Software Requirements Prioritization Practices in Software Start-ups

A Qualitative research based on Start-ups in India

Sravika Kothawar
Rakesh Guptha Vajrapu
This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Sravika Kothawar
E-mail: srko16@student.bth.se

Rakesh Guptha Vajrapu
E-mail: rava16@student.bth.se

University advisor:
Eriks Klotins
Department of Software Engineering

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
ABSTRACT

Context. Requirements prioritization is used in software product management and is concerned with identifying the most valuable requirements from a given set. This is necessary to satisfy the needs of customers, to provide support for stakeholders and more importantly for release planning. Irrespective of the size of the organization (small, medium and large), requirements prioritization is important to minimize the risk during development. However, few studies explore how requirements prioritization is practiced in start-ups. Software start-ups are becoming important suppliers of innovative and software-intensive products. Earlier studies suggest that requirements discovery and validation is the core activity in start-ups. However, due to limited resources, start-ups need to prioritize on what requirements to focus. If they do it wrong it leads to wasted resources. While larger organizations may afford such waste, start-ups cannot. Moreover, researchers have identified that start-ups are not small versions of large companies and the existing software development practices cannot be transferred directly due to low rigor in current studies. Thus, we planned to conduct an exploratory study on requirements prioritization practices in the context of software start-ups.

Objectives. The main aim of our study is to explore the state-of-art of requirements prioritization practices used in start-ups. We also identify the challenges associated with the corresponding practices and few possible solutions.

Methods. In this qualitative research, we conduct a literature review by referring to many article sources like IEEE Xplore, Scopus and Google Scholar to identify the prioritization practices and challenges in general. An interview study is conducted by using semi-structured interviews to collect data from practitioners. Thematic analysis was used to analyze the interview data.

Results. We have identified 15 practices from 8 different start-ups companies with corresponding challenges and possible solutions. Our results show mixed reviews in terms of the prioritization practices at start-ups. From the total of 8 companies about 6 companies followed formal methods while in the remaining 2 companies, prioritization was informal and not clear. The results show that value-based method is the dominant prioritization technique in start-ups. The results also show that customer input and return on investment aspects of prioritization play a key role when compared to other aspects.

Conclusions. The results of this study provide an understanding of the various requirements prioritization practices in start-ups and challenges faced in implementing them. These results are validated from the answers found in the literature. The solutions identified for the corresponding challenges allow the practitioners to approach them in a better way. As this study focused only on Indian software start-up companies, it is recommended to extend to Swedish software start-up companies as well to get a broader perspective. Scaling of sample size is also recommended. This study may help future research on requirements engineering in start-ups. It may also help practitioners who have an intention to begin a software start-up company to get an idea of what challenges they may face while prioritizing requirements and can use these solutions to mitigate them.

Keywords: Requirements engineering, Requirements Prioritization, Software systems, techniques, Software Start-ups, Challenges.
Acknowledgements

“My journey through Master’s program was a transformation period in my life which taught me the path to the zenith, with a naive approach. I started my technical education at Jawaharlal Nehru Technological University, Hyderabad (JNTUH) and Blekinge Tekniska Högskola, Sweden (BTH) has given me an opportunity to develop not only as a student but also reinforced my skills and helped to explore my inner sense and individuality. I take this as an opportunity to express my heartfelt gratitude and would thank my parents Shri. Ganesh Kothawar and Smt. Anuradha Kothawar for not only believing in me but also supporting me through every walk of life. I would also thank my sister Miss. Sowjanya Kothawar for extending her due support and uplifting my confidence. I would like to express gratitude to my supervisor and guide Eriks Klotins for his exceptional support and knowledgeable guidance who has created commendable targets which have constructed a strong path to achieve the targets invariably. I would thank my cousins, family, and friends for their everlasting love and confidence in me.”- SRAVIKA KOTHAWAR

“Firstly, I am thankful to people who helped me to perform this study successfully with proper inputs throughout the thesis. I am also thankful to our supervisor Eriks Klotins, with heartful gratitude towards his support and guidance throughout the thesis. Without his effort, we could not have completed our thesis in the speculated timeline. I also thank the Department of Software Engineering, BTH for trusting us and encouragement throughout the studies. I am thankful to my parents Shri. Vajrapu Koteswararao garu and Smt. Vajrapu Roopa garu to have them in my life and for supporting me. I am so glad towards your love and affection. I am so thankful to my friends and relatives, who taught me love and life.”- RAKESH GUPTHA VAJRAPU
CONTENTS

ABSTRACT ......................................................................................................................... 1

1 INTRODUCTION .............................................................................................................. 5

2 BACKGROUND AND RELATED WORK ........................................................................... 7
  2.1 SOFTWARE START-UPS ............................................................................................... 7
  2.2 REQUIREMENTS ENGINEERING IN START-UPS ....................................................... 8
  2.3 REQUIREMENTS PRIORITIZATION ........................................................................... 9
  2.4 RELATED WORK ....................................................................................................... 14

3 RESEARCH METHODOLOGY ....................................................................................... 16
  3.1 RESEARCH AIM AND OBJECTIVES ....................................................................... 16
  3.2 RESEARCH QUESTIONS .......................................................................................... 16
  3.3 RESEARCH METHODS ........................................................................................... 17
      3.3.1 Literature Review .............................................................................................. 17
      3.3.2 Interview Study ............................................................................................... 19
  3.4 DATA COLLECTION .................................................................................................. 19
      3.4.1 Interview Guide ............................................................................................... 19
      3.4.2 Sampling data ................................................................................................. 20
  3.5 DATA ANALYSIS ...................................................................................................... 21
      3.5.1 Validation ........................................................................................................ 22
  3.6 VALIDITY THREATS ................................................................................................. 23
      3.6.1 Internal Validity ............................................................................................... 23
      3.6.2 External Validity .............................................................................................. 23
      3.6.3 Construct Validity ............................................................................................ 23
      3.6.4 Conclusion Validity ........................................................................................ 24

4 RESULTS .......................................................................................................................... 25
  4.1 LITERATURE REVIEW .............................................................................................. 25
  4.2 INTERVIEW STUDY .................................................................................................. 26
      4.2.1 Demographic Results ....................................................................................... 26
      4.2.2 Interview Data ................................................................................................. 28

5 ANALYSIS AND DISCUSSION ....................................................................................... 32

6 CONCLUSION AND FUTURE WORK .......................................................................... 41
  6.1 FUTURE WORK ......................................................................................................... 41

REFERENCES ...................................................................................................................... 42

APPENDIX ........................................................................................................................... 48
List of Tables

Table-1 Search string and number of articles
Table-2 Frequency of occurrence of practices identified in LR
Table-3 Demographics of interviews
Table-4 Type of Market
Table-5 Prioritization Practices
Table-6 Factors used in prioritization
Table-7 Theme 1: Analysis of prioritization practices
Table-8 Theme 2: Analysis of prioritization factors
Table-9 Theme 3: Analysis of challenges faced in start-ups
Table-10 Solutions to the corresponding challenges
Table-11 Analysis of pros and cons corresponding to solutions
Table-12 Literature review results
List of Figures

Figure 1  Requirements engineering activities
Figure 2  Factors considered while Requirements Prioritization
Figure 3  Overview of Research Method
Figure 4  Events within the Interview Study
Figure 5  Paper publication distribution per year
Figure 6  Percentage of Documents type
Figure 7  Value-based Requirements Prioritization
**List of Appendices**

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A</td>
<td>Literature review results</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Concept Centric Matrix</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Interview Questionnaire</td>
</tr>
<tr>
<td>Appendix D</td>
<td>Color Coding for Interviewees</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

Software startup companies are an interesting phenomenon as they develop innovative software-intensive products under time constraints and limited resources [1]. They constantly search for sustainable and scalable business models [2]. Software startups are not only quite distinct from mature companies, but also from micro-, small-, and medium-sized enterprises, introducing new challenges relevant for software engineering research [3]. More and more software start-up companies are launched into the market every day producing cutting-edge products [4]. However, the failure rate among start-ups is high and is explained by lack of resources, immaturity, multiple influences and dynamic technologies [5] [2]. As a result of these challenges, most start-ups do not survive the first few years of operation and cease to exist before delivering any value [6] [7].

Over the past few years, software start-ups have gathered increased research interest in the software engineering community [8] [9] [10]. However, the role of software engineering practices in start-ups and their impact on product success has not yet been explored in depth [11,12,13,14]. In addition, the state-of-art presented in the literature is not enough to base an understanding of how software engineering practices could help software start-ups [15] [3]. These inadequacies in applying software engineering practices could lead to sub-optimal product quality and be a significant contributing factor to start-up failure [15]. Several studies elaborate the differences between start-ups and established companies, noting that start-ups are defined by limited resources and dynamic technologies [11,1,16]. However, more practices are needed to support a comparison of engineering contexts in different companies, making the transfer of practices from company to company difficult [2, 17]. Therefore, engineering practices specific to start-up’s context are needed.

Earlier studies suggest that requirements discovery and validation is the core activity in start-ups. However, due to limited resources, start-ups need to focus on what requirements to prioritize. If they do it wrong it leads to wasted resources. Requirements prioritization is an activity to balance needs of various stakeholders and to pick a subset of requirements for implementation [20]. In start-ups, it means to balance limited resources, time to market and uncertain ideas for product features [1]. Requirements prioritization has been widely studied in established companies, however, few studies explore how requirements prioritization is practiced in start-ups. This is because start-ups hardly follow any prescriptive methodology and face a unique context in which they need to prioritize requirements that are usually not clear and unstable [19].

Thus, studying this context is important as start-ups face challenge in scoping the Minimum Viable Product (MVP) [3]. Constrained by lack of resources and need to be first in the market, start-ups aim to deliver a product release with the minimum useful set of features, MVP to test if there is market interest to justify further investments in the product development [21]. For small organizations, proper requirements prioritization and selection can mean the difference in not only project success or failure but also overall company survivability [22]. A single requirements catastrophe will drive a small company out of business [23] [4].

In the previous studies, requirements prioritization practices, in general, have been studied extensively. For example, Svensson et al. [24] identify how quality requirements are prioritized in 11 successful companies developing software-intensive systems. Lehtola et al. [25] describe the current state of requirements prioritization in large and medium-sized companies and present the practical challenges involved. However, the research on requirements prioritization in the context of software start-ups is limited. For example, Melegati et al. [18] discuss how requirements engineering processes are executed in software
start-ups by conducting semi-structured interviews with nine different start-up companies where requirements prioritization was discussed in only one small section.

The aim of this study is to explore prioritization practices in start-ups. This paper presents the results of an empirical study that includes data collected through semi-structured interviews with 8 practitioners from 8 different start-up companies in India. As Indian software industry has build up valuable brand equity for itself in the global market and is tremendous innovation and growth in technology, we choose our homeland as a base for our research. The main contribution of this paper is an empirical investigation that identifies the requirements prioritization practices used in software start-ups, the corresponding challenges faced by them and the possible set of solutions to mitigate them. These solutions might be helpful for software practitioners and researchers to build a more comprehensive, empirical knowledge base in order to support forthcoming software start-ups.

The rest of the paper is structured as follows. In section 2 we present the background and related work introducing relevant concepts to understand our study. Section 3 presents the research methodology while results are presented in section 4. The results are interpreted in section 5. Finally, in section 6 a conclusion is presented including major contributions and possibilities for future studies.
2 Background and Related Work

This section presents a detailed background study and related work according to existing literature. While section 2.1 describes the characteristics of start-up companies and how they are different from established companies, section 2.2 gives an overview of requirements engineering activities in software start-ups. Finally, section 2.4 elaborates about the requirements prioritization techniques and the importance of studying them in the context of software start-ups.

2.1 Software Start-ups

A start-up is a newly emerging, fast-growing company which intends to develop an innovative product [5]. Ries et al. [26] defines a software start-up as a human institution designed to deliver a new software product or service under conditions of extreme uncertainty. Start-ups are temporary organizations constantly looking for repeatable and scalable business models [27]. They have limited resources in terms of people and investment, and are run on very tight schedules [3]. In addition to that, they are commonly exploratory in nature, lacking clear requirements, customers, and even business models. Software start-ups are more popular than ever and growing in numbers [1]. Emerging technologies such as smartphones, cloud infrastructure platforms, and enhanced web development tools have contributed to the success stories surrounding start-ups such as Facebook, Twitter, Instagram, Snapchat etc. [28][5]. Software start-ups face multiple influences in the decision-making process as they face pressures from investors, customers, and competitors [4]. As a result, most of the start-ups concentrate more on gaining maximum value for their investments and give less importance to development effort [29]. Software start-ups often make assumptions about the problems and customers they are addressing as well as the market and the solutions they are developing [30]. In start-up companies, the incentives for the employees are determined by the corporate culture and goals of the start-up company [11]. It is also found that software product development is the core activity of software start-up companies [2].

