been working with-in the past and same domain as a key player in software development process.

Research can either employ inductive or deductive approach. The deductive approach entails looking an already existing theory (Guthrie, 2010). Deductive approach the studies looks at creating a theory based on the development of a theory (Guthrie, 2010). The area of software start-up and project management is widely studied. The deductive approach starts with hypothesis while the inductive approach starts with research questions as research foundation (Guthrie, 2010). The inductive approach entails employing a different perspective of the already studied field while deductive approach lays emphasis on causality (Guthrie, 2010). Inductive approach is applied more to the qualitative research while the deductive approach is applied to the quantitative research. The approach is beneficial for acquiring detailed information that will be good in resolving many problems that are faced by the people in different organizations. The right research process and approach can be critical in the long run to succeed and achieve more progress. The issues that are faced by the organizations can be resolved by conducting proper research processes that will be beneficial in the long run and can also provide future research opportunities.

Table 2. *Differences between the Interpretivism, Assumptions and the positivism approach*

(Adapted from Williamson, Bow & Charles Sturt University, 2002)

Assumptions	Interpretivism	Positivism
Reality of nature	Multiple, socially constructed	Single, Tangible, Objective
Research Goals	Weak prediction, understanding	Strong prediction, Explanation
Focus of interest	Deviant, unique and specific	Representative, average, and general
Generation of knowledge		Laws

2.3 Research Execution

The representatives from the start-up case companies selected in the study were interviewed for this face to face interviews were carried out the interview questions were designed and divided into functional themes. The interviewer questions were semi-structured and the sessions were allocated 45- 60 minutes each. The aim was to reach out two to three representatives for the interview from both companies since the main focused was to gather information from the main players of software project management within the company. The interview responses were documented as a word document. After the interviews, transcribed data was double checked to match the main themes and patterns. The most collate patterns formed the themes which were synthesized and given an interpretive analysis method. The results were compiled and discussion was documented. The thesis, in the end provides the conclusions, recommendations, and the future perspectives.

Thematic data analysis is one of the main methods in qualitative data analysis. It entails recording, examining, pinpointing the main themes within a dataset (Guest, MacQueen & Namey, 2012). This thesis also applies thematic data analysis which was collected from the interviews form case companies, this is achieved by examining the patterns which describe the phenomenon under study. The research familiarizes with the patterns and develops codes based on their similarities. During the research process, data is analyzed to get familiarize with it and conduct a rich discussion basing on the findings. It will entail the identification of the explicit and implicit ideas.

2.3.1 Presentation of Case Companies

The anonymity of the companies and the representatives is also important since the study was being carried out entirely for educational purposes and using an empirical research methodology, the companies have been renamed and names of the representatives of the interviews have not been disclosed. The confidentiality of the information and the company status was maintained as a drive of encouraging further research in this category.

2.3.2 Results and Analysis of the Research Data

The main findings of the whole research study are presented. The paper presents the thematic analysis of the transcribed data. Interview responses are also documented in chapter 6 in findings section. Five main themes for the challenges related to software project management are discussed as provided in the interviews guidelines the interview is attached in the appendix section. The Thesis has provided a comparative overview of the companies based on their main goals and objectives and business structure, the current general strategy and their historical background. The comparative section will en-

hance the process of interpreting the common themes and provide a generalized conclusion.

In this regard, project management strategies undertaken by the case companies are discussed later in empirical research chapter. The practice will be efficient in unleashing the project management tools undertaken by the case studies to arrive at their respective strategies. As also discussed earlier the interview questions are analyzed through the application of the thematic analyzing method.

2.3.3 Reliability and Validity of the Data

Reliability in research refers to the accurate and consistent development of the research questions following the objectives and aims of the research (Golafshani, 2003). Under the qualitative study, reliability entails the similarity, stability and the repeating of the measurement method for a given period (Golafshani, 2003). In this regard, consistency was ensured by developing interview questions which were consistent with the objective of the study. Reliability is achieved through re-testing of the instrument for the study of stability (Golafshani, 2003). The thematic data is gauged against the research objectives with the aim of establishing the reliability of the information from the interviewees.

Validity is the measure of the accuracy of the research findings in regards to the objectives of the study (Golafshani, 2003). The validity of research is established and the information from the introduction to the conclusion is following consistency in examining the challenges faced by the software start-up in the project management. The main concepts checked to prove reliability and the validity of the data is through checking of transferability, credibility and precision (Golafshani, 2003). Triangulation is the most important method used in proving the reliability and the validity of the data (Golafshani, 2003). Triangulation is achieved through combining the secondary research (empirical research) and the primary research (literature review and action research) to prove the objectives of the thesis.

2.3.4 Ethical Consideration

Any kind of study has its own ethics. There is always a tension between the privacy of the participant and use of information gathered to make a generalization (Patton, 2000). Ethical principles are important in ensuring the right of every participant is observed (Patton, 2000). The main ethical issues which arose from the use of interviews as the main method of data collection entailed the design used, subjective interpretation of the data and the participant/ researcher relationship. The main objective of the research was to examine the challenges faced by the software start-up in project management. The names of participants have not been disclosed and case companies have been renamed on demand.

2.4 Literature Review

The literature review is about the ideas that are underlying behind or actually supporting the thesis. It is important for getting a very clear picture that shows the process of conducting the research properly. The research focuses on empirical data but at the same time literature review also. These both steps are properly explained as they support the research process well. There are some reasons for the importance of literature review for a thesis that are explained by (Saunders 2009). The first reason for the literature review is that it is a proper tool that is important for preliminary research in order to enhance the research topic and problems. The second important reason is that the literature review actually provides the base for the findings. The process is called critical literature review that was later called as the literature review and it is part of the final research paper (Saunders, 2009).

The findings of the research are also used for properly rationalizing the findings of the research that is very much important. The literature review process allows the people to find the keywords of their topic properly in other research papers in order to highlight the findings of the research that was conducted on the same topic or keywords by other people. It is a very nice starting point as the researcher can get great insight and useful information as the research process is moving forward. The electronic papers are important and serve as a nice source of information and apart from them, there are other important best practices papers and standards that are critical for the research process. The challenges that are faced by the researcher can all be actually resolved by focusing more on the literature review. The more the literature review the better the quality of the research. If the researcher is finding something difficult or any concept is very tough that can be resolved by assessing the information that is presented in appropriate

The literature review shapes the entire structure and process for the research as it allows proper understanding of the different concepts and techniques that are very much critical for the researcher in the long run for this reason, the existing literature on project management, start-ups and software development methodologies has been studied.

2.5 Empirical Method

The aim of the empirical study is the proper understanding of the organization that is why the use of the empirical method for that is critical. It is important for any research methodology to approach the empirical research as the appropriate results will be achieved by focusing on the appropriate techniques. If there are certain theme interviews that are conducted in the particular organization that will be giving an explanation that is very much detailed and giving proper results for approaching of the interview. The research question is clearly trying to find the gap that is present in the case organizations that is why it's important for the researcher to understand the nature and practices of the case organization. The challenges or situations that are faced by the organiza-

tion can be resolved by proper and detailed analysis of the information that is obtained from the research. The interviews that are used for the collection of the data are a rich resource as that will be allowing proper and detailed analysis of the research.

The data that is gathered for proper analysis will be beneficial in the long run. The interviewee's knowledge will be assessed that will also be beneficial in the long run (Hirsjarvi, 2007). The research using the interviews for the questions and the assessment is critical for the properly developing a better understanding of the situation. The interviews are actually conducted in order to focus on the certain areas of the interest of the researcher to further explore more detailed and useful information. The highlighted information from the interview data can help the researcher to find the appropriate solutions to the problems that are being faced and there are many alternatives or ideas that are highlighted as the research process allows more information to be collected rather than restricting the information. (Saunders,2009) have actually clearly highlighted the main challenges that are three major issues in the overall data quality that are present in semi-structured interviews or the in-depth interviews. Firstly, the reliability of the information, secondly there is some form of bias and finally, the third issue or challenge is the validity and generalization.

The actual generalizability or validity of the research findings is very much important for the final research paper. The problems can also be resolved or properly controlled by focusing on conducting more interviews from the same areas. The same question can be asked of several employees or the same group that will be allowing the possibility to generalize the information on all the people. The problems that are present in the generalizability can be properly tackled by focusing on the answers and suggestions.

The research approach that is adopted in this thesis is actually quite generally focusing mainly on the challenges that might be faced by the organization. The themes that are properly analyzed in the research questions or interviews are very much of qualitative and quantitative nature that allows more useful and proper information that is critical for the proper completion of the research process. The data of the interview is divided into categories or themes that will be critical in the long run leading to a proper solution and further it can then be analyzed along with the numerical data that is having empirical nature.

2.6 Software Project management

Project management is a combination of processes, skills and tools also explained in figure 4, these are combined to plan a particular project that includes analysis of risks involved in the project, time management also known as project scheduling, budgeting and allocation of resources. These components combine together to manage a project properly, if the operations of the company are related to software business, the skills and tools are dedicated to software project management particularly. This chapter will

analyze various aspects that are related to software project management. A clear distinction between normal project management and software project management will be discussed. The methodologies and modern tools used in software project management will also be explored. Challenges encountered by modern software project managers in their job will be highlighted and possible solutions suggested or recommended.

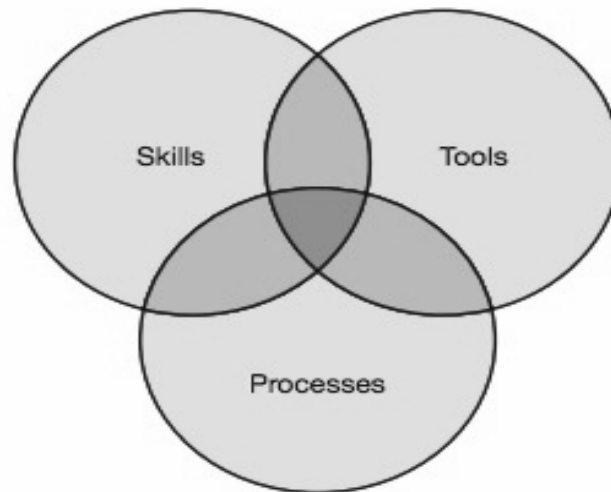


Figure 4. Components of Project management (Adapted from Hobbs 2011)

2.7 Introduction to Software Project Management

A software development company executes its duties in two parts- software creation and software project management. Software creation is the actual processes that lead to the development of the software. Software project management encompasses the processes and activities that are carefully monitored by the software development manager to ensure that the final product is same as that which had been planned for by the founder or management. Every project faces its own share of constraints (Villaforita, 2016). Managing these constraints is the responsibility of the founder, software development manager, management and the software development team. Figure 5 shows how the triple constraints interact. There is a direct interaction between time scope and cost whereas the quantity acts as a uniform entity between the interaction, the major entities however are always time and cost since the scope is always defined in the beginning and often remains unchanged and uniform through the whole project management cycle.

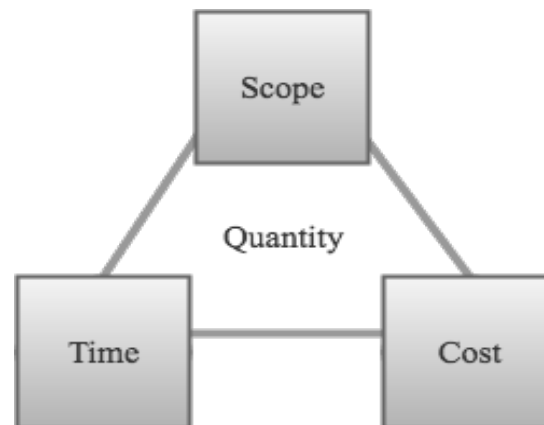


Figure 5. *Triple Constraints for Software-based Projects*
(Adapted from Mall, R., Panigrahi 2017)

Software project management is often spearheaded by a software project manager who is generally responsible for all the activities related to software project management and planning for all projects in a company. The manager is supposed to manage all aspects of software development and ensure that all phases of the software development lifecycle are concluded as and when planned (Schwalbe, 2009). In executing the responsibilities of the office of a software project manager, the office holder will be responsible for managing the software development team and also manage the project. As regards the management of the project, the software development manager will have to undertake activities that include planning of the project, setting the scope of the project, coming up with a budget for the project, managing the resources for the project and scheduling the various tasks and events that need to happen in the course of the project. The activities that relate to the project which must be carefully monitored by the software project manager can be split into three categories: project planning, scope planning and project estimation (Schwalbe, 2009).

2.8 Normal Project Management vs. Software Project Management

Lester (2013) defines project management as a discipline which allows the project team, under the able leadership of a project manager to initiate, plan, execute, control and close the work that had been assigned to a team after having achieved the set goals or met the success criteria that had been forethought. Most of the traditional or normal projects have a defined beginning and a defined end and they are pursued with an aim of producing a unique product, result or service. Additionally, most of these normal pro-

jects are constrained by time, funding or deliverables and they are intended to result into an added value or beneficial change to the project sponsor.

The temporary nature of normal project management undertakings is a sharp contrast from other sub-categories or sub-disciplines of project management which are often semi-permanent, permanent or repetitive activities aimed at producing unique services or products. An example of such a sub-discipline is software project management which is neither temporary nor short-lived. (Mochal,2011) opine that the greatest challenge faced by normal project managers comes from the pressure to achieve all the goals set for the project within the set time, funding or deliverables constraints. Some more specific constraints that normal projects face include the constraints of time, scope, budget and quality. The secondary constraint, which is deemed to be more ambitious, requires that the project manager to optimize the allocation of the required resources and integrate them to achieve the objectives that had been pre-defined.

Traditional or normal project management follows a number of approaches to meet the objectives set for projects. These approaches can be summed up as lean, incremental, iterative, and traditional phased approaches. Of all these approaches lean and traditional phased approach is the most common approaches used in normal project management. Traditional-phased approach has a sequence of steps that need to be followed to complete the project (See Figure 4). These steps or phases include initiation, planning and design, execution and construction, monitoring and controlling and completion of the project (Lester, 2013).

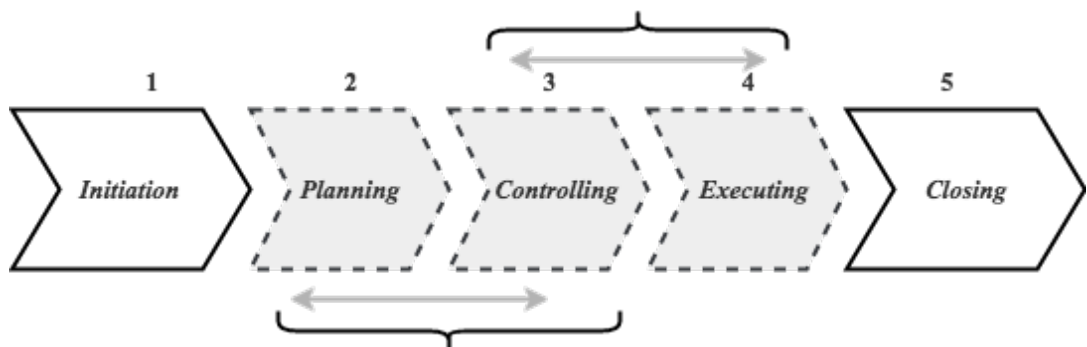


Figure 6. *Traditional five-phased project management cycle*

(Adapted from Mall, R., Panigrahi 2017)

According to (Hobbs, 2011), lean project management is an adaptation of concepts such as lean thinking and lean manufacturing into the discipline of project management. Janes et al, (2016) posit that lean project management is guided by the principle of delivering added value with less waste of resources. Lean project management has some sem-

blance to agile software development in the sense that it uses key techniques that are used in agile software development that include pipelining, the need of having a strong facilitator and blame-free employee engagement. Lean project management focuses on removing bottlenecks that are inherent in traditional project management, with an aim of accelerating growth and increasing productivity (Hobbs, 2011).

Major differences are related to customer involvement and processes, enabling a customer feedback in the whole project management process enables it to be a totally different approach towards completion, most common differences in normal project management and software project management are defined in table 3.

