To Require or not to Require: A Case Study on the Benefits and Drawbacks of Requirements Engineering Practices in Startups

Bachelor of Science Thesis in Software Engineering and Management

Olivier Manzi
Salvatore Spanu Zucca

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

To Require or not to Require: A Case Study on the Benefits and Drawbacks of Requirements Engineering Practices in Startups

[**Context and problem**]: Software startups operate in conditions of uncertainty to develop software intensive products/services. While few startups succeed, the majority fail and poor requirements engineering practices play a significant role in the failure of startups. [**Research and contribution**]: Through an exploratory multi-case study on software startups, we investigate the requirements engineering practices that the they use and their benefits and drawbacks through interviews and surveys. In conducting this study, practitioners can benefit from a list of recommendations on how to make better decisions regarding the practices they adopt and, consequentially, avoid supplying products that fail to meet customer demands. Moreover, academia can benefit from the empirical insight into how startups manage their requirements and the value that they receive from practices.

Supervisor: Jennifer Horkoff
Examiner: Richard Berntsson Svensson

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

To Require or not to Require: A Case Study on the Benefits and Drawbacks of Requirements Engineering Practices in Startups

Olivier Manzi
University of Gothenburg
Gothenburg, Sweden
olivermanzi.school@gmail.com

Salvatore Spanu Zucca
University of Gothenburg
Gothenburg, Sweden
salvatore.spanuzucca@gmail.com

***Abstract*—*[Context and problem]*: Software startups operate in conditions of uncertainty to develop software intensive products/services. While few startups succeed, the majority fail and poor requirements engineering practices play a significant role in the failure of startups. *[Research and contribution]*: Through an exploratory multi-case study on software startups, we investigate the requirements engineering practices that the they use and their benefits and drawbacks through interviews and surveys. In conducting this study, practitioners can benefit from a list of recommendations on how to make better decisions regarding the practices they adopt and, consequentially, avoid supplying products that fail to meet customer demands. Moreover, academia can benefit from the empirical insight into how startups manage their requirements and the value that they receive from practices.**

***Index Terms*—Startup, Requirement Engineering, RE, Sweden, UK, Software Engineering, Practices**

I. Introduction

Software Startups[1] are businesses that work under conditions of uncertainty to *"develop software-intensive products"* that meet the needs of a known or unknown customer segment(s) [1]. Not only do these businesses face a 90% fail rate [2], they also experience significant time, market and competitor pressure, limited resources and inexperienced employees. These factors can lead to poor Requirements Engineering practices[2] and consequently, products that do not fill the market demand which can have far-reaching consequences like devastating losses of: investments, jobs and potential positive social impacts [3].

Requirements Engineering (RE) is a branch of *Software Engineering (SE)* involved with the 'why' and 'what' of a system, the relationships of these factors and their evolution over time and across the *Software Development Life Cycle* (SDLC). RE spans across the entire SDLC as a process concerned with examining real world problems and analysing them to adequately understand and document requirements in a form that allows common interpretation and implementation [4] [5].

[1]The terms startup and software startup will be used interchangeably to refer to software startups.

[2]Throughout this research, we will refer to RE practices as repeatable and structured ways of working followed by the participants involved.

The SE research community, in general, is constantly trying to improve the SDLC. Strides have been made with software development practices designed to keep software projects nimble to change. However Yau and Murphy highlight a difference in focus regarding RE research in large enterprises compared to startups [6]. One reason for the focus on established companies could be the abundance of resources that allow such companies to accommodate research in contrast to startups and their limited means (e.g. less time and labour while competing with more market/competitor pressure).

It is difficult to pinpoint the exact factor(s) that play a role in the failure (or success) of a startup, however studies have indicated that poor RE practices can affect the outcome [7] [8].

A few studies point to the crucial role that RE plays in the success of startups, [1] [7] [9] [10], and it is evident, from a SE point-of-view, that one of their main challenges is the adoption of standard engineering practices like RE [11]. Therefore, it would be of special interest to investigate the benefits and drawbacks of RE practices used by startups.

Our study aims to answer the research questions listed in Table I: **RQ.1** aims to understand which practices, if any, are adopted by software startups as well as the frequency of these practices, and **RQ.2** seeks to understand the advantages[3] and/or disadvantages[4] of such practices, if any, that are used by startups.

Understanding the impacts of RE practices can give practitioners a point of reference regarding which practices to adopt, as well as insights into the current state of practice among software startups. In providing such insight to practitioners, the aim is to also provide future RE studies with a foundation of knowledge that they can build upon.

The **outline of this paper** will go as follows: section II discusses fundamental concepts required to fully understand the research, followed by similar studies described in section III. Section IV explains our research methodology. Section V illustrates the results from our data collection and analysis

[3]Something that improves or promotes.

[4]An unfavourable condition or situation.

TABLE I
RESEARCH QUESTIONS

RQ	Question	Rationale
RQ1	Which RE practices, if any, do startups adopt and how frequently?	In order to know the benefits/drawbacks of RE practices, we have to know which practises are used and their frequency in startups.
RQ2	What are the benefits/drawbacks of RE practices adopted by startups?	We want to know why startups use RE practices and the impact that such practices have on the startup.

which are then discussed in Section VI with consideration to similar studies and a list of recommendations for practitioners. Finally, Section VII concludes our study.

II. Background

A. Requirements Engineering

According to Nuseibeh and Easterbrook, [4], RE is the process of *"identifying stakeholders and their needs, documenting these in a form that is amenable to analysis, communication, and subsequent implementation"*. Additionally, the IEEE Standard Glossary of SE, [12], defines a requirement as *"a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents"*. Furthermore, the RE process can be broken into four stages: elicitation, analysis, specification and validation [13].

1) Requirements elicitation: This is the first phase of the RE process concerned with understanding the problem that the software should solve. It takes place before the development of a software system for mediating between the domain of the system stakeholders and the developers, as well as between the differences in technical language of the two entities.

2) Requirements analysis: Involves the analysis as conceptual modelling, categorization and prioritization of the requirements previously elicited, to avoid potential misinterpretations.

3) Requirements specification: During this phase, specifications related to the requirements elicited and analysed are created to further support the understanding, evaluation and review by involved people.

4) Requirements validation: The process of validation and verification occurs after the development of a solution to ensure the software engineers' understanding of the requirements and that the system meets them as expected.

B. Startups

Ghezzi defines startups as a human institution designed to deliver a new product or service under conditions of extreme uncertainty [14].

While there is no universally accepted definition of startup, this study will define it as businesses working in uncertainty to "develop software-intensive products" with the aim of building a scalable business model [1]. However, classifying startups according to the previous definitions can be inaccurate, thus our study will involve businesses that consider themselves to be startups.

III. Related Work

As far as we know, there is no research that explicitly investigates the benefits and drawbacks of RE practices in a software startup context, making it a research topic in need of understanding.

A. Requirements Engineering in Software Startups

Melegati *et al.* have researched the practices used by startups located in the South-American region, providing an understanding of how startups manage their software requirements [15] [16]. They however lack information on how the different ways of working used by such businesses can be beneficial or counterproductive.

Through a multi-vocal literature review, Tripathi *et al.*, [8], emphasize the importance of standardized SE practices in the development of software products, highlighting the key role that RE plays in building successful products. They presented a detailed overview of the requirements process in software startups.

Rafiq *et al.* [17] examine startups with the goal of investigating their current state of Requirements Elicitation. They aim to bridge the understanding of elicitation in a startup context in contrast to established businesses. The paper highlights the immature level of elicitation practices that startups implement while also providing a list of the practices used. Although the study only looked at elicitation, it emphasized the exploration of other stages of the RE process.

B. Software Startups

Paternoster *et al.* [9] examined 43 studies, their findings partially illustrated the adoption of SE practices, describing the establishment of the RE process as being *"challenging in the context of startups"* and mostly reduced to basic activities. As a result, the study identified RE practices used by startups but lacked insight into the benefit and drawbacks of the respective practices.

In parallel, Berg *et al.* [10] conducted a systematic mapping study (*four years after Paternoster et al.* [9]) with the goal of providing *"an updated view on software startup research"*. The authors looked at studies published between 1994-2017 and ranked RE as the third most popular topic in software startup research with ten papers being published over the course of 23 years, trailing behind management (first) and processes (second) which share a combined total of 30 papers. The authors attested to a need for more research on elicitation techniques as well as negotiation, specification and validation.

Klotins *et al.* [7] examined 88 reports to provide insight into the most relevant engineering areas for software startups. Their study finds that RE is the most discussed SE knowledge area and a central activity in startups. While the study examines SE in startups, the authors discuss the different RE practices that startups adopt in the Elicitation, Analysis and Validation stage of the RE process, explaining in the conclusion that the RE practices adopted are *"often rudimentary"*, consequentially leading to *"unwanted debt, poor product quality and wasted resources on building irrelevant features"*, finally concluding that further work is needed in identifying good RE practices for startups. While the paper highlights RE practices used by startups, it doesn't investigate the benefits and drawbacks that practitioners endure when adopting the respective practices.

Unterkalmsteiner *et al.* [1], with the help of past and current works, propose a research agenda/outline of studies regarding "good" SE practices in startups. The result of the agenda is an emphasis on the research regarding *"efficient and effective RE practices"*.

