Understanding Hypotheses Engineering in Software Startups through a Gray Literature Review

Understanding Hypotheses Engineering in Software Startups through a Gray Literature Review

Jorge Melegati[a,*], Eduardo Guerra[a], Xiaofeng Wang[a]

[a]*Free University of Bozen-Bolzano, Piazza Domenicani 3, Bolzano, Italy*

Abstract

Context: The higher availability of software usage data and the influence of the Lean Startup led to the rise of experimentation in software engineering, a new approach for development based on experiments to understand the user needs. In the models proposed to guide this approach, the first step is generally to identify, prioritize, and specify the hypotheses that will be tested through experimentation. However, although practitioners have proposed several techniques to handle hypotheses, the scientific literature is still scarce. **Objective:** The goal of this study is to understand what activities, as proposed in industry, are entailed to handle hypotheses, facilitating the comparison, creation, and evaluation of relevant techniques. **Methods:** We performed a gray literature review (GLR) on the practices proposed by practitioners to handle hypotheses in the context of software startups. We analyzed the identified documents using thematic synthesis. **Results:** The analysis revealed that techniques proposed for software startups in practice compress five different activities: elicitation, prioritization, specification, analysis, and management. It also showed that practitioners often classify hypotheses in types and which qualities they aim for these statements. **Conclusion:** Our results represent the first description for hypotheses engineering grounded in practice data. This mapping of the state-of-practice indicates how research could go forward in investigating hypotheses for experimentation in the context of software startups. For practitioners, they represent a catalog of available practices to be used in this context.

Keywords: hypotheses engineering, software startups, gray literature review
2010 MSC: 00-01, 99-00

1. Introduction

Traditionally, software engineering is divided into activities, namely, requirements engineering, design, construction, testing, and maintenance. As an example, we can cite the *Guide to Software Engineering Body of Knowledge* [1]. The guide organized the contents of software engineering in 15 knowledge areas, and the first five referred exactly to these steps. Even in the agile methodologies that emerged as a defiant to traditional software engineering methods [2], one can identify these phases in a more iterative and fast way. For instance, we can mention studies on agile requirements engineering (e.g., [3]) and software design in agile development (e.g., [4]).

Recently, a new approach came to complement this culture: experimentation. Bosch et al. [5] argued that there are three different and complementary approaches to software development: requirement-driven, outcome/data-driven, and AI(Artificial Intelligence)-driven. The former approach is characterized by developing software based on defined specifications. In the second, the team focuses on a metric and develops experiments with different approaches to improve it. Finally, in the AI-driven approach, AI techniques are employed to create components based on data collected from previous interactions with the system.

Although Bosch et al.'s outcome/data-driven software development approach is focused on quan-

*Corresponding author
Email addresses: jmelegatigoncalves@unibz.it (Jorge Melegati), eduardo.martinsguerra@unibz.it (Eduardo Guerra), xiaofeng.wang@unibz.it (Xiaofeng Wang)

titative metrics, we can also draw parallels to qualitative techniques employed in developing new products such as problem and solution interviews. The union of quantitative and qualitative methods to guide software development represents an experiment-driven approach, generally referred to as experimentation. It is a process of continuously validating product assumptions, transforming them as hypotheses, prioritizing, and testing them following the scientific method to support or refute them [6]. It comprises several techniques including prototypes, controlled experiments [7], and problem or solution interviews [6].

In software engineering literature, there are some models to guide development following an experimentation approach, e.g., HYPEX (Hypothesis Experiment Data-Driven Development) [8] and RIGHT (Rapid Iterative value creation Gained through High-frequency Testing) [9]. In a previous position paper [10], we analyzed several of these models and draw a parallel between the conventional requirement-driven approach and the experiment-driven one. While the former is based on requirements engineering, design, coding, and testing, the latter is based on identifying, prioritizing, and specifying hypotheses, designing experiments to test them, executing the experiments, and analyzing the results. Based on that, we argued the need for a knowledge area describing how to properly handle hypotheses in a similar way to requirements engineering in conventional approaches, and termed it *Hypotheses Engineering (HE)*. After its publication, the concept has been used to frame an approach targeted to uncertain customer needs [11] and discuss the role of experiments in software organizations [12]. In the same paper, we envisioned some HE activities based on the requirements engineering counterparts: generation, documentation, analysis, and prioritization. However, we based the activity set proposed on speculation without grounding on how software development teams behave.

This paper aims to ground HE in current practice by describing how practitioners understand the concept of hypotheses and identifying types of proposed activities to handle hypotheses. As an initial context to explore, we chose software startups. In recent years, the startup community has been heavily influenced by several popular methods for startup development, e.g., Customer Development Methodology [13] and Lean Startup [14], which are hypothesis-driven and experimentation centric approaches [15, 16]. Software startups are particularly susceptible to these approaches due to the lower cost of running experiments than other types of startups. Therefore, we consider software startups a rich context in instances of the research phenomenon we are interested in investigating. Additionally, narrowing the goal to a specific context allows the research to go deeper through analyzing a smaller data set.

To reach our goal, we performed a gray literature review (GLR) to map HE practices proposed for software startups. Since there is a large practitioner-produced literature for and read by the startup community, analyzing this content represents a more straightforward approach than more direct methods like surveys or interviews with practitioners. Besides that, we could obtain a better reach since online articles are available to a broader audience than the one we could access with a survey. This overview could later be the basis for research employing more direct methods. Through a Web search and applying defined inclusion and exclusion criteria, we identified 95 primary documents and analyzed them using thematic synthesis. As the results, we reached a model describing how practitioners define the concepts of hypotheses and assumptions, its qualities and types, and which activities related to hypotheses they perform. These activities are elicitation, prioritization, specification, management, and analysis.

The remaining of this paper is organized as follows: Section 2 presents the background and related work. In Section 3, we present the research method employed and, in Section 4, the results, which are discussed in Section 5. In Section 6, we present lessons learned from applying GLR in the study. Finally, Section 7 concludes the paper.

2. Background and related work

2.1. Experimentation and hypotheses

In the Software Engineering research, the word "experimentation" has several meanings. The first use that is still referred to nowadays (e.g., [17]), is the use of experiments as a research methodology for scientific inquiries in the Software Engineering research. Basili et al. [18] published an early paper in this regard. Similarly, Wohlin et al. [19] argued the need for systematic scientific research through the use of experiments to investigate phenomena related to software engineering and support the claims made. Additionally, they provided

guidelines to guide researchers while performing experiments in this context.

More recently, the term has been used to describe the use of several techniques to guide the implementation or improvement of software product features based on the evaluation of how (potential) users react to them (e.g., [6, 9]). We adopt this definition in our study. A frequent example of experimentation is A/B testing. In these tests, two different versions of a website feature are implemented, and users are split between the two versions. Data from a defined user action is collected from the two versions and compared to determine which one is better. Several other techniques are included in the context of experimentation, such as iterations with prototypes, gradual roll-outs [7], and problem and solution interviews [6]. Another common term often used in literature is "continuous experimentation" (e.g., [20, 21]), where researchers are generally concerned with controlled experiments and the term "continuous" stresses the repeated use of them during the development process.

The research on experimentation has increased lately [22]. Several studies focused on how companies employed the concept [6, 20, 23, 7]. A common theme discussed in these studies is what prevents companies from further using experiments. Lindgren and Münch [6] investigated ten Finnish software development companies and concluded that, although the concept resonated well with practitioners, they were not using it so often. They found that such limitation was a consequence of issues on the organizational level rather than technological challenges.

Similarly, in a multi-method study, Schermann et al. [20] concluded that, from a process perspective, many organizations perform experiments based on intuition rather than a defined process. In the context of software startups, Gutbrod et al. [23] investigated four German companies. The authors concluded that these companies spent a large amount of time developing the solutions instead of experimenting with potential customers. The main reasons are the lack of knowledge of such possibility and support for identifying, prioritizing, and testing critical assumptions.

In this regard, several authors in scientific literature proposed models to guide experimentation, including HYPEX, QCD (Qualitative/Quantitative Customer-driven Development), and RIGHT. While HYPEX still reflects the requirements-driven approach to a certain extent, the other two models put experimentation and hypothesis in the center, right from the beginning.

Olsson et al. [8] proposed the HYPEX model to make it faster for product management to get feedback from users. The model is divided into six steps, and only in the fifth step appear hypotheses. In the first step, feature backlog generation, the team generates ideas for features that could bring value for the customer, and that should be experimented. In the second, feature selection and specification, the team selects "the highest priority" one for implementation and specifies how it provides value to the user and its business goals. In the third, implementation and instrumentation, the team identifies the minimum slice of the feature that adds value to customers, the so-called Minimum Viable Feature (MVF), implements it, and adds elements to collect usage data. In the fourth, gap analysis, the teams compare the collected data with the expected behavior. Based on that, in the fifth step, hypothesis generation and selection, the team develops hypotheses that could explain differences observed between the expected and actual behavior. If the most probable hypothesis in the previous step is that the feature does not add value, the team moves to the sixth practice, alternative implementation, to build a different implementation.

In contrast, in the QCD model, Olsson and Bosch [24] contended that "requirements are treated as hypotheses that are validated with customers before forming the basis for development." The model is defined by a cycle that starts with hypotheses being selected from a backlog developed with several qualitative and quantitative techniques. Then, a Customer Feedback Technique (CFT) is chosen among qualitative or quantitative techniques. The chosen technique is executed, and the data collected is used to close the loop by updating the hypotheses in the backlog or creating new ones.

Similarly, Fagerholm et al. [9] proposed the RIGHT model for continuous experimentation. It consists of an infinite series of Build-Measure-Learn cycles, where the pieces of learning obtained in one cycle feed the next one. Each cycle contains the following steps: identify and prioritize hypotheses, design an experiment, execute it, that is, deploy an MVP (Minimum Viable Product) or MVF, analyze the data gathered in the experiment, and decide to pivot or persevere.

When comparing these models, a clear pattern emerges: all of them follow a cyclical approach where each cycle consists of a similar sequence of

steps. This sequence is composed of identifying, prioritizing, and specifying hypotheses, designing an experiment, executing it, analyzing the data, and creating or updating hypotheses [10]. There is a clear parallel to the steps commonly used to describe software development: requirements engineering (RE), design, coding, testing, and maintenance. Based on this parallel, in a previous position paper [10], we proposed the Hypotheses Engineering (HE) area defining activities to handle hypotheses in experiment-driven software development in a similar way RE handles requirements in requirements-driven software development.

Since there is some confusion about the terms hypothesis and assumption, it is essential to make explicit how we will use them in this text. "Assumption" is a personal or team-wise, generally implicit, understanding taken as truth without being questioned or proved, and "hypothesis" as an explicit statement that has not been verified yet, but to which an experiment could be implemented to evaluate. We had previously used this definition ([25]), and it is in line with other authors in the literature. For instance, Lindgren and Münch [6] considered that "the assumptions to be tested need to be transformed into falsifiable hypotheses."

2.2. Hypotheses Engineering activities

Based on the parallel with requirements, a set of activities that should be included in HE are envisioned in our previous paper [10], namely, generation, documentation, analysis, and prioritization. However, they are yet to be validated by empirical evidence on how practitioners handle hypotheses in experimentation-driven software development.

The scientific research on hypothesis-related practices is still scarce. Gutbrod et al. [26] proposed the Business Experiments Navigator (BEN) as a tool to map assumptions to experiments and design and select experiments. To argue the need for this tool, the authors reviewed some industry techniques, namely, Assumption Mapping, Prioritization Matrix, Lean Canvas Prioritization, Prioritizing Leap-of-Faith Assumptions Matrix, Question Matrix, Testing Process, and Rapid Experiment Loop. According to the authors, all practices, except for Question Matrix, describe how assumptions could be collected and risk prioritized.