Table 3. *Comparison between normal project management and software project management*

Normal Project Management	Software Project Management
Customers are engaged early in the project but keep off once the project begins	Customers are involved throughout the lifetime of the project
There is heavy upfront design and analysis	There are daily meetings that teams engage in to ensure that they are up to speed with the new requirements.
Teams escalate problems to managers whenever they arise.	Teams work together to solve any arising problems
Anticipation is favored in this model	Adaption is favored in this model
The focus is on the process more than the product	The focus is on the product more than the process
The product is planned intensively before being executed and tested	Testing is done throughout the lifecycle of the project

Contrastingly, Software project management bears characteristics that distinguish it from the traditional or normal project management. Software project management is not a discipline in itself rather it is a sub-discipline within the wider project management discipline. As the name suggests, software project management concerns itself with the management of projects that deal with the development of software, description and comparison is explained in table 5 above. (Stellman & Greene, 2006) define software

project management as an undertaking which is primarily concerned with the planning and leading of software projects. Simply put, software project management looks at how software projects are planned, monitored, implemented and controlled.

Unlike normal projects that have a definite start and end, software projects have a start but they do not have an end. The very nature of the products of the software development project makes it impossible to have a defined end of the software project development journey. Software needs to be closely monitored and updated regularly. The ever-changing nature of the workspace where this software is deployed requires developers to work on their product on a continuous basis and this explains why software developers keep updating and releasing new versions of the same software product. Another marked difference is the fact that software project management follows or uses agile project management, which allows developers to respond fast to the changes in the software sub-industry. (Chin,2004) defines agile project management as an incremental or iterative method used in managing the design of activities of information technology, engineering and other business areas which operate in highly uncertain environments. Agile project management is ideal for software projects because of their uncertain nature and also because of the fact that such projects require unique expertise.

2.9 Software Project methodologies:

Software development methodologies are defined as frameworks that are used in the structuring, planning and controlling the processes of developing a software product (Elsafi & Awad, 2014). Examples of these methodologies include scrum, crystal methods, spiral, agile software development, dynamic system development model, feature-driven development, lean development, rational unified process, extreme programming, rapid application development and waterfall. Different authors classify software development methodologies in a different way. For instance, (Highsmith & Cockburn, 2001), posits that methodologies such as scrum, dynamic systems development model and crystal methods fall under agile software development methodologies. In this section, we will review these agile methods because they are the most common in the software development workspace.

1. Agile Software Development Methodologies

Agile software development methodologies share one characteristic in common; they all attempt to minimize risk by facilitating the development most of the software in short time boxes, which are known as iterations. Each of these iterations run for an average of four weeks and they can each be regarded as being miniature standalone software projects meaning that they comprise of tasks that are necessary to release new software. This is an important distinction between normal projects and software projects. Whereas normal projects only release the final product at the end of the development cycle, soft-

ware projects that use agile development methodologies aim at releasing new software at the close of each iteration (Cockburn & Highsmith, 2001).

2. Crystal Methods

This agile software development methodology, which was developed by Cockburn focuses on people, community, interaction, skills, communications and talents as the primary focus in software development. Cockburn observes that the various teams involved in developing software have diverse skill and talent sets which must be understood and prioritized above the process element, which he regards as being secondary (Bjornvig & Coplien, 2014). Further, he defines the behavior of individuals within teams in terms that include people being unable to act consistently over time, being fond of communicating on a face-to-face basis and getting real answers to real questions and the fact that behavior is highly variable. These findings led him to assert that the adopted software development methodology must be able to fit the agile workspace where these people work and this led to the development of crystal methodology as a type of agile software development methodology (Bjornvig & Coplien, 2014).

3. Dynamic Systems Development Model (DSDM)

DSDM is an agile software development methodology that evolved from rapid application development practices. This approach boasts of being best in coming up with documentation and training for teams involved in agile software development. Further, DSDM is cognizant of the fact that software is never built perfectly the first time and as thus, it considers the software development process as an exploratory endeavor. This methodology is guided by principles that include active user involvement, cooperation and collaboration between the various stakeholders, focuses on frequent delivery of iterations, empower teams to make decisions and an integration of testing within the lifecycle of the project (Geert, 1994)

4. Scrum

This agile software development methodology was developed by Ken Schwaber. It was originally intended to help in improving team productivity which was previously under the stranglehold of process-laden methodologies. Scrum emphasizes on team work. Teams meet in brief daily meetings in which progress is explained, all upcoming tasks are discussed, obstacles are dealt with and all backlog is prioritized over fresh work. The workspace is interactive and assigned tasks are completed in short iterations called sprints (Schwaber, 2002). A scrum master is assigned the responsibility of helping the team to navigate past impediments and deliver the goals set for various sprints. Scrum acknowledges that the challenges encountered in the agile software development workspace cannot be addressed using traditional process-based methodologies (Schwaber, 2002).

2.10 Modern Tools for Software Project Management

Every project, software development included, relies on a number of tools to bring about the envisioned success. Great tools can help in simplifying project activities, facilitate communication with both internal and external stakeholders and improve work efficiency. Though the tools discussed in this chapter are great, they do not replace sound software project management. To achieve the best results from these tools, it is advisable that they are combined with the tenets of sound project management.

There are a number of useful tools that can be used to ensure that software development projects proceed seamlessly. The tools can be classed under different categories including- planning, communication, execution and monitoring tools. These tools belong to the wider category of productivity tools. Examples include Email tools (Gmail, Yahoo etc.), Document sharing tools (Dropbox, Google Drive etc.), Video/web conferencing tools (Zoom, Skype etc.), Repository hosting tools (GitHub, Bitbucket etc.). Team management tools like (JIRA, Trello, Team week, etc.). These tools are ideal for use in agile software development because they enable the developers to be flexible and still maintain organization. Further, these tools enable developers to self-manage and work in such a way that their goals converge to meet the bigger goal set for the project.

These agile tools support the entire software development project by helping the team members to identify the various requirements and split these requirements into smaller tasks. The cycles or iterations are kept short and interactive enabling the team adjust their processes and focus. These tools share a common feature most have a graphical dashboard that makes it possible to check team progress in real time. Some have sophisticated features that make it possible to update the software code at the end of each iteration.

1. Source Control Tools

GitHub makes it possible for developers to follow different paths in coming up with different parts of the software and then merge their codes later. This repository allows software developers to work independently and then merge their codes at the tail end of the project. Many software development teams use GitHub to host and keep their codes neatly organized. Other source control tools include CVS, Subversion, Mercurial and Bitbucket (Straub, 2015).

2. Communication Tools

These tools are further categorized into two email tools (Gmail and Yahoo) and video conferencing tools (Zoom and Skype). There are instances where the team might need to share some piece of information through email. This can be done by using the popular email platforms or use an institution hosted email address or service. Teams can also

have meetings in real-time regardless of their location. This can be facilitated by the use of platforms such as Zoom and Skype (Javadtalab et al., 2015).

3. Continuous Integration Tools

These tools are important because they make it possible for developers to add a layer of processing whenever a code is committed. This is important because it makes it possible for teams to work smoothly together. These tools have plugins that facilitate the running of diagnostic tests meant to ensure that the software is performing correctly. They include tools such as Travis CI, Integrity and Strider (Cuppett, 2016).

4. Team Management Tools

The agile manager can be used to organize teams from the beginning of a project to plan the entire project. This tool gathers information from tools like GitHub and ensures that all tasks are accounted for. Active Collab helps in generating bills and tracking time. This tool can be used to track tasks from conception to completion. Agile bench can be used to track the work done by each member of the team. This tool can help to determine which member is overburdened and help the project manager to reassign some work to ensure equitable distribution of the workload. Other team management tools include pivotal tracker, VersionOne, Jira, Trello, LeanKit, Pmanbox and Axosoft (Parker, 2009).

5. Document Sharing and Documenting Tools

There are instances where the team need to share documents that are essential in facilitating their work. Document sharing tools such as Dropbox and Google Drive can enable them to share and review documents in real time. Documents can be shared in formats that include PDF, PPT, Microsoft Word and Image formats (jpeg, bitmap, png etc.). (Kerzner, 2015).

3. TECHNOLOGICAL ENTREPRENEURSHIP

3.1 Introduction to Technological Start-ups

The 21st Century is considered to be the right age for technology. On the one hand, the tech industry is seen as growing at an astonishing speed and on the other hand several successful tech companies like Microsoft, Google and Facebook reveal the potential that exists in this lucrative industry. A tech start-up can be defined as a new for-profit entrepreneurial venture or business that aims at establishing its market presence and get a share of the revenue that established firms make in the industry.

The tech industry is highly competitive however, the opportunities for new start-ups are countless and these keep emerging with every new day. For any start-up to thrive, it must be established in the right start-up ecosystem. Start-up ecosystems help new businesses to manage growth, increase their competitiveness and create jobs. There are a number of recognized tech start-up ecosystems in the world. The top 20 are shown in Table 2 below:

Table 4. *Worlds top 20 start-up ecosystem with ratings*

(Adapted from startupgenome.com)

Startup Ecosystem	Ranking	Performance	Funding	Market Reach	Talent	Startup Exposure	Growth Index
Silicon Valley	1	1	1	4	1	1	2.01
New York	2	2	2	1	9	4	1.08
Los angelus	3	4	4	2	10	5	1.08
Boston	4	3	3	7	12	7	2.07
Tel Aviv	5	6	5	13	3	6	2.09
London	6	5	10	3	7	13	3.03
Chicago	7	8	12	5	11	14	2.08

Seattle	8	12	11	12	4	3	2.01
Berlin	9	7	8	19	8	8	1.00
Singapore	10	11	9	9	20	9	1.09
Paris	11	13	13	6	16	15	1.03
Sao Paulo	12	9	7	11	19	19	3.05
Moscow	13	17	15	8	2	20	1.00
Austin	14	1	14	18	5	2	1.09
Bangalore	15	10	6	20	17	12	4.09
Sydney	16	20	16	17	6	10	1.01
Toronto	17	14	18	14	15	18	1.03
Vancouver	18	18	19	15	14	11	1.02
Amsterdam	19	15	20	10	18	16	3.00
Montreal	20	19	17	16	13	17	1.05

According to the data in table 2, it can be seen that Silicon Valley is the leading tech start-up ecosystem followed closely by New York City and Los Angeles. For a tech start-up to register success it must be established in an ecosystem that supports start-ups. This section describes in detail various factors that affect the success of technological entrepreneurship.

3.2 Start-up Lifecycle

There are no predefined set of rules and practices or in other words, processes that start-ups should follow however every start-up be it technology-related start-up or the non-technical start-up has a specific life-cycle that the company needs to pass through or follow. According to (Kane,2004), there are six phases that all start-ups go through. (Cann,2013) argue that tech entrepreneurs must be cognizant of the fact that each of the six phases in the life-cycle of their start-ups must be treated with an understanding that they are quite distinct. These six phases include discovery, validation, efficiency, scale, maintenance, and sale or renewal.

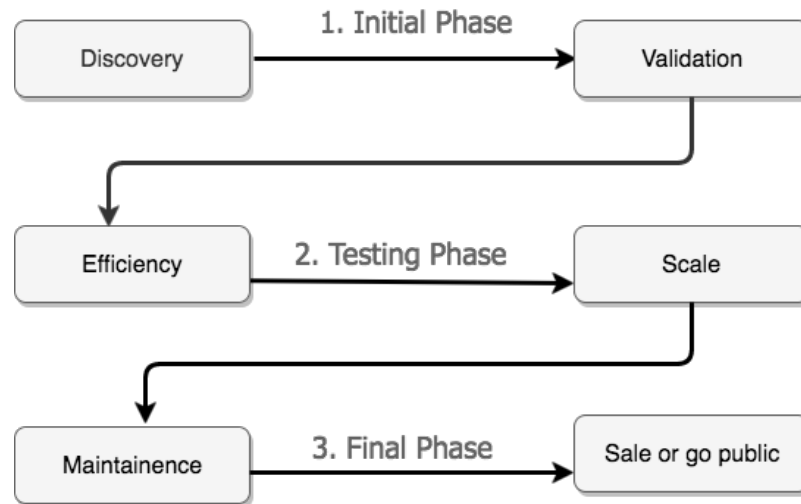


Figure 7. Six phases for life cycle of a start-up (Adapted from Kane 2004).

Figure 2 above is explaining the six phases of a start-up lifecycle. The first phase of the lifecycle is the discovery stage in which the entrepreneur comes with several ideas of the project that the start-up will concentrate on. This phase is also called the ideation stage because it involves the formulation of ideas that are meant to solve a specific problem. For example, the tech company Facebook was a social networking site for students at Harvard University then it later spread to other universities before spreading all over the world (Curran & Curran, 2014). Most of the other major tech companies begun as projects that were meant for a small group of consumers; however, these projects proved useful and they were scaled up and it spread to the rest of the world (Curran & Curran, 2014). Looking back on the Facebook start-up journey, the project started with a minimum viable product (MVP) to survey the market and assess the acceptability of the project.

The second phase of the lifecycle is the validation stage. During this stage, the start-up product or service go from being a hypothetical solution to looking for clients who are ready to pay for the product or service. Proof that customers are willing to pay for the product or service being offered by the start-up is enough validation for the project. The third phase which relates to efficiency requires the technology entrepreneur to look at market studies and settle on a business model that will best suit the start-up and its product or service offering. After establishing efficiency, the entrepreneur needs to look at the possibility of scaling the business idea in a sustainable manner. This means that the tech entrepreneur must ensure that the costs are kept down as the start-up grows its capacity. Tech start-ups like Uber and Airbnb have been able to scale their operations without inflating the cost of doing business (Curran & Curran, 2014).

Immediately after the scaling phase has been executed, it is crucial that the next phase, maintenance, starts immediately. This phase of the lifecycle is important because it is

meant to ensure that the structures that have been built do not collapse. The last and final phase is very important, at this point the tech entrepreneur will have to decide between selling the start-up to a giant tech firm or go public and become one of the giants. If the tech start-up opts to go public, then it must renew its products and service offerings and reinvent itself so as to be able to confront the ever-changing markets (Cann et al, 2013).

3.3 Software development in technological start-ups

Software development process basically consists of a series of steps to be followed in a generic manner known as basic software development process or also known as waterfall model. WM used to be most commonly used model by most software development companies in order to execute their project or product development operations. Since the introduction of start-up methodologies which are also widely used in the software industry as we see a lot of start-ups these days. the basic development cycle has also been evolved and there exists no particular hard and fast rule of thumb in terms of processes and methods to follow since the processes and methodologies a company should follow are molded according to the requirements and needs, this molding includes a number of variables to be considered generally depends in terms of product or projects that a company has been involved into, company structure, size and the software development method they are following which are described later in the chapter.

Software development itself is a sequential process that is carried out after a series of sequential activities or subactivities, each of them is interrelated to each other and may involve a series of complex procedures within themselves. WM which has been widely used by most of the software development companies and has been in practices since ages, in general it usually includes a series of steps to execute in the process after one and another and it's a sequential process. The main steps included in the process model are also defined in Figure 3, briefly include requirements analysis, starting the design, creating an implementation plan and starting of implementation, verification, deployment of the project or product also known as delivery and the final step is maintenance that also includes updates if required. it is necessary to gather basic information and perform a basic analysis as an initial step before start designing the actual product or service, then after the designing the phase of implementation comes where actual operations are performed after the implementation it needs to be verified and tested and if everything goes right it is then deployed. After the deployment it is where the final product is ready the product is maintained, this is called the basic waterfall model for software development and is also explained in figure 8.