Our intent with this research is to provide insights on the benefits and drawbacks of the RE practices used by startups involved with software development.

IV. Research Methodology

This study aims to answer "which" benefits and drawbacks RE practices have in startups. Since startups operate in environments of uncertainty where many uncontrollable factors play a role, one of the simplest ways to conduct our research is to run an exploratory multi-case study using semi-structured interviews and a survey as data collection methods for mainly gathering qualitative data.

A. Case Study Selection

The cases under investigation were startups involved with software development, while the unit of analysis was the requirements practices used by startups. It is not possible to establish strict sampling methods (such as probabilistic sampling) given that it may result in very few samples (if any), due to the unwillingness for startups to collaborate while operating under pressure. Therefore, the sampling method chosen for interviews was non-probabilistic convenience sampling.

Similar to the interviews, the survey's respondents were collected by convenience sampling, so that we would have collected as much data as possible, as long as the participants met our criteria. The survey respondents were required to be currently working within a software startup or employed in the last three years and should have a role that is related to the software product development to avoid data that is irrelevant to the domain of RE. Nonetheless, the criteria was still broad enough to receive a variety of data sources such as CEOs, product owners, developers or even UX personnel.

B. Data Collection

RE practices are very human-focused given that they act as a medium of communication. Their purpose is to state the problem being solved and ensure that the correct implementation is developed. Therefore, to answer our research questions, we conducted interviews with multiple practitioners followed by a post-interview survey[5] which enforced **data triangulation** (the use of multiple data collections methods and data sources to gain effective understanding on the phenomenon from different perspectives).

Interviewees and survey respondents were contacted through social media platforms[6], direct contacts (mail, personal connections), indirect contacts (mutual connections) and related research communities[7]. It should be noted that the survey excluded respondents that took part in the interviews to avoid "duplicate" data.

C. Data Collection Instruments

1) Interviews: The first data collection instrument was semi-structured interview with open questions about the startup and interviewee. The remaining questions regarded the research questions stated. The goal of the interview was to get detailed insight into the context and reasoning that practitioners had regarding the practices used, if any, and the benefits and drawbacks of the respective practices.

The interview guide (Table IV) and rationale (Table V) can be found in Section IX. Although the questions are structured, follow up questions were asked to gain better understanding and context from the answers provided by the interviewees. If permitted, the interviewees were recorded, otherwise one researcher would have transcribed the dialogue whilst the other conducted the meeting. Each interview included an initial and final discussion regarding our ethics, ways of working and what to expect during and after the interview. Afterwards, the dialogues were transcribed and sent to the respective interviewees for confirmation. This provided the interviewees with the opportunity to redact information or make suggestions that could support their points.

2) Survey: After the interviews were completed and analysed, a cross-selectional survey, constructed on the findings from the interviews was deployed. The goal of the survey was to gain a broader set of data points from different perspectives. The criteria for survey sample population was identical to the survey and the sampling method was non-probabilistic convenience sampling in order to obtain as many responses as possible.

The survey's design is shown in Table XII which can be found in Section IX. It consisted of open and closed questions and was pre-tested to be adjusted accordingly.

The survey investigated how often respondents used the practices that emerge from the interviews and what level of satisfaction they assigned to each practice. The data collected for each practice were averaged (*simple average*) according to the responses from the total number of respondents. The results, whether it is the frequency of usage or the satisfaction

[5]The survey will exclude individuals that have already been interviewed.

[6]We picked social media platforms with audiences from Sweden and the United Kingdom.

[7]Software Startup Research Network (SSRN): a network of researchers and practitioners involved in software startups.

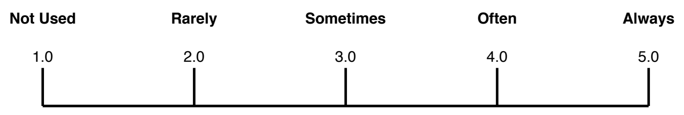

Fig. 1. Survey: Frequency of RE Practices

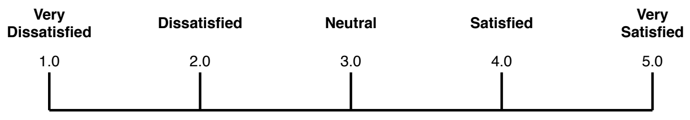

Fig. 2. Survey: Satisfaction of RE Practices

assigned to practices, the results can be compared to the scales shown in Figure 1 (Frequency) and Figure 2 (Satisfaction).

D. Data Analysis

The qualitative data gathered from the interviews was thematically analysed, whereas a combination of thematic analysis and descriptive statistics were used for the survey on account of the open and closed questions. Since the study is exploratory, there were no assumption made regarding the RE practices used by startups prior to the research hence a *data-driven* approach was used to code the interviews [18]. This consisted of two main steps: open coding and axial coding. The first step, open coding, identified our units of analysis (RE practices) and their advantages and disadvantages. The second step, axial coding, consisted of finding a relationship between the codes that emerged from the open coding in all interviews. The second step visualized the collection of codes that emerged and produced the categories found in Section V. It should be noted that open coding was performed separately by the coders to ensure the reliability of the coding process and eventual conflicts were evaluated in conjunction by the authors [19].

E. Ethics

Due to the ongoing global pandemic, COVID-19, interviews were conducted virtually. Given that we wanted to record audio and video for the data analysis section, interviewees were asked to provide consent to the recording of interviews.

Prior to consenting, interviewees were informed about how their data would be handled and what to expect after the interviews were completed.

To ensure the participants authorization in using their provided data, consent and withdrawal are asked for confirmation during the member checking, in which we sent the transcriptions of the interviews back to the participants, allowing them to correct or confirm what they previously said. To prevent the risk of compromising local recordings, interview recordings were stored, securely, in an encrypted online storage.

V. Results

This section will introduce the findings that resulted from the analysis of interview and survey data as described in Section IV; such results have been merged where homogeneous.

A. Demographic data

Data was collected from a total of 14 startups as illustrated in Table II. Nine interviews (ID1-ID9) were conducted with practitioners from eight different startups in Sweden and the United Kingdom while five respondents (S1-S5) took part in the survey. The demographic data also displays market sectors, age, size and if the startups had entered the market.

B. Requirements Engineering Practices

Figure 3 shows the coding tree that emerged from our thematic analysis and their grouping, according to the RE stages described in Section II. Table III illustrates all the categories and practices that surfaced from the data collection instruments; such practices represent actors, ways of working or entities involved in the RE processes of startups, both interviewed and survey participants. The table also shows the number of appearances for each practice (from the interviews) as well as an average frequency and satisfaction[8] (from the survey).

The following subsections will describe the practices used by the population sample, grouped according to the categories that were derived from the data analysis. The ***categories*** will be bold/italic and the **practices** will be bold.

1) *Requirements Elicitation:*

Feedback from colleagues and **Market research** were the most frequently used practices[9] with an average frequency of 4.2. On the opposite end, **Requirements from Management** was "rarely" or "sometimes" used by three of the respondents.

Internal sources

Requirements elicited internally from entities present within the business.

- **Colleagues feedback**
  One of the interviewed case elicited requirements based on direct feedback from the development team. In such instance, the developers that were more familiar with a similar product on a technical level could provide system requirements that non-technical roles may have missed.
- **Requirements from management**
  This way of eliciting requirements was gathered from the interview research method: recorded as the second most frequently common practice in the elicitation stage, consists in the direct solicitation from business analysts, CEOs and other people involved with business and management responsibilities.

External sources

Requirements captured or acquired by inspiration from entities external to the business.

- **Market needs / research**
  Resulted from the interview research method, this method

[8]Practices that were not used by the survey respondents show *N/U* (e.g. Not used).

[9]Excluding **Customer Interviews** and **Meetings with multiple customers** which were mistakenly omitted from the survey.

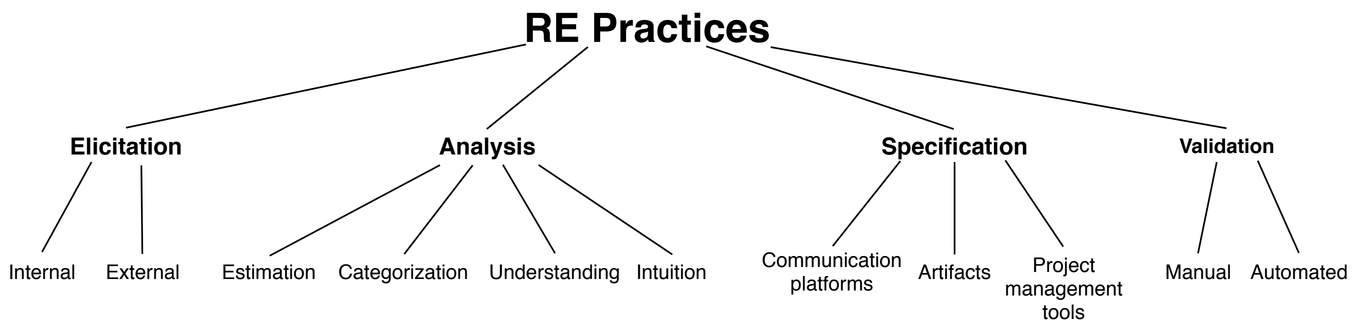

Fig. 3. RQ1 - Code Book

TABLE II
DEMOGRAPHIC DATA

ID	Market Sector	Role	Country	Size (employees)	Age (years)	Reached market
ID1	Biotech	PO	Sweden	N/A	4	No
ID2	Property Management	CTO	Sweden	6	4	No
ID3	Property Management	CEO	Sweden	6	4	No
ID4	Insurance	SET	UK	60	5	Yes
ID5	Consultancy	Mobile Developer	UK	N/A	N/A	Yes
ID6	Educational	Front-end/UX Developer	Sweden	11	5-6	Yes
ID7	Med Tech	CEO	UK	4	2	Yes
ID8	Healthcare	Front-end Developer	Sweden	8	13	Yes
ID9	Educational	COO	Sweden	5	3	Yes
S1	Educational	Software Engineer	N/A	1-5	1	No
S2	Consultancy	CTO	N/A	1-5	1	No
S3	Economic	Software Engineer	N/A	6-10	2	Yes
S4	Healthcare	Software Engineer	N/A	1-5	2	Yes
S5	IT Security	Product Manager	N/A	10+	6	Yes

of elicitation gathers requirements emerged from trends found in the market.