Regarding the Leap-of-Faith Assumptions Matrix, Gutbrod and Münch [27] evaluated how a group of students prioritized a pre-defined set of assumptions about a business model. They concluded that the technique proposed led to reasonable results and, consequently, a learning effect on students.

Regarding hypothesis specification, we proposed the QUESt criteria to improve the quality of hypotheses [28]. We argued that a hypothesis should have a Questioning sense, be Updatable, Evaluable, and Straightforward. Besides that, based on conventional templates for user stories, we proposed a template to specify hypotheses: "If a <role> wants/prefers to <action/characteristic> then <evaluation process> should <evaluation result>." We performed an initial evaluation of the proposed practices with five practitioners reaching promising results.

2.3. Hypotheses Engineering in software startups

Software startups research suffers from a lack of a consistent definition of its object of study. This problem is linked to the fact that, although a commonly used term, "startup" does not have still a clear definition. Practitioners may use it to term their new endeavors simply because of its lack of history. Another motivation could be to give the impression of a technology-savvy project, which brings more attention to the company or product. In the first systematic mapping study (SMS) on the topic published in 2014, Paternoster et al. [29] analyzed 43 studies, including how authors defined software startups. They concluded that there was not an agreement on a standard definition. Therefore, they collected the current themes used to describe the startup context. The list contained 15 items and the most frequent were: lack of resources, highly reactive, innovation, uncertainty, rapidly evolving, and time pressure. In 2018, Berg et al. [30] published a new SMS on the topic. They identified 27 new studies in the period 2014-17. Although they concluded that the scientific rigor increased in the period compared to previous studies, they observed that the lack of a standard definition of startup persisted. They performed an analysis of themes used to define startups similar to what was done in the previous SMS. Their final list consisted of 13 items where the most recurrent themes were: innovation/innovative, uncertainty, small team, lack of resources, and little working/operating history.

Given the lack of a consensus, it is essential to specify the definition of startup we are using. In this study, we define a startup as an organization

that develops innovative software-intensive products, constantly searching for a repeatable and scalable business model. This definition stress that innovation is a key aspect of software startups. Innovation leads to uncertainty in new product development [31], and experimentation is described as an essential element in innovation literature (e.g., [32]). The entrepreneurship literature also advocates it (e.g., [33]). Although not explicitly based on previous research, the most well-known startup development methodology among practitioners, Lean Startup [14], is influenced by experimentation. Several authors [15, 16] investigated the scientific concepts of the Lean Startup approach and emphasized the influence and importance of the experimentation concept. Since many software startups apply Lean Startup [34], they represent a proper and rich context to investigate Hypothesis Engineering.

Previously, we proposed practices to elicit hypotheses in the context of software startups [25]. We investigated where assumptions founders base their product come. We concluded that founders develop an understanding of the market and customers based on their previous experiences and use it to predict how a new product could be developed. Therefore, we used cognitive mapping to elicit this understanding in a graphical form and create hypotheses to be tested. We performed an initial evaluation in two case studies and had promising results.

In summary, scientific research on hypothesis-related practices for experimentation is still in its infancy. So far, there is no clear, comprehensive framework to classify practices, and the proposed techniques are limited to specific activities. Nevertheless, a quick search on the Internet shows that there is a large amount of practitioners-proposed techniques. Such a plethora of practices could serve as a starting point, especially in the context of software startups, for which experimentation is deemed a valuable approach.

3. Research method

3.1. Research questions

To guide this study, we conceived the following research question:

RQ: How is Hypotheses Engineering defined in the context of software startups?

As we observed in the studies published so far, there are diverse goals in the activities employed to handle hypotheses. For instance, elicitation or generation techniques to help practitioners find out which hypotheses the experiments should be built to test (e.g., [25]), or prioritization techniques to facilitate the definition of the testing order [27]. Nevertheless, there is not a clear set of activities regarding hypotheses defined in the literature. Therefore, our first goal is to map the activities which practitioners deem necessary for hypotheses. In this regard, the first sub-research question is:

RQ1: What are the proposed activities of Hypotheses Engineering in the context of software startups?

Then, we will be able to categorize the identified techniques accordingly. Thus, the second sub-research question is:

RQ2: What are the proposed techniques for each Hypothesis Engineering activity in the context of software startups?

3.2. Gray literature review

In their guidelines for conducting GLR, Garousi et al. [35] proposed a list of questions that, if at least one answer were yes, it would indicate the suitability of including gray literature in the review. Regarding this study, four out of the seven questions are answered affirmatively. First, the subject is complex and not solvable, considering only the formal literature and, second, the volume of evidence in formal literature is low. Such aspects were based on the related work presented in Section 2. Such scarcity is a consequence of the novelty of the field and its complexity, as we can observe from each paper's focus in a different aspect. Third, the synthesis of insights and evidence will be useful to industrial and academic communities. To the former, it will give a classification and benchmarks to guide future research and, to the latter, provide a catalog of available practices and a comparison among them. Fourth, the large volume of practitioner sources indicates a high interest in the topic. Therefore, we conducted a GLR based on Garousi et al.'s guidelines [35].

3.3. Documents selection

Since startup practitioners often describe their experiences and search for information online, gen-

erally through blog posts or other electronic available resources, it is reasonable to collect and analyze this type of gray literature. Then, a natural tool for search this literature was the Google web search engine.

To develop the search string, we followed the PICO acronym, generally used in systematic mapping studies [36]. It stands for Population, Intervention, Comparison, and Outcomes. Regarding population, our focus was software startups. Nevertheless, to have a broader set of results, we utilized the term "startup", and the often employed variation "start-up." Several authors performing searches on scientific (e.g., [29]) and gray literature (e.g., [37]) also made the same decision. Besides that, previous studies that searched practitioner-produced literature on the Web about startups (e.g., [37] and [38]) used the other synonyms: "early-stage firm", "early-stage company", and "venture." Regarding the intervention, a natural choice is to search for "hypotheses." As mentioned in Section 2 and observed during initial exploration, a commonly used synonym was "assumption." Since we were not concerned if the authors compared or, even, evaluated their proposals, we did not add query elements regarding comparison and outcomes.

Instead of putting all the synonyms in one search string, as the previous studies have done, we split the synonyms for startup in different search strings. In this way, we wanted to avoid that a page using more than one synonym was considered more relevant by the search algorithm instead of another page that used only one term but is more suitable because of other factors. For instance, a page using the word "venture" and "startup" could have precedence of one using only startup when, in reality, that could not be the case if, for example, the second one describes better an activity. Therefore, we performed searches on Google with four different query strings: one for each synonym and the corresponding spelling variations. We executed the queries in June 2020, in the international site of Google[1] using the Chrome browser on an anonymous tab to avoid the influence of the saved history and cookies in the search results. We configured the Google search engine to return 100 results per page and employed SEOQuake[2] plugin to download the results to a spreadsheet. Table 1 summarizes all the queries performed, including the number of results. We did not perform separated queries for the dichotomy hypothesis-assumption because these terms are often used as synonyms.

Once we collected all the links, one researcher accessed and inspected all the links applying the following inclusion and exclusion criteria:

Inclusion criteria:

- Description of how to perform any activity related to hypotheses.
- The statement should be targeted to startups.

Exclusion criteria:

- Not written in English.
- Simple summaries of books or other frameworks that do not present any new proposal or description of the practice use.
- Not text-based content like videos or slide presentations.
- Duplicated content.

In this step, we removed articles regarding large companies. For instance, we discarded a text[3] on using hypotheses in SAFe, a framework for scaling agile to large companies.

After that, we removed duplicated content. Such a step was necessary because seldom the same text is repeated on different websites and emerged more than once in the search results. We applied the inclusion and exclusion criteria on all the links before removing duplicate content, because comparing text that would not be included would be pointless. The amount of data analyzed prevented the researcher from excluding duplicates (that do not share the same URL) in the first scan of the results.

Then, a second researcher reviewed all the selected links to verify if they really should be included. In the case of disagreement, both researchers discussed if the link should be included or not. In this process, six documents were excluded. After that, we conducted a snowballing approach, following links that authors used to back their arguments or suggested further reading. In this step, we collected eight new documents, for a total of 95 documents. All documents were saved in PDF format.

[1] https://www.google.com/
[2] https://www.seoquake.com/index.html
[3] https://www.scaledagileframework.com/guidance-applied-innovation-accounting-in-safe/

Table 1: Query strings of searches performed on web.

Term for startup	Query string
startup	*(hypothesis OR hypotheses OR assumption) AND (startup OR start-up)*
early-stage firm	*(hypothesis OR hypotheses OR assumption) AND ("early stage firm" OR "early-stage firm")*
early-stage company	*(hypothesis OR hypotheses OR assumption) AND ("early stage company" OR "early-stage company")*
venture	*(hypothesis OR hypotheses OR assumption) AND venture*

Some information was extracted, including title, author, year, contribution type (proposal, experience report, etc.), and the author's background. In case it was signed by a company, information about it as well. The document selection is summarized in Fig. 1.

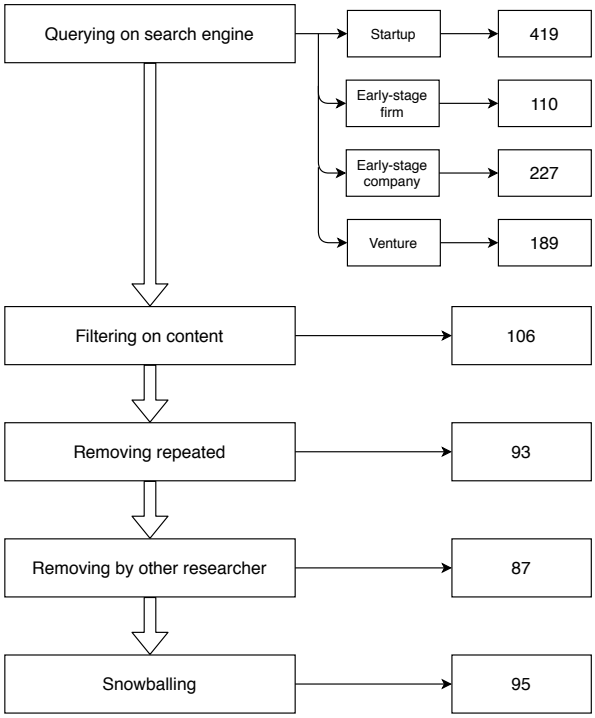

Figure 1: Documents selection

3.4. Data analysis

To analyze the data, we used thematic synthesis, "a method for identifying, analyzing, and reporting patterns (themes) within data" [39]. This definition is aligned with the research goal of identifying the current practices performed in the industry. Besides that, the method has the advantage of combining and organizing results from a large and diverse body of research [39].

For thematic synthesis, we followed the steps proposed by Cruzes and Dybå [40]. In the first step, data extraction, the researcher reads the primary studies to get immersed in data. Details of the publications are also extracted. For our study, these steps were performed earlier in the process of document selection. Besides that, we followed the authors' suggestion, and another researcher checked the extracted data.

The second step consists of coding data when interesting concepts are systematically identified across the entire data set [40]. Since the research goal was to identify the techniques described in practice, a suitable choice was to perform an inductive approach where codes emerged from data. Nevertheless, in the original proposal of HE, we had already proposed categories of HE activities. Therefore, we employed an integrated approach [40], where the proposed activities acted as an initial list of themes and helped the researchers indicating areas where codes could be inductively generated. We performed this step and the following ones using NVivo 12[4].

In the next step, codes are grouped in subthemes, themes, and high-order themes. The codes and themes are compared and analyzed regarding coherency, consistency, distinctiveness [40]. That is, a theme should represent one and only one concept, and the overlapping between themes should

[4]https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home

be minimized. This step is performed iteratively: new themes are created, others are removed, and some are merged.