To understand what challenges are faced by software start-ups, it is very important to understand what constitutes a software start-up [29]. High uncertainty and rapid evolution are the main characteristics of start-up companies [7]. They are less robust and balanced and as a result, they possess less ability in facing challenges when compared to established companies. Start-up companies usually fall short of development time and resources [31]. Despite few cases, over 98% of new software product ideas fail in the market [30]. Majority of these start-ups companies fail within two years from their creation [4]. This has led the researchers to try and identify what factors contribute towards software start-ups success [32][33][34][35].

While established companies execute their development activities using the existing resources and business models, start-up companies search for sustainable and scalable business models [36]. Software start-ups are different from established companies. Established companies own mature processes, talent pool, maturity, partners and a lot more while start-ups lack all these capabilities initially [27]. While established software companies present advantages with respect to start-ups, such as fewer internal communication and coordination problems, a foundation of established products, partners, and customers with a greater shared history and vision [11], start-up companies often take a linear internal communication. While software project failures can lead to business failures in established companies, the situation is more worst in software start-ups as one failed project can put them out of business [37]. Thus, start-ups need effective practices to face those unique challenges [4]. Recent studies suggest agile and lean principles as ways to increase the odds
of succeeding as a start-up. While well-established companies have succeeded in implementing these principles, software start-ups face difficulties in advocating these practices throughout the product lifecycle [38]. Besides this, established companies also succeed with traditional development approaches oriented towards process improvement initiatives. As a result of these challenges and differences, start-up companies have a high risk to fail when compared to mature companies. Recent studies show that about 60% of the start-up companies do not survive more than five years[16]. Therefore, research is essential to support software engineering activities in the context of start-up’s, to guide practitioners in decision making and helping them from making choices which may lead to failure in business [16].

2.2 Requirements Engineering in Start-ups

Requirements Engineering has been recognized as one of the key activities in software projects. Shortcomings in requirements engineering process have often lead to budget and schedule overruns [39]. In start-up setting, this may mean the collapse of the company altogether. Very often the software requirements in established companies are validated by users or customers. Whereas in start-ups, the users or customers are usually unknown. Software start-ups face extreme time-pressure from the market and are revealed to strong competition [15] [40]. Thus, choosing right requirements and adapting to new requests is an essential success factor in this environment [41].

Requirements engineering activities essentially involve gathering, documenting, and managing requirements. Thus, performing every activity throughout the life-cycle of product development is important. With the increasing need to understand the requirements in the software process, requirements engineering is becoming a focus area for research progressively in the field of software engineering. Kotonya and Sommerville[42] suggest traditional requirements engineering model which involve activities like requirements elicitation, requirements analysis, and negotiation, requirements documentation and requirements validation.

![Requirements Engineering activities](image)

The first stage of requirements engineering is requirements elicitation which can also be referred as requirements gathering or acquisition. Here, the requirements are collected from the system users, clients and developers who are identified to be the system stakeholders. The main objective of this process is to find the problem that should be solved and then defining system boundaries [42]. Performing elicitation could be an extremely challenging task for start-ups which focus on a large group of people. However, it is possible to define
what requirements can be developed through interacting with the stakeholders. Thus, from these collected requirements several unavoidable inconsistencies appear like various sources of conflicts, inadequate details to reach the project scope, etc. As it is necessary to analyze detailed requirements at this stage, negotiating with stakeholders is performed to decide what requirements to implement. These selected requirements are then documented and monitored during the development. In the next stage which is validation, requirements are verified and validated before implementation to check if these requirements are complete and consistent. It is also necessary to test if the implemented requirements have accomplished their objectives [42]. Moreover, start-up companies need to focus on requirements prioritization due to their high customer expectations, short timelines, and limited resources. This allows them to deliver the most important functionality of the system to their clients on time [7].

2.3 Requirements Prioritization

Software product quality is determined by the ability to fulfill the needs of customers and users [43]. Thus, eliciting the requirements and identifying the correct requirements prior to the release of the suitable requirements with proper functionality is the major step to product success. Candidate requirements are the most important inputs in building any software product. These can be realized with some time and cost constraints. Requirements Prioritization is an integral part of decision-making [44]. This prioritization helps in distinguishing the crucial requirements from the unimportant ones. The advantages of requirements prioritization are: it helps to estimate the customer’s satisfaction, it reduces rework and plan stability, it helps in providing relative importance to each and every requirement which will, in turn, give requirements with high value with low costs etc. These activities show the importance of prioritizing and deciding what requirements to be included to develop a product [45]. Managing software requirements plays a crucial role in requirements engineering process to deliver high-quality software products [46] [24]. Requirements prioritization, on the other hand, is recognized as an important but challenging activity in software engineering [47]. Shortcomings in requirements engineering are identified as the primary cause of software project failures [48]. Later, the final requirements prioritized might be the basis of product and marketing plans and even be a driving force during the project plan [45]. Ruhe et al. [40] summarized “the challenge is to select the ‘right’ requirements out of a given set of candidate requirements so that all the different key interests, technical constraints, and preferences of the critical stakeholders are fulfilled, and the overall business value of the product is maximized.” It is even possible to rectify the imprecise decisions later using change management, but this process is significantly expensive to correct problems later in development stage [41]. Thus, the cost-effective way of developing a software is to find the optimal set of requirements first and then develop the software. Other benefits of prioritizing requirements are that it can also find the requirements defects such as incorrect, ambiguous and misjudged requirements. They are analyzed from a different perspective where requirements are taken during the review. Thus, requirements are initially vague and become more explicit as understanding on the product grows [40]. Therefore, requirements prioritization is an iterative process and it can be performed in abstraction levels and with different information in different phases of software development lifecycle.

Requirements prioritization can be categorized into two categories: methods and negotiation approaches [49]. Methods are depending on quantitatively assigning values to various factors of requirements while negotiation approaches focus on giving priorities based on stakeholder’s views. Prioritization decisions are made by stakeholders including users, managers, developers, customers etc. considering different aspects like importance, risk, cost, customer input and value [50]. In addition, customers and developers utilize a vast number of methods or practices to select this subset [46] [47].
Despite the benefits offered by various requirements prioritization techniques, it is considered as a challenging activity. It is not possible to consider complete set of requirements in single release due to constraints like limited resources, time to market and budget problems faced by practitioners [22]. These challenges hinder the organization from fulfilling the interests of the stakeholders and enhance the business value [15]. Further, these problems increase the cost of rectification of the system and diminish the product quality [15]. While some practitioners consider requirements prioritization as an easy process and claim that software companies follow systematic prioritization methods, the literature shows that few methods were impractical and have not been applied yet. There is no evidence claiming a prioritization technique as the “right one” [15]. Companies select practices that are suitable for their domains, size, and objectives. Therefore, it is very important to overcome these challenges by selecting the right set of requirements. Moreover, requirements always have dependencies between them and the priorities are relative [51]. These priorities along with factors affecting them keep on varying along with the time. Thus, recognizing the concept of prioritization is not clear [51]. There is a need for professional skill and domain knowledge in implementing requirements prioritization practices.

**Factors to prioritize requirements**

Factors are aspects which are taken into consideration while prioritizing requirements. Some frequently considered factors are Cost, Time, Importance, risk etc. [20]. It is easy to prioritize using single factor. However, in considering multiple factors the high priority ranked requirements would be less priority with some change in factors[20]. It is always essential to consider the factors that affect the software development and those who satisfy the customers with end-product.

- **Cost**: The cost estimations to implement any product is done by the development team. Complexity in implementing the features, reusing the existing code, testing the product costs and documentation are few activities involved when cost is considered [20].
- **Time**: Time is interrelated to many other actors like time required to train the customers, complete all the industry standards, time need for development support of infrastructure etc.[20].
- **Importance**: The importance of requirements differs from every individual and perspective of every stakeholder [20].
- **Risk**: Risks cause delays in the development of the project. Thus, based on the risk estimations, each requirement is ranked [20].

![Figure 2: Factors considered while Requirements Prioritization](image-url)
There are several techniques to prioritize the requirements. To categorize or quantify the requirements scales are used. Four different scale types are described in [49] i.e. Nominal, Ordinal, Interval, and Ratio. Some scales contain more information when compared to others these scales are called richer. The nominal scale includes categorization and classification, i.e. the objects are grouped, and each subgroup is assigned a number or name. the only statistics that can be gathered in this technique is frequency and the mode can be calculated but not the mean or median. An ordinal scale is used to enhance the nominal scale with information about ordering the classes or categories. This scale uses the statistics to calculate the median and non-parametric statistics. Interval scale carries information about the size of the intervals between ordered classes, apparently, this scale has no application in requirements engineering. The ratio scale is richer than other 3 scales, as it possesses ordering, size of intervals, and ratios between entities. Both the parametric and non-parametric statistics can be used, the mean can be calculated. [49]

**Prioritization Techniques**

A number of different prioritization techniques exist in the literature. Aspects are used to prioritize requirements using these techniques. Few prioritization techniques are:

- **Analytical Hierarchy Process (AHP)-** It is introduced by Thomas Saaty (1980), which is an effective tool for dealing with complex decision making and to set priorities. Complex decisions are reduced to a series of pairwise comparisons and then synthesizing the results, AHP helps to capture both subjective and objective aspects of a decision. In addition, AHP incorporates a useful technique for checking the consistency of the decision maker’s evaluations, thus reducing the bias in the decision-making process [20].

- **Numerical assignment [52]-** It is a fundamental technique for prioritizing requirements where several groups of requirements prioritizations are made and then based on their priority the groups are assigned. Difference between groups might be different, while few may be similar. Groups that are common are low, medium, and high. While the requirements are prioritized, the groups are assigned and priority of requirements inside that group will be similar. This technique results in priorities in ordinal scale [49].

- **Bubble sort [52]-** A technique used for sorting elements where two requirements are compared with each other. In case if they are not in sequence, then the requirements can swap with other requirements and compare again. This process is continued until the list is prioritized from higher to lower requirements [52]. The priorities obtained are in ordinal scale [49].

- **$100 method or cumulative voting-** In this technique, the stakeholders are given 100 points which can be hours, money etc. stakeholders dispose the points to the requirements on the basis of its composition [52]. Here results are obtained in ratio scale [49].

- **Ranking -**The requirements are ranged from 1 to n where n is an integer value. High priority is ranked by 1 whereas least priority is ranked by n [52]. The results are obtained in ordinal scale [49].

- **Moscow technique [52]-** Type of numerical assignment which contains 4 priority groups such as: MUST have, SHOULD have, COULD have, Won't have. To prioritize according to priority:
  - MUST have means that requirements in this group must be implemented in the software before they go to release.
  - SHOULD have means that if requirements from this group are implemented then it will be good for the product/software.
COULD have means that if requirement from this group exists then it will be good for the project/software [52].

2.3.1 Importance of Requirements Prioritization in Software Start-ups

Requirements prioritization is an important challenge for start-ups because researchers have identified that start-ups are not smaller versions of large companies[39]. Therefore, the existing prioritization practices cannot be applied directly. The less available literature on requirements prioritization in the context of start-ups adds to the problem area. Software requirements in start-ups are usually not clear and as a result, they struggle to understand them [27]. In addition, the prioritization of requirements has many challenges with either customer-specific requirements or market-driven requirements i.e., issues of balancing the requirements and communication gap between the stakeholders and developers etc. [50]. Without proper communication between the development team and the stakeholders, the product may be less quality, or it may lead to failure of the project. As Start-ups struggle with scarce resources and time pressure, constant prioritization of requirements is important [53]. As requirements prioritization is dependent on stakeholders and start-ups are typically resource constrained, the intersection of software start-ups and requirements prioritization is worth studying.