- **Existing/Potential customer interview**
  The most frequently appearing practice: the majority of the startups we interviewed actively elicited their requirements by directly interviewing their existing/potential customers.
- **Meetings with multiple customers**
  Unlike direct customer interviews, this practice differs in the way that product owner(s), CEOs or developers meet with multiple existing/potential customers at the same time. One interviewee reported to elicit requirement in this way.
- **Competitor Analysis**
  Two cases solicited their requirements by analysing their competitors behavior and achievements; reported to be used by two different interviewees.
- **Industry specialist feedback** (Survey)
  As gathered from the post-interview survey, a single case reports to consult industry specialists for capturing requirements.

2) ***Requirements Analysis:***
In the ***Estimation*** category, the most frequently used practices, according to the survey, are **Time** and **Value estimations** with an average frequency of 4.8 however Time, Effort and Value estimations were more common in interviews. **Hard and Soft requirements** and **UX, Front-end and Back-end** at frequency of 4.4 were the most commonly used practices in the ***Categorization*** category. Moreover, survey respondents were satisfied with **UX,Front-end and Back-end** which received an average satisfaction level of 4.4 followed by the later at 3.8. In contrast, 4/5 survey respondents state that they "did not use" the **MoSCoW** method and 3/5 respondents "did not use" the **Fibonacci Estimation**.

Estimation *Ways of analysing requirements by assigning*

TABLE III
RQ1 - RE PRACTICES USED BY STARTUPS

RE stage	Category	Practice	Number of Appearances in interviews	Average Frequency in survey	Average Satisfaction in survey
Elicitation	Internal	Colleagues feedback	1	4.2	4.6
		Requirements from management	5	3.0	3.8
	External	Market needs / research	2	4.2	4.4
		Existing/Potential customer interview	6	-	-
		Meetings with multiple customers	1	-	-
		Competitor Analysis	2	3.6	4.2
		[s] Industry specialist feedback	1	-	-
Analysis	Estimation	Time estimation	2	4.8	4.2
		Effort estimation	2	4.6	3.2
		Complexity estimation	1	4.4	3.8
		Value estimation	2	4.8	4.2
	Categorization	Hard and Soft requirements	1	4.4	3.8
		UX, Front-end, Back-end	1	4.4	4.4
		MoSCoW method	1	1.6	N/U
	Understanding	Five Whys	1	2.6	2.5
	Intuition	User flow	2	4.4	4.0
		Management decision	5	4.0	3.6
Specification	Communication platform	Email	1	4.2	3.8
		Slack	1	4.0	4.0
		Word of mouth	4	3.8	3.0
	Project management tools	Jira	5	4.0	3.8
		Trello	5	3.0	3.0
		GitHub	2	4.0	4.5
		Asana	1	1.2	N/U
		Confluence	2	2.6	4.0
		[s] Bitbucket	1	-	-
	Artifact	Excel file	1	3.0	2.66
		Diagrams	2	4.0	4.75
		Gherkin feature file	1	1.2	N/U
		User story	2	3.2	4.0
		CDS file	1	1.2	N/U
		Existing system on different platform	4	2.4	3.5
		Word file	1	3.0	3.0
		Whiteboard	1	3.8	4.0
		[s] Wireframe	1	-	-
Validation	Manual validation	Internal demo	7	4.4	4.2
		Beta testers	1	3.8	4.25
		Comparison with existing system on different platform	1	4.0	3.6
		Feedback from end user	1	4.4	4.2
	Automated validation	Unit testing	2	4.0	3.8
		Regression testing	1	2.8	3.33

them an estimation on different dimensions.

- **Time estimation**
  Measured in time, this practice represents a way of analysing requirements based on the time needed for their completion (usually represented in days or weeks). Two interviewees reported to follow this way of estimating requirements.
- **Effort estimation**
  Used by two of our interviewees as a way to analyse and estimate their requirements, it consists in giving an estimation of the effort needed to complete a requirement by assigning a value (the Fibonacci estimation is a variance of this method, which was used by one of the two mentioned participants).
- **Complexity estimation**
  In this method, adopted by one interviewee, requirements are analysed by estimating how complex it may be to develop a feature.
- **Value estimation**
  Interpreted as value given to the business or to the customers, this way of analysing requirements associates them with a numerical value was used by two cases (ID7,ID9).

Categorization
Requirements are analysed by dividing them into categories.

- **Hard and Soft requirements**
  Categorizing requirements by their “importance” and “strictness” to be completed by their deadlines. This is contrary to the functional and non-functional association that the terms “hard” and “soft” requirements have in RE terminology.
- **UX, Front-end, Back-end**

This method of analysis, adopted by one interviewee, consists in assigning requirements according to the different product-development points of view.

- **MoSCoW method**
  This method of requirements analysis consists in prioritizing and categorizing requirements according to the following categories: “Must have”, “Should have”, “Could have” and “Will not have”. It was reported to be used by a single interviewee.

Understanding
Validating requirements such that the relevant stakeholders understand the reasoning behind each requirement.

- **Five Whys**
  One of the interviewed startups followed the interrogative technique of the Five Whys, which consists in asking the question *“Why”* five times in order to understand the motive at the root of requirements as well as its feasibility.

Intuition
This category groups practices that emerge from an individuals intuition/subjectivity.

- **User flow**
  Two startups, according to the data from the interviews, prioritize their requirements by implementing them in order of the flow of usage, defined by the path that a user takes to complete a task.
- **Management decision**
  Within two of the startups interviewed, the requirements were analysed and prioritized according to the sole management decision, mostly based on personal beliefs and intuition. In other cases, the requirements were partially analysed by developers; having however the final decision in the hands of a managerial role.

3) *Requirements Specification:*

Emails were the most frequently used practice in the ***Communication platform*** category to specify requirements with an average frequency of 4.2 although **Slack** received the highest satisfaction (4.0) followed by **Email** at 3.8. In regards to ***Project management Tools***, **Jira** and **GitHub** received an average frequency level of 4.0, however, survey respondents were more satisfied with GitHub which received a satisfaction level of 4.5 followed by the **Confluence** tool at 4.0. **Diagrams** were the most frequent ***Artifact*** with a frequency level of 4.0 followed by **Whiteboards** at 3.8. However, when it came to satisfactions, survey respondents were most satisfied with **Diagrams**, 4.7, followed by **Whiteboards** and **User Stories** which both received 4.0.

Communication platform
Ways that practitioners use to communicate requirements related information between each other.

- **Email, Slack**
  Appearing one time each, emails and slack are two of the communication tools used by startups, in which they would specify and document their requirements.
- **Word of mouth**
  Almost half of the startups interviewed (4/9), specify their requirements verbally within the office and virtually using online meetings.

Artifact
These are the by-products of requirements elicitation and/or analysis.

- **Excel, CDS, Word files**
  One interviewee adopts spreadsheet as an artifact for representing requirements as traceable documentation. Specifically to this case, the product owner in charge of such document did not have a technical background. With the same number of appearances in the interviews (1/9), CDS and Microsoft Word files are used to document requirements. Such files are locally stored and managed by one single person.
- **Diagrams**
  UML, flow diagrams and diagrams with templates internal to the business are adopted as artifacts by two startups, according to the interviews data. These artifacts are used as documentation for the requirements. Additionally, such diagrams were stored in Confluence or simple paper around the office.
- **Gherkin feature file**
  There has been one appearance of requirements being analysed in Gherkin in the interviews: a Domain Specific Language used for writing acceptance criteria by following the *Given-When-Then* statements.
- **User stories**
  Adopted by two interviewees, user stories are short descriptions of a feature from the perspective of the stakeholder who mostly desires its implementation. Specifically to one of the participants who adopts such artifacts, their way of working includes associating each user story to a test script (see unit testing) inspired by an acceptance criteria defined by the customer representative.
- **Existing system on different platform**
  Different interviewed cases (4/9) have reported to use their deployed systems on a different platform as a sort of guidance. Since the deployed systems were built on top of previously managed requirements, the developers would refer to those systems as a form of implicit requirements specifications.
- **Wireframes** These are blueprints for a website. Wireframes visually illustrate the backbone of a project and usually takes place in the earlier stages of the project.