Then, the researchers must explore possible relationships among high-level themes to reach a model. Finally, the trustworthiness of the synthesis is assessed, including if the model obtained is coherent with the evidence. On all steps, another author performed a checking process, reviewing all excerpts coded and how they were grouped in themes. In case of conflict, both authors discussed to reach a consensus.

4. Results

The 95 documents had 83 unique authors, or authoring team, of which 12 were companies and 71 people. In this analysis, when a document had more than an author, we considered them as one. We classified the 83 authors regarding the roles they informed to perform. Since one person could act with different roles like an investor who also gives training to startups acting as an instructor, the total sum is more than the number of unique authors. The most common role with 39 writers is consultant; that is, these authors provide services to help startups developing their products. They generally claim this capacity based on previous experience as practitioners or in accelerators. Their writings portray techniques they employ and advice for startups. The second most common role, with 20 occurrences, is practitioner, that is, startup founders that describe techniques they use. Then, for ten writers, we classified them as authorities that include book authors like Alexander Osterwalder, author of the book *Business Model Generation* that introduced the Business Model Canvas, and Ash Maurya, author of the book *Running Lean* and others. They are a reference to the practitioners' community. Other authors were instructors, professionals who train startups, with seven occurrences, accelerators, or incubators with seven other occurrences, six investors, and two authors from academia.

To classify the contribution type, we used the research type as proposed by Garousi et al. in a multi-vocal literature review on software testing automation [41]. The authors described six types of research contributions:

- Solution proposal: the statement proposes a solution and claims its effectiveness based on simple examples or a line of argumentation.
- Validation research: compared to a weak empirical analysis, these studies present initial empirical evidence regarding the solution effectiveness.
- Evaluation research: compared to a strong empirical analysis, these studies apply "strict and formal experimental method."
- Experience studies: these statements report how the activities have been used in practice.
- Philosophical studies: these reflections propose new structures to view the current practices through, e.g., a new taxonomy or a conceptual framework.
- Opinion studies: these statements represent the authors' opinions regarding practices without proper backing in related work or empirical evidence.

There was also another category to collect those studies that did not fit the previous categories. In our analysis, regarding experience studies, we differentiated those statements about practitioners and those in an educational context that employed practices with students. Table 2 presents the classification results. The majority of the articles (66) were blog posts or other web pages presenting a description or discussion of a proposed solution. In 11 documents, the authors presented their opinion without empirical evidence. Then, we had ten experience reports with startups and other two with students. Finally, in six documents, the authors performed validation research. We also extracted the source publication year, as depicted in Fig. 2. It was not possible to get this information for 17 articles. We believe authors often do not make this information available to avoid the impression of a dated statement.

Table 2: Number of documents by contribution type.

Contribution type	Number of documents
Solution proposal	66
Opinion studies	11
Experience reports	10
Validation research	6
Experience with students	2

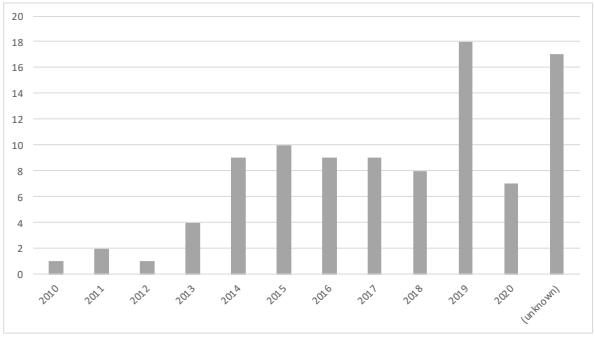

Figure 2: Number of sources by publication year.

The thematic analysis led to a model where the hypothesis is the central theme. Although there was a varied focus and depth of the practices described in the documents, five categories of hypotheses related activities emerged as high-order themes: elicitation, prioritization, specification, analysis, and management. Besides them, another two high-order themes are linked to hypotheses as well: qualities and types. Fig. 3 displays the model that emerged from the data. In the following sections, we describe each high-order theme in detail. As proposed by Cruzes et al. [39], we also use quasi-statistics to bring forward the most frequently occurred practices for each category.

4.1. Definition of hypotheses

In this category, we discuss how practitioners defined the concepts of hypothesis and assumption and how they understood the difference between these concepts. In 16 documents, the authors defined the concept of hypothesis. Present in six documents, the most common definition describes a hypothesis as an "educated guess". This usage is influenced by the definition used by Ries in the Lean Startup book [14]. The second most used definition, present in four documents, describes a hypothesis as the starting point of an investigation. Then, with two mentions each, there are hypothesis as an expectation for the future (here one of the documents also used as the starting point of an investigation), a possible solution to a problem, and a possible explanation for a phenomenon. Finally, one document described a hypothesis as a stage in startup development.

Regarding assumption, two different definitions were suggested in the documents. First, mentioned in six documents, an assumption is described as a statement believed to be true. Then, assumptions are described in three documents as statements that should be true for the idea to work.

Interestingly, there was an attempt to compare the two terms in ten documents. In four of them, assumptions are described as implicit and hypotheses as explicit statements used in the experiment creation. In another three documents, assumptions are suggested as not validated statements, but which risk is accepted. Meanwhile, hypotheses should be tested through experiments. Instead, in the last three documents, the two terms were regarded as synonyms.

4.2. Qualities

Within this theme, we grouped the qualities that practitioners deemed relevant for a hypothesis. The most frequently mentioned aspect was the possibility to test the hypotheses, present in 17 documents. Termed as "testable", this quality suggests that hypotheses should allow tests to be performed to evaluate them. Tristan Kromer [G1] gave an example: *" I can't test whether two magnets dislike each other. I can test if they will physically move away from one another."*

Then, 16 documents mentioned that hypotheses should be **specific**; that is, they have to regard only one aspect for a specific customer type and segment. Cecilia Thirlway [G2] wrote *"Don't be tempted to create a hypothesis that covers more than one aspect of your innovation. While it may feel like it saves time, in actual fact you'll be unable to distinguish which aspect has caused the results of your experiment."*

In 15 documents, the authors mentioned that hypotheses should be **falsifiable**. A hypothesis being falsifiable is related to the possibility of proving that it is wrong. Alex Chuang [G3] wrote *"a good product hypothesis: is falsifiable, which means it can clearly be proven wrong."* In this regard, two blog posts from consulting companies, MaRS [G4] and Bullet [G5], mentioned the philosopher Karl Popper that argued the falsifiability for scientific theories.

Measurable as a trait of a hypothesis is mentioned in 15 documents. It argues that metrics should be indicated, and their analysis could support or refute a hypothesis. Mark Lieberman [G6] mentioned: *"[...] all hypotheses should be quantifiable. In other words, you must be able to predict, account, and analyze your results. A good hypothesis includes both a question and good methodology to uncover the results."*

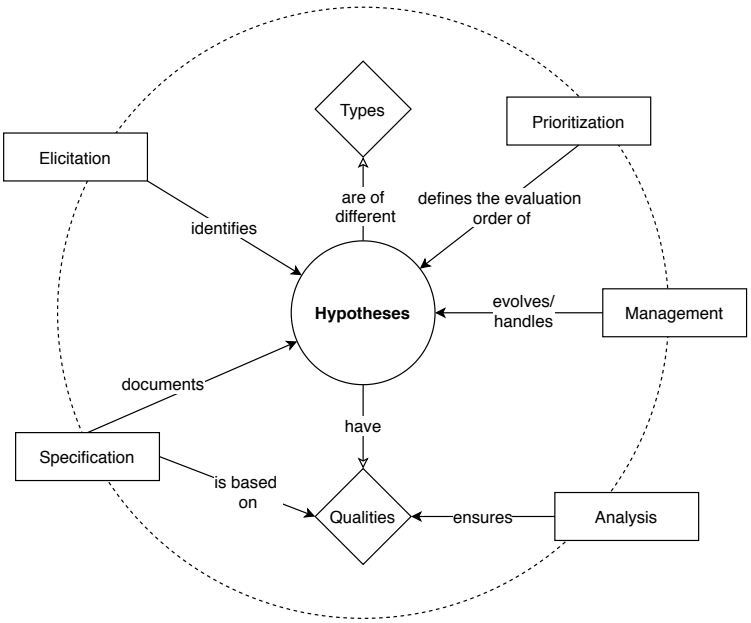

Figure 3: A model of hypotheses engineering in startups.

Then, seven documents suggest that hypotheses should be **time-boxed**; that is, they should specify a time interval to be evaluated. Nico Grey [G7] said: *"If the timebox is too short then the amount of data might be too small, or there might not have been enough time for effects to take place. But if the timebox is too long you are wasting valuable time collecting unnecessary data."*.

A hypothesis should be **non-trivial**, as argued in five documents. That is, it should focus on relevant and risky aspects of the business and bring new information about it. Tristan Kromer [G1] said: *"[the hypothesis] should also be something that is risky. A good experiment will generate an invalid hypothesis about half the time."*

Four documents mention that hypotheses should identify a **causal** relationship. Using Tristan Kromer's words [G1]: *"hypotheses should be clear statements that indicate a causal relationship with a clear actor (i.e., customer)."*

Three documents suggest that hypotheses should be **clear** statements. Startup Drill [G8], a consultant company for innovation, describes in its blog: *"The best way to write your assumptions is to write them on the small post-it, this will also limit the size of the assumption statement so you always have clear assumptions. Write one statement per post-it so they will be more transparent and easy to work within the next steps. "*

Finally, in three documents, it is argued that a hypothesis should clearly state the expected **significance**. Grace Ng [G9] wrote: *"To test this, I set a minimum criterion for what I would accept as validation to further explore this opportunity. I had enough confidence that this was a big problem to set a criterion as high as 6 out of 10."*

Table 3 summarizes the qualities identified and the number of documents that mentioned them.

4.3. Types

Types of hypotheses is a common theme emerged from the data analyzed. They are used to facilitate identification or prioritization of hypotheses. In this section, we grouped similarly identified types. It is important, though, to stress that several authors actually proposed sets of categories that would be able to completely classify all possible hypotheses. For instance, two authors used the fields of Business Model Canvas to classify hypotheses. In our description, such ties were broken but listing all proposed classification schemes would take a lot of space. Besides that, such effort would not bring valuable insights. There were many proposed types that focused on similar aspects and we grouped them together given their similarity and as a way to synthesize the results. Nevertheless, below, instead of ordering all the types, first we describe those that were coded alone since these types

Table 3: Qualities of hypotheses ordered by the number of occurrences.

Quality	Documents
Testable	[G10] [G11] [G12] [G13] [G1] [G9] [G14] [G15] [G16] [G17] [G18] [G19] [G5] [G20] [G21] [G22] [G23] (17)
Specific	[G24] [G25] [G12] [G26] [G1] [G27] [G28] [G29] [G30] [G2] [G31] [G19] [G21] [G32] [G22] [G23] (16)
Falsifiable	[G3] [G24] [G12] [G1] [G29] [G30] [G4] [G6] [G33] [G5] [G34] [G35] [G36] [G37] [G22] (15)
Measurable	[G3] [G24] [G13] [G28] [G29] [G7] [G30] [G2] [G6] [G33] [G20] [G34] [G35] [G22] [G23] (15)
Time-boxed	[G1] [G7] [G30] [G4] [G2] [G36] [G23] (7)
Non-trivial	[G24] [G25] [G1] [G29] [G22] (5)
Causal	[G13] [G6] [G38] [G1] (4)
Clear	[G13] [G1] [G8] (3)
Express significance	[G25] [G9] [G39] (3)

were the most recurrent in the data analyzed.