The Lean start-up method [29] has gained attention among start-ups. The Lean start-up also encourages to experiment a product idea’s potential with Minimum Viable Products (MVP) that are built with the smallest amount of implementation required to validate a product idea [29]. While developing a minimal viable product, developers work heroically but the products don’t meet the customer's requirement. It is unreliable and fails, thus taking longer time in rectifying and often creates further defects which are hard to overcome. Thus, gathering the requirements and prioritizing them at early stages may be useful to the companies [4]. While developing an MVP if start-ups fail then they are dead, whereas big companies can try again. For small organizations, proper requirements prioritization and selection can mean the difference in not only project success or failure but also overall company survivability [22]. A single requirements catastrophe will drive a small company out of business [23] [4].

Minimum viable product (MVP) is the main focus of both business and product development activities in software start-ups [21]. In start-up companies, the biggest challenge is deciding what features or requirements will ship in what order. Developing the MVP will start the prioritization process. An MVP (Minimum Viable Product) is a strategy followed by the start-ups to get the product into the customer's hands as quickly as possible. Start-ups tend to collect a smallest possible group of features that will work as a standalone product to generate maximum customer learning in minimum possible time. In lean and agile start-ups requirements prioritization is a complex activity necessary for generating the product backlog where all the ideas are put forward. After documenting these ideas in a comparable format, they are prioritized following a criterion [29]. In addition, prioritizing business model elements is very important in start-ups at the start of customer validation.

Furthermore, many software start-up companies commonly attribute their failures to their inefficiency in the execution of sales, marketing and delivery [5]. But they fail to recognize that failures and critical issues in product development can lead to loss of customers, terminating return of investments for their founders and investors, and ultimately company failure. Requirements prioritization is one such important but difficult activity in product development [5]. The major difference between an established software company and a start-up is that a start-up company has three phases of product development, beginning from its evolution and before it reaches maturity. These phases are a start-up, stabilization, and growth [5].
The start-up phase can be identified as a period between the product conception and first sale [5]. Though the start-up companies know their market opportunity and have a clear vision, they end up assembling inexperienced developers around them. These developers neglect non-coding issues like requirements elicitation, prioritization, architecture, design etc. Thus, the product becomes unreliable and does not meet the customer expectations [5]. In addition, the requirements become unmanageable during the stabilization phase. Number of requirements or features are requested than what a certain release of a product can deliver and there is no satisfactory way of deciding between them [5]. Moreover, the stakeholders are frustrated by the difficulty of contributing new features and tracking them through the development lifecycle. Thus, there is a need for a business process to capture new requirements, prioritize them and then assess their feasibility and value [5]. These arguments that depict a difference between start-up companies and established companies in terms of product development are the motivation to convince that requirements prioritization actually is a relevant problem for start-ups.

Secondly, the inability to transfer existing requirements prioritization practices to start-ups. Initially, we performed a literature review to understand the state-of-art of requirements prioritization in start-ups. Surprisingly, we found a small number of papers that address the requirements prioritization knowledge in start-ups. Even though many requirements prioritization practices [17] are covered, we identified gaps in practices supporting successful transition through the start-up life cycle, particularly in market-driven requirements engineering, requirements engineering scope definition, alignment between technical decisions and business goals, software architecture etc. Moreover, studies like Eriks et.al [2] show that the ability to transfer results from one environment to another in an applied field of software engineering like requirements prioritization is critical [17]. As a result, we can conclude that the existing studies in requirements prioritization cannot be transferred to start-ups due to an inadequate level of reporting rigor [1]. The main reason for this is as study design details are missing the level of trust in how the study was performed is hard to judge. Therefore, to complete the picture the first step is to know what practices are actually used in the start-up companies [1].

A software product can go wrong if the right requirements are not prioritized at right time. The authors of papers [54] and [55] report that the existing requirements prioritization techniques applicable to established companies are too cumbersome and depend mainly on business value and cost. As the established companies differ from start-up companies in terms of the size of software, number of features, the time gap between releases, market value, volatility and other human behavioral factors, the existing prioritization factors may not be transferred directly to the start-up companies. Moreover, as start-up companies have high risk coexisting with high growth [29], prioritizing practices that focus more on risk and other factors are needed. Therefore, keeping the limited availability of resources in mind, the researchers need to study more about prioritization practices in software start-ups.

Software start-ups use light-weighted techniques than others in an informal way. In start-ups, requirements are mostly market-driven and are hardly documented. So, they use light-weighted practices than other well-known practices. By using these practices, they tend to support and trace knowledge, by opting open-source solutions that require little or no training and maintenance [1].
2.4 Related Work

This section briefly describes a selection of empirical studies based on requirements prioritization in general and requirements engineering practices in software start-ups as the researchers have not found any study focusing specifically on requirements prioritization in the context of software start-ups. The researchers also compare their work with closely related efforts of other people’s work in the area.

Svensson et al. [24] identify how quality requirements are prioritized in 11 successful companies developing software-intensive systems. The authors also discuss the importance of requirements prioritization in software product development and the need to find the right balance among competing for quality requirements. The results show that quality requirements are given less importance when compared to functional requirements. This study focuses only on quality requirements whereas our focus is on both functional and non-functional requirements. Moreover, the studies also differ in terms of their context where the prioritization take place.

Lehtola et al. [25] describe the current state of requirements prioritization in large and medium-sized companies and present the practical challenges involved. The authors identify prioritization as an ambiguous concept and the need to prioritize the customer preferences to make complex context-specific decisions. The results show that only a few current practices are systematic and face challenges in paying attention to all the relevant factors that influence priority. The study has similarities in results when compared to our research but does not focus on start-up companies.

Herrmann et al. [56] conduct a systematic literature review to investigate how the existing methods approach the problem of requirements prioritization based on benefit and cost. The results of this study identify few under-researched issues sketching an agenda for future research in this area. Martinez et al. [57] explore the relationships among prioritizing requirements, understanding requirements and individual cognitive profiles. The results of the conducted case study show similarities and differences when visual and non-visual people use visual or non-visual software requirements specifications. Kauppinen et al. [47] conduct a case study where two requirements prioritization methods are evaluated in the industrial product development projects. The study involved two cases where pair-wise comparisons technique was evaluated for prioritizing user needs and Weigher's method for changing requests. The results show that the requirements prioritization for market-driven product development is challenging. Karlsson et al. conducted a self-experiment to compare six prioritization techniques [58]. About 13 QR were prioritized by all the three authors using each technique. Karlsson et al. concluded that the AHP was the most promising technique due to providing the most trustworthy results, and it includes a consistency check [58]. However, the main problem with AHP was identified as scalability. Bebensee et al. [59] identify Binary Priority List (BPL) as an important requirements prioritization method that requires more research in detail. The author discusses how BPL can be used and assesses its process quality by comparing it to another prioritization techniques. The results show that BPL method is suitable for medium-scale companies.

Paternoster et al. [60] conduct a systematic mapping study to structure and analyze the literature on software development in start-up companies and in one of the sections they focus on requirements engineering. They identify that the requirements in start-ups are market-driven rather than being customer specific. The authors acknowledge the importance of involving the customer/user in the process of eliciting and prioritizing requirements according to their primary needs. Melegati et al. [18] discuss how requirements engineering processes are executed in software start-ups by conducting semi-structured interviews with nine different start-up companies. The author Jan-Martens his article [27] conducts a literature review on software engineering in start-ups and is showing a lack of empirical research and the need for proper guidance on the handling of requirements in start-ups. The author uses an evaluation-based research approach consisting of a survey and multiple-case
study. The outcomes of the study help in understanding requirements engineering in start-ups and propose a guideline for start-ups to gather requirements efficiently.

The focus of the above-mentioned studies was either on the evaluation and comparison of different prioritization techniques or on requirements engineering practices in start-ups but not on how start-ups prioritized requirements in the industry. Except Paternoster [60], none of the other papers have similarity with the problem area focused by researchers. This paper presents a study with a primary focus on how software requirements are prioritized in software start-up companies. Even though paternoster [60] identified the factors of prioritization for start-ups, a more detailed focus on prioritization practices and challenges was missing.
3 Research Methodology

3.1 Research Aim and Objectives

Aim: The overall aim of this research is to investigate state-of-art of requirements prioritization practices in software start-ups.

Objectives: The following 4 objectives were set to achieve the aim:

- To identify the requirements prioritization practices currently implemented in start-up companies.
- To identify the challenges faced by practitioners in implementing the corresponding requirements prioritization practices in start-ups.
- To identify the possible solutions that could mitigate these prioritization challenges.
- To identify the pros and cons of the identified solutions.

3.2 Research Questions

To achieve our goal and to drive the study we formulate the following research questions:

RQ1: How is requirements prioritization currently practiced in start-up companies?

Motivation: The motivation behind selecting this research question is to know the current requirements prioritization practices being used by the software practitioners in start-up companies. The answers to this question provide a base to know what challenges exist based on the corresponding practice i.e. RQ2

RQ2: What are the various challenges faced by the practitioners while implementing these practices in start-ups?

Motivation: The motivation behind selecting this research question is to identify various challenges which are currently being faced by the software practitioners in prioritizing requirements in start-up companies. These answers provide a base for answering RQ3.

RQ3: What are the possible solutions to mitigate these challenges?

Motivation: The motivation behind selecting the questions 3 and 3.1 is to identify few possible solutions for the corresponding challenges/problems faced by different software practitioners. Thus, the solutions identified for the challenges in RQ3 allow the practitioners to approach them in a better way.

RQ3.1: What are the pros and cons of the proposed solutions?

Motivation: This RQ addresses the pros and cons of the solutions identified in RQ 3. These pros and cons might be helpful for the practitioners in software startups to select more efficient requirements prioritization practices.
3.3 Research Methods

A qualitative research approach was carried out for conducting the empirical study. This approach is useful when the purpose is to explore an area of interest, and when the aim is to improve the understanding of phenomena. In addition, qualitative research is directed primarily at collecting and analyzing data. The aim is to achieve information depth rather than breadth [61] in an inductive way [62]. Since the nature of our study is explorative and the main purpose of our study is to gain an in-depth understanding of requirements prioritization in software start-ups, a qualitative approach is considered suitable.

The selection of research method to be used is very important. Several elements like research question nature, researcher experience, study scope etc. should be considered. Wohlin et.al [63] mention many research methods like action research, case study, experimentation, ethnography and grounded theory. The applicability of each of these methods alongside other quantitative methods is examined carefully.

Finally, we selected literature review and interview study as our research methods. While literature review was selected to gain a basic understanding of different requirements prioritization techniques, the interview study method is used to answer our research questions. The Interviews were conducted with 8 different software start-ups. A much-detailed explanation regarding each of these methods is provided in the following subsequent sections.

3.3.1 Literature Review

A Literature review was conducted to locate and identify relevant data, documents and other sources on our topic [64]. The main reason for performing literature review as a first step is to identify relevant scientific data related to the requirement prioritization techniques practiced and the challenges faced while implementing them. Though literature review and systematic literature review (SLR) follow the same principles, we have chosen literature review over SLR due to time and resource constraints. The literature review provided a clear understanding of the techniques in use and the complete terminologies related to
prioritization factors. The data explored can be used to formulate the closed and open-ended questions for the interview. It is also used in the confirmation studies such as data triangulation to validate the results of the interview where both the perspectives of the researcher and practitioner can be compared to achieve quality results. The following steps give a detailed outline and design of the literature review:

Data sources-A well-formulated Keyword driven search approach was used to identify references. We use Google Scholar, Scopus, and IEEE Xplore as databases for identifying relevant papers [63]. These databases contain data in the form of journal articles, conference papers, books and scholarly articles which are said to form the basis for literature review [64]. We selected Scopus as a database to aid the search strategy as it is a valid source for finding information about present and ongoing research in computer science and related areas. Moreover, this database is updated and facilitates filters to narrow the search.