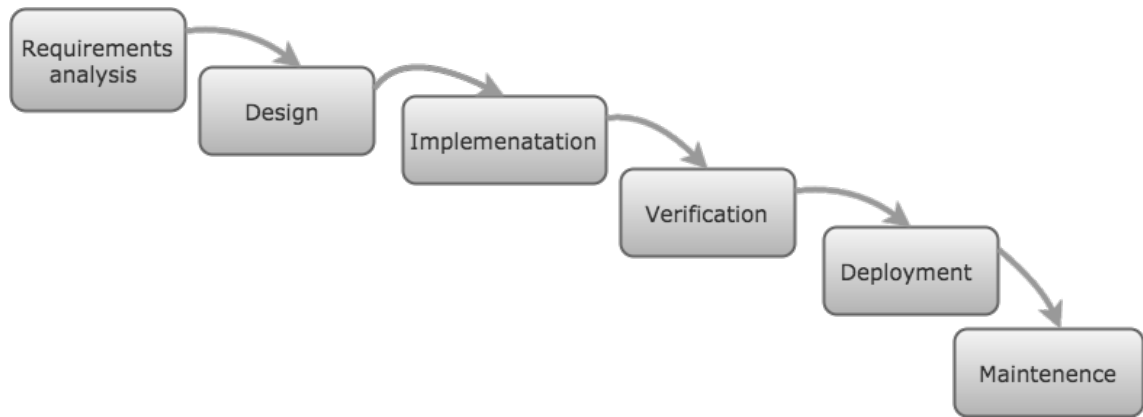


Figure 8. *Basic waterfall model for software development process*
(Adapted from Schwaber, K., & Beedle, M., 2002)

Since start-ups are newly created firms with a limited history of dealing with the demands of high and volatile markets, these firms must approach software development in a different fashion compared to established firms. The dynamic and unpredictable nature of the workspace where tech start-ups operate require tech entrepreneurs to act quickly and establish a niche from where the start-up can get sustainable income. Granted, over sixty percent of start-ups in the US fail to survive past the five years' mark simply because of factors that include missed market windows and poor engineering practices (Nobel, 2011). Some of the process related characteristics for software development in start-ups are defined below:

3.3.1 Process Management

As stated above, software development in start-ups has characteristics that make it distinct from software development in established firms. Firstly, process management in tech start-ups is evolutionary and agile. In this regard, (Coleman and O'Connor, 2008) opined that start-ups must be flexible enough to accommodate the frequent changes that are characteristic of most start-up ventures. The authors further observed that agile methodologies are ideal for start-ups because they can be leveraged to ensure that the start-up embraces change and redrafts its business strategies in line with the developments in the industry.

3.3.2 Customer's involvement in Product Development

The second characteristic of software development is related to customer involvement and customer's feedback in product development process, software development in start-ups are usually driven by the demands by customers who act as designers. For a

product or service being offered by a start-up to have success in the market, the start-up must pay keen attention to the interest of its target customers. The tech entrepreneur must ensure that the service or product brought to the market is constantly optimized to achieve the best problem/solution fit. According to (Blank, 2013), tech entrepreneurs must focus on the process of customer development before releasing the product or service to the market. The problem that the software is aimed at solving must be real for the product to be accepted in the market. Since customers are the ones that steer the requirements products and services being offered by start-ups, it is therefore important for tech developers to use architecture and designs that allow for easy updating and removal of functionalities (Ries, 2011).

3.3.3 Organizational structure

Organizational structure is also one of the significant characteristic in terms of software development in start-ups since the teams tasked with driving the various software development agendas in start-ups often adopt loose organizational structures which are free from the cumbersome and time-wasting traditional organizational hierarchies that are characteristic of established tech firms (Yu et al., 2012). The overall team performance and success can easily be enhanced by empowering the team members and encouraging them to commit to continuous learning, a characteristic that will enable them to quickly adapt to new and emerging trends within the industry. Additionally, it is important for the team to possess the requisite expertise, which will by and large help to counterbalance the possible shortage of resources in the start-up (Carmel, 1994).

3.3.4 Tools and training

The last characteristic is related to tools and training that software start-up companies usually use, start-ups are very fortunate because of the existence of state-of-the-art technologies and tools that can be used in designing and delivering solutions that are tailored to tackle the problems faced by its target customers (Crowne, 2002). Employees working in these start-ups prefer using the technologies that can easily and quickly adapt to the changes in customer needs and preference. Additionally, the use of easy-to-manipulate tools in the start-up workspace facilitates the lowering of start-up training and management costs (Crowne, 2002).

3.4 Common software development practices and methodologies

(Chemuturi, 2013) defines software development methodology as a framework that is adopted in structuring, planning and controlling the entire process of developing a software. Tech start-ups need to follow methodologies that will not only allow them to cut cost but also to hasten software development and improve on their turnaround time.

There are many different methodologies that can be used by software start-up firms to develop their software or projects which are in general their products. Examples of these methodologies include Dynamic Systems Development Model (DSDM), Crystal Methods, Extreme Programming (XP), Lean Development (LD), Joint Application Development (JAD), Rational Unified Process (RUP), Feature Driven Development (FDD), Rapid Application Development (RAD), Spiral, Waterfall (a.k.a. Traditional) as described before in this chapter, Scrum and Systems Development Life Cycle (SDLC). Out of all these system development methodologies, agile development methodologies such as Crystal Methods, Scrum, RAD and DSDM are the most common in the start-up workspace. Aside from agile software development methodologies, lean development is also often adopted by technology start-ups as their software development methodology of choice (Mall et al., 2017).

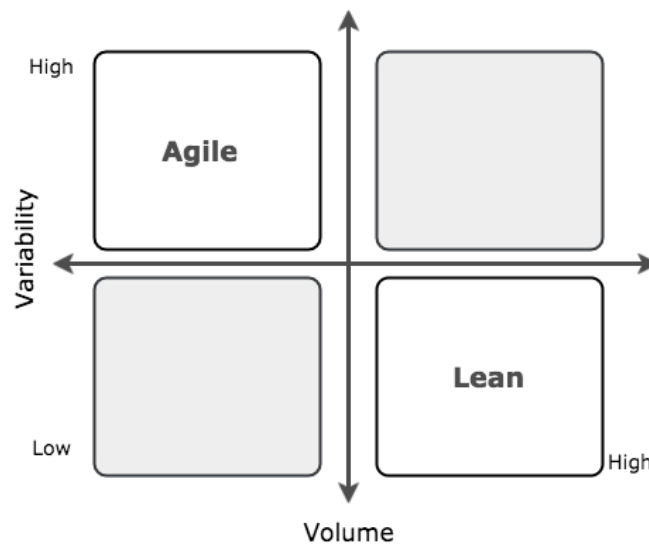


Figure 9. *Lean vs Agile in terms of volume and variability*
(Adapted from Highsmith, J., & Cockburn, A., 2001)

In general software companies choose to either follow lean development methodologies or go for agile depending on the needs, structure, size and nature of the company. Figure 4, explains the difference between agile and lean software methodologies depending upon the properties of the company. Agile methodologies are commonly used where there are more variables involved and lean methodologies on the other hand are useful when the volume of tasks is high and there are less number of variables involved in the process. Most common differences between lean and agile software development methodologies are described further in Table 5.

Table 5. *Major differences between lean and agile methodologies*

Agile	Lean
Teams are self-organized, leaders act as facilitator's	Leaders act as foundation only they are capable of changing anything in the process
Workflow is iterative	Workflow is continuous , not iterative
Customer satisfaction is ultimate goal process is slower as compared to lean methodologies	Fast process, delivery for project or product is made as fast as possible
Key measure in agile is customer's satisfaction	Lean focuses on output and key measures are productivity and cost

Agile software development methodologies are structured in such a way as to counter the risks that are associated with developing software in relatively short deadlines which are commonly called iterations. Each of these iterations lasts for between one to four weeks and they operate as miniature software development projects fully fitted with tasks that are necessary to release periodic mini increments that include requirement analysis, planning, design, testing, coding and documentation. Although some iterations may not necessarily lead to the release of a new software, at the very least, they help to update the project and bring it closer to its official release (Martin, 2014). After each iteration is concluded, the software development team reevaluates the project to determine the next steps that need to be taken to bring the final product to the market.

Agile software development methods are ideal because of the fact that they emphasize on real-time communication both horizontally and vertically in the start-up workspace. When it comes to communication, agile teams use face-to-face communication to seek clarifications, share ideas and voice their concerns. Clearly, this mode of communication has its own share of benefits. Considering that information is a critical resource in the fast-paced agile development work environment, it is therefore important for information to flow freely and faster from senders to recipients and there is no faster way to achieve this than by face to face communication (Martin, 2014). Agile teams are often located in bullpens which have all the personnel needed to deliver a complete software; these include programmers, coders, business analysts, testers, actual customers, technical writers, interface designers and the management.

In lean software development, the project focuses on the creation of software that is tolerant to changes. The measurable objective of LD methodology is to cut cost by

building software with one-third development hours, one-third human effort, one-third investment that would normally be spent in developing similar software through traditional means. LD is guided by principles that include prioritizing customer satisfaction, using needs to determine the technology, providing the best value for money, leveraging team effort to deliver team success and appreciating the fact that true success relies on active customer participation (Levine, 2009).

3.5 Managerial and Organizational Practices in Software Start-ups

The managerial and organizational practices adopted for a software start-up must be in tandem with the nature and needs of the start-up. The management in any organization plays an important role in coming up with policies that guide every operation in the organization. Additionally, organizational practices will be guided by the organizational culture, which is largely determined by the managers that are at the helm of the start-up. In most cases, start-ups are managed by the founder or the tech entrepreneur; whose key role is to provide and reward all the factors of production. Simply put, these founders are often found doing a bit of everything needed to get their start-ups off the ground. In this regard, the tech entrepreneurs habitually hire fellow generalists who will operate under the guide of the “all hands-on deck” policy to help the start-up navigate through its initial stages (Gulati & DeSantola, 2016).

However, (Gulati and DeSantola,2016) observes that with time, these start-ups expand and with this expansion the firm becomes complex and need that its staffs be assigned formal tasks. The founder must appoint some of the generalists to handle different aspects of the organization. For example, he will need to have a project manager, a HR manager, finance manager, operations manager, programmers, technical writers among others. As this happens, founders must work with their managers to come up with formal structures needed to drive the start-up to its envisioned success. Specialization within the start-up workspace is beneficial in the sense that it allows the staff to use their knowledge to tackle their assigned tasks more efficiently. Moreover, specialization ensures that future growth and expansion is achieved through structures that will monitor these new developments.

As the start-up software firm moves from hiring generalists to concentrating on hiring domain experts, it is advisable for the management to look from within the firm and see if they can promote some of the generalists to occupy the office of domain experts. This can be achieved by training the generalists to take up specialist jobs. In cases where the firm must hire domain experts from outside the organization, such should be done, albeit cautiously. The experts hired must have skills that will enable them to mingle with the generalists without raising tensions. The organizational culture determined by the management will play a cardinal role in shaping a workspace that promotes cordial working

relationship which will translate into the staffs working together constructively. It is important for the management to ensure that the early employees (the generalists) feel valued and engaged even as domain experts come in (Gulati & DeSantola, 2016).

Another critical role that the founder and the management team must be party to is the cultivation of a learning mindset among the employees. The founders must find ways of encouraging generalists to hone their skills in specialized areas and also look for ways in which the domain experts can offer mentorship to the generalists with a goal of helping them to grow in specialized niches. Creating a flat organization, which is absent of titles and bureaucracy, which often stifle quick decision making, is the desire of every founder. However, this is not practical for start-ups that must grow and expand their operations. To handle the requirements of the expanding firm, the founders must come up with structures, titles, roles and responsibilities (Gulati & DeSantola, 2016).

3.6 Software Quality Assurance Practices

The cost associated with software defect and errors can potentially derail the whole progress and cause serious losses to the founder of the tech start-up. For this reason, it is highly advisable that the start-up embraces best practices in software quality assurance. (Galin,2004) defines software quality assurance as a set of activities aimed at ensuring that the software engineering or software development process proceeds seamlessly. Unlike normal product quality assurance, software quality assurance is often complicated owing to the fact that software is both invisible and intangible (Galin,2004). However, industry experts have come up with certain best practices that can be adapted to reduce the chances of coming up with defective software. The software quality assurance process and its best practices have been explained in figure 10.

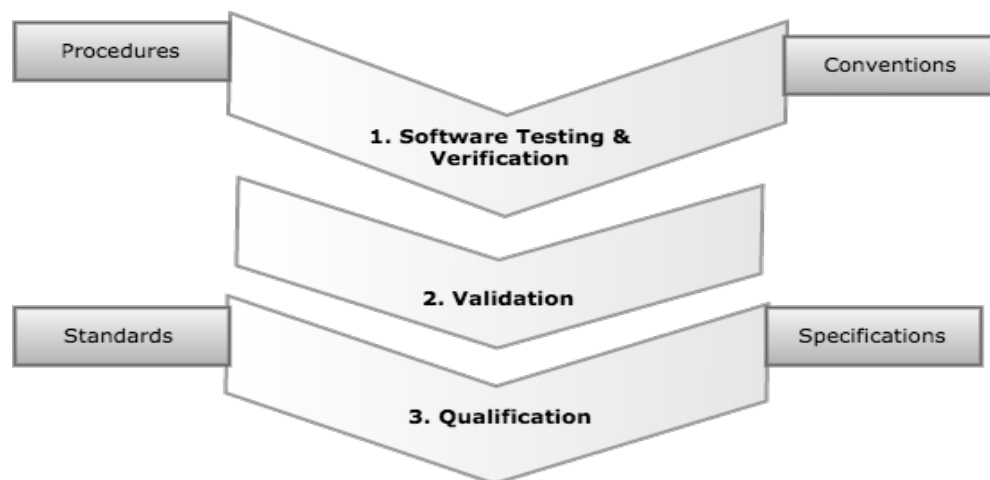


Figure 10. *Software Quality assurance process*

The software quality best practices can be clustered into three broad groups verification, validation and qualification (Galín, 2004). During the process of verification, the product of a given phase is evaluated to see if it satisfies the conditions that had been set at the start of that phase. Validation looks at the system at the end of the development process to check if it satisfies certain requirements that had been outlined by the development team. Qualification normally comes at the end of the software development lifecycle and it looks at the software to determine whether it is suitable for use. These three processes are considered best practices because they are critical in ensuring that software released is free from errors and defects. (Galín, 2004).

3.7 Most Common challenges faced by software start-ups

Just like the other types of start-ups, tech start-ups face challenges that can potentially cause them to fail in actualizing the vision of its founders. Statements such as “a smooth sea never make the best sailors” and “survival for the fittest” fit well with most tech start-ups. Founders often experience challenges that make entry and establishment into the software industry difficult. Luckily, difficult and impossible are not synonymous with. There are so many examples of start-ups that managed to overcome the barriers of entry into the industry and went forward to joining the giants in the industry. As regards survival for the fittest, the founder must ensure that the start-up has what it takes to sail through the initial phases of business. Common challenges faced by software start-ups include fierce competition, unrealistic expectations, hiring suitable staff, impaired decision making, poor financial management, untrusting customers (Tyler, 2015).

3.7.1 Fierce Competition

It is argued by (Giardino et al., 2014) that tech start-ups face fierce competition from established firms within the industry. It is common place to find giants engaging in fierce competition. Start-ups that come into this hyper competitive environment must ensure that they have a unique value proposition which will in turn assure them of gaining competitive advantage and thrive in the competitive business environment.

Early stage start-ups often need to come up with very good differentiation strategy that is sustainable and profitable in long term for the business, one example for this situation could be newly formed resource sharing ideas that have been recently introduced by some transportation companies like car sharing and ride sharing that is a very unique and effective service and pricing model to offer transportation services.

3.7.2 Unrealistic Expectations

The second challenge that start-ups usually face is related to founders some founders have unrealistic expectations which they project to their teams and this places pressure on

the teams to over deliver within a relatively short time. To counter this, founders must subject all expectations to the metric of sustainability. In the event that the expectation is not deemed sustainable, then such an expectation needs to be revised.

In Most common situations the start-ups try to sell the ideas or services and promise deliverables that are not realistic or clients often expect them to deliver a lot within limited amount of resources they have this make it difficult and a major challenge for early stage start-ups couple the unrealistic expectations in terms of deliverables and performance.

3.7.3 Building competent teams

The third usual common challenge is that some start-ups fail to hire suitable candidates who can drive the start-up to success. The team hired by the founder must be comprised of sufficiently qualified individuals who can work together within the established structures. Fourth, founders often run into challenges finding partners who share their vision. Entering into partnerships with such individuals can hinder decision making especially when certain critical decisions need to be made by the partners (Giardino et al., 2014).

Hiring or building competent teams within the start-ups also become a most common challenge because mostly start-ups have limited amount of funding and resources through which they are unable to offer market competitive salaries and perks and finally end up in situation where they need to hire fresh or inexperienced team members which can be a tricky challenge in terms of performance of the company.