The most commonly mentioned type is **customer hypotheses**, present in 15 documents. This type of hypothesis focuses on verifying if the target customer that founders think will have an interest in their product or service exist and will pay for it. Although using the term "customer segments hypothesis", Brian Laung Aoaeh [G18] summarized: *"The customer segments hypothesis forces you to answer the questions 'Who are my customers?' and 'What problems do my customers face?' The hypothesis brief should discuss customer problems, types, and archetypes respectively."* As an illustration, we can mention Alex Chuang's example [G3]: *"I believe restaurant owners will use our lightweight video resume app at least twice a month to hire servers quickly and they will convert to paid subscriptions after a 30-day unpaid trial because our product helps them hire 50% faster."*

Then, in 12 documents, there is the **problem hypotheses** type. This type of hypothesis is concerned with if a problem the product or service is trying to solve is a real problem for the target customer. Alexander Cowan [G11] mentioned *"Do the problems you're solving really exist? Is it more of a 'job to be done' or a need, desire? How important is the problem or problems? How is the customer solving them now? With what alternatives?"* An example from the same document was *"If we ask nonleading questions about how cycling might be even better for regular cyclists, we'll consistently hear that they wish they could more confidently, more easily try out new routes."*

In a similar sense, in 12 documents, **value hypotheses** are suggested, that is, if the proposed product or service would be able to provide value for the target customer. For instance, Cecilia Thirlway [G2] wrote: *"The value hypothesis is designed to test whether your product or service provides potential customers with enough value once they are using it (and therefore, whether they would be willing to pay for it)."* As an illustration, we can give an example from Alexander Cowan [G11]: *"There are HR Managers in charge of recruiting technical talent, and they need to screen recruits for the specific technical skills in a job description. Currently, they do their best by checking references and asking a few questions, but if we offer a way to automate quizzing for a specific job description, then we'll observe HR Managers creating and using quizzes and standardizing on use of the platform for new hires."*

Then, 13 documents suggest different **market-related hypotheses** types. In this group, hypotheses are related to several aspects of selling the product or service. They include if the channels proposed are adequate, the pricing strategy, and growth and distribution mechanisms. For instance, Alexander Cowan [G11] provided a template: *"If we get [customer segment or persona] to a landing page with a demo, [x]% will sign up for [our email product announcements, a free trial]."*

Finally, in 10 documents, there are different types of **product-related hypotheses**. This group aggregates suppositions regarding the feasibility, the team capacity of developing the idea, and the viability. Two authors also mentioned feature-level hypotheses, that is, regarding if a determined new feature is useful or not to customers. To illustrate, we can use a hypothesis example provided by Nico Grey [G7]: *"If I add a notification feature that allows the waiter/waitress to set reminders to add in his/her tips, then I am going to see a 10 percent*

increase in the number of users opening the app four times or more in a week over the next three months."

Table 4 summarizes the identified hypotheses types and the number of documents that mentioned these types.

Table 4: Hypotheses types.

Type	Documents
Customer	[G3] [G11] [G40] [G1] [G41] [G42] [G14] [G43] [G18] [G44] [G33] [G45] [G46] [G47] [G48] (15)
Problem	[G11] [G9] [G14] [G49] [G43] [G17] [G50] [G51] [G8] [G45] [G52] [G53] (12)
Value	[G11] [G29] [G42] [G2] [G43] [G18] [G45] [G52] [G54] [G55] [G48] [G56] (12)
Market-related	[G1] [G29] [G2] [G43] [G18] [G52] [G54] [G55] [G48] [G56] [G14] [G11] [G17] (13)
Product-related	[G40] [G17] [G44] [G45] [G47] [G42] [G49] [G43] [G57] [G11] (10)

4.4. Activities

In the data analysis, five categories of activities related to how to handle hypotheses emerged. They are elicitation, prioritization, specification, analysis, and management. In the following sections, we detail each of them. Here, we used the term activity for HE, following the tradition already employed in Requirements Engineering (e.g. [42]).

4.4.1. Elicitation

This activity groups practices that practitioners can use to reach hypotheses.

The most common type of practice is the **use of canvases or maps**, present in 30 documents. These practices are characterized by the use of graphical artifacts composed of specific fields that should be filled based on the product idea. For instance, the most mentioned canvas was the Business Model Canvas. Osterwalder et al. [43] proposed this artifact based on their ontology on the business model [44]. The canvas is divided into nine elements named key partners, key activities, key resources, cost structure, value proposition, customer relationships, channels, customer segments, and revenue streams. It can be used to represent a business model and, according to the authors, envision innovations on it. In the context of startup hypotheses, the documents in this type suggest that the founder or team fills the canvas and develop hypotheses based on its elements. Similar proposals are made using other canvases like Lean Canvas [45] and Value Proposition Canvas [46].

The second most common type of practices in this category is the use of a **pre-defined set of questions or aspects** about the product. This practice type is present in 25 documents. For these practices, the authors suggest different lists of questions that could lead or serve as initial set of hypotheses. For instance, Alex Pawlowski [G51] said:

"The following questions should guide you to bridge the gap and find potential answers to your most pressing issues around: Customer problem: which customer problems are to be solved? Product fit: can our product solve the customer problem? Business Model fit: If so, can we make it viable and profitable? Insights: is there a learning curve for lessons learned?"

The next most common type is based on **team sessions**, where the team is gathered and execute some form of facilitation technique to reach hypotheses. This type of practice was found in 14 documents. The most common example from this type is brainstorming sessions. Also, in 14 documents, **individual techniques** are suggested to reach hypotheses, e.g., "five whys", and pre-mortem - to imagine if the company failed and the founder has to describe why.

Finally, in three documents, **interviews** either with customers (problem or solution interviews) or with experts are suggested as a hypothesis elicitation technique. Table 5 summarizes the hypotheses elicitation techniques identified and the number of documents that referred to them.

Besides these types of techniques, in ten documents, it is suggested the execution of some techniques before creating hypotheses, e.g., to use empathy-building techniques or to observe the current state of the market or the user behavior. For instance, Alexandre Azevedo [G65] mentioned: *"If you want to be successful in developing your business, start by being an expert in your customers' lives. The better you know your customers, the easier it will be for you to generate powerful insights that will guide you on the development of your busi-*

Table 5: Hypotheses elicitation practices ordered by the number of occurrences.

Type	Documents
Use of canvases or maps	[G40] [G32] [G47] [G58] [G30] [G45] [G59] [G31] [G22] [G57] [G24] [G60] [G1] [G61] [G41] [G62] [G63] [G64] [G18] [G65] [G51] [G66] [G67] [G48] [G68] [G9] [G50] [G69] [G17] [G70] (30)
Pre-defined set of questions or aspects	[G62] [G26] [G71] [G72] [G14] [G73] [G28] [G74] [G43] [G75] [G50] [G51] [G8] [G21] [G76] [G34] [G77] [G78] [G79] [G9] [G80] [G8] [G81] [G46] [G82] (25)
Team sessions	[G9] [G80] [G83] [G50] [G8] [G70] [G84] [G20] [G79] [G22] [G74] [G85] [G28] [G24] (14)
Individual techniques	[G40] [G61] [G29] [G14] [G22] [G86] [G38] [G87] [G88] [G28] [G89] [G90] [G65] [G27] (14)
Interviews	[G24] [G49] [G36] (3)

ness model hypothesis."

4.4.2. Prioritization

This category groups practices related to creating an order in which the hypotheses should be tested, or at least identifying the riskiest one.

The most common type of practice in this category is prioritizing **based on two qualitative dimensions**, usually employing matrices. It was found in 15 documents. These practices are characterized by suggesting to get one's hypotheses set in a physical form (e.g., sticky notes) and put them in a matrix according to two axes based on their understanding. The axes labels varied from the probability of the hypotheses being false vs. the impact of being false in the probability of idea success to risk vs. effort. It is important to notice that none of the documents presented clear guidelines of how systematically compare hypotheses regarding these dimensions. Such an assessment is implicitly left to founders.

In 10 documents, founders are suggested to prioritize hypotheses relying on their own discretion or intuition, that is, their **gut-feeling**. Then, in seven documents, it was suggested to prioritize **based on one qualitative dimension** either the risk or the impact to the users, again without clearly stating how to compare hypotheses on this regard.

In five documents, it is argued that there is a standard, **pre-defined order** in which hypotheses should be tested. For instance, Jeff Bussgang [G91] mentioned *"Generally speaking, I find that the consumer value proposition tests are the most important initial tests to focus on. Once you nail the consumer value proposition, the go to market plan can flow and once both of those components are locked down, the profit formula can be tested."*

Less voiced approaches, in only one document each, are **talk to domain experts** and **find assumptions at the beginning of the chain**. While the first is self-explaining, the latter argues that hypotheses are naturally linked to, or based on, others. It is then straightforward to find the one at the beginning of the chain and start evaluating it. Table 6 summarizes the identified practices for prioritization, including the number of documents that mentioned them.

Table 6: Hypotheses prioritization practices ordered by the number of occurrences.

Type	Documents
Based on two qualitative dimensions	[G58] [G28] [G74] [G62] [G57] [G86] [G50] [G8] [G76] [G69] [G89] [G14] [G22] [G20] [G39] (15)
Using gut-feeling	[G9] [G80] [G28] [G74] [G88] [G49] [G30] [G17] [G32] [G47] (10)
Based on one qualitative dimension	[G81] [G74] [G62] [G49] [G44] [G70] [G79] (7)
Pre-defined order	[G41] [G91] [G54] [G55] [G48] (5)
Talk to domain experts	[G62] (1)
Find the beginning of the chain	[G71] (1)

Besides that, in five documents, it was proposed some form of **team-based prioritization** process. A technique suggested by some authors is dot-voting in which each team member has some points, for instance, three, and he or she can distribute it among the hypotheses that he or she thinks should be prioritized first. The dots can be split among hy-

potheses as the team member wishes: all can go to just one, an even or an uneven distribution among the selected hypotheses. Other authors did not suggest a specific technique just mentioned a discussion among team members.

Finally, one author criticized the lack of quantitative metrics for prioritizing hypotheses. Sam McAfee [G41] suggested the use of a model to calculate the risk associated with hypotheses and prioritize based on that. This model could be based on financial or historical data.

4.4.3. Specification

In this category, we grouped practices and guidelines of how to specify a hypothesis. The most common type of practices is the use of a **template**. In 33 documents, some form of templates is suggested. We divided the templates in two big groups regarding if they considered only the hypotheses itself or included also details of how to validate it either through the use of a metric or an experiment. In some documents, both types of templates are mentioned. The most common template type, within 23 documents, is considering the way to test hypotheses. They range from simple statements such as *"because we believe X, if we do Y, we expect Z to happen"* cited by James Bircher [G10], to highly detailed experiment specifications as Abdo Riani [G92] proposed: *"my goal is to solve [describe problem] with [describe solution] for [describe ideal buyer] by [describe your execution plan within a specific time frame] and to acquire [number of customers] that will generate [amount of money] through [describe marketing channel(s)] by [deadline]."*

Regarding hypotheses specification templates that do not mention metrics or experiments, as shown in 14 documents, they generally start with the phrase "we believe" to stress the idea of uncertainty. Sylvia Lai [G79] proposed a template with an example: *"We believe that [sharing more information about the driver's experience and stories] For [the riders] Will [make riders feel more comfortable and connected throughout the ride]."* Table 7 displays the documents describing templates for hypotheses.

In three documents, the use of acronyms to evaluate hypothesis specification is proposed. They are SMART [G70, G92] (specific, measurable, achievable, realistic, and time-bound) and HOPE [G6] (hypothesis, objective, prediction, and execution).

Table 7: Hypotheses specification templates ordered by the number of occurrences.