Search strategy-A document search was initiated in the Scopus database using the field “Article title, Abstract, Keywords”. The search string was refined with AND &OR operators along with synonyms and terms related to keywords [64]. Table-1 represents the initial (S1, S2) and final (S3) set of search strings. The search is further refined after applying all inclusion and exclusion criteria. The final search string formulated depending on the research questions is (TITLE-ABS-KEY (requirements AND prioritization ) AND TITLE-ABS-KEY ( technique ) OR TITLE-ABS-KEY ( method ) OR TITLE-ABS-KEY ( approaches ) OR TITLE-ABS-KEY ( process ) OR TITLE-ABS-KEY ( procedure ) AND TITLE-ABS-KEY ( software AND systems ) AND TITLE-ABS-KEY ( requirements AND engineering ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) ) AND ( LIMIT-TO ( LANGUAGE , "English" ) ) AND ( LIMIT-TO ( SRCTYPE , "p" ) OR LIMIT-TO ( SRCTYPE , "j" )).

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Search String</th>
<th>Articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>TITLE-ABS-KEY (requirements AND prioritization)</td>
<td>2,027</td>
</tr>
<tr>
<td>S2</td>
<td>(TITLE-ABS-KEY (requirements AND prioritization) AND TITLE-ABS-KEY (software systems))</td>
<td>339</td>
</tr>
<tr>
<td>S3</td>
<td>(TITLE-ABS-KEY (software systems) AND TITLE-ABS-KEY (requirements AND prioritization) AND TITLE-ABS-KEY (requirements AND engineering) AND TITLE-ABS-KEY (techniques OR methods OR process OR approach OR procedure))</td>
<td>186</td>
</tr>
</tbody>
</table>

Study selection- A total of 186 articles were identified from the final search string and after applying the following inclusion criteria 132 article were retrieved.

Inclusion Criteria
- Literature Databases: Scopus, Google Scholar, IEEE Xplore.
- Year: 1996-2017
- Subject Area: Computer Science
- Document type: Conference Paper, Article
- Search space: Title, Abstract, Keyword
- Language: English
- Quality of publication: Peer-review

Data extraction and synthesis- The data from literature review was analyzed following the guidelines of the article [65] [66]. The author used concept-centric approach and author-centric approach to generate the results of the literature review. While the author-centric approach is a summary of relevant articles, concept-centric approach determines to organize framework of the review from an author-centric approach [65] [66]. Initially, articles were

18
studied, and a table was created that followed author-centric approach. Appendix B consists of the articles and the respective techniques found in them. But to analyze the data, a concept-centric approach was needed. This was obtained by converting the author-centric approach into a concept-centric matrix as listed in the appendix B. These approaches were used to understand different concepts of techniques and how much frequency of occurrence for each practice.

3.3.2 A Qualitative Interview Study

An interview is a commonly used qualitative research method [67]. The main motivation behind selecting interview as the research method is to investigate the requirements prioritization practices in software startups. Interviews are conducted with a wide range of objectives ranging from collecting historical data from the memories of the interviewees, to collect opinions or impressions about something, to identify a terminology used in a particular setting etc. The main motivation behind selecting interviews to collect qualitative data is that they provide a better and in-depth knowledge about a particular issue or study from multiple perspectives. In addition, interview studies typically achieve higher response rates than surveys [63].

Since the topic of our research is requirements prioritization in software start-ups we consider developers, leaders/founders, requirements engineer/analyst, CTO etc. as our target population. This is because in start-ups the prioritization takes place at different levels depending on the type of organization [18]. The nature of our interviews is exploratory study. The explorative study is generally used as a pre-study to a more thorough investigation to assure that important issues are not foreseen. In other words, they not only answer the basic research questions but also provide new possibilities that could be analyzed [65].

3.4 Data Collection

We used semi-structured interviews as the source of data collection where the interviewers handle the questionnaire instead of the respondents themselves. These interviews were conducted for this study through either telephone or skype. They provide more flexibility to record and gather unexpected information when compared with structured interviews and is less expensive to be used extensively when compared to purely unstructured interview [67]. A single round of semi-structured interviews was conducted with all the interviewees using a questionnaire consisting of both open-ended and closed-ended questions. While the former was used to understand the requirements prioritization techniques in the practitioner’s perspective, the latter aims at analyzing the organization’s views on techniques, challenges faced, solutions used and benefits from these practices. The following steps provide details of the interview design:

3.4.1 Interview Guide

An interview guide was prepared initially by the researcher with the help of some notes written on it to steer the interview.

Preparation- A single round of interview was planned to conduct for the selected set of interviewees. Prior to this, the participants are contacted through social media and an invitation letter is sent to each interviewee through e-mail that not only addresses the research motive but also requesting them to provide details regarding the time, date and medium to conduct the interview. Further as per the convenience and availability of both researchers and interviewee, the interviews were scheduled. The interviews were planned to be conducted in a time frame of 30-45 minutes.
Purpose of research- A brief introduction was given by the researchers about their personal details (name, education, Institute) followed by the purpose and objectives of the interview. Later the interviewee is explained about the problem domain, important terminologies, and techniques. This is to ensure that the practitioner shows interest in answering covering unexpected information. They are also ensured that the details provided by them will strictly be used for education purpose and be kept confidential.

Questionnaire- The interview starts with general questions that provide basic information about the participant like name and age of the organization, job position, experience, type of software development approach etc. Later a discrete set of project specific questions are posed that answer the research questions RQ1, RQ2, and RQ4. The former consists of close-ended questions while the latter consists of a mixture. Both open-ended and close-ended questions are formulated with the help of literature review identified for this research area. Furthermore, before preparing the final questionnaire, a draft was submitted to the supervisor. After a couple of evaluations and review, a final draft of the questionnaire was obtained and used for the interview. A snapshot of the interview guide is presented in the Appendix.

Execution- A total of 8 interviews were conducted as planned above by the researchers themselves. Out of these 8 interviews, 6 of them were conducted through a phone call and 2 interviews through Skype. Face-to-face interviews were not possible as most of the interviewees were in India. Both the telephonic and skype interview were audio recorded, keeping the internet connectivity issues in mind. The first interview was conducted in the presence of the supervisor to gain some experience to compete for the remaining interviews. While one researcher was responsible to pose questionnaire and manage the conversation, the other researcher was taking notes of important keywords and topics that were relevant to answer the research questions. The further process of converting the raw data of interview is explained in the next sections.

Recording- Recording data is an important step for conducting interviews as it helps the data to be secured and be used multiple times in future to write the field notes. If the interviewee consents to audiotape the interview, the entire interview data collected through telephone or skype calls is audio recorded. The data was recorded, with the name of the interviewee, in multiple databases and cloud (google drive) to avoid loss of data during a system crash. Besides, the recording also gives an opportunity to the researcher to assess his/her interviewing skills [67].

Transcription- Transcription is another important step that needs to be done prior to analysis of the interview data. It is very difficult to analyze the raw data and draw qualitative results from it. Therefore, we manually analyzed this data by going through multiple times and writing field notes. Though several transcription tools are available, manual transcription of the data was preferred as it provides an opportunity to learn what exactly the interviewee is addressing.
This text data generated from the audio data is written as field notes in Microsoft Word processor and saved in the database with the name of the interviewee [68].

3.4.2 Sampling of the data

The sampling of data is necessary for understanding the population from which they are drawn [69]. They aim at the development of generalized conclusions. Sampling methods are divided into two categories: probabilistic and non-probabilistic sampling. Convenience sampling method is used to select the interviewees for the study which is a non-probabilistic method [70]. In the non-probabilistic method, the probability of selection of each member
for the interview is uncertain or random. Since our research is an explorative study and the research object being software start-ups which is too scattered, we select this method. On the other hand, convenience sampling method is selected because of its simplicity and ability to save money and time for researchers. Other non-probabilistic methods were not selected due to insufficient knowledge in software start-ups and being unfamiliar with them. Therefore 8 interviewees were selected who have experience in requirements engineering and prioritization in software start-ups. These interviewees are from 8 different start-up companies in India. The main reason for selecting Indian-based start-ups due to the personal contacts of the researchers within the software start-up industry.

3.5 Data analysis

Thematic analysis, a qualitative data analysis method used for most of the empirical software engineering research, is applied to analyze the interview results. Based on the frequency and relevance of the collected data, data is identified as themes and codes. Braun and Clarke define it as- "Thematic analysis minimally organizes and describes the dataset in rich detail and frequently interprets various aspects of the research topic [68] ". A.Lacey et.al [71] model of thematic analysis is referred to analyze the qualitative data collected from interviews. In other words, the interview text data is studied, and the data is coded and categorized into themes [72].

The motivation behind selecting thematic coding is that it provides a sharp analysis that answers particular research questions. In addition, this method accompanies the research questions by providing an investigation of interview data from two perspectives: one coding based perspective providing sufficient information; second from the research question perspective strengthening the consistency between them. In addition, quantitative analysis method called “frequency counting” was used to analyze the interview data regarding
prioritization practices. This method is used as it provides a number how often a certain term has been observed in the data. With the number, we can show that one term is more or less relevant than another [73].

A detailed description of the interview data analysis is explained in the further sections [72]. Thematic analysis has the following steps of analysis:

**Acquiring the data**- With some prior knowledge of data, preparation of notes which help in making ideas for coding is done. The data collected from interviews can be analyzed and interpreted to draw valid observations from it, and thus answer our research questions. All the interviews that were audiotaped are transcribed on the same day using Express Scribe software. A free version of express scribe software is used for transcription as this it improves the efficiency of transcription using the speed control option. A word processor is used to collect and export the transcribed data and the file is saved in multiple databases in the name of the interviewee. Later all the text files are studied to understand and recollect the viewpoint of the interviewee.

**Generating the initial codes**- After transcribing and organizing the data, the text data is coded manually by examining every line of answers of the interviewees.

**Reviewing the themes**- Codes are identified based on the frequency of the words and their relevancy. Furthermore, these codes are highlighted using colors.

**Defining and naming the themes**- The codes identified are clustered into themes depending on their relevancy with research objectives. Furthermore, the efficiency of data analysis is ensured by using the same color to highlight the texts under the same theme. This grouping of codes into themes also depends on the frequency of its occurrence. A snapshot of color coding the transcribed texts are shown for Interviewee2 in the appendix.

**Generating the report**- The generated codes are written into themes and then a narrative description of data is written that makes an argument related to the research questions.

For example: What factors do you consider while prioritizing requirements?

“……we identify a distinct set of business values like sales, customer satisfaction, marketing, and rate them on a scale of 1-10, ummm…. then we identify all the requirements to be prioritized and scale them on 1-10 and based on agreement from various stakeholders …. and finally, we use a prioritization matrix with business values and risk…… cost is also considered as a key factor”. The interviewee compares their previous practice saying, “I must say that this approach of a ranked list is a bit easy to use than the early requirement review meeting.”

Codes in yellow color- Prioritization practices (Themes)

Codes in green color- Factors (Themes)

3.5.1 Validation

Triangulation is important to increase the accuracy of the empirical research. To validate the qualitative analysis, we used the following 3 methods to validate data [69].

- **Method triangulation**: Checking the consistency of the results generated from various data collection methods said to be method triangulation. Here, the collected data from the literature is used as an evidence to validate the results obtained from the study.
- **Researcher triangulation**: Involving all the researchers related to this in the analysis process. Here two researcher’s perspectives are used to validate the results.

- **Data Triangulation**: involving various job roles data is used to validate the data. Here roles like project manager, requirements analyst, developer etc data are used.

3.6 Validity Threats

Validity threats to any study denote the trustworthiness of the results and the extent to which the results are free from researcher bias [69]. Validity threats must be addressed in all the phases of interview study to ensure the quality of the study results. There are four types of threats validity of our study namely: Construct Validity, Internal Validity, External Validity and Conclusion Validity.

3.6.1 Internal Validity

This threat is concerned with the factors that influence the relations (input-to-output process) in the study [69]. The possible internal validity threats to our study are:

Interview Questionnaire: One possible threat to internal validity is the misinterpretation of the interview questions by the participants. To address this threat, the questionnaire was prepared based on the guidelines stated in [74]. Additionally, to ensure the understandability of the questions, the questionnaire was reviewed by the Supervisor and the modifications are done to enhance the quality of the questionnaire.

Interview Respondents: The results of the interview are highly dependent on the participants selected as inexperienced people may lead to incorrect interview results. To mitigate this threat, we have selected highly experienced and knowledgeable people in the study area as the interview participants. The purpose of the interview along with the example of expected outcomes is explained to the respondents. The sample size is feasible to generalize the results as it helps the researcher to maintain a close relationship with the interviewees thus, helps in obtaining open and above-board information. This can help in mitigating some of the tendency and validity threats that inherent in qualitative research.