3.7.4 Lack of resources and Funds

Poor financial management is also a challenge faced by many start-ups since in the beginning, the start-ups are operating with limited funding and resources. Small start-ups may not have the capacity and skills needed to manage the finances of the start-up properly. As the start-up expands, the founder must consider hiring a finance manager who will help in managing the financial resources of the firm. One of the most common fundamentals of a successful company is to have a good management of resources and accounts, which acts as backbone for the businesses so the funds and financial management need to be done or in other words accounts are needed to be handled in timely manner since nearly 50% of the early stage start-ups fail because of poor financial management (Giardino, C. et al., 2014).

Improper or poor management of accounts and financial resources like book keeping, cash flows and sales makes it difficult to forecast the profitability and long term success of the companies, which often misleads the management and company soon run out of funds and financial resources resulting in not able to afford the employees or able to pay them for long term, this can result in losing the trust on employees and soon the com-

panies main operations are effected as well sine hiring and putting new resources to handle the operations is not often the best practice for successful businesses.

Last but yet most important, the lack of trust on the part of customers is yet another challenge that start-ups have to deal with. Most customers readily trust established businesses. Start-ups can gain the trust of these customers by ensuring that they structure their operations with the customer in mind. Being customer-centric will attract and retain customers to the start-up (Tyler, 2015).

4. LITERATURE REVIEW ABOUT MOST COMMON CHALLENGES IN SOFTWARE PROJECT MANAGEMENT

This chapter describes the most common challenges that have been described through various researchers in the existing literature. Due to the fact that every start-up has its own organizational practices and different nature of the operational business they can differ from each other in various aspects however due to the characteristics of software industry, which include it being in an endless state of flux, means that this industry is influenced by a number of challenges. Software development managers and teams must be cognizant of these challenges and come up with solutions tailored to circumvent these challenges. Early stage start-ups are becoming very popular nowadays as they also help in building economic growth of any company to a reasonable extent. The software industry is highly competitive and as thus the workers in this industry must possess both industry-specific skills and relevant software development expertise (Sliger & Broderick, 2008). Some of these challenges include:

4.1 High competition from global companies

The software industry is highly competitive. Founders often run into barriers to entry especially if the start-up is venturing into the development of software that other companies are either developing or might develop soon. Giants in the industry are best placed to outcompete new entrants into the industry. Therefore, founders and tech entrepreneurs must ensure that the software hits the market faster than their closest competitors (Sliger & Broderick, 2008).

Mostly software start-ups often tend to introduce a product rather than offering services, which in a way is more difficult and highly competitive, they need to come up with a very powerful strategy in order to compete with the existing market giants also in the beginning because of limited resources its often not possible to start the business globally which makes it one of the major challenge initially that start-ups face, however nowadays because of growing number of opportunities around the world there are a number of possibilities that enable efficient operations and incubation of early stage start-ups by offering free office space , access to free resource and funding platforms (Chin, G. 2004).

The most common example can be companies which try to introduce their own product and enter the market first time face a lot of tough competition from the companies espe-

cially from Silicon Valley because it already has a lot of technical giants who might have already worked on introduced a similar idea or product. Entering into the market as a new entrant is one challenge and coming up with a sustainable differentiation strategy is the other one because of which so many start-ups usually end up selling the idea or whole company in some tenure of the bigger operators in the industry, the main reason being is not being able to create a sustainable differentiation strategy or sometimes lack of funds and resources to proceed further with the idea.

4.2 Lack of sufficient software-specific expertise

The software development process can prove to be time-consuming especially if the start-up does not have sufficiently qualified individuals to work on developing the software. Most firms do not have the luxury of time, meaning that they cannot train their staff to acquire the requisite skills before embarking on the development of the software. For this reason, firms are forced to headhunt staffs that possess the skillsets needed for the project. Hiring such employees can be costly thereby inflating project cost (Sliger & Broderick, 2008).

As a start-up in general because of the age and limitation of the resources which is most of the time due to financial or funding issues in the beginning, mostly all the companies face a lack of expertise and grip over all the technologies in the industry, since software development is also an iterative process that includes continuous learning and progress, so individuals and companies also learn by adopting a certain methodology or process and adapt the lessons learned by applying it through their processes (Tyler, J. 2015). This makes it difficult for early stage start-up companies to be technically mature enough and compete with rest of the competition in the industry. However, this does not mean that the products or services offered by early stage start-ups are not technically competent or of great quality, it only applies to the general software development practices and process that is a major challenge for early stage start-ups in their development process.

4.3 Testing and bug fixes leading to numerous software iterations

Software development is accomplished through a series of periodic deliverables called iterations. These iterations ensure that the software is thoroughly checked throughout the lifecycle of the project to ensure that the actual outcome of the project matches the expected outcome. It is commonplace to discover bugs and issue throughout the life of the project. Consequently, the development team must ensure that these issues are resolved before the software goes live. Sadly, not all bugs can be fixed and some slip

through and go unidentified contributing to high levels of customer dissatisfaction with the such software (Sliger & Broderick, 2008).

According to a survey, software start-ups generally claim to follow a testing mechanism within their software development process but usually in real they really don't. Since sometimes due the fact that most projects are not in line with the costs so it sometimes becomes challenging for early-stage start-up companies to follow test-driven development process for the processes or product development (Galín, D., 2004).

5. EMPIRICAL RESEARCH

Empirical research is actually the indirect and the direct method of observation of the phenomena under study (Gartner, 2015). Empirical research requires the researcher to have a set of questions or at least one research objective which is answered with the help of the data collected from the research (Gartner, 2015). The main objective of this thesis is to analyze and discuss the main challenges so the empirical research was focused on two software companies in Finland, both case companies from which the empirical data has been collected are based in Helsinki Finland.

The empirical study will be applied by gathering data through interviews and checking the findings against the above objectives and research questions. This part of the research will be focusing on the two case organizations Company X and Y operating in Helsinki Finland the actual names of the companies have been renamed. The proper research information gathered from the companies highlights various issues and problems that are being faced by the businesses in this particular domain and suggest the appropriate strategies that will be beneficial for the long-term growth and success. This section basically provides information about the case companies and further described the empirical research carried out with them through semi structured interviews. The basic flow of the empirical research is explained in Figure 11. The process actually consisted of ideas generations from literature review and action research then, empirical research actually combines the ideas and data collected from case companies, the findings an empirical data was analyzed to then construct the theoretical framework for solutions and conclusion of the thesis.

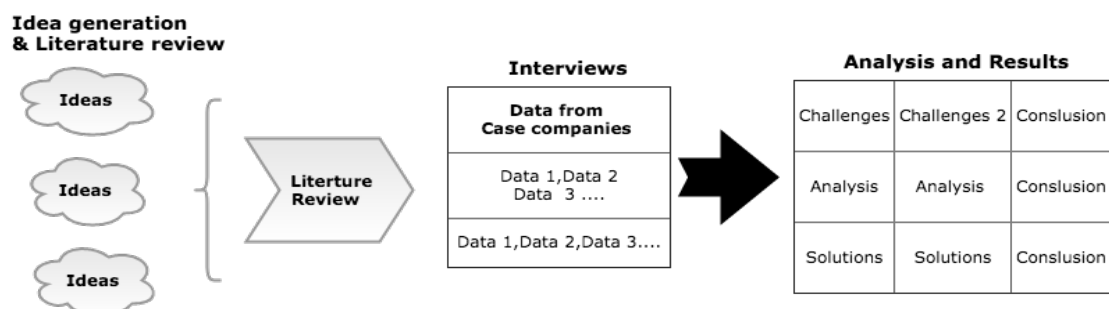


Figure 11. Empirical Research process to gather data from Case companies

5.1 3 Interviews

In order to collect data from the case companies in the empirical research the interview questions were designed using 5 themes related to the software project management.

The themes for the interview questions were actually the main elements required for software project management, these were related to structure of the teams, project scheduling, tools and techniques used within the company, software development methodologies and project management methods used by the company and communication tools.

The interviews were conducted by asking the people from the organizations the target interviewees were (CTO, project manager or Software developers) they were asked about the views and experiences. The interviews were designed as semi-structured interviews that lasted actually around 45-60 minutes for each person this allowed the interviewer to get sufficient detailed information from the respondents about the research. The interviews were about the different important and critical themes that are critical for the entire research process. The themes for the interview questions can be found in Appendix 1 at the end of the thesis.

5.2 Interview Execution

The semi-structured sort of interview questions was most appropriate and suitable method for the empirical research process. The interviews that are collected are done as the empirical data collection for the process. The main source of data were project managers, CTO and software developers from the case companies this allowed to actually gather relevant information from the key players related to software project management themselves.

5.3 Case Company 1

5.3.1 Introduction to the company

Case company 1 is renamed as company X and is a Helsinki based software business which is offering its software services in different parts of Finland and it is basically a project based company. The business was founded in 2013 and has been operating since then offering a wide range of services to their customers. The company was founded by 3 co-founders from Finland as CTO, CEO and a head of sales as major shareholders of the company in 2013. The company started with initial funding from TEKES (Finnish funding company for Innovation), which is basically a government based agency funding the start-ups in Finland . The software team consisted of 4 team members in the beginning and company employed 10 employees working fulltime with 6 sales team members. The company offers its services all over Finland but major clients are from Helsinki capital area (Espoo, Vantaa, Kerava), Tampere and Lahti. At the moment company employs 8 members in its software development team, with a project manager and head of technology or CTO and 22 sales team members across Finland. The company

has managed to run their third funding round and there are major shares form one of the major companies in Finland the S-group.

5.3.2 Business Structure

The company has a business to business (B2B) model and the customers are basically small or medium businesses offering their services to the consumers, so basically company X has a platform which their customers use to build customized booking systems to be used by their clients. The clientele of the company generally includes, hair saloons, doctors, dentists and beauticians offering beauty treatment services of different kinds most of them are from capital part of Finland and Tampere but they do have some clients from Oulu and Turku as well.

The company main business is to offer Software as a Service (SaaS), and the main mission of the company also is to help small and medium-size service based businesses to move towards digitalization. The company has built a platform basically using PaaS architecture, which helps the customers to build a customized booking system and appointment system with customized calendar options depending on the needs, this service also includes building the personal web pages and digital marketing services for the business. The platform has been tested and they are currently using its second version with major improvements. This is included in the basic contract and the customers are charged with one-time setup fee and then monthly fee included in the for the usage of the service. The projects that the company handles are additional services for the clients which includes, mobile booking apps, customized application programmable interfaces (API) and digital marketing strategies that help them to improve their websites and results in increased sales. The solutions that company offers include:

- Online calendar and booking system
- Mobile booking system
- Rest Api and integration
- Website development
- Digital marketing solutions
- Search engine optimization

The main objective being to help small medium business manage their customers and move towards digitalization, this eventually results in increased sales and more conversions to sales. The company is one of the lead solutions providers for e-business. The company's major revenue is form the contracts that are made with help of sales team and minimum length of basic contract is 1 year that includes booking calendar, customized web page and SEO related to the web page. At the moment company X has more than 25 clients form Finland and they are managing 20 projects at the moment.

5.3.3 Organizational Structure and structure of Software Development Team

Company X has 30 employees in total at the moment which comprises of 8 team members in the software development team and 22 sales persons all across Finland in different cities, the main structure of the organization is also explained in figure 10. The software development team is managed by CTO who also acts a project manager and lead by 2 team leads or senior software developers, the team is divided into 2 groups and has one team has 2 developers and other has 3 developers, 2 software developers a creative designer and managed by a team lead. The sales team has 22 people at the moment and is being laded by head of sales, the main duties of sales people in the company is to sell the services to the customers and get more contracts. The sales team is paid a basic salary and bonus which is based on the number of sales contracts from customers they get during the month. The company's human resource and financial matters are managed by company's CEO and all the accounting is outsourced to a third party company.

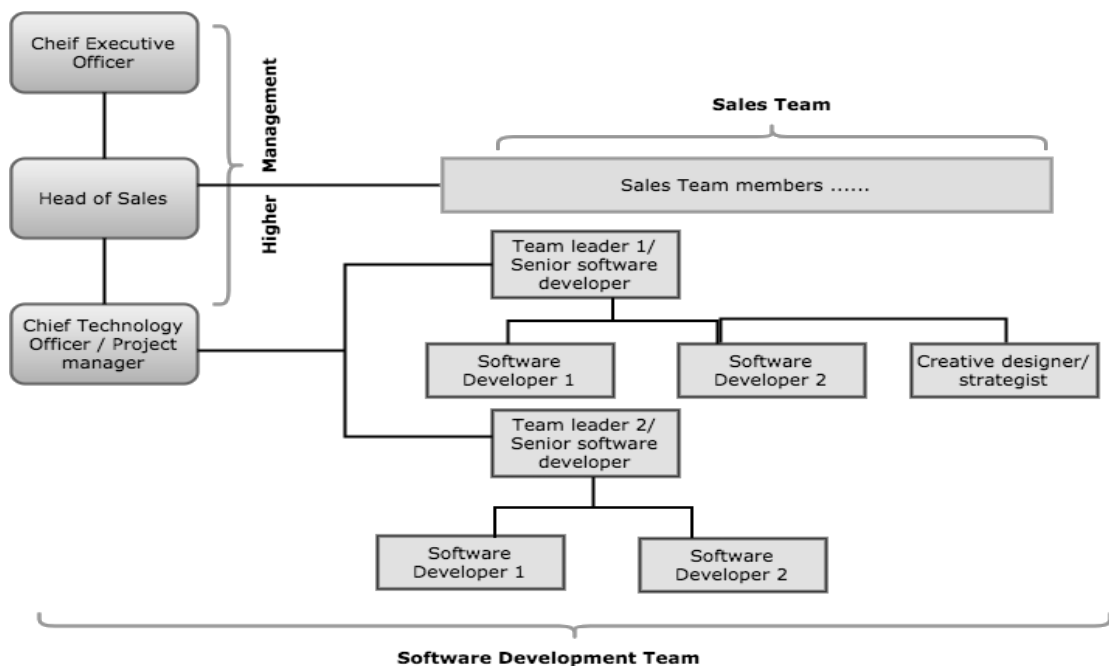


Figure 12. Organizational Structure of Case Company X

5.4 Case Company 2

5.4.1 Introduction to the Company

Case company 2 which is renamed as company Y is a product based IT company that is offering cleaning services through its platform which customers can use to order clean-

ings for homes and moving cleaning in Finland. The company also has a platform and mobile app which is used by cleaners to see and grab available works or cleanings. The cleaners are not employed by company but they have a freelancing contract so they can work and grab cleanings when they have time and are paid on hourly basis accordingly. The clients are both the cleaners and customers who order cleanings through their online booking system and website. The company was initially founded by 2 members as CTO and CEO, with one operations manager and a website with a prototype in 2015 with initial funding from a giant company in Finland as investors. At the moment company has 3 members in their management team and CTO, CEO and a COO, with one operations manager and a customer service representative, there is also a digital marketing specialist in the company and he software development team consists of 4 software developers. At the moment the company offer sits services in Helsinki, Espoo, Vantaa, Sipoo, Oulu, Porvoo and Tampere areas with ongoing development for Estonia and Sweden. The company has recently also signed a deal and got an investment from one of the largest groups in Finland.

5.4.2 Business structure

The company's main business is offering cleaning services to the clients, which is ordered through their online booking system and website they have a B2C model. The other product of the company is cleaner and client management system also called by the company as "INTRA", which is used by cleaners who are working as freelancers to see and grab cleanings if available, they also have a mobile application for this purpose and the cleaners also use this to calculate the amount of work and send invoices to the company. The cleaners can invoice company Y through their own company or use third party services such as "Ukko.fi" and "eazy.fi" to send invoices for the amount of work they have done. The cleaners are not actual employees of the company and they sign freelancer's agreement by which they can take work anytime if there is a free cleaning available. All this is managed by their "INTRA" which offers services like cleaner management, invoice management, customer management, complain management and order management.

The cleanings are orders either as one-time or subscription cleaning which has 3 options (weekly, once a month or once in 2 weeks), the clients can pay either by invoicing method or credit card, once the order is placed by their website or online booking app, the order management can be handled through "INTRA" by cleaners, operations team and their accounting department for invoicing payment to freelancers. The main products of the company include:

- Online booking system to order cleaning services
- Customer and cleaner INTRA
- Mobile application for cleaners / freelancers

At the moment the company is developing their product to expand and offer services in Tallinn(Estonia) and Sweden. They are also working to develop their booking module to offer services for B2B (for office cleanings) under the franchising license of their recent investors.