Template type	Documents
With experiment description or metric	[G10] [G3] [G24] [G11] [G60] [G38] [G9] [G85] [G7] [G14] [G68] [G17] [G44] [G19] [G33] [G59] [G20] [G79] [G36] [G93] [G92] [G22] [G23] (23)
Without experiment description or metric	[G12] [G61] [G29] [G14] [G57] [G49] [G2] [G90] [G31] [G84] [G79] [G36] [G37] [G22] (14)

One interesting excerpt is from the consulting company MaRS [G21] that highlights the importance of writing the hypotheses as a team to enforce a shared understanding. It is suggested that *"do not delegate the work of writing the briefs. Sit down as a team and ensure that what you document reflects a shared understanding of the issues you aim to test."* A similar argument is present in another document as well.

4.4.4. Analysis

In a smaller number, some practices are grouped under this category. They occur once the hypothesis is elicited and specified, and focus, for instance, on analyzing the quality of the hypotheses statements developed. In seven documents, it is suggested that practitioners should inspect the created hypotheses and try to **break hypotheses in small ones**. Daniel Tenner [G81] suggests *"before trying to answer the questions, you first break them down into sub-questions."* In two documents, a **cross-dependency analysis** is proposed to understand better how hypotheses are interconnected. In both cases, the importance of this activity before prioritizing is emphasized. Finally, practitioners are recommended to **check the hypotheses** concerning some attributes. This suggestion is related to the acronyms observed in the specification category but, instead of being part of that process, it is suggested that it should be done when hypotheses statements are ready. Table 8 presents a summary of the sources that suggested these techniques.

4.4.5. Management

The final category of activity is to manage hypotheses. First, four documents ([G25], [G94],

Table 8: Hypotheses analysis techniques ordered by the number of occurrences.

Practice	Documents
Break hypotheses in smaller ones	[G11] [G81] [G14] [G2] [G51] [G59] [G39] (7)
Cross-dependency check	[G83] [G39] (2)
Attributes checking	[G19] [G23] (2)

[G39], and [G34]) highlight the importance of an **iterative process to improve hypotheses**. In this regard, we can mention Ted Ladd [G25] that wrote *"failure to confirm a hypothesis should prompt the entrepreneur to shift some of the conditions of the hypothesis and try again. The entrepreneurial process is recursive, where each iteration alters the assumption being tested. What is unethical heresy to a researcher is a necessary pivot for entrepreneurs designing a new venture."*

The need for **tracking of hypotheses** is specified in one document. Alex Sherman [G90] mentioned that some colleagues from his company said that *"it can seem daunting to be in a new space [tracking assumptions], but this helps frame and prioritize the team to come together on the most important thing we need to learn"*. They *"found that if not everyone participating didn't have the shared context the quality of the assumptions varied and weren't always applicable."*

Finally, one document proposes the concept of **hypotheses backlog**. Melanie Hambarsoomian [G53] wrote: *"I've created what I've called a 'Hypothesis backlog'. It is a collection of opportunities to improve the product — some guesses, some validated by research. But likely, the prioritized ones need further discovery and problem validation. It is not a backlog in the sense of a stream of work for developers or a roadmap for the Product (but could end up becoming part of these things if prioritized and validated)."*

5. Discussion

Based on the developed model, we are able to answer our research questions. Regarding RQ1, "What are the proposed activities of Hypotheses Engineering in the context of software startups?", the emerged model depicted the five categories of HE activities: elicitation, prioritization, specification, analysis, and management. These activity categories are similar to those described for Requirements Engineering (RE). Nuseibeh and Easterbrook [47] defined RE as the process of discovering the purpose for which a software system was intended and documented the stakeholders' needs to be used further in the software development process. Usually, RE is described as a knowledge transfer activity from domain experts to system development teams [48, 49, 50]. Below, we will further develop this argument comparing the identified activities for HE in software startups and those commonly described in RE. For each activity, we also discuss the techniques employed, summarized in the corresponding tables, answering RQ2, "What are the proposed techniques for each Hypothesis Engineering activity in the context of software startups?"

Hypotheses elicitation is described as an activity focused on identifying critical uncertain aspects of a business idea that should be validated. In the documents analyzed, the most common practices are the use of canvases or maps, and a pre-defined set of questions. This use is different from RE, where interviews are the most used [51] and considered the most effective [49] technique to elicit requirements. It is essential to notice, though, that requirements elicitation research has already pointed in the direction of creativity or green-field requirements [50].

Prioritizing hypotheses is generally described as finding the riskiest ones, those that, if not validated, could lead to the company failure. The common practices are based on two-dimensional matrices, usually comparing an estimate of the probability of not being valid and the impact of such a fact on business viability. Such a focus represents a subtle but defining difference to requirements prioritization. In the latter, the focus is on finding the requirements with higher value to the business success [52] and in the former, finding the hypotheses that represent the highest risks to business success.

Hypotheses specification is generally made through the use of templates to document hypotheses in a written form. The predominance of templates manifests a clear influence of user stories. Beck [53] initially proposed a user story as a lightweight requirement documentation technique to be used in the Extreme Programming methodology. Nevertheless, it was better described by Cohn [54] that popularized the most common tem-

plate and de-facto standard [55]: "As a <role>, I want <goal>, [so that <benefit>]." Several other authors proposed similar templates. Wautelet et al. [56] analyzed 20 templates from scientific sources and 65 from practitioner ones, and summarized in a unified model: "As a <Role>, I want/want to/need/can/would like <Task> so that to <Goal>." In comparison, the templates to specify hypotheses are more concerned with highlighting the uncertain sense of them. Such concern is demonstrated by the fact that, for instance, several templates start with "we believe that."

Another interesting finding is the two different approaches regarding the boundaries of specification, namely, to include or not the experiment itself. Some templates are restricted to the question addressed by that hypothesis. Meanwhile, other templates also give details on how to implement the experiment to evaluate the hypothesis. For instance, some templates mention the metrics that should be evaluated.

Compared to the previous activities, the importance of hypotheses analysis practices was less emphasized in the reviewed documents. Such a disparity may be a consequence of considering the practices proposed for elicitation and specification sufficient to reach a good set of hypotheses. This fact is similar to RE validation and the research on it [57], which indicates that there are many possible advances to make for handling hypotheses.

Hypotheses management is another area underserved in the reviewed gray literature. One possible reason is that the primary focus of the documents analyzed is in early-stage startups, and management is probably only needed in a later stage when several hypotheses have been previously tested and are currently handled. Besides that, most documents are short and focused only on one aspect, probably to maintain the readers' attention. Therefore, they tend to focus on practices that the readers consider more impactful to attain their attention.

Regarding the practices, one important result is revealed by the contribution types. There was no study classified as evaluation research in the documents analyzed, that is, employing strict and formal methods. Such a result indicates that the techniques currently used are based on practitioners' experience and are not systematically evaluated.

Finally, there is a distinction between HE and data-driven RE to be made. The latter is generally referred to as the explicit and implicit use of user feedback in an aggregated form to support requirements decisions [58, 59]. For instance, developer teams can use software data usage or even customers' comments on app stores to elicit and prioritize requirements. Nevertheless, in this sense, the goal is still a knowledge elicitation about what users want rather than validating a pre-defined hypothesis.

In summary, despite the similar steps, HE and RE have different goals that are, to a certain extent, complementary like the development approaches they serve: experimentation and requirement-driven. RE is focused on eliciting the knowledge that users have and how the software would help them, and HE leads towards knowledge creation, prioritizing the riskiest elements to the validity of a business or feature idea. For instance, HE, and the experiments performed based on the hypotheses, could serve as validation for requirements.

Regarding the HE literature, our results represent the first step towards better describing it. The original position paper [10] argues for its existence based on a comparison between experiment-driven and requirements-driven software development and a consequence need of techniques to handle hypotheses as those that act on requirements. Therefore, in that paper, the activities were speculative based on Requirements Engineering. This paper, on the other hand, describes the activities based on what practitioners proposed.

5.1. Threats to validity

Although Garousi et al. [35] did not mention a specific discussion on threats to validity in GLR studies, we think such discussion is essential. We followed the seminal scheme to assess threats to validity described by Runeson and Host [60] composed of four aspects: construct validity, internal validity, external validity, and reliability. Even though the authors originally proposed this scheme to case studies, Garousi et al. [41] used them for a multivocal literature review but using conclusion validity rather than reliability. Wohlin et al. [19, p.103] claim that reliability is the counterpart, in qualitative studies like ours, of conclusion validity for quantitative studies.

Construct validity is related to the constructs under study and if how they are described in data represents what the researchers have in mind [60]. A significant threat in our study is the hypothesis-assumption dichotomy and the lack of a uniform understanding of these concepts among practitioners. To mitigate this threat, we discussed this point

in our results. Another concern is the definition of startup itself. As discussed earlier, there is no unique, agreed-upon definition of startup. Nevertheless, the data showed that innovation is a common element on which practitioners concentrate, which is in line with the scientific literature's understanding, as discussed in Section 2.3. Therefore, the threats to construct validity were minimal.

Internal validity is related to causal inferences made when, for instance, a factor is said to be determined by a second one, but, in reality, it is determined by a third one not considered in the study. Since the study is descriptive, this risk is minor. Nevertheless, a related threat is if the classifications obtained in the data analysis were sound, not influenced by potential bias or presumptions of the researchers. To mitigate this threat, two researchers inspected the analysis, and disagreements were discussed among all researchers until they reached a consensus.

External validity is related to how much the results could be generalized to the population. For this study, it means if the results represent what happens in all software startups. Using the results of Google indicates that these documents are probably the ones that practitioners are considering. Although it is possible that some software startups indeed employ different practices, it is less probable that they utilize a technique completely different from the categories described in the results.

Finally, reliability is concerned with how the results depend on the researchers that performed the study. Such an aspect is related to the study being reproducible. Therefore, to improve this aspect, we followed published guidelines for the utilized method. Besides that, we documented and presented all steps performed. Nevertheless, just one researcher doing the first screen of the links may threaten our study. Such a decision was taken given the amount of data available and consequent work to read all documents. To mitigate this threat, the researcher that inspected all documents included those that he was not sure about the relevance. In such a way, we tried to minimize false negatives but allowed false positives to be re-evaluated by a second researcher. Also, the high number of documents considered in the study and an evident theoretical saturation reached indicate reasonable results.

6. Challenges and lessons learned on gray literature review

In this section, we describe the challenges faced and the lessons learned of applying the gray literature review as the main research method. We grouped them into two aspects: document selection and data analysis.

6.1. Documents selection

The first challenge in a GLR is the number of candidate documents and how to select them. In a systematic literature review (SLR) constrained to academic publications, one could evaluate the candidates based on a series of metadata such as title, venue, year, and abstract. In gray literature, there is no such information, or it is limited: titles may not fully represent the content, not all documents display the publication date, and it is not common to present a summary or abstract. Therefore, it is usually necessary to read or scan the whole document to evaluate whether it should be included.

Therefore, we had an enormous amount of work to evaluate all links. Nevertheless, the use of separated search strings allowed us to compare the results with different synonyms. We noticed that the majority of the documents selected (74, or 82, if we include those obtained from snowballing, out of 95) were already obtained in the search with the term "startup." The other synonyms had a much smaller contribution. Besides that, we noticed that different terms are generally used in distinct contexts. For instance, the search with "early-stage firm" did not contribute to the final set of documents. We noticed that most of the results were concentrated on academic papers, in the field of Economics and Finance, or related to Firm Laws. The other two synonyms contributed fewer documents as well, and several of them, eight in the case of "venture", were already found using the term "startup". Besides that, the documents found for the other synonyms did not add any new element to the emerged model; that is, we can say that theoretical saturation was reached.

In summary, based on our experience, we would say that different communities use distinct terms. Thus, differently from an SLR and depending on the research goal, authors of GLR may focus on specific terms instead of targeting a comprehensive set of synonyms. For instance, researchers could analyze a sample of results for different terms and

reason if a limited set of terms would lead to satisfactory results.