Project management tools
Platforms in which the practitioners document, store and track requirements.

- **Jira, Trello, Asana, Confluence**
  Used with equal frequency by startups according to the interviews data, Jira, Asana and Trello are project management tools widely adopted to document and display requirements digitally for development teams. In one case, Confluence was used to store artifacts.

- **GitHub, Bitbucket**
  These are version control systems that include features for project management and storing documentation, although these are not the primary services offered by the platforms. Github was reported to be used by two interviewees, while Bitbucket by a survey participant.

*4) **Requirements Validation**:*
When ***Manually Validating*** requirements, survey respondents rated **Internal Demonstrations** and **Feedback from End User** as the most frequently used practices, on average, with a frequency level of 4.4, however validating products with **Beta testers** was rated with the highest level of satisfaction at 4.25.

Automated validation
Requirements validated with automation tools and systems.

- **Unit testing**
  A validation method used by two interviewees to automate the requirement testing of the single units of the software system, each associated with user stories (and consequently, a requirement) to ensure that the developers implement the right solution.
- **Regression testing**
  Regression testing is a form of testing that examines the functionality of a system from a broader prospective. It detects discontinuities of the system functioning when new features are introduced, which could potentially break the preexisting ones. Having regression testing linked to user stories allows the validation of requirements. This way of working was reported to be used by one interviewee.

Manual validation
Methods of validating requirement that do not involve automated tests.

- **Internal demonstration**
  Resulting as the most common requirement validation practice according to the interviews (7/9), implementations are demonstrated to management roles who decide, manually, whether the implementation is satisfactory or not. It belongs to the manual verification category as there is no automation involved, and its success is often based on people's judgement. A common entity involved with demonstration are **system prototypes**, often requested by the management under periodical milestones as deadline.
- **Beta testers**
  In one of the interviewed cases a beta version of the system would be sent to representatives of the customers to assess its correctness and to give further feedback.
- **Comparison with existing platforms**
  In the case of an interviewee having previously implemented and deployed a product for another platform, it would then be used as reference for validating the new product to be deployed on a different ecosystem *(e.g. Windows and Mac:OS)*
- **Feedback from end users**
  A participant to the interviews collects feedback with surveys to assess their requirements validity, receiving direct feedback from end users as reviews and scores.

C. Benefits and Drawbacks of RE Practices

This subsection presents the benefits and drawbacks of the RE practices that resulted from both interviews and survey. The practices that did not present any benefits or drawbacks are not discussed. Tables VI (Elicitation), VII (Analysis), VIII (Specification) and IX (Validation) shows the results that originated from the data collection instruments which can be found in Section IX.

*1) **Requirements elicitation**:*
Elicitation practices were grouped into two categories: ***Internal*** and ***External*** elicitation as shown in Figure 3. Survey respondents, on average, rated their satisfaction with using **Feedback from colleagues** at 4.6 or between "Satisfied" and "Very Satisfied" in contrast to **Requirements from management** which received the lowest average satisfaction at 3.0 or "Neutral".

Requirements elicited from management were common in startups (5/9 interviewed cases). By management, in this case, we refer to people covering roles such as CEO, CTO and/or Project Managers/Owners. Respondents in non management positions agreed that this method has its advantages since the idea initially came from the CEO or Founder. As one interviewee (ID9) puts it, *"eliciting effective requirements from someone in a management role depends on how similar they are, as an individual, to their target end user(s)/customer segment(s)"*. If the management role and customer segment are not similar, then eliciting requirements can be misleading and cause frequent changes or pivots to the requirements. Nonetheless, if managers have similar needs and interests to their customers, then they can be a useful and reachable source of elicitation.

Feedback from development team has its advantages when it comes to eliciting requirements before and after entering the market. Members that have previous work experience with products that are similar to the startup's provide valuable insight into the requirements that are currently unknown. Colleagues also provide system requirements which can compensate for the non-technical knowledge of management figures. The disadvantages occur when there is no procedure for establishing requirements since employees may begin to implement their own requirements without consulting the rest of the team.

Interviewing customers is advantageous because the requirements allow you to *"build something that someone is willing to pay for"*. Moreover, **Meetings with multiple customers**, while similar to customer interviews, allow startups to get a broader understanding of the general demand, that is, a consensus on which requirements are demanded by the majority of their customer representatives and which are only desired by a few.

Lastly, **Competitor Analysis** is helpful because startups can use them as a form of inspiration. Furthermore, Competitor Analysis allow startups to see which features the market demands.

*2) **Requirements analysis**:*

Practices in the Analysis stage were grouped into four categories: Estimation, Categorization, Understanding and Subjectivity (Figure 3). While the majority of categories received no general praise or critique, the Estimation category was seen as inaccurate at times, especially when the development team had little involvement in the analysis.

Time Estimation was seen to be advantageous because of how flexible it is when it comes to adjusting estimations accordingly, however, it also has its disadvantages because of how difficult it is to estimate the requirements accurately.

Complexity Estimation was seen to be advantageous because of how well it supported trade-off dialogue. By understanding how complex a requirement is to implement, a startup could evaluate whether they'd receive a bigger return of investment if they implement many non-complex requirements or one complicated feature.

The **MoSCoW** method is beneficial when it comes to getting started with requirements analysis. This is because it allows the startup team to filter out the requirements and focus more on those that were higher up in priority. However, under the same light, the method presents the disadvantage of being too *"simplistic"* since it misses bigger-picture details.

An interviewee describes the **Five Whys** technique to be beneficial in analysing requirements since it highlights the root problem that the requirement tries to solve, hence supporting room for discussion.

The **User Flow** method is beneficial to startups as a prioritization technique, as it allows the development team to understand the order in which requirements will be used by the end users. This is very useful when startups do not have an end user yet.

The **Management Decision** as a method of analysing the requirements is common in the majority of cases interviewed (5/9 interviewees). On the other hand, the survey respondents assigned the practice an average frequency of 3.0 ("sometimes"). The benefits of management roles being the sole participant in requirements analysis is that it is much faster than having discussions with groups. Nonetheless, the disadvantages of management roles single-handedly analysing requirements outweighed the benefits. One case (ID2) explains that "when management roles were solely responsible for analysing requirements, it led to constant changes in requirements prioritization and consequently, lower developer productivity". Moreover, when the individuals in management positions lack technical background, the estimations, organization and deadlines assigned to requirements may lack accuracy. To emphasize the point, one case (ID8) elaborates that when developers are not part of the analysis stage, then there lacks negotiation between management and developers (*who will implement solutions for such requirement)* which negatively affects the accuracy of estimations and deadlines for requirements (e.g. effort, time and complexity estimations). In two of the cases, however, developers carried out requirement estimations and left the prioritization to the management positions That way, developers were not assigned inaccurate expectations and managers were able to evaluate trade-offs.

*3) **Requirements specification**:*

On a broader level, the practices in the specification stage were divided in categories as Communication platforms, Project management tools and Artifacts as shown in figure 3.

Communication platforms. One such platform is **Slack**, a business communication platform where users can chat. A drawback that results from Slack is that specified requirements are easily lost and forgotten within the chats between startup members. Another medium of communication is **word of mouth** which is a widely used among startups (4/9). It is a quick method for specifying requirements however a drawback to this practice is that the majority of specifications are easily lost when they are not written down on an artifact. Moreover, given that word of mouth is purely verbal, the requirements specified set acceptance criteria that are difficult to remember and verify.

Project management tools. In general, using multiple different project management tools to document requirements makes it difficult to follow specifications from a developer's perspective: one case (ID1) mentioned the difficulties faced when trying to asses requirements that were stored on different tools. This situation occurs when startups switch to different tools during the development stages but do not transfer all requirements to the new tool. From a developer perspective, the interviewee (ID1) mentioned that it became difficult to track requirements and from a management position, maintaining requirements on all tools becomes burdensome. One such software is **Jira**, a widely used tool by startups (5/9 interviewees), which provides good structure for storing and tracking requirements although hard to maintain without a person dedicated to such task. Another similar tool is **Trello**. A disadvantage that interviewees presented with this tool was the flexibility that the tool provided to documenting requirements. Given that there was no structure to conform to, startups often documented requirements vaguely which left room for developers to make their own interpretations and implementation accordingly which wasn't always successful. Similarly to Jira, it is hard to be fully maintained without a person dedicated to the task. **GitHub** offers features to complement code with documentation, confirmed to be a benefit since it leads to more accessible specifications. This is because the specifications are stored in the same platform as the code, making it easier for developers to access.