5.4.3 Organizational Structure and structure of Software Development Team

Case company Y has 3 members in the management team, the CEO, CFO and COO, the cleaners/ freelancers are managed by operations team, there is a customer support representative, a part-time accountant, and a digital marketing expert in the company. The software development team is managed by CTO who also takes care of project management in the company, the software development team consist of 4 developers at the moment with 2 frontend developers and 2 backend / full stack developers one of which is working remotely and is the oldest software developer of the company. The structure of Case company Y is also explained in figure 11 below.

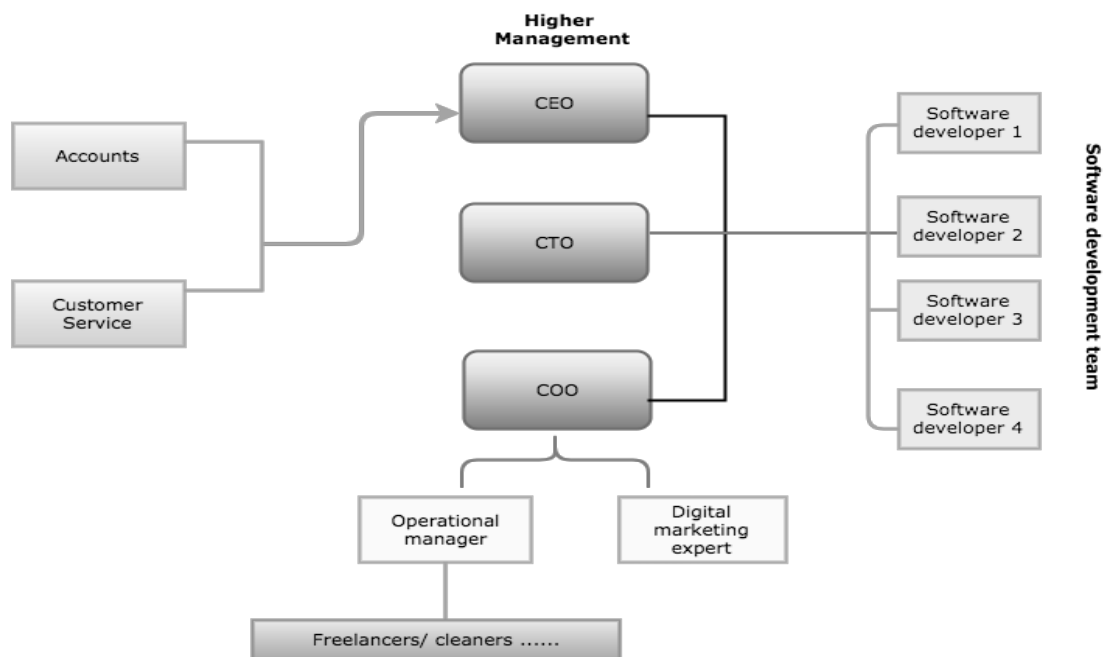


Figure 13. Organizational structure of case Company Y

5.5 Comparison of Case Companies

Although both companies selected for the study are start-up companies and related to operations in software industry, the domain and structure is however, a bit different since “Company X” is purely as project based company and “Company Y” is a product

based company that develops service based system as a product for their customers to use as a platform to order their services.

Company X is older in terms of operations and also has more employees than Company Y, however the technical hierarchy or the technical operational structure of the Company Y seem more systematic since they are using Scrum as a software development methodology and the team is more managed by organization of regular scrum meetings etc. Company X, on the other hand has a different technical hierarchy and their operations mainly rely on sales team since they do services for clients also they don't stick to particularly one software development methodology as they use a combination of Scrum and Kanban, also although the team is managed by team leaders but the internal and external communication about the services seem to be a bottle neck at the moment in terms of development practices for the company More information and data gathered about the structure and practices that were gathered from the case companies is presented in section 6.

6. FINDINGS AND RESULTS

The findings gathered from the case companies from the interviews that were conducted have been documented and described in this chapter. In order to cover all aspects and areas of software project management the interview questions were structured and divided into 5 functional themes related to software development which are explained in table 6:

Table 6. *Functional themes for data gathering and target interviewees from case companies.*

Functional Theme	Target Interviewee
1. Organizational Structure and Business Domain	Project Manager or CTO
2. Software Development and Project management	Project Manager or CTO
3. Main tools and platform used within the company	Project Manager or CTO
4. Team Management	CTO and Software Developer
5. Major challenges and pitfalls that need improvements	Team lead, Software developers and CTO

6.1 Organization structure and business domain

1. How old is your company and what particularly is business about and primary location of customers?

“The company was founded in 2013 initially with 3 co founders and we offer services in Finland, our customers are mostly from Helsinki, Espoo, Vantaa, Tampere and Lahti, but we do have clients from other parts of Finland as well. We offer Saas based services to our customers through our online bookings systems and web applications.” [CTO Case company 1]

“The company started its operations in August 2015, we are offering a platform for customers to book cleaning services and we operate in Helsinki, Vantaa, Espoo, Lahti,

Sipoo, Porvoo, Hanko, Turku and Tampere, soon we are expanding in Sweden and Tallinn (Estonia). Our customers are basically both freelancers cleaners and customers who want to order cleaning services through our system.” [CTO Case company 2]

2. What is the structure of the software development team in the company?

“At the moment we have 8 team members in total in our software team with a CTO and 2 team leaders and all of them are software developers.” [CTO Case company 1]

“We have 3 fulltime developers with 2 frontend developer and 1 backend developer in our Helsinki office and one lead developer works remotely from Slovakia so 4 developers in total and one Project manager / CTO.” [CTO Case company 2]

3. What is the size of the development team and how many team leads or project managers?

“Team of 5 software developers are who are managed by 2 team leads and one Chief Technology officer CTO. “ [CTO Case company 1]

“We have 4 developers who are managed by me as a Project manager and one Chief technology officer CTO.” [CTO Case company 1]

4. Do you have multiple teams working on the projects?

“Yes, we do have 2 team basically working on different projects and managed by the team leads/ senior software developers [CTO Case company 1]

“Since we are product based company we have one team working with different parts of projects, and our projects are basically our products that include booking system, intra, website and mobile application. We do have some designing stuff outsourced so third party company who are doing that for us.” [CTO Case company 2]

5. How many projects usually the teams in the company manage?

“Depends on the workload usually the projects are divided in to exciting and new, teams are managing multiple projects at a time.” [CTO Case company 1]

“We only have one team in the organization so basically they work same time on single project break down in (backend + frontend) at the moment the team is working on 2 projects are both are in-house products.” [CTO Case company 2]

6. What are your main customers or what particularly are end users for your software products?

“Our customers mainly include small business that are offering services to consumers, our business is focused on both B2B and B2C, most of them are beauty saloons, bars and hair dressers.” [CTO Case company 1]

“Our main customers are people who want to order home cleaning and moving cleaning for their homes and then the cleaners who work as freelancers to grab cleanings and work and are paid hourly.” [CTO Case company 2]

6.2 Software Development and Project management

1. Which software development methodology or model is used by your company?

“We are pretty much into agile and we use combination of Scrum and Kanban.” [CTO Case company 1]

“We use Scrum”. [CTO Case company 2]

2. Do you use sprints? How long are your product development sprints?

“Yes, we do sometimes we have longer development sprints depending on project requirements but usually its 2-3 weeks long.” [CTO Case company 1]

“Yes, we have sprints and our sprints are 2 weeks long.” [CTO Case company 2]

3. Do you have scrum meetings or daily standups or sprint reports, how long are they?

“We have daily standups within the team 2-3 mins long for each member, and once in a week usually on Tuesday scrum of scrum with all teams to discuss what are they up to.” [CTO Case company 1]

“We have daily scrum standups usually 2 minutes for each member, sprint reports are automated form project management software.” [CTO Case company 2]

4. How do you test your products or projects, is there a dedicated software testing or quality assurance team in the company?

“The testing is done by our development team itself, we don’t have a separate software testing team in the company at the moment, we use automated testing frameworks.” [CTO Case company 1]

“We don’t have a testing team in our company, the development teams test the updates and features on their end manually and we do use one software for manual testing called use trace that tests the applications for frontend.” [CTO Case company 2]

5. What is the customer involvement in the software development cycle in your company do you get feedbacks after each phase etc.?

“We do get continuous feedback since the customers require a lot of customizations depending upon their nature of business so they are involved from the beginning at every step.” [CTO Case company 1]

“We get the feedback and error reports etc. from the users but once the updates or features are deployed we don’t involve continuous integration in our development phase.” [CTO Case company 2]

6.3 Tools and platforms

1. What tools are being used for your company to manage the sprints or whole product development cycle or particularly by what software you manage the projects?

“We are using “JIRA” as main project development tool, with “confluence” for documentation and for bug reporting we use “ASANA”.” [CTO Case company 1]

“We use “JIRA” as main project management and sprint planning tool and for bug reporting we use “Bugsnag”.” [CTO Case company 2]

2. How do you communicate with the team about ongoing progress, updates, bottlenecks etc. within the team do you use any software for that?

“We use Slack as a communication tool within the company and issues and bottlenecks are also discussed during scrum meetings.” [CTO Case company 1]

“Slack is used for development communications and updates within the teams and other team members of organization.” [CTO Case company 2]

3. Do you document your products or do you provide documentation for the customers regarding the products? If yes what software you use or its just simple pdf, word etc.? how well documented you think they are?

“We document our own processes and functionalities for our system which is not so well documented but can be improved we use Confluence for that and its only accessible to the internal team, but for individual client products we don’t have the documentation at this point it only basic structure for development team to understand the basic requirements like platform, functions etc.” [CTO Case company 1]

“We have documentation for all our products since the backend is same for all of them and we use API docs and JIRA’s confluence tool to do that to make it understandable for the development team.” [CTO Case company 2]

4. Do you use any testing platforms or soft wares to test your products for example automated testing etc.?

“Within the team we do, we use automated testing frameworks like Jasmine and Karma but depends on the urgency sometimes I feel that testing is the phase often ignored at times especially in start-ups.” [CTO Case company 1]

“We use Use trace to test our frontend and for backend we use unit testing platform like Phantom.js” [CTO Case company 2]

5. Do you use any platform or software for bug reporting?

“We use ASANA and mostly Google docs to document individual crucial bugs for client products.” [CTO Case company 1]

“We use “**Jira**” to document bugs assign it to the developers.” [CTO Case company 2]

6.4 Team management

1. What do you think about your product development cycle and methodology as a whole? Do you think there is room for improvement?

“As whole sometimes I feel that sprints seem short, clients have some un-realistic and complex requirements and there are some bottlenecks so sprints should be extended or should be 3 weeks rather 2 weeks to include proper testing etc.” [Developer from company1]

“I just need to add more testing cycles or at least software testing time frames to the project schedules so that there is enough time to properly test the features we build this will eventually reduce bugs and save time.” [Developer from company2]

2. Do you think the teams and team size are enough to manage these projects or they are overburdened?

“Sometimes the project deadlines are tough and we are overburdened but since that’s the part of start-up methodology so its manageable.” [Developer from company1]

“Over burdened not really...Most of the time we are not overburdened since the project planning is being done in realistic way but yes sometimes there are times when deadlines are tough.” [Developer from company2]

3. Do you think that each time the requirements are clear and easy to understand or there can be more improvement to these?

“They are clear usually most of the time since our team leads are the ones understanding them form clients and we usually get clear guidelines and requirements.” [Developer from company1]

“Most of the time they are clear.” [Developer from company2]

4. Do you think that deadlines are realistic and easy to manage and deploy the projects with the planned deadlines?

“If the project planning is done right its always easy to manage the deadlines and in our case yes usually they are, but sometimes there are clients who want really fast results and have some surprisingly unusual requirements.” [Developer from company1]

“Yes, the deadlines are realistic and if there is something we usually extend the sprint times mentioning that for example X and y were the issues that caused these.” [Developer from company2]

5. Do you think there is enough knowledge and support provided within the team and from the higher management to build the product etc.?

“Yes, I think there is sometimes since start-ups offer a lot more to learn, you need to learn at your own but most of the time there is enough support.” [Developer from company1]

“Yes, there is but sometimes you need to do things and manage at your own too like testing etc.” [Developer from company2]

6.5 Major challenges and main pitfalls that needs improvement

1. Do you think that there is a lack of planning or requirements are sometimes unclear from the client’s perspective or from the product development team that should be improved?

“In B2C a lot of times there are unclear requirements, or un-usual stuff, also changes that takes a lot of time and effort.” [Developer from company1]

“Not really I think most of the time they are quite clear off course there are exceptions at times but not so often.” [Developer from company2]

2. Do you think that there is a lack in software quality insurance especially in start-ups that should be improved?

“Yes, I totally agree with the fact, and this is because of limited resources and size of teams usually.” [Developer from company1]

“Yes, there is a clear lack of software quality assurance in start-up companies because of the limited people they have and commonly because of limited funds that can be improved.” [Developer from company2]

3. Do you think working with start-ups is more challenging? and what are the differences you see in terms of project management or product development while working with start-ups as compared to traditional software companies?

“Of course because of the nature of start-ups, size, resources, experience, expertise, its quite more challenging than working with a stable medium or large size companies, but on the other hand it also allows you to learn more.” [Developer from company1]

“Yes, working with start-ups is quite different and challenging as well in different ways, but it allows also to learn more as you are required to work in different roles at the same time with limited time and resources etc”. [Developer from company2]

4. What do you think is the most crucial thing that you face related software project management working in start-up environment?

“Deadlines and quality is most crucial and both of them are relatively difficult to manage together.” [Developer from company1]

“Sometimes the resources are limited so there is huge amount of work to be done in sprints, so the project planning can be improved by including only the crucial parts of projects tasks in sprints.” [Developer from company2]

5. Any suggestions or comments regarding this interview or for the research as a whole?

“The research seems to be a bit challenging since project management itself is quite a vast topic, but I hope you come up with some really nice solution to come up the challenges in project management with start-ups.” [Developer from company1]

“Not as such. Just that the topic is quite interesting and do share the views with us on how to overcome and improve the challenges and pitfalls in software project management specially while working in start-ups.” [Developer from company2]

7. DISCUSSION

The research questions were divided into themes and were designed on the basis of fundamentals of project management as discussed in chapter 6, the data was gathered in the form of answers from the interviews from two case companies and this section provides the detailed analysis and details of the results and also in-depth solutions for the main challenges.

The interviews were conducted with the main players in terms of usage of software project management of Case company X and Case company Y. They gave insights for challenges in software project management into each business and also provided an insider's look at the organizational structure, the major issues the companies faced, and the project management styles they utilize. The main interviews were conducted with Chief technology officers and software developers who are the key players in software project management. The five key themes which have been addressed in chapter 6 provided the main empirical data.

7.1 List of main Challenges and Pitfalls with Software Project Management in Start-ups

The two companies are again very similar in the tools they use for their development. Both take a holistic approach to software and program usage and allow employees to use whatever tools they see fit for the project. Similar to methodology, the risk of the company's rapid growth is that they will lose the freedom they emphasize in favor of structure and organizational cohesion. The team management is one of the few areas where Case company X and Case company Y begin to differ. Much of this difference is due to the age of each organization and the target customers.

Case company X is two years older than Case company, and those two years really make a difference in the way the teams are managed and the organizational growth. Similar to the difference in team management, the two companies face very different obstacles as they continue to grow. On one hand, Company X needs to determine its place in the business-to-business market and make sure that their efforts grow and help them become broader. Case company Y, on the other hand, needs to continue to growing to keep up with the current market demands. Moreover, they need to figure out how to make more money so they can provide their employees with the resources needed to do their jobs.

The Major common challenges that have been analyzed during the empirical research process have been classified into 3 major groups which are also illustrated in Figure 14 they have been initially identified and classified according to the main themes in the data gathering process.