6.2. Data analysis

Regarding analysis, performing thematic synthesis is more challenging with gray than white literature. In academic papers, the peer-review process leads to some forms of consensus around some concepts. This effect does not happen in the literature produced by practitioners. We could observe the influence of some authors or seminal methodologies like Lean Startup [14] or Business Model Canvas [43], but different practitioners had distinct interpretations of the same concepts. Besides that, there is a recombination phenomenon where authors collect influences from several proposals and build new ones. These newly proposed concepts are similar but different from those that influenced their creation, making it hard to create themes grouping these concepts.

In summary, there is a divergent mechanism in place where a term can have several meanings. This phenomenon hinders the grouping of concepts needed in thematic synthesis. Therefore, the researcher must pursue a more intensive and in-depth reading-and-comprehending operation to search the underlying meanings.

7. Conclusions

The availability of a large amount of usage data and the rise of customer development methods such as Lean Startup in innovative contexts led to a new approach to software development based on experiments. In this situation, several techniques were proposed to handle hypotheses in a similar way that RE activities handle requirements in conventional software development. We called these activities with the collective name of Hypotheses Engineering. Given that experimentation is essential in innovative contexts such as software startups, this context was a reasonable choice to further study HE activities and techniques. To achieve that, we collected practitioners' statements found online and analyzed them using thematic synthesis.

This work's main contribution is a model that grounds Hypotheses Engineering in data from practice describing how practitioners in software startups perceive the concept of hypothesis, including its qualities and types, and how they perform or suggest to conduct the activities related to hypotheses. In this process, we identified five activities: elicitation, prioritization, specific, analysis, and management, that group the practices found. Besides that, we identified the qualities that practitioners deemed relevant for hypotheses and the types of hypotheses. Researchers could use the model and associated concepts as a conceptual basis to investigate hypotheses and experimentation related phenomena. Practitioners could use the results as a catalog of techniques and a guideline to combine different practices into an overall custom-made method. Following Garousi et al.'s advice, we plan to publish these results in a more friendly venue and format for the practitioners' audience. Finally, researchers and practitioners could use our results to frame the development of new practices to HE in software startups.

This work stems several branches for future research. Our work described the way practitioners perform HE activities in software startups, but we cannot determine *if* and *why* the associated techniques are effective. Future studies could evaluate several aspects of these techniques, such as effectiveness and usability. An interesting investigation would be to analyze these aspects for techniques in different startup development stages. Future research could also use the identified categories as benchmarks to search for or build potential new techniques and practices. Similar studies could be performed for other software development teams, like in large organizations, to verify if the activities these teams perform are similar or if others emerge in different contexts. In the future, once literature builds around this concept, we expect that a multivocal literature review could be a valuable future work.

Sources

[G1] T. Kromer, Templates suck, here's our lean startup template, https://kromatic.com/blog/templates-suck-heres-our-lean-startup-template/, accessed: 2020-06-03.

[G2] C. Thirlway, What makes a good hypothesis?, https://www.solverboard.com/blog/what-makes-a-good-hypothesis, accessed: 2020-06-04 (2019).

[G3] A. Chuang, The ultimate step-by-step guide to validating your startup idea, part two, https://medium.com/startup-grind/the-ultimate-step-by-step-guide-to-validating-your-startup-idea-part-two-882b9105bd, accessed: 2020-06-08 (2016).

[G4] MaRS, 7 principles lean startups can learn from the scientific process, https://www.marsdd.com/

news/7-principles-lean-startups-learn-from-scientific-process/, accessed: 2020-06-11 (2015).
[G5] Bullet, Lean startup zappos: the basics, https://www.bullethq.com/lean-startup-zappos-how-zappos-validated-their-business-model-with-lean/, accessed: 2020-06-12.
[G6] M. Lieberman, Hypothesis testing for entrepreneurs, http://blogs.oregonstate.edu/thestartupadvantage/category/hypothesis/, accessed: 2020-06-04 (2016).
[G7] N. Grey, Writing a good hypothesis, https://help.glidr.io/en/articles/1648327-writing-a-good-hypothesis, accessed: 2020-06-10.
[G8] Startup Drill, Defining the problem to work with, https://startupdrill.co/blog/problem-assumptions/, accessed: 2020-06-04 (2018).
[G9] G. Ng, A guide to validating product ideas with quick and simple experiments, https://www.smashingmagazine.com/2014/04/a-guide-to-validating-product-ideas-with-quick-and-simple-experiments/, accessed: 2020-06-09 (2014).
[G10] J. Birchler, Lean startup best practices: Write hypotheses you can learn from., https://medium.com/@jamesbirchler/lean-startup-best-practices-write-hypotheses-you-can-learn-from-7e119afa0080, accessed: 2020-06-08 (2016).
[G11] A. Cowan, Your lean startup, https://www.alexandercowan.com/creating-a-lean-startup-style-assumption-set/, accessed: 2020-06-12.
[G12] B. Yoskovitz, How to structure good hypotheses for your lean startup, https://www.instigatorblog.com/good-hypotheses/2011/05/05/, accessed: 2020-06-03 (2011).
[G13] P. O'Malley, Formulate a hypothesis & design an experiment to test it, https://openclassrooms.com/en/courses/4544561-learn-about-lean-startup/4710706-formulate-a-hypothesis-design-an-experiment-to-test-it, accessed: 2020-06-09 (2020).
[G14] N. Torabi, Running hypothesis driven experiments using the mvp, https://uxdesign.cc/the-product-manager-and-the-mvp-a0c618b0d8fa, accessed: 2020-06-04 (2019).
[G15] E. Garbugli, The importance of creating testable hypotheses in b2b customer development, https://leanb2bbook.com/blog/importance-creating-testable-hypotheses-b2b-customer-development/, accessed: 2020-06-03.
[G16] R. Dawson, Using testable hypotheses to bring lean startup into the enterprise, https://rossdawson.com/blog/using-testable-hypotheses-to-bring-lean-startup-into-the-enterprise/, accessed: 2020-06-11 (2015).
[G17] J. Kylliäinen, Idea validation: Steps and tools for testing your idea, https://www.viima.com/blog/idea-validation, accessed: 2020-06-04 (2019).
[G18] B. L. Aoaeh, A note on startup business model hypotheses, https://www.tekedia.com/note-startup-business-model-hypotheses/, accessed: 2020-06-04 (2017).
[G19] T. Budayici, Hypothesis-driven practices to build better features, https://medium.com/swlh/hypothesis-driven-practices-to-build-better-features-b5cc53c9fd24, accessed: 2020-06-04 (2020).
[G20] M. Pillich, G. Melgers, Think like a startup: a 5-step guide to lean experiments, https://hike.one/update/think-like-a-startup-a-5-step-guide, accessed: 2020-06-05 (2016).
[G21] MaRS, Value proposition and blank's customer discovery method—phase 1: State your hypotheses, https://learn.marsdd.com/article/blanks-customer-discovery-method-part-1-the-customer-development-model-in-value-proposition/, accessed: 2020-06-11.
[G22] The Growth Revolution, From assumption to hypothesis done right, https://thegrowthrevolution.com/from-assumption-to-hypothesis-done-right/, accessed: 2020-06-25.
[G23] T. Torres, The 5 components of a good hypothesis, https://www.producttalk.org/2014/11/the-5-components-of-a-good-hypothesis/, accessed: 2020-06-25 (2014).
[G24] T. Kromer, Assumption vs. hypothesis — to the death!, https://medium.com/@Kromatic/assumption-vs-hypothesis-to-the-death-df1ebc63e749, accessed: 2020-06-08 (2018).
[G25] T. Ladd, A fatal flaw in the lean startup method: Building a hypothesis, https://www.forbes.com/sites/tedladd/2019/10/18/a-fatal-flaw-in-the-lean-startup-method-building-a-hypothesis/, accessed: 2020-06-03 (2019).
[G26] T. Kastelle, How to make good lean startup hypotheses, http://timkastelle.org/blog/2016/02/how-to-make-good-lean-startup-hypotheses/, accessed: 2020-06-03 (2016).
[G27] R. Higham, The mvp is dead. long live the rat., https://hackernoon.com/the-mvp-is-dead-long-live-the-rat-233d5d16ab02, accessed: 2020-06-09 (2016).
[G28] L. Szyrmer, How to identify your riskiest assumption, https://www.launchtomorrow.com/2019/12/how-to-identify-your-riskiest-assumption/, accessed: 2020-06-03 (2019).
[G29] V. Todorov, Three simple tips for formulating hypotheses for startup business, https://www.linkedin.com/pulse/20140721125030-7214044-three-simple-tips-for-formulating-hypotheses-for-startup-business, accessed: 2020-06-03 (2014).
[G30] A. Maurya, How to identify a lean startup, https://blog.leanstack.com/how-to-identify-a-lean-startup-204855635220, accessed: 2020-06-11 (2010).
[G31] Future Founders, What is customer discovery? a 4-step guide to building the right product for the right customers, https://futurefounders.com/news-article/what-is-customer-discovery-4-step-guide-to-building-the-right-product-for-the-right-customers/, accessed: 2020-06-11 (2017).
[G32] P. Noack, How to put your business model to the test?, https://venture-leap.com/2020/05/04/how-to-put-your-business-model-to-the-test/, accessed: 2020-06-25 (2020).
[G33] A. Chuang, The quick and dirty guide to validating your startup idea, https://medium.com/swlh/the-quick-and-dirty-guide-to-validating-your-startup-idea-c6be6cd91f51, accessed: 2020-06-04 (2016).
[G34] Yarandin, Testing business hypotheses: The lean startup, http://www.yarandin.com/en/testing-

business-hypotheses-lean-startup, accessed: 2020-06-11 (2017).

[G35] N. Brisbourne, Good startup hypotheses must be falsifiable, http://www.theequitykicker.com/2015/02/09/good-startup-hypotheses-must-falsifiable/, accessed: 2020-06-11 (2015).

[G36] A. Maurya, The 7 habits for running highly effective lean startup experiments, https://leanstack.com/7-habits-for-running-highly-effective-experiments/, accessed: 2020-06-11 (2015).

[G37] S. Runner, Startup experiments, https://startuprunner.com/startup-experiments/, accessed: 2020-06-06 (2014).

[G38] A. M. Helmenstine, What are the elements of a good hypothesis?, https://www.thoughtco.com/elements-of-a-good-hypothesis-609096, accessed: 2020-06-09 (2019).

[G39] C.-T. Chu, Bridging the gap between lean startup in theory and in practice, https://ml.posthaven.com/bridging-the-gap-between-lean-startup-in-theory-and-in-practice, accessed: 2020-06-05.

[G40] A. Osterwalder, How to test your idea: Start with the most critical hypotheses, https://www.strategyzer.com/blog/how-to-test-your-idea-start-with-the-most-critical-hypotheses, accessed: 2020-06-09 (2017).

[G41] S. McAfee, The only proven method to identify your riskiest assumption, https://medium.com/startup-patterns/the-only-proven-method-to-identify-your-riskiest-assumption-a240c6403a67, accessed: 2020-06-09 (2015).

[G42] P. Goossens, Tips: how i structure business assumptions for a new innovation project, https://www.boardofinnovation.com/blog/structure-business-assumptions/, accessed: 2020-06-10.

[G43] CobuildLab, Start correctly your business with a value hypothesis and growth hypothesis, https://cobuildlab.com/blog/value-hypothesis-and-growth-hypothesis/, accessed: 2020-06-11.

[G44] Board of Innovation, Validation guide - 24 ways to test your business ideas, https://info.boardofinnovation.com/hubfs/Validation%20Guide%20compressed.pdf, accessed: 2020-06-04.