Artifacts. Multiple interviewees agreed on artifacts being easier to adopt and follow when everyone in the team agrees with their usage. Although a single case (ID5) affirmed that **not documenting requirements** at all is faster, other cases stated that documenting requirements ensured that their acceptance criteria were met. It should be noted that documenting requirements demands a general effort from all startup employees to be successful. Storing requirements on **Excel spreadsheet files** makes accessing and reading requirements difficult for developers, specifically confirmed by one case in which the product owner managed all the requirements and the entire spreadsheet individually. **CDS files** are highly unstructured text files that easily become too long and hard to maintain. Using an **existing system on different platform** makes less confusion on the developers as it can be used as artifact for reference when specifying requirements. **Gherkin feature files** are another example of artifacts that make unit testing easier and give active feedback on implementation as they follow a Behavior Driven Development (BDD) development methodology and given-when-then statements. As a case mentions, the Gherkin file is beneficial because it facilitates unit testing. **Diagrams** were said to be clearer than word of mouth due to their traceability and persistence (written over verbal), although developers happen to be unfriendly with them, defining such artifacts as business savvy rather than software-oriented. **User stories** present disadvantages when too short: *"one sentence does not describe the requirements enough and demands further clarifications"*; they were also stated to often lack details which makes it hard to understand the big picture of the system.

4) Requirements validation:

We categorised practices into manual and automated Validation according to Figure 3.

Manual validation. One of the cases of our study, which did not adopt any structured practice for the validation of their requirements, affirmed that the benefit of validating manually in a *verbal* manner based on *trust* reduces the friction at work, hence less time and people involved. Validating requirements manually can be easily accomplished in the *early stages of development* as the features are basic, compared to an advanced stage of implementation. Among the drawbacks, it is likely to *miss features out of scope* and does not perform well when scaled. When manually validated by the management, it can lead to *time losses* and it becomes harder when the development is delegated to an *off-shore team*.

Internal demonstrations, frequently used by the majority of cases (7/9) studied, these are quick ways of validating requirements by showing a functioning prototype of the system to the management. This method was beneficial to startups because of how quickly it could be done; as a drawback instead, two cases have reported the frequent prototype demands from management roles have led to increased pressure on the development team and consequently lowering its productivity. Additionally, internal demonstrations do not necessarily highlight whether all components of a system are working as the should, which makes it difficult to verify requirements from a big picture perspective (e.g. integration testing).

One case reported that performing manual verification using the **existing system on a different platform** leads to less confusion.

Direct **feedback from end users** captured through surveys or interviews exposes flaws from the customer's perspective and consequently leads to new and better requirements.

Automated validation. Practitioners reported difficulties with following too many types of documentation when referring to acceptance criteria for validating requirements. **Unit tests** are used to build test scripts which are beneficial when associated with single user stories. However, they present a drawback with the need of team's general effort to perform well and they might not always be followed by the developers.

VI. Discussion

This section intends to discuss and answer the research questions introduced in Section I; according to our findings, we will conclude the discussion with a list of recommendations to software startups on how to manage their requirements in a SDLC, followed by the threats to the validity of our research and how the threats were mitigated.

A. *Answer to R.Q.1: RE Practices*

The first goal of our research was to investigate which RE practices startups use to manage their requirements. The findings will be discussed according to the four phases of the RE process cycle.

In general, the cases studied used numerous practices in all stages of their RE process and, often, in combinations.

Elicitation. A recurrent phenomenon is the *Internal* elicitation of requirements from management positions as well as employees. One case (ID2) reports "*We talk to potential customers and perform market studies, but besides that, what should be developed is in the CEO's head who mainly thinks what the customers want*": presenting the influence that the management role has in the elicitation process. This occurrence is primarily manifested in startups which do not yet have their product in the market (ID1,ID2,ID3). As Melegati *et al.* present in their research, *"Ideas come from anyone in the company"* [15]. In a similar stroke, Tripathi's study finds internal sources to be the most common source of requirements elicitation, especially during the initial stages of development when customers are unknown [8].

Contrary to the interviewees, the respondents from the survey were more prone to elicit requirements from Colleagues. One limitation that presents itself in the survey is that respondents who are in positions of management may have misinterpreted the requirements elicitation of fellow managers as "requirements from colleagues" even though the requirements were coming from another management position (see threats to validity).

It should be noted that acknowledging the contradictions in interviews and surveys highlights the necessity for a larger sample of interviews and survey respondents. Without a larger number of cases to study, it is difficult explain why some practices have contradicting results.

In the *External* elicitation category, Market Needs was used the most often by survey respondents in contrast to interviewees who commonly used Customer Interviews to elicit requirements. The survey did not mention Customer Interviews or Meetings with Multiple Customers which may explain the contrasting results among interviewees and survey respondents and remains a limitation in this study. Moreover, Customer Interviews was the most common Elicitation practice among all cases interviewed which is in line with previous studies by Tripathi's [8] and Rafiq's [17].

Analysis. We found that Time and Value estimation were the most common practices among startups in the interview and survey methods to estimate requirements. These finds tie well with Tripathi's study wherein "value to customers" played a significant role among cases [8].

It caught our attention that the practices Hard and Soft Requirements and UX, Front-end and Back-end, only appeared once, in total, among all nine cases interviewed although both scored a 4.4 frequency of usage which means that there is a clear inconsistency between the results from the two collection instruments. However, there is no data from related studies to compare this finding with and not enough evidence to explain why this occurs.

Specification. Table III shows that startups do not use one particular method for specifying but rather different ones over time. Similar to our findings, Tripathi *et al.* add that startups do not usually have "clear documentation processes and use various tools such as physical and electronic boards" [8]. Jira is the management tool that is mostly used (5/9 interviewees, 4.0 usage frequency) followed by Trello (5/9 interviewees, 3.0 usage frequency).

As resulted from both, our post-interview survey and Melegati's research [15], wireframes are seldom used (S2).

Several similarities with the related work are visible when comparing results on requirements specification: Trello and agile project management tools (Table III), as well as an isolated case of low appliance of requirements traceability: "*We like live documentation because we are developers, documentation is in our code. We are not sitting here writing documents, it is not our job.*"(ID5): this phenomenon similarly manifested in Melegati's study as *"we did not spend much time documenting because information is in people"* [15].

It is notable that the most common artifact is Existing Systems on Different Platforms according to the interviews, which presents an inconsistency with the results gathered from the survey: Diagrams result to be the most occurring artifact from survey's respondents, while Existing Systems were "rarely" used.

Validation. Startups frequently do manual tests to cover the basic usability of the system and its user flow, with internal demonstrations from the developers to the management (7/9 interviewees, 4.4 usage frequency) and less commonly as feedback from end users. Unit testing is the preferred way of automating the requirements validation.

Analogous[10] results can be seen when comparing to the related work: *"MVP, mock-ups, prototypes, surveys"* [15] were recurrent entities in the validation process, equivalent to *Internal demo* and *Feedback from end user* in Table III.

B. *Answer to R.Q.2: Benefits and Drawbacks*

The second goal of this study was to investigate the benefits and drawbacks of RE practices. Firstly, we will show general findings related to the advantages and disadvantages of RE practices; secondly, a discussion on findings related to RE practices in each phase of the RE process is presented. The single **practices** will be denoted by bold text, while the coded ***categories*** are displayed with italic bold text.

Among the several advantages and disadvantages, the following phenomena are common and General to most of the interviewees:

- **G1** There is a common appreciation for the flexibility of individual roles making quick decisions which skip the time-cost and friction that exists in established businesses.
- **G2** Practitioners show a strong desire for structure and general agreements: noncompliance to a way of working can lead to the vanishing of results expected from its usage.
- **G3** Having existing customers allows startups to adopt further beneficial practices, later in the RE cycle, that would not have been possible to adopt otherwise (e.g. a startup without customers does not have a beta-tester that they can use in the requirements validation).

1) ***Elicitation***:

Due to the unlikeliness of having existing customers, startups rely heavily on eliciting requirements ***Internally***. A similar conclusion was reached by Melegati and Goldman who point out that such a scenario depends on the type of product being developed and/or the market that the startup operates within. A client-product is built for a single customer whereas a user-product is operated by multiple customers. When startups are working with client-products, there is usually a customer involved which facilitates the elicitation stage. This is contrary to startups that build user-products that are targeted to a broader, inaccessible or unknown, segment of customers [16].

Additionally, soliciting **requirements from colleagues** within the startup is beneficial when they have prior experience with similar products or a deep technical understanding. Although this practice should not be used as a primary source of requirements, it is an effective method for startups to gather requirements especially when the individual responsible for the RE process is not technical.

One case (ID9), in which requirements are elicited from a management role, reports significant benefits given that such people shared similar interests with their customer segment. In other words, they were also users of their own products which

[10]Similar

allowed them to anticipate requirements that end users may have. In contrast, ***internally eliciting*** requirements is particularly disadvantageous for startups when external validation, to some extent, is excluded. Multiple cases (ID1, ID2,ID6) report that when **management** positions are the source of requirements elicitation, it can lead, pivots and mismatch of customers needs presenting an overall number of cons outweighing the pros.

The majority of interviewees (6/9) described **Existing/Potential customer interviews** to be beneficial because they come directly from those people that would use the product the most. One case (ID3) elaborated that "*it was one way to build something that someone was willing to pay for*".

Existing customers are beneficial and allow for more practices to be adopted, however, if startups do not have any, the next best solution is to interview *potential* customers which can unlock the other practices that require existing customers. One case (ID1) reported to **interviewing potential customers** which mitigated the risks and drawbacks derived by the lack of existing customers.