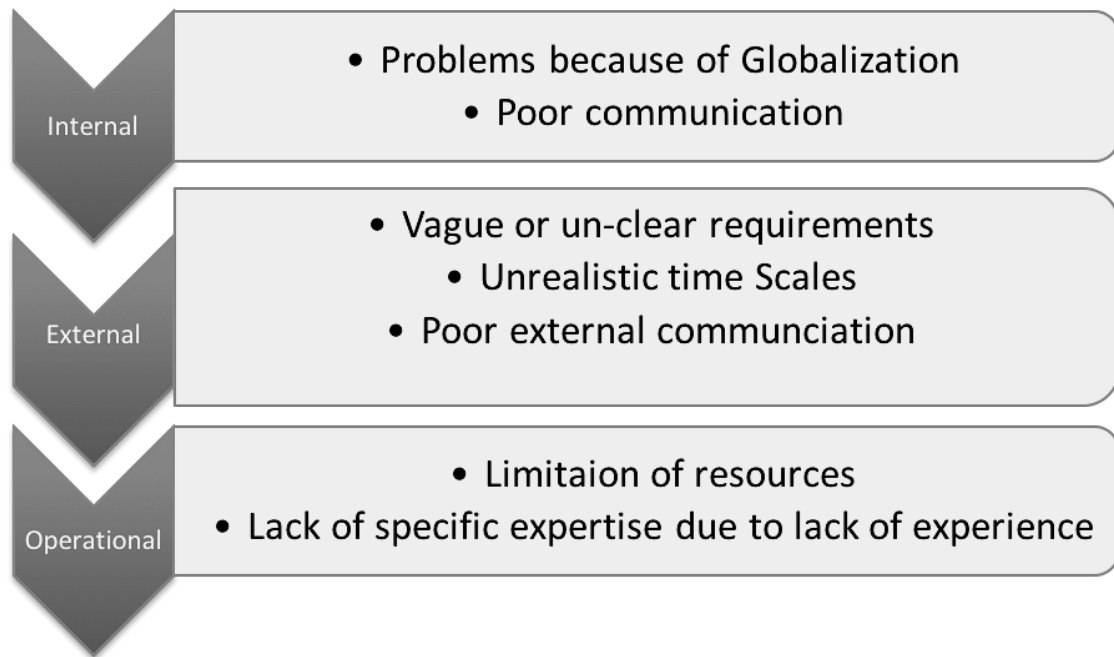


Figure 14. *Classification of major challenges in software project management in start-ups based on the empirical data gathered*

Figure 14 explains the classification of major challenges that are faced by software start-up companies in project management, they are mainly classified according to the functional areas into 3 main categories internal challenges that include challenges like poor communication, and challenges occurring due to the impact of globalization, external challenges that includes unclear requirements, challenges like unrealistic time schedules and deployment dates and poor communication to clients, then finally the operational challenges that include limited resources and challenges like lack of expertise due to operating nature of start-ups.

7.1.1 Lack of necessary Tools and Infrastructure

According to the data from both case companies collected from theme 3 and section 6.3 for tools and platforms as a start up it is clear that both the case companies lack as some point in adapting the basic necessary tools and platforms, instead they use non standard practices which is a significant challenge in software start up environment. Software development is not an easy undertaking. It requires a lot of infrastructure and legal pro-

cedures to be undertaken. Significant human and financial resources are needed for maintaining, enhancing, and implementing the infrastructures and the legacy systems (Jalote and Pankaj, 2002). These areas of legal and infrastructures are completely dominated by the pioneers in the software start-ups industry. It is not easy for the start-ups to venture, replace and maintain. The stakeholders who conduct legal processes and requirements for the software companies are so acquitted to the well-established companies and may require an extra tip to support the work of a software start-up project in the legalizing sector (Jalote and Pankaj, 2002). Moreover, software start-ups need a lot of infrastructures to set up, test, roll out and maintain. With less support from the concerned stakeholders, these projects end up stale or sold to the domineering company in the local or international market (Thomas and Cagley, 2010).

From the interview analysis the infrastructure is a serious problem for the case companies as also was reckoned by the developer working with company X because of the nature of start-ups, size, resources, experience, expertise, it is more challenging than working with a stable medium or large size companies, but on the other hand it also allows you to learn more. This means that experienced developers are willing to work with the software project development pioneers than the new entrants. It is a limiting factor and a great challenge for the software developers in the start-up stage. More so, software project management tools are very important in ensuring the success of the software. The start-up's developers opt for cheap and easily accessible project management tools such as Microsoft Excel and the Microsoft project. The greater challenge following the given tools is that they can easily share information in the extranet.

It compromises the security and the original idea of the developer. The desktop software project support version lacks the advantage of providing real-time data. Looking at the dynamism of software projects much information is shielded from the developer. By the time the project is completed. It could be completed with a more secure and sophisticated version of the same. Indulging in the web-based solutions for software project management is the best option for the developer. The developer faces the problem of increased costs and manpower for managing and interpreting the impact of the real data on the progress of the project. "Yes, I totally agree with the fact, and this is because of limited resources and size of teams usually."

7.1.2 Lack of specific Expertise for the Software Development

According to the data form in section 6 and from the theme 6.2 and 6.3, it can be seen that software developers in both the case companies face a very common problem while gathering the required team for communication in project management. The software team is required for the process of forming ideas, organizing, encouragement, and for guiding each other (Stellman, Greene and Jennifer, 2005). For instance, project developers must have adequate experience and knowledge for the process of controlling, motivating, and planning the product or the service. The team is needed to assist the devel-

opers in carrying out the objectives of the team with the aim of realizing the set objective. A team which lacks the necessary expertise may not realize the set goals and objectives (Reinertsen, 2009). A developer is likely to operate alone in case the team members fail to cooperate towards the objective of the project. These circumstances may overwhelm the project developers and lead to cost or time overruns. A team which is well experienced in the different areas of project management is more confident and shares knowledge for improving the success factor of the project (Reinertsen, 2009).

Trust and clear communication enhances the nature of interactions and communication between the developer and the team to achieve project deliverables, do testing, do programming and draft the required model. The project managers need to be experienced in handling issues such as finding options, solving problems, cost estimation, risk analysis and problem-solving. The project team must be capable of undertaking the given responsibilities in the team. Project management requires experience of undertaking varied expertise and multiple perspectives. For instance, project managers are equipped with large tasks such as language, culture, skills and good knowledge. The project management has the responsibility of ensuring each team is assigned a task with clear layout of handling unacceptable results conflicts and misunderstanding (Ong, Nee, & Xu, 2008).

Team members, on the other hand, must function as teams but handle individual tasks. The problem of lack of enough software team was reckoned by the respondent from the company 1: "The testing is done by our development team itself, we don't have a separate software testing team in the company at the moment, we use automated testing frameworks." A similar response was also acknowledged by the interviewed person in the company 2: "We don't have the testing team in our company, the development teams test the updates and features on their end manually and we do use one software for manual testing called use trace that tests the applications for the front end." The start-up software project developers are continuously facing a challenge of hiring the competent and experienced taskforce. It is tedious and time-consuming to search for developers and management to implement the project and the ideas (Ong, Nee, & Xu, 2008). Moreover, the already well-established software project managers and developers are looking for competitive salaries which may not be possible to accomplish. In this regard, the project ends up incurring costs and time overruns in quest to look for experienced and competent employees.

The information was confirmed by the software developer for the case company 2: "Yes, working with start-ups is quite different and challenging as well in different ways, but it allows also learning more as you are required to work in different roles at the same time with limited time and resources etc." It is a very significant challenge since a developer may be experienced at one phase of the project but lacks enough skills to undertake the second phase of the project. As a result, the developer encourages in a self-learning and training session to gain skills for the required phase. Consequently, the

project is delayed which affect the entire projected deliverables and milestones. It is known that technology start-ups companies incur the problem of hiring or over scaling. Consequently, the software starting companies end up failing to premature hiring and over scaling.

7.1.3 Mismanaged Resources

The project developers and the team lack adequate well-updated information regarding the project and its requirements as gathered from section 6. Some project requires a small team while others require large teams to execute the activities of the software development. Current demands in the software projects developments are complicated the whole process of developing a software is created to ease the operations of an end user. The well-developed software is one which the user experience is easy to log in and navigates that is why currently, most organizations are opting for online e-commerce businesses.

Hence, software capable of supporting such functionalities is on the demand. It is a complicated undertaking requirement for the complicated users and the basic users at the start –up period. Basing on the resources, experiences and the legal requirements to be undertaken. The information was reckoned by a developer from the case company 2: “Yes, there is a clear lack of software quality assurance in start-ups companies because of the limited people they have and commonly because of limited funds that can be improved. “The project manager may misuse the project resources by gathering unqualified personnel to run the activities of the project. In this regard, unqualified personnel is hired to run the project which leads to project failure.

7.1.4 High competition because of Globalization

Globalization is redirecting business from the local market to the international market (Robert and Frees, 2003). Globalization has its own advantage and disadvantages. However, when it comes to the start-up software project management the concept of globalization becomes a challenge (Robert and Freese, 2003). Most software developers search for uniqueness and wonderful functionality to penetrate faster in the global markets (Joseph and Gulla, 2012).

For the case companies X and Y, the challenge is the lack of enough information on how many software developers are working on the similar projects. Moreover, it is not easy for a software developer to apply for guesswork and determine who is close to the same idea. The pressure becomes more real during the marketing period of the software. Technology and Malware are created daily. It is a concept which braces software start-ups on the toes wondering about the viability of the software. How to market without selling the initial developer idea is also a challenge. Most developer's ideas are stolen by the well-established organization with a fair share of the consumer markets (Joseph and

Gulla, 2012). In this regard, globalization poses a great challenge and competitions to the software start-ups. The competitions adequately affect the start-ups in terms of hosting, service level agreements, customer retention, customer reach and pricing structures.

The start-up's software developers are deemed as starting very low in term of pricing. In short run, the developer may fail to flourish and end up selling the pertinent rights to more established organizations dealing with software development. Therefore, in the case companies' analysis from the interview respondents, it is evident that the respondents complain of unusual stuff happen during the start-up period. The respondents specifically quoted issues relating to the experience, the skilled force, the insurance and the registration of the product. Most specifically, the extent of the resources required to run the company. Most of the start-ups lack enough finances to market and legalize the product to venture in the world markets. In this regard, they are easily lured by the well-established product software companies in the same line of products. Company case 1: "In B2C a lot of times there are unclear requirements, or unusual stuff, also changes that takes a lot of time and effort. "The concept of globalization also brings about geographically dispersed project team.

Software development projects must be undertaken with the high level of collaboration and interaction. A software project requires multi teams each working towards the perfection of a specific area. In this regard, the team is the third parties, the sub-contractors, the vendors, the clients and the employees. The start-up's companies lack enough teams to offset and support the functions of the projects. A lot of resources are required to gain support from the third parties, sub-contractors, vendors, clients and employees. They end up outsourcing some of these stages and consequently compromising the output of the project. It is more so in case the outsourced parties lack transparency and visibility.

7.1.5 Vague Requirements

Any step of the software development lifecycle requires clear and well-gathered requirements (Bertram, 2016). It is important to gather information, evaluate and analyze the customer requirements in the both the short run or the long run of the software project management. Most of the time the project developers engage in software development with no clearly defined goals and they end up incurring time and cost overrun this also the same case with Case company X and Y as depicted in section 6 through the theme 6.4 and 6.5. A thorough research is required since customers are rarely sure of the software specifications they prefer in the market. In this regard, the specifications keep on changing which may affect the project lifecycle. Lack of proper research pre-empt the developed software into inefficiencies, ineffectiveness, increased costs and delays.

The problem becomes real when the developer starts the project without considering its challenges and limitations. Understanding the end user requirements is very important in designing successful software. The need for continuous customer involvement was reckoned by the developer from the company case 1: "We do get continuous feedback since the customers require a lot of customizations depending upon their nature of business so they are involved from the beginning at every step." The information regarding the need for continuous feedback from the customer was also reckoned by the respondent from the company case 2: "We get the feedback and error reports etc. from the users but once the updates or features are deployed we don't involve continuous integration in our development phase."

7.1.6 Unrealistic Time Scale

Software project incurs failure due to lack of proper estimation of the time required to complete the project deliverables (Wagner, 2013). The success factor of a software project is when it is delivered faster and timely. Cost overrun can cause the software product to become out-dated even before it is rolled out in the market (Wagner, 2013). It also requires research, evaluation, and analysis of all the requirements in the ground. Lack of realistic timeframe leads to superficial designs with poorly coded systems which fail to function in the first product testing. Inelastic timeframes also occur from the stakeholders benefiting from the project. They exert pressure on the developers and the project managers to complete the project development within their given timeframe. It is a problem which is incurred in the case the developer, development team and the stakeholders do not conduct regular meetings towards evaluating the requirement of the project in terms of costs and time. Consequently, the final product becomes faulty, or in case both cost and time overruns due to resources needed for rectifications.

Moreover, working under pressure can produce software of low quality which does not meet the requirements of the end user. The importance of using realistic timeframe was acknowledged by the respondent from the case company 1: "If the project planning is done right it's always easy to manage the deadlines and in our case yes usually they are, but sometimes there are clients who want really fast results and have some surprisingly unusual requirements." Also the respondent from the company case 2: "Yes, the deadlines are realistic and if there is something we usually extend the sprint times mentioning that for example X and y were the issues that caused these." Scope creep is the general overview of the project in the starting phase. In case the scope increases above what is project then scope creep occurs. Scope creep mainly occurs in the testing phase when the customer's specifications are not met and the team needs to go back to the drawing board. In this regard, the software project needs to be adequately prepared through thorough research and analysis. Lack of proper preparation a scope creep is likely to occur. The increase in scope creeps leads to both project and cost overrun. Unrealistic time-

scale in project management for the start-up company is confirmed by the respondent from the company case1: “In B2C a lot of times there are unclear requirements, or unusual stuff, also changes that takes a lot of time and effort.”

7.1.7 Poor Communication

Poor communication is another common challenge faced by software start-ups development team in the project according to the data gathered in section 6 from section 6.5. It is the main reason why many projects fail. The developers and the managers plus the team fail to communicate effectively concerning the needs and the requirement of the projects. As a result, the disagreement leads to delays and increased resources for correcting the mistakes made. It is, therefore, very important for the requirements of the project to be shared frequently with the stakeholders and the team involved in the project. Most developers avoid communicating to the team members as a strategy of avoiding conflicts in views and the methodology. The stakeholders also prohibit programmers from disclosing about the progress of the project which may lead to project failure in the long run (Donohoe, & Software Product Lines Conference, 2000). Poor communication is a recipe for lack of progress of the most software in the start-up phase. Project failure from poor communication can severely affect the functionality, the schedule and the development of the project. Inefficiencies and lack of effectiveness in the development of the project are also caused by lack of proper communication within the team and all the stakeholders required for the management of the product. Lack of end-user involvement in the project also shows a lack of communication.

The project developer must continuously discuss how the product is used with the end user. If the user is not aware of how to use the product, then the product may not deliver up to the expected results. Poor communication also occurs when the end user is not involved in the testing phase, the analysis phase and the design phase (Petersen et al., 2007). Lack of proper communication with the end user can lead to project failure since users may keep changing the specification hence leading to project failure. Good communication ensures that the requirements of the end user are taking into consideration before the final software product is produced. Lack of communication can lead to users not getting the expected quality and functionality and hence leading to project delays. Rolling out the project with a lot of functionality that the user does not know will lead to the system collapsing at the end. Users must be actively involved in every stage of the project. They should be given time to suggest areas they feel when improved they function better and vice versa (Kurtz and Snowden, 2003).

7.1.8 Poor Planning

For a software project to be successful than adequate planning needs to be undertaken. Planning should be undertaken after researching about the resources required in every

phase. According to the data from section 6.5 planning seems to be one of the key challenges in project management of case companies. Complete planning ensures that the project is undertaken within the schedule and within the timeline (Reisman, 2000). A successful software project is the one which has met all the projected deliverables and the milestones. The deliverable schedule, the cost of the development and the required resources are all the factors considered in the project planning phase. The project developer may not adequately plan for the project if the expected requirements are not met. The project flow is constantly interrupted within may ruin the entire programming process. All the phases of the software development lifecycle are interrelated. A phase is dependent on another phase which may lead to project failure. The results of poor planning are prolonged scheduled and increase in the cost of the product. At the end of the project, the scheduled will be affected, the requirement will not be met and the costs will be increased.