3.6.2 External Validity

This threat is concerned with the extent to which the study results are generalizable and are of interest to people outside the study sample [69]. One of the main validity threat to our research could be that we have only considered software startups from India. To ensure the threat to generalizability is addressed, we have documented the study context, research method, data collection and data analysis procedures in a detailed format stating examples of the interpretation. This guarantees the applicability of the study results to other similar contexts. Furthermore, we have conducted 8 interviews as a part of our study and collected evidence from a sufficient sample of various management roles which helped us to gain in-depth knowledge in the research area. Thus, ensuring the results cover a broader perspective and can be applied to relevant levels of an organization.

3.6.3 Construct Validity

Construct Validity deals to the extent are the identified measures really matching the researcher's mind and satisfy the research questions formulated [69]. One of the possible
threats to construct validity in our study is the misinterpretation of the interview questions by the participants which may lead to incorrect results to the formulated research questions. To mitigate this threat, we have performed data analysis simultaneously to the data collection. Further, we have mailed the participants with backup questions in case of any query. The analyzed results were constantly sent for feedback to the supervisor to ensure the credibility of the results and the stated modifications helped in improvising the future interviews.

3.6.4 Conclusion Validity

This validity is concerned with the researcher’s ability to draw conclusions from the results [69]. To ensure that the interpreted results are precise and correct, the data analyzed by the first researcher is validated by the second researcher and vice-versa. Further, this analysis is reviewed by the supervisor. In addition, because the respondents of the interview vary with their level of experience, education and organization scope their understanding of the interview questions may not be same. To mitigate this threat the respondents were given a clear understanding of the context and motive prior to the interview. Moreover, open-ended questions used for the interview further mitigate the threat.
4 RESULTS

4.1 Literature Review

This section presents and discusses the findings of the literature review. We begin by presenting an overview of the selected articles and then discuss the findings of this review in detail.

A total of 132 papers were identified from the search string and applying the inclusion criteria. 87 papers remained after applying the following exclusion criteria. This includes 3 papers that are not published in English language, 21 papers that do not have any link with the research questions and discuss other aspects. After reading full abstracts, around 12 papers were excluded that do not discuss any specific technique. Only the most recent, complete and improved versions were included and the rest of the 11 duplicates are excluded. Figure-5 and Figure-6 represent the number of papers published in a year and percentages of selected study respectively.

Since there is no paper that specifically focus on requirements prioritization in the context of startups, these papers were studied separately. 25 papers were identified from the literature that focused on startups. Around 10 papers that focused on software development and software engineering were selected.

![Figure-5 Paper publications distribution per year](image)

![Figure-6 Percentages of Documents type](image)
Table-2: Frequency of occurrence for practices identified in LR.

<table>
<thead>
<tr>
<th>S.no</th>
<th>Practices</th>
<th>Frequency of occurrence in LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Analytical Hierarchy Process(AHP)</td>
<td>40</td>
</tr>
<tr>
<td>2.</td>
<td>Quality Functional Deployment(QFD)</td>
<td>15</td>
</tr>
<tr>
<td>3.</td>
<td>Planning Game</td>
<td>19</td>
</tr>
<tr>
<td>4.</td>
<td>Binary Search Tree(BST)</td>
<td>13</td>
</tr>
<tr>
<td>5.</td>
<td>Cumulative Voting ($100 test)</td>
<td>16</td>
</tr>
<tr>
<td>6.</td>
<td>Cost-Value Approach</td>
<td>13</td>
</tr>
<tr>
<td>7.</td>
<td>Top Ten Requirements</td>
<td>5</td>
</tr>
<tr>
<td>8.</td>
<td>Win Win</td>
<td>8</td>
</tr>
<tr>
<td>9.</td>
<td>Pair-wise Comparisons</td>
<td>9</td>
</tr>
<tr>
<td>10.</td>
<td>Priority Groups</td>
<td>7</td>
</tr>
<tr>
<td>11.</td>
<td>Numerical Assignment</td>
<td>12</td>
</tr>
<tr>
<td>12.</td>
<td>Ranking</td>
<td>5</td>
</tr>
<tr>
<td>13.</td>
<td>Value-oriented prioritization</td>
<td>1</td>
</tr>
<tr>
<td>14.</td>
<td>Software Engineering Risk Understanding and Management</td>
<td>1</td>
</tr>
<tr>
<td>15.</td>
<td>Minimal Spanning Tree</td>
<td>8</td>
</tr>
<tr>
<td>16.</td>
<td>Wiegers’ Matrix approach</td>
<td>13</td>
</tr>
<tr>
<td>17.</td>
<td>Value-based Prioritization</td>
<td>3</td>
</tr>
</tbody>
</table>

Around 17 prioritization techniques were retrieved from the selected 87 studies. These practices were repeated several times in these papers. In addition, the literature identifies many methods as frameworks for basic and important techniques. AHP (Analytical Hierarchical Process) was identified as the most promising [58] and highly investigated technique with a frequency of 40. However, the numerical assignment was identified as the most widely used technique in industry [20]. The results of LR are tabulated in Appendix A which provide much detail about the prioritization practices and their corresponding challenges.

4.2 Interview Study

4.2.1 Demographical Results

Demographic information regarding the subjects involved in the study is important to report as their background could probably affect the results in a different way. In this study, an interview guide was used by the researcher had a set of both general and project-specific questions. These general questions gathered information about the participant job role, experience in working with requirements prioritization and the size of the organization. Further, these questions can be categorized into open and close-ended questions. The following demographic results are presented from the general questions that help the researcher to verify the suitability of the interviewee for the interview.

Table-3 Demographics of interviewees

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Job Role</th>
<th>Experience</th>
<th>Application Domain</th>
<th>Number of Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviewee 1</td>
<td>Software Developer</td>
<td>8 years</td>
<td>Tool development and management software</td>
<td>20</td>
</tr>
<tr>
<td>Interviewee 2</td>
<td>CEO</td>
<td>6 years</td>
<td>Healthcare</td>
<td>50</td>
</tr>
<tr>
<td>Interviewee 3</td>
<td>Project Manager</td>
<td>10 years</td>
<td>Mobile Applications</td>
<td>15</td>
</tr>
<tr>
<td>Interviewee</td>
<td>Business Analyst</td>
<td>3 years</td>
<td>Mobile applications</td>
<td>40</td>
</tr>
</tbody>
</table>
Since previous studies show that subjects with different roles/perspectives prioritize requirements differently, it is important to consider interviewees with different job role. The interviewees in this study belonged to 6 different roles. All the interviewees involved in the prioritization process directly. For example, the software developer participated in the weekly meetings for prioritizing requirements whereas the CEO or CTO involved in selecting an approach for their organization. Out of the 8 interviewees, 7 of them had clear knowledge about how requirements are prioritized at their organization than the 8th interviewee. Further 5 of the interviewees had previous experience in dealing with requirements at large or medium scale companies. One of the interviewees who was discussing his previous experience claims that “there is a lot of difference between how we used to prioritize the client requirements earlier and how we are dealing with it now...At x there was a separate team, or an analyst placed or was responsible for it.... but now it’s either the team leader or project manager that prioritizes the user needs after one or two meetings with the clients...”

Table-3 shows the different stakeholders identified for the interviews. A quite important observation from the demographic results shows that most of the start-up companies do not employ a separate requirement engineer or analyst to gather and prioritize requirements. But the presence of different stakeholders ensured the researcher’s results to be unbiased.

It is important to consider different organizational characteristics like the size of the organization, application domain, and type of market etc. to understand the practical context and constraints under which requirements prioritization is performed. Table-4 presents the results of the type of market situation that gives information about how the organization’s market looks like i.e. whether it is market-driven or bespoke. Market-driven software development refers to the situation where the development costs of a generic product are divided among many buyers on an open market and where the potential profit is rewarded to the producer (e.g. someone orders you to make a website for him). It is different from customer-specific development (also called as bespoke), where one single customer pays all the development costs and the resulting product is specific to the needs and wishes of that one customer (e.g. Apple is making the new iPhone and must consider needs of millions of customers). [114].This information is collected from a closed question categorizing the products into two groups namely market-driven and bespoke.

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Market Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviewee 1</td>
<td>bespoke</td>
</tr>
<tr>
<td>Interviewee 2</td>
<td>market-driven</td>
</tr>
<tr>
<td>Interviewee 3</td>
<td>bespoke</td>
</tr>
<tr>
<td>Interviewee 4</td>
<td>market-driven</td>
</tr>
<tr>
<td>Interviewee 5</td>
<td>market-driven</td>
</tr>
<tr>
<td>Interviewee 6</td>
<td>bespoke</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>Interviewee 7</td>
<td>bespoke</td>
</tr>
<tr>
<td>Interviewee 8</td>
<td>bespoke</td>
</tr>
</tbody>
</table>

The techniques employed by the practitioners to prioritize requirements for developing and maintaining each of these products varies largely. This is further illustrated in the next section where the techniques practiced at start-ups are elaborated.

### 4.2.2 Interview Data

This section describes the results discovered during the analysis of the interviews. These results are drawn from the responses provided by the 8 interviewees. In addition, the interviewee data is elaborated using the codes that are identified from the transcribed data. Furthermore, these codes are categorized into themes for analysis.

#### A. Interviewee 1

The interviewee 1 describes a framework for prioritization that combines both core business values and risks. “……we identify a distinct set of business values like sales, customer satisfaction, marketing, and rate them on a scale of 1-10, ummm…. then we identify all the requirements to be prioritized and scale them on 1-10 and based on agreement from various stakeholders …. and finally, we use a prioritization matrix with business values and risks……. cost is also considered as a key factor”. The interviewee compares their previous practice saying, “I must say that this approach of a ranked list is a bit easy to use than the early requirement review meeting”

**Practices:** “value-oriented”, “weighing risks”, “business values”, “ranking”, “prioritization matrix”, “requirement review meetings”.

**Challenges:** “dependency between requirements is not considered”, “scalability problem if large number of requirements are considered”

**Solutions:** “consider risk, cost and rank for individual requirements”, “Document analysis”

**Pros and Cons:** “less disagreements on product requirements”, “stakeholders participate well”, “understandability of requirements”

#### B. Interviewee 2

The interviewee 2 believes that initially, it is important to collect all the requirements instead of concentrating on what requirements to prioritize. For example, “it is important to do requirements prioritization to get a data set which is a representative of the target market as possible. It depends on how to prioritize requirements if there is a certain technique or the input for this technique the output will be wrong.”

**Practices:** “cost-benefit analysis”, “value-based approach”, “100$ test”

**Challenges:** “complex communication with customers”, “difficulty in assigning values”, “100$ has no dependencies”

**Solutions:** “getting the goals of the customer”, “shifting the goals in between people”, “use simple methods”.

**Pros and Cons:** “customer value understanding”, “sometimes using simple techniques may lead to scalability problems”

#### C. Interviewee 3
The interviewee works as a product manager for a software company that provides healthcare management incentives. The interviewee describes “the PM is the key person who could drive some properties and weightages across the requirements”. The interviewee also shares his experience working with an automated tool JIRA for prioritizing.

**Practices:** “business case analysis”, “weighted critical analysis”, “value-based”

**Challenges:** “ensuring that clients understand priorities”, “differences in client’s priorities”, “less resources”, “JIRA limitations”

**Solutions:** “workshops to put requirements in the form of user stories”, “grouping user stories into epics”

**Pros and Cons:** “increase in return of investment”

D. Interviewee 4

**Practices:** “quality function deployment”, “pair-wise comparisons”

**Challenges:** “not scalable”, “does not cater for inconsistencies”, “time taking”, “complicated results”

**Solutions:** “using combination of different practices”

**Pros and Cons:** “reliable results”

E. Interviewee 5

**Practices:** “evaluate the requirements value based on customer expectation and implementation costs”, “Project meetings to discuss and reach a common consensus.”