*2) **Analysis**:*

Practitioners in startups affirm ***estimations*** to be often inaccurate, especially when a decision is taken without the involvement of both management and development parts. Management roles that dictate requirements analysis have led to conflicting opinions regarding the benefits and drawbacks. One case (ID2) mentions that having management roles singlehandedly analyse requirements likely leads to constant changes to the prioritization because of hasty decisions. Two interviewees (ID1, ID7), in management positions, explain that when there is a *single decider* in the requirements analysis stage, the decision process becomes much faster. However one of the two interviewees (ID1) acknowledges that if the management role has full "power" in deciding then it can lead to hasty decisions at the cost of the developers re-implementing solutions.

In contrast, two cases (ID5,ID8) mention that developer voices are crucial to the analysis stage since these are the people that will implement the requirements. Giardino *et al.* underline the significant impact that *empowerment* has on the performance of a startup team [20]. One significant dimension that the study considers to empower a team is "influence in decision making". Sole **management decisions** in the requirements analysis phase also exclude requirements negotiation which can lead to incorrect estimations and deadlines by managers that lack the technical knowledge. Furthermore, when managers and developers are involved in the analysis of requirements, then startups benefit from technical and business-oriented perspectives which mitigates the previously mentioned risks.

Although unpopular, analysing requirements for a better ***understanding*** is advantageous as it prevents the risk of excessive pivoting.

*3) **Specification**:*

Among the ways of documenting requirements, tools and artifacts that do not allow for a good structure or provide such features (see G2 in this subsection), are highly counterproductive from an effort and time perspective (e.g. Microsoft Word and CDS plain text files). Due to the difficulties of changing ***Project management tools*** later in development. Therefore, it is highly favorable to carefully choose a tool in the early stages of a startup and stick with it in a long-term perspective.

Although easy to apply, less time-consuming and volatile[11] methods of communication for requirements specifications are strongly disadvantageous when they are used for the sole goal of creating documentation *(e.g. using Slack to document a requirement or verbally defining an acceptance criteria)*. Managerial decision pivoting manifests on the choice for project management tools when the usage of more than one single instrument creates maintenance and compliance issues. Although one case reported that it is beneficial to avoid documenting requirements because it "saves time", such a statement is countered by the number of disadvantages that resulted from the lack of requirements documentation as found in numerous other cases.

User Stories as an artifact for requirements documentation are disadvantageous when used vaguely and with ambiguity: practitioners feel the need to understand the requirement from a more detailed and big-picture view. Such artifacts, however, present strong benefits when complemented by the usage of unit testing scripts associated to each user story.

Startups that have **systems existing on different platforms** as an artifact for requirements specification can benefit from the practice since it makes further requirements less ambiguous.

*4) **Validation**:*

Among the ways of validating requirements, practitioners have shown strong satisfaction towards involving end users in the validation process, retaining such practices as highly beneficial due to the quality of feedback they receive. Although ***manually validating*** requirements has its benefits, our findings show that such practices may not "assure" the validation of the system as a whole. In other cases, the effort required to manually validate requirements in **internal demonstrations** consumes time, which management roles usually don't have. Most of the disadvantages of manual validation can be compensated by the adoption of supplementary automated validation methods.

An interesting similarity between our results and Melegati's is the partial unwillingness of startups to perform automated validation because *"the time spent to perform validation for a more straightforward task could be higher than the feature development"* [15]. One startup (ID7) mentions that *"We never needed them [automated test]... I would go and use the product that they implemented... we did not use a formal process because we trusted and we were engaged with each other as a relationship."* while another case (ID1) explains that automated testing is not necessary when the product is still in its early stages of development.

[11]Which are of rapid exchange, hard to track and exchanged in methods not designed for solely providing structure.

Sometimes, practices are simply ignored by employees and this is brought to attention by two cases (ID1, ID5) which faced challenges when they tried to implement an acceptance criteria due to the employees unwillingness to comply.

As visible in Table III, one of the startups introduced a regression testing pack to ensure the functionality of the system when adding the implementation of a new feature into the code base.

Relying on **beta testers** and closely involving end users with the validation process is an effective method of validating and eliciting requirements; however not accessible by startups that do not have potential/existing customers.

Although contradicting, our results have shown a need for both speed and structure in the RE practices used: such factors need to be well-balanced to minimise the potential risks given by adopting practices which only benefit in speed (resulting in a potential mishandling of requirements) or in structure (leading to slow times of development and consequential missing of the opportunity to enter the market in the right moment).

Due to the small sampling population and data collected, there was no pattern that emerged regarding the evolution of RE practices in startups that had entered the market and those that had not. Nonetheless, a majority of the cases interviewed emphasized the need for more structure in their process but lacked a concrete plan on the steps needed to "evolve". A similar study by Grahla *et al.* investigated the evolution of RE practices in startups over time [21]. In their study, the majority of startups do not adopt an "engineering approach" to their RE process in an attempt to reduce any hindrance that the approach may have on delivery of their product(s). They do, however, point out that over time, startups begin to establish more structure in their RE processes once they become more "customer-oriented".

C. Recommendations for Practitioners

To summarise the findings of this study, a list of recommendations to startups will be provided below.

1) Concerning the elicitation of requirements, if a startup does not have existing customers, it is highly advised to obtain potential customers early in order to adopt further practices that may benefit the startups RE cycle in the long-run. Relying *entirely* on intuition, thoughts or subjective ideas of a single individual (e.g. CEO) is discouraged. This practice is not beneficial as it lacks external feedback and can consequentially affect the other phases of the RE process. On the other hand, eliciting requirements purely from external sources is not possible for startups which do not have existing customers. It is therefore recommended to elicit requirements internally and externally: such way of working leaves space to the internal ideas and eventual external confirmation (e.g. market needs or potential customers). In the case of startups that have existing customers, it can be beneficial to establish a close relationship. As one case (ID9) explains, startups become more familiar with their end users and this can help them "impersonate" customers when eliciting requirements.
2) Analysing requirements should be approached from a variety of perspectives, as supported by the idea of agile cross-functional teams [22]: startups face numerous disadvantages in the long-run when a decision is taken by either management or developer team, and not both. Estimations, deadlines and understanding are likely to be inaccurate without a general agreement from both groups.
3) When specifying requirements, practitioners should avoid unstructured methods of specification: digital (e.g. Email and Slack) and verbal communication can easily lead to a loss of specifications.
4) When validating requirements, it is recommended to validate manually and automatically. While manual validation provides quick feedback, automated tests compensate for the blind spots that arise from manually testing requirements. The two categories complement one another. Moreover, involving beta testers in the validation phase leads to the elicitation of richer requirements prior to the start of a new RE cycle.

Startups cannot control all factors that play a role in their success but they can decide how to conduct their process [21]. Our recommendations give some feedback on how to do so.

D. Threats to validity

Construct validity. Theoretical terminology may be lost in practical translation leading to a misunderstanding of concepts. To mitigate this threat, the researchers provided definitions for SE terms and complimentary examples during the interviews. Moreover, questions in the interviews and survey were accompanied by explanations/examples to clarify assumptions during the interviews. To mitigate the potential risk of survey respondents misunderstanding the terminology used when referring to the RE phases, descriptions and examples were provided in all the survey questions

Internal validity. The convenience sampling method chosen for the survey may not be representative of our population. Furthermore, by sending out the survey through email and social media platforms, there was no control regarding which survey respondents answered the forms, as well as their country. To mitigate the risk of including participants from different economical and social environments from Sweden and the United Kingdom, the survey was shared to private groups and connections limited to such countries. When creating the survey, questions regarding the *customer interviews* and *interviews with multiple customers* practices that resulted from interviews, were unintentionally omitted. This means that respondents could not provide their opinions regarding these practices, however, the survey mitigated this issue by providing respondents with a field to enter practices that were not originally shown in the survey. Another limitation regarding the survey (refer to Table XII) is that open questions were not mandatory, which explains the lack of explanation from survey respondents as to why they assigned certain values to practices. The purpose of the surveys is to validate the results of the interviews. To mitigate the risk of collecting data that is less representative, we excluded interview participants from the survey.

External validity. It should be noted that this study is conducted in Sweden and the UK which may limit the external validity given the different social and economical settings that may exist in other geographical locations. An external threat to validity is that some of the interviewees may be demographically similar since they originate from the same social network (e.g. personal contacts). We mitigate the threat by diversifying the age and role of the participants in our research. Furthermore, data triangulation is applied to reduce the threats to reliability of the research by conducting multiple research methods (interview and survey).

Reliability. A threat to validity exists from thematically analysing the interview transcripts which can potentially lead to author bias and unreliable codes. To mitigate the threat, the authors will apply the Inter-Coder Agreement whereby authors code the data independently and come together to discuss and resolve difference [19]. For example, in some cases, interviewees implicitly answered interview questions without being aware. In such cases, the researchers would note the answers and associate them with the appropriate questions via thematic analysis and resolve conflicts together.

VII. Conclusion

Software startups develop innovative products while subject to pressure and uncertainty. Due to the context and environment in which these business operate, they generally adopt unstructured practices to manage their requirements. It is crucial for practitioners in such businesses to understand the benefits and drawbacks of the RE practices they adopt. After conducting an exploratory multi-case study with 14 startups, we have found that startups can gain substantial benefits from involving potential or existing customers in their RE process and eliciting requirements from both internal and external sources. Analysing requirements effectively should involve all stakeholders. Moreover, when validating requirements, it is beneficial to closely involve end-users.