Poor planning can compromise the processes undertaken for the project to the completion. A sensitive area to be affected is the testing phase. The developer must confirm the usability of the project upon completion. Nevertheless, the acceptability of the project is the end user by checking that the requirements of each phase are adequately met. Testing fails with poor planning. The users do not know what areas to test the software. Most likely the user was not informed about the functionality of every area. The need for proper project planning is confirmed by the developer from the company case 1: "Deadlines and quality are most crucial and both of them are relatively difficult to manage together." The information is also confirmed by the project manager from the company 2: "Sometimes the resources are limited so there is the huge amount of work to be done in sprints, so the project planning can be improved by including only the crucial parts of projects tasks in sprints."

7.1.9 The problem of Poor management

However, there seems to be no problems such related to management level issues in the case companies but this is a significant challenge in project management, a manager is a person entrusted to oversee the development of the software project. The goal of the project manager is ensuring success through undertaking a practical approach to planning and solving all the problems experienced in the project. Most of the problems facing the software development managers are the inability to predict and manage possible risks which may affect the development of the project.

The manager fails to maintain and establish the required momentum to run the activities of the projects. Managers fail if they cannot balance the resources, effort and time entrusted in the project. Hence, the team members lack the basis of providing excellent to the individual deliverables. Managers ensure the project is based on the strategic framework and solid foundation. Most projects fail because of the important elements to run the project were not taken into consideration. A project manager fails if he or she

does not provide guidance to the rest of the team concerning the effective paths that they are likely to follow (Petersen et al., 2007). Project managers fail due to the inability to conduct the feasibility analysis of the project. Hence, they initiate the project without considering the exact needs and the specific needs which are important to the customer. The project managers in this regard, fail to study ROI associated with the project leading to poor estimation of the benefits and the costs (Crowne, 2002).

Another reason why project fails is the inability to determine the project requirements by the managers. The managers are expected to plan for the requirements through interviewing the beneficiary of the project (Ivarsson and Gorschek, 2010). In this regard, managers fail due to lack of configuration management system utilized for the version and tracking monitoring. The use of defect tracking tool and the automated build systems.

Managers also fail because they lack to use regression testing tools and reporting tools which are automated. Managers' lack to invest of the necessary tools required to perfect the product development process. They do not put into consideration tools which present project and product visibility (Ambler, 2002). Managers also fail when they lack effective tools for mundane and repetitive tasks. Managers fail to lead the software development process into progress when they do not consider the technical and the documentation required to run the activities of the project. The manager must undertake all the necessary precautions to ensure that the legal requirements for the starting the project are taken into consideration. The design process fails due to lack of the specific requirements which are well documented in the project requirements (Crowne, 2002).

7.2 Common challenges related to Software Project Management faced by both Case Companies

The most common challenges have been discussed already and identified in the previous section but there are some common challenges faced by both of the case companies related to effective software project management that are common in every type of business, these common challenges have been described in Figure 15. Figure 15 explains the most common challenges that are faced by software start-ups however it explains that left side indicates the challenges that are internal and right-hand side explains the challenges that are common and external and are due to nature of software start-ups and external business processes.

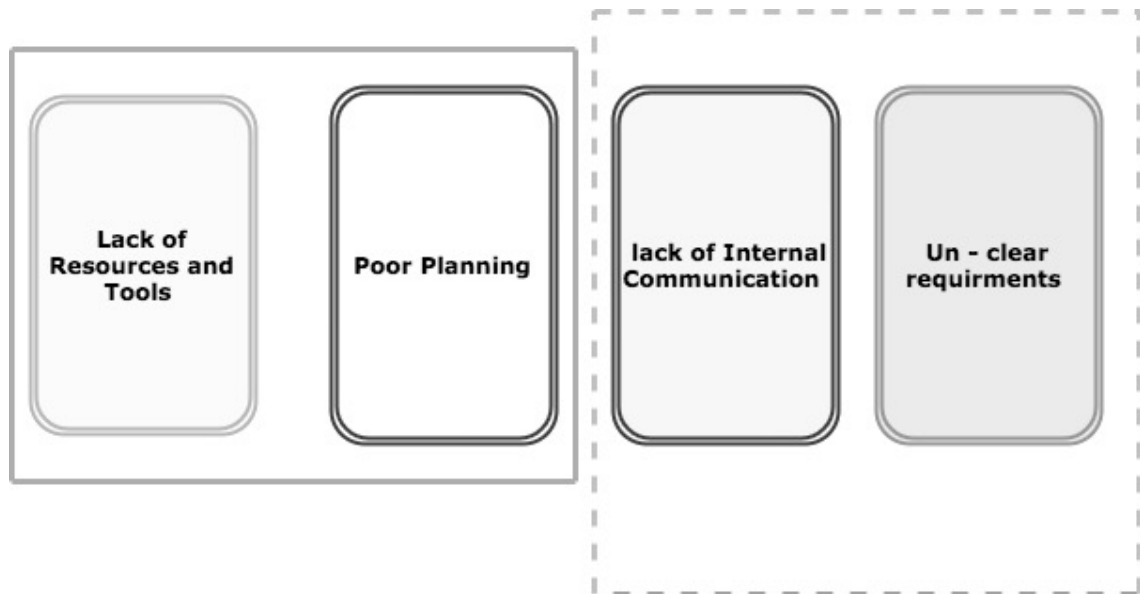


Figure 15. Common challenges related to software project management in software companies

7.3 Criteria for Solution

As discussed in the above section the start-up based software developers face a lot of challenges. The challenges arise from the issues ranging from the concept of globalization. It is the desire to reach a wider market without considering the risk to be incurred. The main challenges discussed in this thesis regarding start-up in the software development are poor planning, poor communication, scope creep, lack of necessary tools and infrastructure, lack of expertise to perform each task of the project development, mis-managed resources, and vague requirements.

The main criteria to solve the problems is to address each issue discussed and the most optimal method of reaching the optimal solution. The criteria is basically a narrative of the solution of each of the problem discussed in the above question. All the problems from the interviews as provided in thematic view above will be handled individually for providing the required solution in project management. The solution will be provided after conducting the necessary research on the internet. The author will consider the information provided by other authors and also give personal insight regarding the problems and the required solutions.

The following will be the main themes which will be exhaustively discussed with the aim of providing a solution. The first, entail setting of clear requirements about the project deliverables and expected milestones before initiating the development process. The

second process is sticking to the initial requirements of the project. The third process is ensuring testing and re-testing of each phase of the software development process. The fourth process and criteria for the solution are through ensuring transparency and accountability through effective communication. The last process is through ensuring scheduling and keeps track of all the timelines for the project. These criteria for solution will be discussed in length in the following subsection.

7.4 Proposed Solutions

7.4.1 Setting of Well-defined and Clear Initial Requirements

It is recommended that for the software developer to start working on the project must ensure clear and realistic objectives are set. It means that the developer and the manager must undertake comprehensive requirements of the project before indulging into it. All the specifications of the end user must be taken into consideration before starting to build the software. Setting clear and realistic goals is ensured by conducting thorough research on the requirements of the customers. It is a common problem that faces many project developers that customers at the start-up have no idea what they need. The issue is knowing how to perform analysis and ask the right information (Ivarsson and Gorschek, 2010). The requirements should be clearly documented in a well-written document. To overcome the problem of the vague requirement is advisable for a developer to have enough time during the start-up of the project aiming at understanding the scope, the deliverables and the objectives of the project. It is also recommended that the developer visualizes the customer's point of view and make viable assumptions through understanding the risk of the project and also the benefits of the project to the end user. It is accomplished through having a concrete vision statement. The vision has user document benefits and the specific functions and the intentions of the project in user usability. Well the customer specification and sign off, think about and read about the customer requirements. Having a clear picture of the recommendations of the both parties easily reaches the target of the project (Sutton, 2000).

7.4.2 Dealing with Unrealistic Customer Requirements

The software developers are always demanded by the customers to demonstrate the importance of the software in providing customer solutions. The managers have to obligation to optimize the solution and the problem constantly (Edward Kit, 1992). It is, therefore, advisable to know what customers require. Through the use few functional requirements, the managers can be able to test the software. The developers must involve the end user by prioritizing and eliciting their primary needs in the design plan. Also checking of alternative solution is an optimal solution in ensuring that the software

meets customer needs. Going hand in hand with the customer is important in adapting to disruptive change affecting the design process.

The use of design patterns and architecture is advisable to check on the factors to remove or update in the software development process. The use frequent code refactoring is required for ensuring each phase of the project is consistent with the requirements of the customer. The developer is also recommended to apply the functionality of quality assurance to ensure quality is maintained. The automatic testing methodology is the most visible in achieving the needs of the customer (Carmel, 1994).

It has been common that most customers see the issues and the problem of the software after the prototypes and the progress is established. At that stage, the original plan of the developer become more visible to the customer who is able to pinpoint the mistakes which occurred in the project. The external environment of the software development is changing progressively requiring the need for reshaping the original plan of the software. The key to solving such problem is to oversee the problem and ensure a backup is already in place and it is working (Yu et al., 2012).

It is recommended that the project developer enlightens the customer about the development process and is entry process in the software development. It is well conducted through incorporating, analyzing, receiving customer value and change results. It is recommended that the project developer set reliable and achievable milestones for each of the development phase (Jim and McCarthy, 1995). For instance, each phase 75% completion should be corrected. Before indulging to changes the stakeholders involved in the process of software development must understand the changes needing to be corrected and must be given an opportunity to provide their rationale regarding the change. The stakeholders may be the funders of the software projects hence notifying those means that the project is capable of escaping the problem of cost overrun (Yogendra, 2002).

7.4.3 Managing Unrealistic Time Frame

The problem of unrealistic time frame occurs in the project when the project is initiated with conducting a thorough analysis of the resources and the scope of the project. The unreasonable timeframe is usually committed by the project developer who lacks the required expertise to understand the requirements of the project (Jim and McCarthy, 1995). The major problems arising from unrealistic timeframes are quality defects and project delays.

The most viable procedure for overcoming the problem of unrealistic timeframe is through understanding what the project requires and drawing a plan of the resources,

deliverables against the time frame. It is recommended that in each phase of the project the developer must be capable of modeling the worst-case scenarios, the middle-case scenarios, and the best-case scenario. Enough time must be segregated for quality inspection and testing of the software against resources constraints. Ensure project deadlines are well communicated by drawing of figures and evidence of the expected risk in each phase (Li, 2007).

7.4.4 Improving Poor Communication

It is common that project developers and customers may fail to understand one another. The technicality of the task may cause the developers fail to reach the target of the customers. It can lead to severe miscommunication and lead to confusion. It is therefore recommended that each party has the accurate understanding of the tasks and the deliverables required for the meeting the project objectives (Dan and Conde, 2002). It is well conducted through ensuring every member of the team get access to the information discussed regarding the progress and the expectations of the projects. It is important for the project developer to be consistent while performing his or her duties. It is important to provide all the stakeholders with a copy of the information and ensure that the stakeholders they stick to the information (Blank, 2013).

Poor communication in the project development for the start-up also arises due to lack of understanding of the politics of the working place. It is important to undertake and understand the power of coalitions, negotiation, and conflict among the stakeholders, the developer, and the project manager must be functional and operational (Davis, 2005). It is important for the manager to understand on how to frame the agendas of the meeting for addressing the pressing issues in every stage of the product development. In this regard, the managers or the project developers must source for all the information required to a developer the project and know who has the information which might be helpful in addressing the project. The manager is also required to think systematically and establish social networks through the building of the relationship and cultivating allies. The group might be reliable in solving major issues derailing the success of the project (Ries, 2011).

7.4.5 High Completion and impact of Globalization

Software development is a process which is highly affected by the concept of globalization. The proliferation of computers and the increase of internet penetration has made almost every individual gain access to the internet and requires software for one use or another. Globalization affects the performance of the software based on the team composition and the input of every member in the project. Team management is very important in improving the performance of software (Edward and Hasted, 2005). The most

critical factors for controlling software development global teams are the team management, and effectiveness of the communication, adequacy of the tools and the selection of the right team to make the product. Most important is the strategy of controlling the external factors which may positively or negatively impact the performance of an organization. It is important for the manager to consider the barriers and the pressure likely to occur when teams are located in the different locations (Coleman and Connor, 2008).

The cultural diversity, expectation and the behavior of each team must be taken into consideration. It is also recommended for the team member to promote communication through the creation of tools which aid in enhancing communication with the team in the global locations. A developer is expected to create tools which help his team to remain connected (Edward and Hasted, 2005). The team must be competent and have enough skills regarding the team at hand. It is also important to allow the team to bond with one another and create a synergy towards solving the problem. In case the software development has a global outlook. It is advisable to select the team which matches with the distributed environments. It is also recommended maintain accountability and transparency with the team members and to the team as a group. The team manager is expected to install collegiality, trust, teamwork, and the right attitude to the members through challenges and guidance. It is also expected that the team members work progress and staff availability to monitor for effective accomplishment (Paternoster et al., 2014).

The setting of quality expectations and goals is important in managing teams for the managers. The managers must also monitor the activities of the team personally. The developer must communicate project goals to the team internally for each member to be aware of their requirements (Luke and Hohmann, 2003).

All the time frame and schedules with the output as desired must be communicated to the team. The team members should be encouraged to communicate to the manager and the rest of the team of areas they feel they cannot contribute earlier enough. Distributed teams are properly managed through advocating for accountability. Team members are also rewarded by promotion and appraisals. An antecedent to success is through ensuring cohesiveness. In these managers leading effectiveness is measured through the ability to blend, to promote interactions within the coworkers, to ensure performance assessment through using of real-time assessment, through the elimination of waste. It is good to understand the team with the diverse cultural background where they originate. Miscommunication through barriers in language and culture is the key to limiting team identity, cohesion and undermining trust (Nobel, 2011).

7.4.6 Tools and Infrastructure

Most of the software development start-up company look for development and technological tools without considering the working experiences and legacy they possess within the team or within the company. Product and product-specific requirements must be taken into consideration when choosing the software tools and infrastructures. The developer must undertake tools which are flexible and accommodative to the change in the software development process of the company. For instance, a developer can select general-purpose tools such as notification system, scheduling system, planning systems, tracking problem reporting, and configuration management (Fairlie and Kauffman 2014).

It is also recommended for the managers to look for easy to implement tools which are capable of analyzing team performance, time, project information and data. Open source applications are more recommended for the start-up companies because of the price effectiveness. The right tools have proven to overcome the problem of lean and agile methodology failures in companies. Effective tools are also important in improving the user experience. The best tools and infrastructure for the software development process requires a well-established and known framework. The framework must be flexible with the capability of meeting market needs and product changes. Effective tools provide room for experimentation and prototyping (Fairlie and Kauffman, 2014).

Tools are also effective in encouraging timely feedback from the end users and customers making the external communication effective. The understanding of user experience of the tools at the start-up is efficient in controlling any loss to be incurred in case the end user changes specifications. The tools they company chooses should focus on the core functionalities and continuous value delivery. The tools must be efficient in improving communication and interactions with the members of the team. Good and perfect infrastructure is important for establishing efficient metrics regarding customers demand and feedback. The use of efficient tools is important in handling fast-paced and facilitating product development for any software business (Fairlie and Kauffman, 2014).

7.4.7 Effective Management Process

In order to make the product, process and projects successful, the project development process must be opportunistic, evolutionary and agile. The ability to respond to changes requires the use of evolutionary and agile methodological approaches. Time and cost overruns are highly reduced through the adaption of incremental and iterative approach. Lean methodology is a variant to agile but most of the companies still claim to use agile methodologies what matters here is whatever methodology company uses, they should try to follow it properly. It entails understanding the software development product process and identifying each phase and the probable issues likely to emerge from each

phase. The use of short-term objectives is understandable and adaptable to changing needs. The project developer and the management can ensure the use of development tools and technology without considering experience but it needs the firm to stick to a particular methodology in order to be successful in the longer run (Ries, 2011).

7.5 Lessons Learned

Starting up a project is not a simple process specially for a new company or a start-up. It is an undertaking which has its own advantages and the disadvantages. The software developer most of the time or the project manager carries the entire burden of the project. The customers are the boss in the process and they dictate the nature of software development. There are so many challenges that the software project managers encounter. Most of the challenges are caused by lack of enough planning and an adequate time speculation. Without research, it is not possible to speculate how much is needed in terms of needed resources and finances as well. The first thing to consider as a software project developer is to improve communication internally as well as externally. Poor communication leads to project failures. Communication is ensured when all the stakeholders meet at least in a month to discuss the progress of the project and within the teams internally there is a regular discussion about what's being done and what are the short and long-term future goals of the company.