**Challenges:** “difficulty in deciding the cost due to absence of a valid method in early development phases”, “obligations between the person who implements the requirement and person who identifies its effect”

**Solutions:** “whiteboarding sessions”, “document analysis”.

**Pros and Cons:** “stakeholder’s improvement”

F. Interviewee 6

**Practices:** “priority groups”, “planning game”

**Challenges:** “inconsistencies in the decision making”, “differences in stakeholder views”.

**Solutions:** “using focus groups and brainstorming”, “whiteboarding sessions”

**Pros and Cons:** “modernizing the product with new features”, “not scalable”

G. Interviewee 7

The interviewee describes their practice as “we have an environment where the stakeholders are given freedom to participate in the prioritization…. anyone can come up with their own idea. The stakeholders need to understand the values they are assigning, and risk associated with it…this is done on the basis of how much effort is needed, ROI, probability to be true”
**Practices:** “collect ideas”, “assigning values to each suggested feature”, “prioritize them based on ROI and cost”, “use backlog for important requirements”.

**Challenges:** “inability to organize goals hierarchically”, “poor pre-requisite handling”

**Solutions:** “using sprints”, “user stories”, “group interviews”

**Pros and Cons:** “better report generation”

**H. Interviewee 8**

The interviewee claims that “…to be honest we do not follow any formal process. Most of the times it’s me who is responsible for collecting all the requirements and sort them in such a way that brings some business impact for us. Initially, I prefer to complete the small and important tasks and later using ranks we prioritize other stuff. It is an advantage if the person who works with the requirements is creative besides being skillful”. The interviewee said that they use an agile-like process and are still working in order to find the best suitable practice of prioritizing.

**Practices:** “ranking”, “intuitive priority”, “pushed to backlog/feature request”, “schedule into sprints”

**Challenges:** “natural dependencies exist”

**Solutions:** “having daily stand-ups”, “maintain backlogs”, “operating in sprints”

**Pros and Cons:** “ability to build features”, “satisfy the new and existing customer”

The above interviewee results can be summarized in the following tables.

**Table-5: Prioritization practices in interviews**

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Requirements Prioritization Practices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Value-oriented</td>
</tr>
<tr>
<td>2</td>
<td>Cost-benefit analysis, 100$ test</td>
</tr>
<tr>
<td>3</td>
<td>Business case analysis, Weighted critical analysis, Value-based</td>
</tr>
<tr>
<td>4</td>
<td>Quality functional deployment(QFD), Pair-wise comparisons</td>
</tr>
<tr>
<td>5</td>
<td>Value-based</td>
</tr>
<tr>
<td>6</td>
<td>Priority groups, Planning game</td>
</tr>
<tr>
<td>7</td>
<td>Value-based, Maintaining a Backlog</td>
</tr>
<tr>
<td>8</td>
<td>Ranking, Intuitive priority</td>
</tr>
<tr>
<td>Practice</td>
<td>Factors</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Value-based</td>
<td>Value, risk, cost, time.</td>
</tr>
<tr>
<td>Ranking</td>
<td>Value, risk, cost, time, ROI</td>
</tr>
<tr>
<td>Cost-benefit analysis</td>
<td>Cost, ROI, value</td>
</tr>
<tr>
<td>100$ test</td>
<td>Cost, ROI, value</td>
</tr>
<tr>
<td>Business case analysis</td>
<td>Importance, value, customer-input, ROI</td>
</tr>
<tr>
<td>QFD</td>
<td>Customer input, stakeholder, value</td>
</tr>
<tr>
<td>Pair-wise</td>
<td>Customer input, stakeholder, value</td>
</tr>
<tr>
<td>Planning game</td>
<td>Customer input, business-value, stakeholder</td>
</tr>
<tr>
<td>Maintaining a backlog</td>
<td>Importance</td>
</tr>
<tr>
<td>Intuitive priority</td>
<td>Customer input, cost, value, ROI</td>
</tr>
<tr>
<td>Weighted Critical</td>
<td>Importance, value, customer-input, ROI</td>
</tr>
<tr>
<td>Prioritization matrix</td>
<td>Value, Risk, cost, customer-input, ROI</td>
</tr>
</tbody>
</table>
5 ANALYSIS

The following three sub-sections present and discuss one research question each, corresponding to the research questions in section 3.2.

RQ1) What are the current requirement prioritization practices in software start-ups?

In analyzing Research Question 1 (RQ1), this section examines how requirements are prioritized in software start-ups. As discussed earlier, we opted thematic coding for data analysis. Therefore, we identified the following four themes: prioritization practices; factors/criteria; challenges and solutions. These themes were selected based on our research objectives. The codes identified in the raw data of each interview are mapped to the relevant themes based on our knowledge from the literature review. In the next step, these codes are counted manually and are assigned weights. These weights denote the number of occurrences for each code in each interview. Further, we calculate the sum of these occurrences for analyzing the most frequently used code in the empirical study. The following tables give a detailed overview of the themes, codes identified and the summation of their occurrences. The second column in each of these tables presents the summation of occurrences of the corresponding code in each of the 8 interviews. For e.g.: number of occurrences of each code in (interview 1 + interview 2 +……interview 8).

Table-7 Theme 1: Analysis of Prioritization practices

<table>
<thead>
<tr>
<th>Codes</th>
<th>Σ of occurrences from each interviewee</th>
<th>Total no of occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value-based</td>
<td>1+1+1+0+0+1+0+0</td>
<td>4</td>
</tr>
<tr>
<td>Ranking</td>
<td>1+0+0+0+0+0+0+1</td>
<td>2</td>
</tr>
<tr>
<td>Cost-benefit analysis</td>
<td>0+1+0+0+0+0+0+0</td>
<td>1</td>
</tr>
<tr>
<td>100S test</td>
<td>0+1+0+0+0+0+0+0</td>
<td>1</td>
</tr>
<tr>
<td>Business case analysis</td>
<td>0+0+1+0+0+0+0+0</td>
<td>1</td>
</tr>
<tr>
<td>QFD</td>
<td>0+0+0+1+0+0+0+0</td>
<td>1</td>
</tr>
<tr>
<td>Pair-wise</td>
<td>0+0+0+1+0+0+0+0</td>
<td>1</td>
</tr>
<tr>
<td>Planning game</td>
<td>0+0+0+0+0+1+0+0</td>
<td>1</td>
</tr>
<tr>
<td>Maintaining a backlog</td>
<td>0+0+0+0+0+0+0+1</td>
<td>1</td>
</tr>
<tr>
<td>Intuitive priority</td>
<td>0+0+0+0+0+0+0+1</td>
<td>1</td>
</tr>
<tr>
<td>Weighted Critical</td>
<td>0+0+1+0+0+0+1+0</td>
<td>2</td>
</tr>
<tr>
<td>Prioritization matrix</td>
<td>1+0+0+0+0+0+0+0</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 8 Theme 2: Analysis of prioritization factors

<table>
<thead>
<tr>
<th>codes</th>
<th>Σ of occurrences from each interviewee</th>
<th>Total no of occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>3+2+1+1+0+0+2+4</td>
<td>13</td>
</tr>
<tr>
<td>Risk</td>
<td>2+0+0+0+0+0+1</td>
<td>3</td>
</tr>
<tr>
<td>Cost</td>
<td>1+4+0+1+0+0+0+2</td>
<td>9</td>
</tr>
<tr>
<td>Customer-input</td>
<td>3+2+2+4+1+4+1+3</td>
<td>20</td>
</tr>
<tr>
<td>ROI</td>
<td>2+2+0+0+0+0+2+2</td>
<td>8</td>
</tr>
<tr>
<td>Importance</td>
<td>1+0+1+0+0+0+2+0</td>
<td>4</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>0+0+0+2+0+2+0+0</td>
<td>4</td>
</tr>
</tbody>
</table>

Our results gathered mixed reviews about the prioritization practices in start-ups. After interviewing 8 different interviewees corresponding to 8 different start-up companies, we found that interviewees 1 to 6 followed formal methods of prioritization like QFD, pair-wise comparisons, ranking etc. while interviewees 7 and 8 had informal ways to prioritize. These results are a reflection of the resource and time constraints in start-ups [76]. This difference can be explained by our focus i.e. we focused only on prioritization in start-ups. From table- 7 it can be observed that Value-based prioritization practice was used more frequently among the 8 interviews (4) followed by ranking (2) and weighted critical analysis (2). Thus, value-based requirements prioritization method is dominant for prioritizing software requirements in start-ups. This finding is not in line with P.Berander et al. [20] which stated that numerical assignment is the most common method used. This difference can be interpreted and rooted in difficulties to assign numbers to product value aspects [75].

**Value-based requirements prioritization**: The value-based prioritization technique has been used by three out of the eight practitioners interviewed. All the three interviewees described that this technique involved the project manager and other members of the organization like developers, testers etc. The technique involved worksheets containing a list of requirements that need to be prioritized. While the project managers assessed these requirements based on value, risk, and effort, the other project members had to assess them based on value. However, two differences were observed between these organizations in terms of implementing this technique. Firstly, two categories of ordinal scales based on which the assessments were observed. While two interviewees used high, medium and low importance the other one interviewee used level 1, level 2 and level 3. Though the implementation of this method is in line with S.I Mohamed [75], a significant difference was observed in terms of the criteria for prioritization i.e. time was also used alongside cost while assessing requirements based on effort. This could be due to startups facing limited time to develop. Usage of any complex frameworks of value-based requirements prioritization like TOPSIS or VBIERP [82] were not observed.

**Ranking**: It is the second most used technique among the eight practitioners we interviewed. Interviewee 8 describes ranking as a simple technique where the CEO or customer himself ranks the requirements on an ordinal scale of 1-n based on criteria that differ from project to project. He also explains that “I usually prefer to finish small and important tasks done first. But after that that the way we rank them is more difficult.” However, interviewee 1 used this technique as a sub-technique for implementing value-oriented prioritization method. Since their focus was on market-specific development (table-), they used risks and core business values like sales, marketing etc. as prioritization criteria. The interviewee explained that "sometimes, there are few requirements whose numerical values are tied and it becomes difficult to prioritize them .. ". This could be due to the involvement of multiple stakeholders.
[20]. Another efficient technique that stands alongside ranking is **weighted critical analysis**. This technique is also used as a sub-process for value-oriented prioritization technique by interviewee 1. Both the interviewees describe this technique as allotting weights to each requirement and prioritizing them after establishing criteria. They were observed to use spreadsheets and whiteboards for. Though literature review identifies AHP as the most promising technique for prioritizing [58], our interviews identify only pair-wise comparisons which is a sub-process of AHP. The informal ways of prioritization techniques identified from interviewee 7 and 8 are an intuitive priority and maintaining product backlog. Intuitive priority was observed as a technique where the requirements are prioritized by either the CEO or project manager or business analyst based on expert opinion. The interviewee 7 describes maintaining a product backlog containing a different set of feature ideas and prioritizing them based on customer needs. Another important observation from our study is regarding selection strategy for different prioritization approach. In the majority of the interviews, it was either the project manager or CEO responsible for selecting a prioritization technique. This finding is in line with M Crowne et.al [5] acknowledging that important decisions in start-ups are made by project owners on an ad-hoc basis.

It is important to report inputs used for an approach when investigating or exploring them in the industry. Prioritization can be performed with different aspects/criteria like value, cost, time etc. While analyzing the answers for the questionnaire like “**What factors do you consider for prioritizing requirements?**”, we obtained some valuable results that present different prioritization criteria like value (business values like sales, marketing), cost (of implementation), time (software development), risk (internal and external), return of investment, importance (of the requirement for the product) are the factors considered for analysis. An important observation from the selected studies is that most of the interviewees considered the clear definition of the criteria (eg. Business values like sales, marketing for value).

Table-8 shows that start-ups consider customer input as the main criteria for prioritizing requirements. This can be interpreted that on a surface, start-ups fail due to lack of customer interest in the product. However, RP is critical to select the right subset of features for implementation. Thus, shortcomings in prioritization are a potential root cause for lack of customer interest. Our finding confirms earlier results by Ahrend [77] acknowledging that start-ups failure is more likely due to lack of customers instead of product failure. The results are also different from the arguments of paper [36] stating start-up companies in the initial stages focus more on the target-market i.e. the requirements are more market-specific and therefore, customer and final users are not involved in requirements prioritization. The customer input criteria is followed by business value and the return of investments as the dominant criteria. The analysis of these practices is done based on factors for prioritizing and type of product developed which are the themes identified from coding interview data.