Furthermore, this study provides researchers with empirical data on the current state of RE practices in software startups and insight into the opinions of practitioners.

Future work. Our findings warrant an investigation on a larger startup population sample as it can lead to a broader understanding of RE practices and the benefits and drawbacks faced by practitioners. Moreover, future work regarding the relationship between user and system requirements and how they can be integrated in a startup that lacks technical expertise might prove important. Lastly, this research primarily studied successful startups, that is to say, startups that were still active. It would be interesting for future studies to investigate unsuccessful startups and their experience regarding RE practices and their benefits and drawbacks.

VIII. Acknowledgment

Salvatore takes the opportunity to show special regards to the family and close friends, as well as gratitude to the professors from previously attended institutions who gave encouragement in the pursuit of this education.
Olivier would like to thank his family and close friends for always being there to support him, when necessary. He would also like to thank the Computer Science and Engineering Department at the University of Gothenburg for the eye-opening experience.
Lastly but most importantly, we both thank **Jennifer Horkoff** for supervising this research and continuously giving valuable advice during our work.

References

[1] Michael Unterkalmsteiner, Pekka Abrahamsson, Xiaofeng Wang, Anh Nguyen, Syed Shah, Shahid Sohaib, Guido Baltes, Kieran Conboy, Eoin Cullina, Denis Dennehy, Henry Edison, Carlos Fernández, Juan Garbajosa, Tony Gorschek, Eriks Klotins, Laura Hokkanen, Fabio Kon, Maria Ilaria Lunesu, Michele Marchesi, and Augustin Yagüe. Software startups - a research agenda. *E-Informatica Software Engineering Journal*, 2016:89–124, 10 2016.

[2] Failory. The ultimate startup failure rate report, 2020. [https://www.failory.com/blog/startup-failure-rate], Accessed 16/06/2020.

[3] Startup Genome. The 2015 global startup ecosystem ranking, 2015. [https://startupgenome.com/blog/the-2015-global-startup-ecosystem-ranking], Accessed 16/06/2020.

[4] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: A roadmap. *Proceedings of the Conference on The Future of Software Engineering*, 03 2000.

[5] Pamela Zave. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, 29:315–321, 12 1997.

[6] Alex Yau and Christian Murphy. Is a rigorous agile methodology the best development strategy for small scale tech startups? 2013.

[7] Eriks Klotins, Michael Unterkalmsteiner, and Tony Gorschek. Software engineering in start-up companies: An analysis of 88 experience reports. *Empirical Software Engineering*, 05 2018.

[8] Nirnaya Tripathi, Eriks Klotins, Rafael Prikladnicki, Markku Oivo, Leandro Pompermaier, Arun Kudakacheril, Michael Unterkalmsteiner, Kari Liukkunen, and Tony Gorschek. An anatomy of requirements engineering in software startups using multi-vocal literature and case survey. *Journal of Systems and Software*, 146, 08 2018.

[9] Nicolò Paternoster, Carmine Giardino, Michael Unterkalmsteiner, Tony Gorschek, and Pekka Abrahamsson. Software development in startup companies: A systematic mapping study. *Information and Software Technology*, 56, 10 2014.

[10] Vebjørn Berg, Jørgen Birkeland, Anh Nguyen, Ilias Pappas, and Maria Letizia Jaccheri. Software startup engineering: A systematic mapping study. *Journal of Systems and Software*, 144, 06 2018.

[11] S. M. Sutton. The role of process in software start-up. *IEEE Software*, 17(4):33–39, 2000.

[12] Ieee standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, page 62, 1990.

[13] Pierre Bourque, Richard E. Fairley, and IEEE Computer Society. *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*. IEEE Computer Society Press, Washington, DC, USA, 3rd edition, 2014.

[14] Antonio Ghezzi, Marcelo Cortimiglia, and Rafael Bortolini. Lean startup: a comprehensive historical review. *Management Decision*, pages 1–20, 08 2018.

[15] Jorge Melegati, Alfredo Goldman, Fabio Kon, and Xiaofeng Wang. A model of requirements engineering in software startups. *Information and Software Technology*, 109, 02 2019.

[16] Jorge Melegati and Alfredo Goldman. Requirements engineering in software startups: a grounded theory approach. 06 2016.

[17] U. Rafiq, S. S. Bajwa, X. Wang, and I. Lunesu. Requirements elicitation techniques applied in software startups. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 141–144, 2017.

[18] Jessica DeCuir-Gunby, Patricia Marshall, and Allison Mcculloch. Developing and using a codebook for the analysis of interview data: An example from a professional development research project. *Field Methods - FIELD METHOD*, 23:136–155, 05 2011.

[19] Udo Kuckartz and Stefan Rädiker. *Analyzing Intercoder Agreement*, pages 267–282. Springer International Publishing, Cham, 2019.

[20] Carmine Giardino, Nicolò Paternoster, Michael Unterkalmsteiner, Tony Gorschek, and Pekka Abrahamsson. Software development in startup companies: The greenfield startup model. *IEEE Trans. Software Eng.*, pages 585–604, 2016.

[21] Catarina Gralha, Daniela Damian, Anthony Wasserman, Miguel Goulão, and João Araújo. The evolution of requirements practices in software startups. pages 823–833, 05 2018.

[22] Kangmin Lee and Joon Mo Ahn. The impacts of cross functional team on innovation performance and openness. 01 2018.

IX. Appendix

A. *Interview guide*

Table IV highlights the main interview questions and some probing questions. Nonetheless, given that the interview is semi structured, follow-up questions may be conducted to get a richer understanding.

B. *Data Collection Protocol*

Table V illustrates the purpose for the questions asked in the interview and the research questions associated with each question. Some questions (opening questions) may not be directed towards a particular research question but act as a point of context.

C. *Company Blurb*

We are two Software Engineering Bachelor students at the University of Gothenburg in our last year of study. We are writing our thesis on how software startups decide what to build and how they follow through the process of building the right product for their customers. Our goal is to get a better understanding of how practitioners like yourselves perceive this process and its components. We would like to interview you and/or your startup team(s) regarding how you manage to meet your customer demands in the context of a software startup. The interview will be anonymous which means that your personal and business related information of the startup will not be disclosed. Furthermore, the interview may be recorded so that we focus more on the content and quality of the interview however, if you wish to not be recorded, we will take notes during the interview instead. Once the interview is complete and we have completed our analysis of the inputs from all our interviewees, we will send you an e-mail with our interpretation of your interview so that you may confirm or correct our findings in case of inaccuracies. Finally, due to the current threat of Covid-19, we are happy to conduct interviews via Skype, Zoom, and the like as you prefer. We are flexible in dates and times, but we aim for interviews in March. If you like, we will also provide you with the final version of our thesis with all the summarized results of our research. We hope that it may be useful to our participants.

TABLE IV
INTERVIEW GUIDE

Question Number	Question
1	What sector(s) is your startup in?
2	What does your startup do in detail?
3	What is your role in the startup? What responsibilities do you have?
4	Are you familiar with Requirements Engineering (RE)?
5	Do you elicit requirements and if so, how are they elicited?
6	Are there any benefits/drawbacks to your method of eliciting requirements?
7	Do you analyze requirements and if so, how are they analyzed?
8	Are there any benefits/drawbacks to your method of analysing requirements?
9	Do you specify your requirements and if so, how are they specified?
10	Are there any benefits/drawbacks to your method of specifying / documenting requirements?
11	Do you validate requirements and if so, how are they validated?
12	Are there any benefits/drawbacks to your method of validating requirements?

TABLE V
INTERVIEW GUIDE RATIONALE

Question	Purpose	Reasoning
1	Opening Question	Understanding the startup from an external perspective.
2	Opening Question	Understanding the startup product.
3	Opening Question	Understand the perspective of the respondent.
4	RQ1 RQ2	Understanding of their (startup) knowledge regarding RE. Allows use to provide an example if necessary (construct validity).
5 6	RQ2	Find out if the startup uses RE practices for eliciting requirements and if they do use them, what are the benefits/drawbacks? If they don't use them (practices), why and what are the benefits\drawbacks?
7 8	RQ2	Find out if the startup uses RE practices for documenting requirements and if they do use them, what are the benefits/drawbacks? If they don't use them (practices), why and what are the benefits\drawbacks?
9 10	RQ2	Find out if the startup uses RE practices for requirements analysis and if they do use them, what are the benefits/drawbacks? If they don't use them (practices), why and what are the benefits\drawbacks?
11 12	RQ2	Find out if the startup uses RE practices for validating requirements and if they do use them, what are the benefits/drawbacks? If they don't use them (practices), why and what are the benefits\drawbacks?