[G45] A. Makkawi, Lean canvas examples: Create your lean hypotheses in [30 mins flat], https://buildsuccessfulstartups.com/lean-canvas-examples/, accessed: 2020-06-11 (2020).

[G46] P. Stenius, H. Wang, A data-driven approach to validate product-market fit with early adopters, http://www.reddal.com/insights/a-data-driven-approach-to-validating-product-market-fit/, accessed: 2020-06-25 (2014).

[G47] A. Osterwalder, 5 lean startup essentials to reduce risk and uncertainty, https://www.strategyzer.com/blog/posts/2015/4/23/5-lean-startup-essentials-to-reduce-risk-and-uncertainty, accessed: 2020-06-25 (2015).

[G48] S. Blank, The leanlaunch pad at stanford – class 2: Business model hypotheses, https://steveblank.com/2011/03/15/the-leanlaunch-pad-at-stanford-class-2-business-model-hypotheses/, accessed: 2020-06-25 (2011).

[G49] N. D. Stevanović, What is your riskiest assumption?, https://mvpworkshop.co/blog/validate-riskiest-assumption/, accessed: 2020-06-03 (2017).

[G50] G. Ng, Why lean startup experiments are hard to design, https://www.lean.org/leanpost/Posting.cfm?LeanPostId=193, accessed: 2020-06-04 (2014).

[G51] A. M. Pawlowski, Startup series: Part ii — the art of customer discovery and hypotheses testing, https://medium.com/startup-avenue/startup-series-part-ii-the-art-of-customer-discovery-67e0bed0527, accessed: 2020-06-04 (2019).

[G52] V. Todorov, Assumption vs hypothesis in lean start-up practice, https://www.linkedin.com/pulse/20140707174655-7214044-assumption-vs-hypothesis, accessed: 2020-06-11 (2014).

[G53] M. Hambarsoomian, Use a hypothesis backlog to capture and refine your problems, https://medium.com/@melhambo/use-a-hypothesis-backlog-to-capture-and-refine-your-problems-6ed0c4d499a2, accessed: 2020-06-12 (2017).

[G54] T. Griffin, 12 things about product-market fit, https://a16z.com/2017/02/18/12-things-about-product-market-fit/, accessed: 2020-06-25 (2017).

[G55] A. Rachleff, Why you should find product-market fit before sniffing around for venture money, https://www.fastcompany.com/3014841/why-you-should-find-product-market-fit-before-sniffing-around-for-venture-money, accessed: 2020-06-25 (2013).

[G56] B. Hardin, Value hypothesis and growth hypothesis, http://bretthard.in/post/value-hypothesis-and-growth-hypothesis, accessed: 2020-06-25 (2012).

[G57] J. Gothelf, The hypothesis prioritization canvas, https://jeffgothelf.com/blog/the-hypothesis-prioritization-canvas/, accessed: 2020-06-03.

[G58] D. Kander, How to diagnose your riskiest assumptions, https://dkander.wordpress.com/2013/05/07/how-to-diagnose-your-riskiest-assumptions/, accessed: 2020-06-09 (2013).

[G59] A. Fichtner, Hypothesis-driven development, https://hackerchick.com/hypothesis-driven-development/, accessed: 2020-06-05.

[G60] T. Kastelle, What assumptions underlie your business?, http://timkastelle.org/blog/2016/01/what-assumptions-underlie-your-business/, accessed: 2020-06-09 (2016).

[G61] R. Hall, What is the riskiest assumption test and why are startups embracing it?, https://clutch.co/app-developers/resources/what-is-riskiest-assumption-test, accessed: 2020-06-03 (2020).

[G62] Founder Institute, 3 steps to finding the biggest risks to your startup (and how to eliminate them), https://fi.co/insight/3-steps-to-finding-the-biggest-risks-to-your-startup-and-how-to-eliminate-them, accessed: 2020-06-10 (2018).

[G63] D. Adams, D. Loomis, Lean startup for b2b, https://www.iicie.com/uploads/White-Paper/14677499905Lean-Startup-for-B2B-AIM-White-Paper.pdf, accessed: 2020-06-03 (2015).

[G64] J. Lokitz, How to identify the riskiest assumption of your (innovative) idea, https://designabetterbusiness.com/2016/06/21/how-to-identify-the-riskiest-assumption-of-your-innovative-idea/, accessed: 2020-06-03 (2016).

[G65] A. Azevedo, What is your business model hy-

pothesis?, https://thetractionstage.com/2018/10/24/designing-your-hypothesis-through-the-business-model-canvas/, accessed: 2020-06-04 (2018).

[G66] S. Blank, Why build, measure, learn – isn't just throwing things against the wall to see if they work – the minimal viable product, https://steveblank.com/2015/05/06/build-measure-learn-throw-things-against-the-wall-and-see-if-they-work/, accessed: 2020-06-05 (2015).

[G67] K. Kelso, Every startup pitch in the world boils down to two core ingredients, https://www.linkedin.com/pulse/every-startup-pitch-world-boils-down-two-core-kris-kelso, accessed: 2020-06-25 (2018).

[G68] R. Goldberg, Run user experiments to validate your hypothesis, https://www.ibm.com/garage/method/practices/think/practice_run_user_experiments/, accessed: 2020-06-11.

[G69] D. Shah, The most important thing to do before building your startup, http://www.onstartups.com/the-most-important-thing-to-do-before-building-your-startup, accessed: 2020-06-05 (2014).

[G70] Workshop Tools, Riskiest assumption canvas, https://www.wrkshp.tools/tools/riskiest-assumption-canvas, accessed: 2020-06-04.

[G71] E. v. d. Pluijm, How to find your riskiest assumption, https://medium.com/wrkshp/how-to-find-your-riskiest-assumption-7c9ef811622d, accessed: 2020-06-03 (2019).

[G72] D. Amato, Start slow, learn from failure & embrace reality, https://discover.rbcroyalbank.com/techto-october-edition-start-slow-learn-from-failure-embrace-reality/, accessed: 2020-06-25 (2019).

[G73] F. Emprechtinger, Testing hypotheses, https://www.lead-innovation.com/english-blog/testing-hypotheses, accessed: 2020-06-04 (2019).

[G74] I. Nousis, Introducing the riskiest assumption canvas, https://uxdesign.cc/riskiest-assumption-canvas-73ec0e2e0abc, accessed: 2020-06-10 (2019).

[G75] R. Hall, Why your rat (riskiest assumption test) is the real mvp, https://medium.com/mindsea-development-inc/why-your-rat-riskiest-assumption-test-is-the-real-mvp-177d66cde3e1, accessed: 2020-06-11 (2019).

[G76] Tim, How to validate critical assumptions through mvp's, https://founder-hacks.com/how-to-validate-critical-assumptions-mvp/, accessed: 2020-06-11 (2019).

[G77] Hub of tech, How to be mature as a startup, https://huboftech.io/how-to-be-mature-as-a-startup/, accessed: 2020-06-25 (2019).

[G78] F. Basrai, New venture checklist, https://medicine.yale.edu/ihy/resources/new_venture/, accessed: 2020-06-25.

[G79] S. Lai , 5 steps to a hypothesis-driven design process, https://www.invisionapp.com/inside-design/hypothesis-driven-design-process/, accessed: 2020-06-11 (2018).

[G80] R. Hoover, There will be wrong assumptions, https://ryanhoover.me/post/45112508553/there-will-be-wrong-assumptions, accessed: 2020-06-09 (2013).

[G81] D. Tenner, How to evaluate and implement startup ideas using hypothesis driven development, https://danieltenner.com/2017/02/01/how-to-evaluate-and-implement-startup-ideas-using-hypothesis-driven-development/, accessed: 2020-06-03 (2017).

[G82] T. Reiss, Assumption / validation flowchart, https://medium.com/product-ponderings/assumption-validation-flowchart-9dc42293b612, accessed: 2020-06-09 (2017).

[G83] D. Cohen, Using hypothesis-driven development, https://medium.com/startup-grind/using-hypothesis-driven-development-64154ca4e179, accessed: 2020-06-03 (2019).

[G84] N. Legay, Validate before you build: Increase your odds of startup success, https://voltaeffect.com/validate-build-startup-success/, accessed: 2020-06-05 (2018).

[G85] M. Reulman, Before the mvp: How to craft a hypothesis statement, https://thinking.philosophie.is/how-to-craft-an-mvp-hypthesis-statement-31e7895e4017, accessed: 2020-06-10 (2015).

[G86] N. d. Witte, Hypothesis and critical assumptions, https://sites.google.com/site/moocmodules/research-for-entrepreneurs/hypothesis-and-critical-assumptions, accessed: 2020-06-03 (2014).

[G87] T. Tunguz, The founder's null hypothesis, https://tomtunguz.com/null-hypothesis-for-entrepreneurs/, accessed: 2020-06-04 (2013).

[G88] P. O'Malley, Prioritize which experiments to run, https://openclassrooms.com/en/courses/4544561-learn-about-lean-startup/4705111-prioritize-which-experiments-to-run, accessed: 2020-06-10 (2020).

[G89] C. LaCalle, 6 ways to de-risk key assumptions, https://www.dreamit.com/journal/2019/5/9/6-ways-to-de-risk-key-assumptions, accessed: 2020-06-25 (2019).

[G90] A. Sherman, 7 tools and tips for tracking your experiments, https://builttoadapt.io/7-tools-and-tips-for-tracking-your-experiments-74d20c1b1f0e, accessed: 2020-06-11 (2018).

[G91] J. Bussgang, Founders should test the hypotheses that matter most, https://blog.usejournal.com/founders-should-test-the-hypotheses-that-matter-most-aeaa00988f1b, accessed: 2020-06-10 (2019).

[G92] A. Riani, How to set smart startup goals at the idea stage, https://www.forbes.com/sites/abdoriani/2020/05/26/how-to-set-smart-startup-goals-at-the-idea-stage/, accessed: 2020-06-06 (2020).

[G93] A. Osterwalder, Validate your ideas with the test card, https://www.strategyzer.com/blog/posts/2015/3/5/validate-your-ideas-with-the-test-card, accessed: 2020-06-08 (2015).

[G94] Applied Technologies, How to start developing a software product and what is a startup?, https://www.appliedtech.ru/en/s-chego-nachat-razvitie-programmnogo-produkta-i-chto-takoe-startap.html, accessed: 2020-06-05 (2019).

[G95] D. Sutevski, How to test business ideas using null hypothesis, https://www.entrepreneurshipinabox.com/1399/business-ideas-null-hypothesis/, accessed: 2020-06-05.

References

[1] P. Bourque, R. E. Fairley, I. C. Society, Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0, 3rd Edition, IEEE Computer Society Press, Washington, DC, USA, 2014.

[2] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas, Manifesto for agile software development (2001). URL http://www.agilemanifesto.org/

[3] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, Computers in Human Behavior 51 (2015). doi:10.1016/j.chb.2014.10.046.

[4] M. Alshayeb, W. Li, An empirical study of system design instability metric and design evolution in an agile software process, Journal of Systems and Software 74 (3) (2005) 269–274. doi:10.1016/j.jss.2004.02.002.

[5] J. Bosch, H. H. Olsson, I. Crnkovic, It Takes Three to Tango : Requirement , Outcome / data , and AI Driven Development, in: Software-intensive Business Workshop on Start-ups, Platforms and Ecosystems (SiBW 2018), CEUR-WS.org, Espoo, 2018, pp. 177–192.

[6] E. Lindgren, J. Münch, Raising the odds of success: the current state of experimentation in product development, Information and Software Technology 77 (2016) 80–91. doi:10.1016/j.infsof.2016.04.008.

[7] A. Fabijan, P. Dmitriev, C. McFarland, L. Vermeer, H. Holmström Olsson, J. Bosch, Experimentation growth: Evolving trustworthy A/B testing capabilities in online software companies, Journal of Software: Evolution and Process 30 (12) (2018) e2113. doi:10.1002/smr.2113.