Explaining the prioritization techniques based on market type is considered important as the suitability of techniques varies for different markets. Software organizations deal with various requirements for their businesses that are either market-driven or bespoke. Since the bespoke companies focus on the design, functionality, and capability of the product, they are observed to focus on factors like cost, the return of investment and stakeholder input aspects of requirements prioritization. From the results, we observe that companies 1, 2, 6, 7, 8 that are bespoke use requirements prioritization practices like 100$ test, Pair-wise comparisons, priority groups, etc. On the other hand, start-ups that focus on delivering market-driven products use prioritization practices where stakeholders and value play a key role.

**RQ2) What are the various challenges faced by the practitioners in start-ups for requirements prioritization?**

The theme 3 and theme 4 identified for our study are challenges and solutions respectively. Qualitative analysis was considered for two themes as the interviewees discussed only those
challenges or solutions relevant to their corresponding prioritization used in their organization. Since most of the challenges and solutions mentioned in the interviews were unique, a qualitative analysis was chosen.

**Table-9 Theme 3: Analysis of challenges faced in start-ups**

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Requirements Prioritization practices</th>
<th>Challenges</th>
</tr>
</thead>
</table>
| 1           | Value-oriented                         | • Lack of support for requirements dependencies  
|             |                                       | • Assigning values to changing requirements  
|             |                                       | • Difficult to apply on a large set of requirements.  
|             |                                       | • Difficulties to communicate different value perspectives among different stakeholders |
| 2           | Cost-benefit analysis, 100$ test       | • Lack of support for requirement dependencies  
|             |                                       | • Assigning values to changing requirements  
|             |                                       | • Lack of communication. |
| 3           | Business case analysis, Weighted critical analysis, Value-based | • Lack of technical knowledge of customers  
|             |                                       | • The inability of expressing the needs correctly.  
|             |                                       | • Changing priorities of customers |
| 4           | Quality functional deployment(QFD), Pair-wise comparisons | • Lack of proper communication  
|             |                                       | • Time-consuming |
| 5           | Value-based                            | • Difference of opinions between person who implements requirements and the ones who examine its effects  
|             |                                       | • Difficult to predict costs |
| 6           | Priority groups, Planning game         | • Difference in stakeholder’s views  
|             |                                       | • Inconsistency in decision making  
|             |                                       | • Lack of proper communication |
| 7           | Value-based, Maintaining a Backlog     | • Lack of knowledge in assigning priority to requirements  
|             |                                       | • Inability of the stakeholder in understanding and handling requirements |
| 8           | Ranking, Intuitive priority            | • Lack of resources  
|             |                                       | • Not suitable for multiple stakeholders.  
|             |                                       | • Unreliable results. |

In analyzing research question 2 (RQ2), this section describes what challenges are faced by the companies corresponding to their practices of requirements prioritization. Table-9 shows what challenges were addressed by the interviewees. The results show that lack of proper communication between the customers and developers and conflicts between stakeholders’ due to multiple views are the major challenges faced by software start-ups. This data is not in
line with [56], that claims scalability is the major challenge faced in requirements prioritization.

Interviewees 8 and 7 had requirements prioritization practices that were not clear and followed informal approaches. After giving a detailed overview of their practices, it was observed that the decisions for identifying and assigning priorities were taken by the high-level managers or owners themselves. The informal techniques explained by interviewee 7 and 8 have many challenges as the requirements are prioritized based on expert opinion. The decision makers could be wrong.

![Figure-7: Value-based Requirements Prioritization](image)

RQ3) What are the solutions to mitigate these identified challenges?

We conducted interviews from 8 start-up companies based in India and more than 50% of the interviewees claimed requirements prioritization to be a challenging task prior to developing a quality product. The interviewees described various challenges they faced while implementing prioritization practices at their company and certain views to overcome them. The goal of this research question is not presenting a best practice but to identify few approaches or solutions that might help the practitioners to utilize these practices efficiently. Since lack of communication was the major drawback of the practiced prioritization methods, the solutions suggested by the interviewees mostly are different ways to ensure proper communication between stakeholders and customers. The following are the solutions gathered from interviews.

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Lack of dependencies</td>
<td>• Dependency problems can be avoided by using a combination of 2 or more techniques.</td>
</tr>
<tr>
<td>• Assigning values to changing requirements</td>
<td>• Considering the cost, risk and ranking each requirement may be beneficial in obtaining business values like sales, marketing etc., for the requirement.</td>
</tr>
<tr>
<td>Problems</td>
<td>Solutions</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Problem scaling large set of requirements</td>
<td>Revisiting different prioritization models for assessing their capability.</td>
</tr>
<tr>
<td>Lack of communication.</td>
<td>Ensuring proper communication between the client and the developer for getting the exact goals of the customer</td>
</tr>
<tr>
<td></td>
<td>Improving communication between stakeholders and customers</td>
</tr>
<tr>
<td></td>
<td>Conducting daily stand-up meetings will resolve the conflicts and maintaining backlog helps in verifying the release of requirements.</td>
</tr>
<tr>
<td>Lack of technical knowledge of customers</td>
<td>Educating the customers who lack knowledge of what requirements provide them value or profits</td>
</tr>
<tr>
<td></td>
<td>Training the stakeholders TO understand the customer needs and their priorities</td>
</tr>
<tr>
<td></td>
<td>Ability to identify what requirements prioritization factors suitable for a particular product being developed.</td>
</tr>
<tr>
<td></td>
<td>Conducting workshops and putting them in the form of user stories.</td>
</tr>
<tr>
<td>The inability of expressing the needs correctly.</td>
<td></td>
</tr>
<tr>
<td>Changing priorities of customers</td>
<td></td>
</tr>
<tr>
<td>Time-consuming</td>
<td>Techniques that involve groups can be handled efficiently by conducting regular meetings, providing equal importance to every stakeholder’s view to reaching proper consensus.</td>
</tr>
<tr>
<td></td>
<td>Combination of various techniques.</td>
</tr>
<tr>
<td>Difference of opinions between person who implements requirements and the ones who examine its effects</td>
<td>Focus group meetings are to be held as the conflicts between the views of various stakeholders can be resolved.</td>
</tr>
<tr>
<td></td>
<td>Whiteboarding sessions</td>
</tr>
<tr>
<td></td>
<td>Gathering goals from the customers to select the most valuable.</td>
</tr>
<tr>
<td>Difficulty to predict costs</td>
<td></td>
</tr>
<tr>
<td>Difference in stakeholder’s views</td>
<td>Group interviews will help in gathering the views of every</td>
</tr>
</tbody>
</table>
- Inconsistency in decision making
- Avoid miscommunications.
- Avoiding the techniques that confuse the stakeholder’s ratings, or categories of requirements.
- Conducting workshops helps in writing the user stories in agile methodology and are given priority while later discussions.
- Brainstorming sessions teach the importance new set of features and help the stakeholders to prioritize the requirements which they think can be a best for the product.
- Lack of resources
- Spending only a limited amount of resources for each individual requirement.
- While considering different factors to obtain the right set of requirements, use of various practices gives reliable results.
- Daily standups, backlog

<table>
<thead>
<tr>
<th>Lack of knowledge in assigning priority to requirements</th>
<th>Inability of the stakeholder in understanding and handling requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conducting workshops helps in writing the user stories in agile methodology and are given priority while later discussions.</td>
<td>Brainstorming sessions teach the importance new set of features and help the stakeholders to prioritize the requirements which they think can be a best for the product.</td>
</tr>
</tbody>
</table>

Besides the above solutions, one of the interviewees also suggested the need for having an idea of how the customer uses the results of any collected data. Interviewee 1 as a developer shares his experience to support the above claims “I had an external customer, but he just wanted to have the reports as quickly as possible. the thing is that when you do this kind of analysis you can do it manually, then it may take 10 people to manage it or 3 months or you could divide the system which removes the manual work to a considerable extent and you can repeat the data analysis on diff data sets so many times. But the customer was only interested in one of analysis, right? so it didn’t really need to have this data analyzed all the time. So, there was a miscommunication.”
RQ3.a) What are the pros and cons of these solutions?
The pros and cons of the solutions are analyzed and are reported as follows in table 11.

**Table-11: Analysis of pros and cons for the corresponding solutions**

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Solution</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviewee 1</td>
<td>Combination of various techniques. Considering the cost, risk and ranking each requirement may be beneficial in obtaining business values like sales, marketing etc., for the requirement.</td>
<td>Most accurate results, from full analysis perfect decisions, are gathered.</td>
<td>Slow and costly. Manipulation of numbers is possible according to the stakeholder’s importance or approval hurdles.</td>
</tr>
<tr>
<td>Interviewee 2</td>
<td>Using a combination of two or more techniques. Ensuring proper communication between the client and the developer for getting the exact goals of the customer</td>
<td>Relatively fast. Suitable for smaller teams.</td>
<td>Time-consuming if requirements are more.</td>
</tr>
<tr>
<td>Interviewee 3</td>
<td>Workshops to put requirements in the form of user stories. Educating the customers who lack knowledge of what requirements provide them value or profits</td>
<td>Maximizes ROI</td>
<td>Time consuming to implement</td>
</tr>
<tr>
<td>Interviewee 4</td>
<td>Combination of various techniques.</td>
<td>Collect reliable results</td>
<td>Sometimes confusion with techniques</td>
</tr>
<tr>
<td>Interviewee 5</td>
<td>Whiteboarding sessions and focus groups</td>
<td>Helps in educating every stakeholder</td>
<td>Time-consuming if requirements are more.</td>
</tr>
<tr>
<td>Interviewee 6</td>
<td>Group interviews</td>
<td>Clear reports, Uncovering a rich set of requirements in short period of time.</td>
<td>Scalability issues</td>
</tr>
<tr>
<td>Interviewee 7</td>
<td>Brainstorming</td>
<td>Improvising the features frequently, Determining various</td>
<td></td>
</tr>
<tr>
<td>Interviewee 8</td>
<td>Daily standups, backlog</td>
<td>Ability to build new features, Satisfying the customer needs</td>
<td>-</td>
</tr>
</tbody>
</table>
6 CONCLUSION AND FUTURE WORK

The study explores the state-of-art of requirements prioritization practices in the context of start-ups. A research based on thematic analysis was proposed and conducted. As a result, few new conceptual findings were presented containing elements from start-ups context. These elements affect how requirements prioritization practices are conducted. A list of used practices was also provided by conducting interviews from 8 start-up companies based in India.

The research study makes four contributions. First (C1), provides a deeper understanding of requirements prioritization in start-ups perspective. The work focused on understanding the current practices of requirements prioritization used by the software start-ups through interviews. Further, the practices that are majorly being used are been listed. Secondly (C2), describes the extent of identified challenges which are being faced with prioritizing requirements are been reviewed. Third (C3), the research also focusses on the methods and processes which are currently being implemented by the practitioners to overcome those challenges. Fourth (C4), this research mainly focuses to identify the better solutions to overcome the major challenges faced by software companies. Fifth (C5), the work also focuses on identifying the solutions i.e., to find the Pros and Cons of the solutions through interviews.

The main findings of this research are as follows: The findings for RQ1 are value-based RP practice and Ranking was commonly used. Value and customer input are the dominant criteria while prioritizing requirements. The findings for RQ2 identify lack of proper communication between customers and organization and conflicts due to stakeholder’s multiple views as the main challenges for prioritizing requirements in start-ups. The findings for RQ3 mostly focus on mitigating the major challenge faced in RQ2. Therefore, these solutions suggest the practitioners to focus on the kind of prioritization practice and software development methodology in use.

The results of this study may help future research on requirements engineering in start-ups or other small-scale industries. It may also help practitioners who have an intention to begin a software start-up company to get an idea of what challenges they may face while prioritizing requirements and can use these solutions to mitigate them. The solutions identified for the challenges faced in using these practices allow the practitioners to approach them in a better way. The emergence of the software start-up research area reflects the fact that we need to better understand this phenomenon to learn valuable lessons and accumulate valid knowledge to benefit future software initiatives.