TABLE VI
INTERVIEWS: REQUIREMENT ELICITATION - BENEFITS AND DRAWBACKS

Category	**+/-**	**Practice**	**+/-**
Internal	N/A	Colleagues feedback	(+) Eliciting from members with product experience is great when product hasn't reached the market yet (-) Too many members being elicited becomes hard to track
		Requirement from management	(+) Eliciting requirements from management is good because it's their idea (-) Management may not always know what the users want (-) Management frequently changes requirements which may reduce productivity (-) Subjectivity of management may not represent customers (-) If Management imagination is not similar to customer segment, then elicitation can be misleading
External	N/A	Market research	[s] (+) Essential in the educational sector
		Existing/Potential customer interview	(+) Requirements come straight from the people that will use our products (+) You build something that someone is willing to pay money for
		Meetings with multiple customers	(+) Meetings allow for a broad understanding of customer demand
		Competitor Analysis	(+) Good form of inspiration

TABLE VII
INTERVIEWS: REQUIREMENTS ANALYSIS - BENEFITS AND DRAWBACKS

Category	+/-	**Practice**	+/-
Estimation	(-) Estimations can be inaccurate	Time Estimation	(-) Difficult to estimate and may not be accurate (+) Flexibility of startup allows for easy adjustments
		Complexity Estimation	(+) Provides trade-off for how much to implement
Understanding	N/A	MoSCoW method	(-) MOSCOW too simplistic (+) MOSCOW helps to get started
		Five Whys	(+) Helps with understanding purpose of requirement
Intuition	N/A	Management Decision	(+) Time to make decision is faster (-) Freedom and speed to make decisions may lead to wrong decisions and leading to wasted work (-) Priority of requirements may constantly change causing teams to not be less productive (-) Management with low technical depth may estimate efforts and deadlines inaccurately (-) Lack of requirement negotiation between management and dev team leads to poor estimations
		User Flow	(+) Allows for a better understanding of which order the features should be developed

TABLE VIII
INTERVIEWS: REQUIREMENTS SPECIFICATION - BENEFITS AND DRAWBACKS

Category	**+/-**	**Practice**	**+/-**
Communication platforms	(+) Easy and quick way to specify requirements	Slack	(-) Requirements will get lost in the chat (-) Requirements are easily forgotten
	(-) Hard to manage and track requirements	Word of mouth	(-) Continuous change of specifications to due word of mouth (-) Word of mouth specifications get lost (-) Word of mouth is hard to follow because acceptance criteria is not clear
Project management tools	(-) Hard to follow with too many types of documentation.	Jira	(+) Provides structure (-) Hard to maintain without a dedicated person
	(-) Using multiple platforms at the same time is difficult to follow and maintain.	Trello	(-) Hard to maintain without a dedicated person
	(-) Hard to change tool later in development.	GitHub	(+) Documentation near code [GitHub projects] is more accessible
Artifacts		Excel fie	(-) Difficult to read specifications on tools hardly accessible
	(+) Easier to specify requirements in a method that everyone agrees with.	Diagram	(-) Specifications from BA are not developer-friendly which makes it hard to read and understand (+) Diagrams are more clear than word of mouth because clearer(-) Hard to maintain it gets outdated when requirements are updated
	(-) If team members don't use specifications, acceptance criteria are missed.	Gherkin feature file	(+) Provide active feedback on implementation and easy to unit test
	(-) Vague requirements means they are forgotten or developers implement different interpretations	User story	(-) User stories with one sentence are very ambiguous and cause confusion (need for constant clarification) (-) User stories without detail do not provide big picture which makes it harder to understand larger goal
		.CDS file	(-) CDS becomes too long (-) CDS is hard to structure
		Existing system on different platforms	(+) Referring to the system on different platform meant less confusion
		Whiteboard	[s] (+) Facilitates general understanding during meetings

TABLE IX
INTERVIEWS: REQUIREMENTS VALIDATION - BENEFITS AND DRAWBACKS

Category	**+/-**	**Practice**	**+/-**
Manual Validation	(+) Verbal/trust creates less friction at work (+) Good enough when in early stages of development because of features are basic (+) More formal validation not needed because system is not critical	Internal demo	(-) Some features may be on different platforms, making it hard to verify whether they have been met correctly on all of them (-) If management requests demos frequently, it leads to pressure and lower productivity (+) Demo is fast
	(-) Manual verification can miss features out of scope (-) Too much reliance on beta testers showing up most of the bugs	Comparison with same system on existing platform	(+) Validating via existing platform meant less confusion
	(-) Testing takes a lot of management work hour (-) Hard to verify with off-shore dev-team	Feedback from end users	(+) Makes better requirements elicitation
Automated Validation	(-) Hard to follow with too many types of documentation (-) Using multiple platforms a the same time is difficult to follow and maintain	Unit test	(-) Test scripts need general effort to work (+) Test scripts on user stories good way of working (-) Not used by developers

TABLE X
SURVEY: RE PRACTICES USAGE FREQUENCY

RE stage	Practice	Not Used	Rarely	Sometimes	Often	Always	Average
Elicitation	Colleagues feedback		1		1	3	4.2
	Requirements from management		2	1	2		3.0
	Market research		1		1	3	4.2
	Existing/Potential customer interview	-	-	-	-	-	-
	Meetings with multiple customers	-	-	-	-	-	-
	Competitor Analysis		1	1	2	1	3.6
Analysis	Fibonacci estimations	3	1		1		1.8
	Time estimation				1	4	4.8
	Effort estimation				2	3	4.6
	Complexity estimation				3	2	4.4
	Value estimation				1	4	4.8
	Hard and soft requirements			1	1	3	4.4
	UX, Front-end, Back-end				3	2	4.4
	MoSCoW method	4			1		1.6
	Five Whys	2		1	2		2.6
	User flow				3	2	4.4
	Management decision			1	3	1	4.0
Specification	Email			1	2	2	4.2
	Slack	1			1	3	4.0
	Word of mouth			2	2	1	3.8
	Jira		1		2	2	4.0
	Trello	1	1	1	1	1	3.0
	Github	1			1	3	4.0
	Asana	4	1				1.2
	Confluence	2	1		1	1	2.6
	Excel file		2	1	2		3.0
	Diagrams		1		2	2	4.0
	Gherkin feature file	4	1				1.2
	User story		1		1	3	3.2
	CDS file	4	1				1.2
	Existing system on different platform	2	1	1		1	2.4
	Word file	1	1	1	1	1	3.0
	Whiteboard		1	1	1	2	3.8
Validation	Internal demo			1	1	3	4.4
	Beta testers	1			2	2	3.8
	Comparison with existing system on different platform			1	3	1	4.0
	Feedback from end user			1	1	3	4.4
	Unit testing			2	1	2	4.0
	Regression testing	2		1	1	1	2.8

TABLE XI
SURVEY: SATISFACTION WITH RE PRACTICES

RE stage	**Practice**	**Not Used (-)**	**V. Dissatisfied (1)**	**Dissatisfied (2)**	**Neutral (3)**	**Satisfied (4)**	**V. Satisfied (5)**	**Average**
Elicitation	Colleagues feedback					2	3	4.6
	Requirements from management				2	2	1	3.8
	Market research				1	1	3	4.4
	Existing/Potential customer interview	-	-	-	-	-	-	-
	Meetings with multiple customers	-	-	-	-	-	-	-
	Competitor Analysis(s)				1	2	2	4.2
Analysis	Effort estimation	4			1			3.0
	Time estimation				1	2	2	4.2
	Effort estimation			2	1	1	1	3.2
	Complexity estimation			1	1	1	2	3.8
	Value estimation				1	2	2	4.2
	Hard and Soft requirements		1		1		3	-
	UX, Front-end, Back-end				1	1	3	4.4
	MoSCoW method	5						-
	Five Whys	3		1	1			2.5
	User flow	1			2		2	4.0
	Management decision				3	1	1	3.6
Specification	Email				2	2	1	3.8
	Slack	2			1	1	1	4.0
	Word of mouth			2	1	2		3.0
	Jira				2	2	1	3.8
	Trello	1		2	1		1	3.0
	Github	1			1		3	4.5
	Asana	5						-
	Confluence	3			1		1	4.0
	Excel file	2	1		1	1		2.66
	Diagrams	1				1	3	4.75
	Gherkin feature file	5						-
	User story				2	1	2	4.0
	CDS file	5						-
	Existing system on different platform	3			1	1		3.5
	Word file	1	1		1	2		3.0
	Whiteboard	1			2		2	4.0
Validation	Internal demo				1	2	2	4.2
	Beta testers	1			1	1	2	4.25
	Comparison with existing system on different platform				2	3		3.6
	Feedback from end user				1	2	2	4.2
	Unit testing			1	1	1	2	3.8
	Regression testing	2		1	1		1	3.33

TABLE XII
SURVEY QUESTIONS

ID	Question	Type	Rationale
S1.1	What area does your startup operate in?	Open	Demographic
S1.2	What role do you have in the startup?	Open	Demographic
S1.3	How old is the startup?	Open	Demographic
S1.4	What is the size of the startup?	Open	Demographic
S1.5	Has your product reached the market?	Open	Demographic
S[2-5].1	How often are the following RE practices adopted at work? [Multiple Choice Matrix]	Closed	R.Q 1
S[2-5].2	If used, how satisfied are you with the practices listed? [Multiple Choice Matrix]	Closed	R.Q 2
S[2-5].3	If the practices used at your startup are not listed above, could you list them?	Open	R.Q 1
S[2-5].4	Can you provide some rational for why you are satisfied or very satisfied with your current practices?	Open	R.Q 2
S[2-5].5	Can you provide some rational for why you are dissatisfied or very dissatisfied with your current practices?	Open	R.Q 2