[8] H. H. Olsson, J. Bosch, From Opinions to Data-Driven Software R&D: A Multi-case Study on How to Close the 'Open Loop' Problem, in: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE, 2014, pp. 9–16. doi:10.1109/SEAA.2014.75.

[9] F. Fagerholm, A. Sanchez Guinea, H. Mäenpää, J. Münch, The RIGHT model for Continuous Experimentation, Journal of Systems and Software 123 (2017) 292–305. doi:10.1016/j.jss.2016.03.034.

[10] J. Melegati, X. Wang, P. Abrahamsson, Hypotheses Engineering : first essential steps of experiment-driven software development, in: IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE), 2019, pp. 16–19. doi:10.1109/RCoSE/DDrEE.2019.00011.

[11] S. Gottschalk, E. Yigitbas, G. Engels, Model-based hypothesis engineering for supporting adaptation to uncertain customer needs, in: B. Shishkov (Ed.), Business Modeling and Software Design, Springer International Publishing, Cham, 2020, pp. 276–286. doi:10.1007/978-3-030-52306-0_18.

[12] K. Power, Improving Flow in Large Software Product Development Organizations : A Sensemaking and Complex Adaptive Systems Perspective, Ph.D. thesis, NUI Galway (2019).

[13] S. Blank, B. . . B. Dorf, The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company, K & S Ranch, 2012.

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

[15] D. L. Frederiksen, A. Brem, How do entrepreneurs think they create value? A scientific reflection of Eric Ries' Lean Startup approach, International Entrepreneurship and Management Journal 13 (1) (2017) 169–189. doi:10.1007/s11365-016-0411-x.

[16] R. F. Bortolini, M. Nogueira Cortimiglia, A. d. M. F. Danilevicz, A. Ghezzi, Lean Startup: a comprehensive historical review, Management Decision (August) (aug 2018). doi:10.1108/MD-07-2017-0663.

[17] F. Dalpiaz, I. van der Schalk, S. Brinkkemper, F. B. Aydemir, G. Lucassen, Detecting terminological ambiguity in user stories: Tool and experimentation, Information and Software Technology 110 (December 2018) (2019) 3–16. doi:10.1016/j.infsof.2018.12.007.

[18] V. R. Basili, R. W. Selby, D. H. Hutchens, Experimentation in software engineering, IEEE Transactions on Software Engineering SE-12 (7) (1986) 733–743. doi:10.1109/TSE.1986.6312975.

[19] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.

[20] G. Schermann, J. Cito, P. Leitner, U. Zdun, H. C. Gall, We're doing it live: A multi-method empirical study on continuous experimentation, Information and Software Technology 99 (February) (2018) 41–57. doi:10.1016/j.infsof.2018.02.010.

[21] S. G. Yaman, F. Fagerholm, M. Munezero, J. Münch, M. Aaltola, C. Palmu, T. Männistö, Transitioning Towards Continuous Experimentation in a Large Software Product and Service Development Organisation – A Case Study, Vol. 10027, 2016, pp. 344–359. doi:10.1007/978-3-319-49094-6_22.

[22] F. Auer, M. Felderer, Current State of Continuous Experimentation: A Systematic Mapping Study, Proceedings of the 2018 44rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (2018) 335–344doi:10.1109/SEAA.2018.00062.

[23] M. Gutbrod, J. Münch, M. Tichy, How Do Software Startups Approach Experimentation? Empirical Results from a Qualitative Interview Study, in: M. Felderer, D. Méndez Fernández, B. Turhan, M. Kalinowski, F. Sarro, D. Winkler (Eds.), Product-Focused Software Process Improvement, Springer International Publishing, Cham, 2017, pp. 297–304. doi:10.1007/978-3-319-69926-4_21.

[24] H. H. Olsson, J. Bosch, Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation, in: Lecture Notes in Business Information Processing, Vol. 210, 2015, pp. 154–166. doi:10.1007/978-3-319-19593-3_13.

[25] J. Melegati, X. Wang, Hypotheses Elicitation in Early-Stage Software Startups Based on Cognitive Mapping, Springer International Publishing, 2020, pp. 211–220. doi:10.1007/978-3-030-49392-9_14.

[26] M. Gutbrod, J. Munch, M. Tichy, The Business Experiments Navigator (BEN), 2018 IEEE International

Conference on Engineering, Technology and Innovation, ICE/ITMC 2018 - Proceedings (2018). doi:10.1109/ICE.2018.8436389.

[27] M. Gutbrod, J. Münch, Teaching Lean Startup Principles : An Empirical Study on Assumption Prioritization, in: Software-intensive Business Workshop on Start-ups, Platforms and Ecosystems (SiBW 2018), 2018, pp. 245–253.

[28] J. Melegati, X. Wang, QUESt: new practices to represent hypotheses in experiment-driven software development, in: Proceedings of the 2nd ACM SIGSOFT International Workshop on Software-Intensive Business: Start-ups, Platforms, and Ecosystems - IWSiB 2019, ACM Press, New York, New York, USA, 2019, pp. 13–18. doi:10.1145/3340481.3342732.

[29] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, P. Abrahamsson, Software development in startup companies: A systematic mapping study, Information and Software Technology 56 (10) (2014) 1200–1218. doi:10.1016/j.infsof.2014.04.014.

[30] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, L. Jaccheri, Software startup engineering: A systematic mapping study, Journal of Systems and Software 144 (February) (2018) 255–274. doi:10.1016/j.jss.2018.06.043.

[31] S. Salomo, J. Weise, H. G. Gemünden, NPD planning activities and innovation performance: The mediating role of process management and the moderating effect of product innovativeness, The Journal of Product Innovation Management 24 (4) (2007) 285–302. doi:10.1111/j.1540-5885.2007.00252.x.

[32] S. H. Thomke, Managing Experimentation in the Design of New Products, Management Science 44 (6) (1998) 743–762. doi:10.1287/mnsc.44.6.743.

[33] W. R. Kerr, R. Nanda, M. Rhodes-Kropf, Entrepreneurship as Experimentation, Journal of Economic Perspectives 28 (3) (2014) 25–48. doi:10.1257/jep.28.3.25.

[34] J. Pantiuchina, M. Mondini, D. Khanna, X. Wang, P. Abrahamsson, Are software startups applying agile practices? The state of the practice from a large survey, in: Lecture Notes in Business Information Processing, 2017. doi:10.1007/978-3-319-57633-6_11.

[35] V. Garousi, M. Felderer, M. V. Mäntylä, Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, Information and Software Technology 106 (2019) 101–121. doi:10.1016/j.infsof.2018.09.006.

[36] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology 64 (2015) 1–18. doi:10.1016/j.infsof.2015.03.007.

[37] N. Tripathi, E. Klotins, R. Prikladnicki, M. Oivo, L. Pompermaier, M. Arun Sojan Kudakacheril, A., Unterkalmsteiner, K. Liukkunen, T. Gorschek, An anatomy of requirement engineering in software startups using multi-vocal literature and case survey, Journal of Systems and Software 146 (2018) 130–151. doi:10.1016/J.JSS.2018.08.059.

[38] S. S. Bajwa, X. Wang, A. Nguyen Duc, P. Abrahamsson, "Failures" to be celebrated: an analysis of major pivots of software startups, Empirical Software Engineering 22 (5) (2017) 2373–2408. doi:10.1007/s10664-016-9458-0.

[39] D. S. Cruzes, T. Dybå, P. Runeson, M. Höst, Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example, Empirical Software Engineering 20 (6) (2015) 1634–1665. doi:10.1007/s10664-014-9326-8.

[40] D. S. Cruzes, T. Dyba, Recommended Steps for Thematic Synthesis in Software Engineering, 2011 International Symposium on Empirical Software Engineering and Measurement (7491) (2011) 275–284. doi:10.1109/ESEM.2011.36.

[41] V. Garousi, M. V. Mäntylä, When and what to automate in software testing? A multi-vocal literature review, Information and Software Technology 76 (2016) 92–117. doi:10.1016/j.infsof.2016.04.015.

[42] I. Sommerville, Integrated requirements engineering: a tutorial, IEEE Software 22 (1) (2005) 16–23. doi:10.1109/MS.2005.13.

[43] A. Osterwalder, Y. Pigneur, Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, Wiley, 2013.

[44] A. Osterwalder, Y. Pigneur, C. L. Tucci, Clarifying Business Models: Origins, Present, and Future of the Concept, Communications of the Association for Information Systems 16 (July) (2005). doi:10.17705/1cais.01601.

[45] A. Maurya, Running lean : iterate from plan A to a plan that works, O'Reilly, Sebastopol, CA, 2012.

[46] A. Osterwalder, Y. Pigneur, G. Bernarda, A. Smith, Value proposition design: How to create products and services customers want, John Wiley & Sons, 2014.

[47] B. Nuseibeh, S. Easterbrook, Requirements engineering, in: Proceedings of the conference on The future of Software engineering - ICSE '00, Vol. 1, ACM Press, New York, New York, USA, 2000, pp. 35–46. doi:10.1145/336512.336523.

[48] P. Zave, M. Jackson, Four Dark Corners of Requirements Engineering, ACM Transactions on Software Engineering and Methodology 6 (1) (1997) 1–30. doi:10.1145/237432.237434.

[49] A. Davis, O. Dieste, A. Hickey, N. Juristo, A. Moreno, Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review, in: 14th IEEE International Requirements Engineering Conference (RE'06), IEEE, 2006, pp. 179–188. doi:10.1109/RE.2006.17.

[50] A. Sutcliffe, P. Sawyer, Requirements elicitation: Towards the unknown unknowns, 2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings (2013) 92–104doi:10.1109/RE.2013.6636709.

[51] A. Ferrari, P. Spoletini, S. Gnesi, Ambiguity and tacit knowledge in requirements elicitation interviews, Requirements Engineering 21 (3) (2016) 333–355. doi:10.1007/s00766-016-0249-3.

[52] P. Achimugu, A. Selamat, R. Ibrahim, M. N. R. Mahrin, A systematic literature review of software requirements prioritization research, Information and Software Technology 56 (6) (2014) 568–585. doi:10.1016/j.infsof.2014.02.001.

[53] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley Publishing Company, 1999.

[54] M. Cohn, User stories applied: For agile software development, Addison-Wesley Professional, 2004.

[55] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, S. Brinkkemper, The Use and Effectiveness of User Sto-

ries in Practice, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 9619, 2016, pp. 205–222. doi:10.1007/978-3-319-30282-9_14.

[56] Y. Wautelet, S. Heng, M. Kolp, I. Mirbel, Unifying and extending user story models, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8484 LNCS (2014) 211–225. doi:10.1007/978-3-319-07881-6_15.

[57] S. Maalem, N. Zarour, Challenge of validation in requirements engineering, Journal of Innovation in Digital Ecosystems 3 (1) (2016) 15–21. doi:10.1016/j.jides.2016.05.001.

[58] W. Maalej, M. Nayebi, T. Johann, G. Ruhe, Toward data-driven requirements engineering, IEEE Software 33 (1) (2016) 48–54. doi:10.1109/MS.2015.153.

[59] X. Franch, C. Ayala, L. López, S. Martínez-Fernández, P. Rodríguez, C. Gómez, A. Jedlitschka, M. Oivo, J. Partanen, T. Räty, V. Rytivaara, Data-driven requirements engineering in agile projects: the Q-rapids approach, Proceedings - 2017 IEEE 25th International Requirements Engineering Conference Workshops, REW 2017 (2017) 411–414doi:10.1109/REW.2017.85.

[60] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2) (2009) 131–164. doi:10.1007/s10664-008-9102-8.