6.1 Future Work

1. Requirements re-prioritization is an area that requires some future work in terms of start-up companies.
2. A proper survey for collecting information about practices and challenges of requirements prioritization in start-ups can provide valuable and reliable results.
3. Performance comparison of the top techniques practices in the perspective of start-ups found in this research.
4. This research focused only on Indian software start-up companies and it is recommended to extend it to Swedish software start-up companies as well to get a broader perspective. Scaling the sample size for an interview is also recommended.
5. Evaluating the identified solutions for reliability is another aspect of future research.
References


M. Aasem, M. Ramzan, and A. Jaffar, “Analysis and optimization of software


### Table-12 Literature Review Results

<table>
<thead>
<tr>
<th>S.no</th>
<th>Technique</th>
<th>Description</th>
<th>Challenges</th>
<th>Literature</th>
</tr>
</thead>
</table>
| 1.   | Analytic Hierarchy Process (AHP) | A pair-wise comparison matrix used to calculate the importance of every single requirement. | • Tidious for large number of requirements.  
• Not expandable  
• Not suitable for large-scale requirements. | [50] [22] [20] [25] [56] [58] [24] [57] [78] [79][80] [81] [82] [83] [84] [85] [86] [87] [88] [89][90] [91][92] |
| 2.   | Quality functional deployment    | Matrices are used in this technique to sequentially represent the client’s expectations and how developers meet those expectations. | • Mostly applied to small subsystems.  
• Typically applied to small subsystems.  
• does not cater for inconsistencies and not scalable. | [24] [57] [80] [83] [84] [85] [87] [88] [89] [90] [91] [92] [93] [94] |
<p>| 3.   | Planning Game                    | In this technique, requirements are categorized by the clients in three classes like essential, conditional, and optional. The processes depend on criteria: client’s business value and developer’s technical risks. | Not scalable for large number of requirements. | [47][24][50][25] [78] [80] [82] [84] [94] [95] [96] [97] [98] [99] [100] [75][101] |
| 4.   | Cumulative Voting ($100 Dollar Test) | In this technique, stakeholders are given 100 imaginary units or $100 note to invest on elicited requirements. Later, total expended money per each requirement is divided by total no of stakeholders involved in | Is not appropriate for enormous number of requirements. | [50] [25] [81] [82] [84][97] [98] [102] [103] [104] [105] [106] |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 5. | **Cost-value Approach** | Requirements prioritization is done depending on their perceived value and implementation cost. | - Not scalable.  
- Time-consuming. |
|   |   |   | [57] [78] [80][81][90][95] [96][102] [107] [108] [109] [110] |
| 6. | **Top ten requirements** | Important top-ten requirements from a set of requirements according to stakeholders are chosen in this technique. | As ranking process do not utilize weights, this technique can be conflicting and ambiguous. |
|   |   |   | [25] [82] [102] [104] [105] |
| 7. | **Win Win** | The goals of each stakeholder are considered as win conditions and if every stakeholder approves, then those win conditions become agreements else iterations take place. | When biased stakeholders are involved consensus are difficult to reach. |
|   |   |   | [25] [80] [82] [97] [105] |
| 8. | **Pair wise comparisons** | Ranking of requirements are done by comparing in pairs till the important requirements appear at the top of the stack. | This technique is time consuming, difficult, and unreliable results. |
|   |   |   | [47] [81] [89] [94][75] [107] |
| 9. | **Priority Groups** | The information gathered from priority groups is used where categorizing the requirements is based on the stakeholder’s importance. | Do not have the strength of indicating consistency for the judgment of decision makers |
|   |   |   | [20] [58][80] [86] [94] [111] |
| 10 | **Numerical Assignment** | Grouping requirements into different categories i.e. high, medium, low is done in Numerical assignment. | Due to the categorization of the technique it led to criticism among the stakeholders. |
|   |   |   | [24][50][25] [82] [95] [96] [102] [104] [107] [112] |
| 11 | **Ranking** | Numbering of requirements is done from 1 to n to rank requirements. | - Scaling with enormous number of requirements is not possible. |

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

49
Relative difference between the ranked items is not shown, it is difficult to order different stakeholder’s views. Thus, it is suitable for single stakeholder.

<table>
<thead>
<tr>
<th></th>
<th>Value-oriented prioritization</th>
<th>Mapping of requirements to identified business values is done and stakeholders ratings are considered to prioritize requirements.</th>
<th>this technique is not applicable for large-scale projects.</th>
<th>Do not consider requirements dependencies for computational procedure.</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Software Engineering Risk understanding and management</td>
<td>Estimates are given to cost, benefit, development risk, and operational risk reduction to accomplish prioritization process.</td>
<td>Dependencies between requirements are not handled.</td>
<td>[103]</td>
</tr>
</tbody>
</table>

Appendix B

Concept-centric Matrix
### Concept-centric Matrix

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

### Concept-centric Matrix

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

### Concept-centric approach

<table>
<thead>
<tr>
<th>Practice</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analytic Hesitancy Process (AHP)</td>
</tr>
<tr>
<td>2</td>
<td>Quality Function Deployment</td>
</tr>
<tr>
<td>3</td>
<td>Planning Game</td>
</tr>
<tr>
<td>4</td>
<td>Binary Search Tree (BST)</td>
</tr>
<tr>
<td>5</td>
<td>Cost-benefit Analysis (CBA)</td>
</tr>
<tr>
<td>6</td>
<td>Expert Systems (ES)</td>
</tr>
<tr>
<td>7</td>
<td>Fuzzy Logic (FL)</td>
</tr>
<tr>
<td>8</td>
<td>Genetic Algorithms (GA)</td>
</tr>
<tr>
<td>9</td>
<td>Heuristic Algorithms (HA)</td>
</tr>
<tr>
<td>10</td>
<td>Neural Networks (NN)</td>
</tr>
<tr>
<td>11</td>
<td>Petri Nets (PN)</td>
</tr>
<tr>
<td>12</td>
<td>Simulation (Sim)</td>
</tr>
<tr>
<td>13</td>
<td>Soft Computing (SC)</td>
</tr>
<tr>
<td>14</td>
<td>Swarm Intelligence (SI)</td>
</tr>
<tr>
<td>15</td>
<td>Concept Centric Approach</td>
</tr>
</tbody>
</table>
Appendix B

Interview Questionnaire:

Thank you for agreeing to participate in this small interview or discussion where we would like to study on how requirements are prioritized in software start-up companies. This work serves as a part of my research project for my Master's Thesis at Blekinge Institute of Technology, Sweden. This interview contains few generic questions about your organization, following with few project specific questions. "Requirements Prioritization is used in software product management to determine which important requirements to be involved in a certain release and to minimize the risk in development." By this interview, we gain knowledge on various Requirements Prioritization practices. Thus, this study helps us to identify major challenges faced while implementing prioritization in your organization or in software start-up companies to serve you better in future. Your answers will add valuable information in my research. The interview takes around 30-45 minutes for completion. All answers you provide will be kept strictly confidential.

The following question can be answered with some general knowledge in the field. If you're not sure about any question you can skip to the next one.

General questions:

1. Can you please introduce yourself and your current role in this company?
2. Could you please describe three main responsibilities you have in your current role?
3. How much experience do you have in this position?
4. How long have you been working in this company?
5. Did you work in any established company prior to this? Could you please share your experience there?
6. What is your experience in Requirements Engineering?
7. Do you get your requirements from a specific customer or try to understand needs of a wider audience?
8. Could you please tell me where the products of your company are used and how?
9. In what kind of products are you involved in? (market-driven, product-driven, service-driven)

Project specific questions:

10. Can you tell me about an interesting project where you were involved recently?
11. Could you describe a project you were involved in that did not go as planned?
Project specific questions:

10. Can you tell me about an interesting project where you were involved recently?
11. What Software Development method was applied for this project?
12. How do you gather requirements for this project?
13. Could you please guide me through how you select requirements for implementation? What roles are involved?
14. What factors play a role when prioritizing requirements?
   a. User input
   b. Quality
   c. Cost
   d. Importance
   e. Return of investment
   f. No criterion
   g. Others

15. Did you use any requirement prioritization method in any of your project? For example, AHP, grouping, cumulative voting, ranking and others
16. What changed? Did the practice provided results you expected?
17. Could you walk me through how you applied the practice?
18. So, this technique/method is applied for every project or specific ones?
19. What was your experience when a combination of techniques was used? How were the results different from previous once?
20. What challenges/problems you faced while implementing this technique?
21. How were these problems mitigated?
15. Did you use any requirement prioritization method in any of your project? For example, AHP, grouping, cumulative voting, Ranking and others.

16. What changed? Did the practice provided results you expected?

17. Could you walk me through how you applied the practice?

18. So, this technique/method is applied for every project or specific ones?

19. What was your experience when a combination of techniques was used? How were the results different from previous once?

20. What challenges/problems you faced while implementing this technique?

21. How were these problems mitigated?

22. Why do think it is important to mitigate these problems?

23. How do you think these solutions can improve the practice?

24. How do you think these solutions affect the process negatively?

25. Does your organization use any software tools to prioritize requirements? If yes, what are those tools and how do they affect the process?

26. Any additional information on this topic which you would like to discuss?
Appendix C

Color Coding

Hello, this is Sarah, a healthcare management system. We are a predictive and prescriptive analytics company, focusing on population health management. We are an early-stage young company where we identify the risky population ahead of the time before they land up in the hospital, what we are doing is the company. We work with hospitals and health insurance in the United States. Presently, the focus is on the US market, what kind of international programs hospitals can design for risky population to increase quality and reduce their cost. We are a healthcare management company, we have building products for risk models. We have an engineering sector it’s Healthcare, we have developers, system analysts, business analysts working with clients, 10+ clients and more.

Yes, it is a software company which develops products which mainly focus on healthcare companies.

How do you do requirements gathering, and how do you prioritize them and do you use any tools or practices?

Absolutely, so we do it, so they are several ways to collect requirements. We research the market place in the current problem, so healthcare has 2 types of regulations federal and regulatory requirements. So, as a healthcare company, it’s very important that we are aware of the internal subject matter experts, these are people who can provide us with the information. This is one thing we do, and this is where the basic tenants are used to go in each release, what is the thing we need to get done, and try to evaluate the results.

In experience on an existing product. We can model the clients, each client has some set of requirements and none of the requirements match for the same product because everyone is different. In this environment, we use the three clients and the product that is already in the market. We do the bundling session where requirements are written on board and discussed what is common across all the requirements, what is required must solve business object and what is the priority of the business object. If this priority is high, then it is something that we need to work on. If it is the mandatory, allowed for the three clients have a common requirement because it is a product based and not a service based company, we work on a scale, what do some clients tell us, if a product is consumed by 7 clients they have by majority of the business up to client, some clients might not take this priority but what is the majority we take. Some clients do not even know what is priority, so we educate
Color Coding for interviewee 2

services translated that nobody should care any form of social security number. Social security number just like any other customer information, it should not be stored. For the client database, it should be encrypted. It is important to ensure that the data is secure and not accessible by unauthorized personnel. This is to prevent any unauthorized access to personal information. The client database should be regularly updated and monitored to ensure that it is secure and up-to-date.

The client database should also be designed to handle different types of clients, such as new clients, returning clients, and clients with special needs. This allows for a more personalized approach to customer service. The database should also be designed to handle different types of financial transactions, such as deposits, withdrawals, and transfers. This allows for a more efficient and streamlined process for handling financial transactions.

The client database should also be designed to handle different types of inquiries, such as account balances, transaction history, and account statements. This allows for a more efficient and streamlined process for handling inquiries. The database should also be designed to handle different types of customer service requests, such as account changes, account closures, and account suspensions. This allows for a more efficient and streamlined process for handling customer service requests.

The client database should also be designed to handle different types of customer complaints, such as account issues, service issues, and product issues. This allows for a more efficient and streamlined process for handling customer complaints. The database should also be designed to handle different types of customer feedback, such as positive feedback, negative feedback, and neutral feedback. This allows for a more efficient and streamlined process for handling customer feedback.

The client database should also be designed to handle different types of customer communications, such as e-mails, phone calls, and in-person communications. This allows for a more efficient and streamlined process for handling customer communications. The database should also be designed to handle different types of customer requests, such as account requests, product requests, and service requests. This allows for a more efficient and streamlined process for handling customer requests.

The client database should also be designed to handle different types of customer preferences, such as account preferences, product preferences, and service preferences. This allows for a more efficient and streamlined process for handling customer preferences. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.

The client database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments. The database should also be designed to handle different types of customer segments, such as new customers, returning customers, and high-value customers. This allows for a more efficient and streamlined process for handling customer segments.
Color Coding for interviewee 2

Color Coding for interviewee 2