It is also important to provide every member of the team with the details of the project to prevent lack of knowledge about any updates. Frequent testing and bug fixing is the only process of ensuring a successful project. Through the testing, a developer is capable of understanding the requirements of the customers and how to integrate them in the second phase of the project; doing so prevent the project from experiencing costs and time overrun. Through proper communication, a developer understands the basic bottlenecks facing the projects. The politics are effective in solving problems which arise in the course of the project. The project manager is expected to gather all the tools required to run the project. The more and flexible the tools are the more efficient and effective is the process of managing software development. Update tools are also important in providing real-time data and information.

8. SUMMARY AND CONCLUSION

The main objective of the thesis was to analyze the challenges faced by the software start-up in project management by taking into account the existing literature about the topic and taking into account the empirical data gathered during the research process from case companies so the study can be helpful and beneficial to other software start-up companies.

In order to achieve the main objective of the thesis during the research process the interviews were conducted to gather the empirical data from start up software companies. The Software developers and the managers from the two cases of start-up companies were involved in providing the empirical data. The study employed the use of thematic analysis to develop the results of the interviews which were back up by secondary information through literature review and from similar experiences by working in one of the software start-up. The main challenges according to the data were lack of internal communication, lack of internal resources, poor planning and un-clear requirements. Major challenges that were analysed from the empirical data and from the literature review were divided into 3 main categories, internal challenges, operational challenges and external challenges which are further divided into 4 major challenges that are common in almost every software start up company and they are lack of resources, poor planning, lack of internal communication and sometimes un-clear requirements about the products or services.

Start-up software companies also face a major challenge from the external stakeholders since most of them have some external financial stake involved. Some of the policies and customer requirements may not also be very effective to develop the idea for the growth of the company. The start-up companies also have unrealistic timelines mostly. The software developer plans for the project using scalable tools is able to accomplish success. Tools are also efficient in managing time and resources required to run the activities of the project. Most software start-up experiences the problem of insufficient cash flow; Moreover, increase in income necessitates them to invest globally.

Most software project developers face the problem of lack of time management during the development up to the implementation phase working in the start-ups. Most of the projects and product end up incurring time and cost overflow. It is attributed to delays incurred by including new consumer requirements in the prototypes or delays associated with redesigning. Most of the start-ups in the process of software development lack enough finances to start and develop their ideas. The start-up also requires a lot of finances to develop implement and sustain the functionality of the software project or

product in the market. As a result, some of the software developers' end up selling the ideas to the well-established organizations or others choose to maintain the ideas but become obsolete. Most of the software developers claim lack of experienced expertise in the different stages as the main problem that they face during the software start-up process.

The software development teams may have the expertise to work on a single phase but may lack the capability to work on the subsequent phase causing a gap in the market. Planning is an important undertaking to a software project developer. Without proper planning, the developer is likely to experience loss and delays. Planning starts from the ground through conducting research to identify what are the requirements and the specifications of the customers regarding the projects. Planning has also proven efficient in estimating the project resources. Through such estimation, the developer understands which phase requires a particular project and which phase does not. Moreover, software planning is important in providing a framework which serves as a guideline for the entire process. Most software start-up fails because they do not plan what is required in the start-up. Through proper planning, the realistic timeframe is met. Planning is essential in providing a design of the entire project development process and the expected deliverables plus milestones. Communication and transparency are enhanced through proper planning and projects.

The issues of costs and time can be sorted out through considering the important factors which may affect the development of the project in the different phases. Proper management is also an important factor which ensures the success of the project. Proper management means all the project milestones and deliverables will be achieved since the management is well versed and equipped with the right skills to handle the task at hand. Management must understand and appreciate the role played by each member in the project design and development. The management is concerned with understanding the input of every member of the team and proving motivation which can be both intrinsic and extrinsic.

The higher management in the start-up companies must also understand the team in terms of diversity and distribution. The company must have what it takes to manage distributed teams. Each member of the team must be given the opportunity to input about the team progress must the input should be carefully designed not to affect the progress and the expected milestones and deliverables. These can be solved easily these days through automated communication tools like Slack which has been used by many companies as an internal and external communication and reporting tool.

Unrealistic expectations from the customers is very a common challenge in software development process which is faced by start ups in the software development industry. Customers' expectations keep on changing depending on the market dynamics. The developer must calve a product which is smart enough to entice the customers and

maintain importance. In this regard, the aspect of software testing, realistic timeframe, proper management, good communication and proper planning must be taken into consideration during the start-up period by the software entrepreneurs.

Since because of the nature of lean start-up architecture there are a lot of challenges that a newly formed company can face apart of the ones that have been analysed form this study but if we talk about software start-up companies in particular since almost 90% of them start by offering products or services in in software industry most of the challenges can be solved by doing effective project management which is a key challenge that almost every software start-up company face these days. Effective Project management can enable companies to utilize limited resources they have to achieve high end results that can be sustainable, problems like lack of communication, and non clear requirements for products or services can easily be solved by adopting a particular management methodology within the company and could be helpful in solving key challenges and key issues related to the development and sustainability of the company resulting in long-term success and profitability.

8.1 Proposals for Future Research

The research was conducted to mainly address the major challenges in project management faced by start-up companies since the study take into account that effective project management or at least in case of software start-ups if the firm manages to identify the major challenges and effectively manages to perform project management in proper manner most of the major issues related to the development process can be solved.

This study was limited to analyze the most common and major challenges in software start-up companies by taking into account the most common and traditional working practices and software methodologies used in the industry, since there are a lot of latest methodologies introduced these days for project management and trends are changing really fast, the study can be used as a starting point to further develop best practices and methodologies needed for effective operations of software start-ups or starting point for developing a study for helping the new software start-up companies in defining the best practices for effective project management in newly formed businesses.

REFERENCES

- Ambler, (2002). "Lessons in Agility from Internet-Based Development," IEEE Software, vol. 19, no. 2, 66–73.
- Buchanan, D. A. (2011). "The SAGE handbook of organizational research methods".
- Bjornvig, G., & Coplien, J. O., 2014. "Lean architecture: For agile software development." Hoboken, N.J: Wiley.
- Blank, S., 2013. "Why the Lean Start-Up Changes Everything." Harvard Business Rev., vol. 91, no. 5, 2013, p. 64.
- Cann, R., Jansen, S., & Brinkkemper, S., 2013. "Software business start-up memories: Key decisions in success stories."
- Carmel, E., 1994. "Time-to-Completion in Software Package Startup." Proc. 27th Hawaii Int'l Conf. System Sciences: 498–507
- Chemuturi, M., 2009. "Software estimation best practices, tools & techniques: A complete guide for software project estimators." Fort Lauderdale, Fla: J. Ross Pub.
- Chin, G. (2004). Agile project management: How to succeed in the face of changing project requirements. New York: AMACOM.
- Cockburn, A., & Highsmith, J., 2001. "Agile software development, the people factor." Computer, 34(11), 131-133.
- Coleman G., & O'Connor, R., 2008. "An Investigation into Software Development Process Formation in Software Start-ups." J. Enterprise Information Management, 21 (6); 633–648.
- Crowne, M., 2002. "Why Software Product Startups Fail and What to Do about It." Proc. Int'l Eng. Management Conf. (IEMC): pp. 338–343.
- Cuppett, M. S., 2016. "Managing Data Platforms to Support Continuous Integration". Berkeley, CA: Apress
- Curran, K., & Curran, N. 2014. "Social Networking Analysis." London: SpringerLink
- Dan and Conde (2002). "Software Product Management: Managing Software Development from Idea to Product to Marketing to Sales."
- Davis (2005). "Just enough requirements management: where software development meets marketing."

Davis (2005). "Just enough requirements management: where software development meets marketing."

Edward and Hasted. (2005). "Software That Sells: A Practical Guide to Developing and Marketing Your Software Project."

Edward Kit (1992). "Software Testing in The Real World."

Fairlie and Kauffman. (2014). "Index of Entrepreneurial Activity." Kauffman Foundation."

Fairlie, (2014). "State of Entrepreneurship Address." Kauffman Foundation.

Golafshani, N. (2003). "Understanding Reliability and Validity in Qualitative Research. *The Qualitative Report*, 8(4), 597-606.

Galín, D, 2004. "Software quality assurance: From theory to implementation." Harlow u.a.: Pearson/Addison Wesley.

Geert, P., 1994. "Dynamic systems of development: Change between complexity and chaos." New York: Harvester Wheatsheaf.

Giardino, C. et al., 2014. "What do we know about software development in startups?" Blekinge Tekniska Högskola, Institutionen för programvaruteknik.

Gulati, R. & De Santola, A., 2016. "Startups that Last." Harvard Business Review. Retrieved from <https://hbr.org/2016/03/start-ups-that-last> [Accessed on 17.02.17].

Highsmith, J., & Cockburn, A., 2001. "Agile software development: The business of innovation." *Computer*, 34(9), 120-127.

Hirsjärvi, S., Remes, P. & Sajavaara, P. (2007). *Tutki ja kirjoita*. 13th edition. Kustannusosakeyhtiö Tammi. Helsinki. p. 448.

Hobbs, D. P., 2011. "Applied lean business transformation: A complete project management approach." Ft. Lauderdale, Fla: J. Ross Pub.

Howe, K. R. (2003). "Closing methodological divides: Toward democratic educational research." Dordrecht: Kluwer Academic.

Ivarsson and T. Gorschek.(2010). "A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations." *Empirical Software Eng.* vol. 16, no. 3, pp. 365–395.

- Janes, A., Succi, G., & Springer-Verlag GmbH., 2016. "Lean Software Development in Action." Berlin: Springer
- Javadtalab, A., Shirmohammadi, S., Hosseini, A. M., & University of Ottawa., 2015. "An End-to-End Solution for High Definition Video Conferencing over Best-Effort Networks."
- Kane, P., 2004. "The lifecycle of a technology company. Boston, Mass."
- Kammerhofer, J. (2012). "Positivism." New York: Oxford University Press.
- Kerzner, H., 2015. "Project management 2.0: Leveraging tools, distributed collaboration, and metrics for project success." Hoboken, New Jersey: Wiley
- Kurtz and D.J. Snowden. (2003). "The New Dynamics of Strategy: Sense-Making in a Complex and Complicated World," IBM Systems J., vol. 42, no. 3, 2003, pp. 462–483.
- Lester, A., 2013. "Project Management, Planning and Control." Elsevier Science. Elsevier Science: Butterworth-Heinemann
- Levine, M. K., 2009. "A tale of two systems: Lean and agile software development for business leaders." Boca Raton: CRC Press.
- Li, (2007). "The Role of Value Proposition and Value Co-Production in New Internet Startups: How New Venture e-Businesses Achieve Competitive Advantage." Portland Int'l Center for Management of Engineering and Technology (PICMET), pp. 1126–1132.
- Luke and Hohmann (2003). "Beyond Software Architecture: Creating and Sustaining Winning Solutions."
- Mall, R., Panigrahi, C. R., & Pati, B. R., 2017. "Software Development Methodology for Cloud Computing and Its Impact."
- Martin, R. C., 2014. "Agile software development principles, patterns, and practices." Harlow: Pearson Education Ltd
- Menard, S. W. (2008). "Handbook of longitudinal research: design, measurement, and analysis." Burlington, MA: Elsevier/Academic Press.
- Mochal, T., & Mochal, J., 2011. "Lessons in Project Management. Berkeley, CA: Apress."
- Nobel, C., 2011. "Why Companies Fail, and How Their Founders Can Bounce Back." Harvard Business School.

Parker, G. M., 2009. "Team leadership: 20 proven tools for success." Amherst, MA: HRD Press.

Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P., 2014. "Software development in startup companies: A systematic mapping study. *Information and Software Technology*", 56(10), 1200-1218.

Pekkola, S. & Nieminen, N. (2015). "Comparison of Functional, Structural, and Dynamic Business-IT Alignment Models: A Case Study." *Proceedings of the 21st Americas Conference on Information Systems (AMCIS'15)*.

Ries, E., 2011. "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses." New York: Crown Business.

Saunders, M., Lewis, P. & Thornhill, A. (2009). "Research methods for business students." 5th edition. Pearson Education. United Kingdom. p. 614.

Schwaber, K., & Beedle, M., 2002. "Agile software development with Scrum" (Vol. 1). Upper Saddle River: Prentice Hall.

Schwalbe, K., 2009. "Introduction to project management." Boston, Mass: Course Technology.

Sliger, M. & Broderick, S., 2008. "The Software Project Manager's Bridge to Agility." London: Addison-Wesley Professional.

Stellman, A., & Greene, J., 2006. "Applied software project management." Beijing: O'Reilly.

Straub, B., 2015. "Git Fundamentals: GitHub Integrations." Nashua, New Hampshire: Skillsoft Corporation.

Sutton, (2000). "The Role of Process in Software Start-ups." *IEEE Software*, vol. 17, no. 4, 2000, pp. 33–39.

Tyler, J., 2015. "Building great software engineering teams: Recruiting, hiring, and managing your team from startup to success."

Villafiorita, A., 2016. "Introduction to Software Project Management. New York: CRC press."

Williamson, K., Bow, A., & Charles Sturt University. (2002). "Research methods for students, academics, and professionals: Information management and systems."

Yogendra, (2002). "Aligning Business and Technology Strategies: A Comparison of Established and Start-up Business Contexts," Proc. Int'l Eng. Management Conf. (IEMC), pp. 2–7.

Yu et al., (2012). "Entrepreneurial Success for High-Tech Start-ups: Case Study of Taiwan High-Tech Companies," Proc. 6th Int'l Conf. Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 933–937.

APPENDIX A: INTERVIEW QUESTIONS FROM THE CASE COMPANIES

Theme 1: Introduction organization structure and business domain (Interviewee: CTO or project manager)

- How old is your company and what particularly is business about and primary location of customers?
- What is the structure of the software development team in the company?
- What is the size of the development team and how many team leads or project managers?
- Do you have multiple teams working on the projects?
- How many projects usually the teams in the company manage?
- What are your main customers or what particularly are end users for your software products?
-

Theme 2: Software Development and Project management (Interviewee: CTO or project manager)

- Which software development methodology or model is used by your company?
- Do you use scrum, Kanban or any other methodology?
- Do you use sprints? How long are your product development sprints?
- Do you have scrum meetings or daily standups or sprint reports, how long are they?
- How do you test your products or projects, is there a dedicated software testing or quality assurance team in the company?
- What is the customer involvement in the software development cycle in your company do you get feedbacks after each phase etc.?
- Have you ever faced difficulties with project scheduling and deadlines? if yes what are the bottlenecks mainly (planning, over estimations, un realistic deadlines etc.)
- Have you ever faced a situation during the project when it ran over budgeted, and crossed the cost estimates, or have you ever ran out of funding for the projects?

Theme 3: Tools and platforms (Interviewee: project manager):

- What tools are being used for your company to manage the sprints or whole product development cycle or particularly by what soft wares you manage the projects?
- How do you communicate within the team about ongoing progress, updates, bottlenecks etc. within the team do you use any software for that?
- Do you document your products or do you provide documentation for the customers regarding the products? If yes what software you use or its just simple pdf, word etc.? how well documented you think they are?
- Do you use any testing platforms or soft wares to test your products for example automated testing etc.?
- Do you use any platform or software for bug reporting?

Theme 4: Team management (Interviewee CTO and developers)

- What do you think about your product development cycle and methodology as a whole? Do you think there is rom for improvement?
- Do you think the teams and team size are enough to manage these projects or they are over burdened?
- Do you think that each time the requirements are clear and easy to understand or there can be more improvement to these?
- Do you think that deadlines are realistic and easy to manage and deploy the projects with the planned deadlines?
- Do you think there is enough knowledge and support provided within the team and form the higher management to build the product etc.?

Theme 5: Major challenges for software development which can be improved (Interviewee: Team lead, developers or CTO)

- Do you think that there is a lack of planning or requirements are sometimes unclear form the client's perspective or from the product development team that should be improved?
- Do you think that there is a lack in software quality insurance specially in startups that should be improved?
- Why do you think working with startups is more challenging and what are the differences you see in terms of project management or product development while working with startups?

- What do you think is the most crucial thing that you face related software project management working in startup environment?
- Any suggestions or comments regarding this interview or for the research as a whole?