A Systematic Mapping Study on the Use of Software Engineering Practices to Develop MVPs

Silvio Alonso, Marcos Kalinowski, Marx Viana, Bruna Ferreira, Simone D.J. Barbosa
Department of Informatics
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
{smarques, kalinowski, mleles, bferreira, simone}@inf.puc-rio.br

***Abstract*—[Background] Many startup environments and even traditional software companies have embraced the use of MVPs (Minimum Viable Products) to allow quickly experimenting solution options. The MVP concept has influenced the way in which development teams apply Software Engineering (SE) practices. However, the overall understanding of this influence of MVPs on SE practices is still poor. [Objective] Our goal is to characterize the publication landscape on practices that have been used in the context of software MVPs. [Method] We conducted a systematic mapping study using a hybrid search strategy that consists of a database search and parallel forward and backward snowballing. [Results] We identified 33 papers, published between 2013 and 2020. We observed some trends related to MVP ideation and evaluation practices. For instance, regarding ideation, we found six different approaches (e.g., Design Thinking, Lean Inception) and mainly informal end-user involvement practices (e.g., workshops, interviews). For evaluation there is an emphasis on end-user validations based on practices such as usability tests, A/B testing, and usage data analysis. However, there is still limited research related to MVP technical feasibility assessment and effort estimation. We also observed a lack of scientific rigor in many of the identified studies. [Conclusion] Our analysis suggests that there are opportunities for solution proposals to address gaps concerning technical feasibility assessment and effort estimation. Also, more effort needs to be invested into empirically evaluating the existing MVP-related practices.**

***Index Terms*—MVP, minimum viable product, software engineering, systematic mapping.**

I. Introduction

In Lean Startup [1], Eric Ries presented a methodology for allowing entrepreneurs to develop new products using validated learning about the customer. What initially began to be used in startups has gained popularity and nowadays many software companies are adopting parts of this methodology to achieve high levels of innovation. Its main focus relies on the identification and implementation of a product that adds real value to the customer [2]. Indeed, a key concept to Lean Startup is the Minimum Viable Product (MVP).

A recent systematic mapping about the term MVP [3] showed that there are many definitions; the most popular one was provided by Ries, who defines an MVP as "a version of a new product, which allows a team to collect the maximum amount of validated learning about customers with the least effort". That mapping also reported that the key factors related to MVPs are technical characteristics of the product and market and customer aspects, such as "minimum set of features", "customer feedback", and "minimum effort" [3].

Lean Startup advocates for building MVPs as experiments to perform the "build-measure-learn" cycle as fast as possible [1]. MVPs are being used in many different contexts, such as: startups [4], universities [5], industry-academia collaborations [6], [7], and established enterprises [8], [9]. Nevertheless, there is not much evidence on how this use is affecting software engineering practices. Lindgren and Münch [10] found that, even though the principles of continuous experimentation resonated with industry practitioners, the state of practice is not yet mature. Filling the gap between the theory in software engineering and the practical use of MVPs is a relevant issue that will benefit practitioners from both industry and academia.

In this paper, we synthesize existing work on practices employed in the development of software products using MVPs. We report on a secondary study yielding 33 selected publications to pursue our research objective that is to understand Software Engineering practices that have been used to develop software product MVPs. We characterize these practices by answering six Research Questions (RQs). RQs 1–4 are related to four software MVP development activities, respectively: ideation, technical feasibility assessment, effort estimation, and evaluation. RQs 5–6 are related to characterizing the reported evidence. The RQs are detailed in Section III.

Our results show that most papers concern practices regarding MVP ideation and evaluation. For ideation we found the use of several approaches (e.g., Design Thinking, Lean Inception) and an emphasis on informal end-user involvement practices (e.g., workshops, interviews) and lightweight documentation (e.g., features, user stories). Regarding evaluation there is an emphasis on end-user validations based on different practices (e.g., usability tests, A/B testing). Furthermore, we highlight gaps concerning technical feasibility assessment and effort estimation and a need for further empirical studies.

The remainder of this paper is organized as follows. Section II presents the background and related work. Section III presents the overall goal of our research and our research questions. Section IV defines the systematic mapping protocol. Section V presents the results, which are further discussed in

978-1-6654-2705-0/21/$31.00 ©2021 IEEE
DOI 10.1109/SEAA53835.2021.00017

Section VI. Finally, Section VII concludes the paper.

II. Background and Related Work

In this section, we outline the domain focusing on software engineering and the role of MVPs. Related work concerns other secondary studies conducted on closely related topics.

A. Software Engineering and MVPs

Software engineering is concerned with all aspects of software production, from the early stages of system specification to maintaining the system after it has gone into use [11]. As an engineering discipline, it focuses on applying the right theories, methods, and tools to get results with the required quality within the proposed schedule and budget.

The need for product development methodologies to be adapted to the scope, size, complexity, and changing requirements in the initial phase of a software project is widely recognized. However, there is still little guidance on how startups and innovation companies can adapt their process to rapid changes in context [12]. Many innovation companies carry out problem-solving experiments to learn more about different solution options, often leading to delivering a product that is very different from the original idea [13]. This learning is typically associated with the construction of a Minimum Viable Product (MVP), a lean version of the product to validate a new technology or elicit customer requirements [1], [13].

Eric Ries [1] proposed the Lean Startup methodology for business and product development. He defines a product creation process that can be summarized by combining business-oriented hypothesis experimentation and iterative product launches. Building a product iteratively, taking into account the needs of initial customers, can reduce risks such as expensive launches, failure to use, and low adherence. In Lean Startup, Build-Measure-Learn is the fundamental principle to transform ideas into products, measure how customers respond, and, finally, know whether to give up or persevere [1]. A fundamental concept of Build-Measure-Learn is building MVPs based on quick feedback obtained from the initial users. Ries [1] defines an MVP as: "a version of a new product, which allows a team to collect the maximum amount of validated learning from customers with minimum effort."

Several other definitions have been proposed [3], and practitioners and researchers often face the problem of selecting the most appropriate one. Moreover, the influence of MVPs on software engineering practices is still poorly understood.

B. Related Work

To the best of our knowledge, only one systematic mapping study was conducted focusing solely on MVPs, but we identified papers that present secondary studies concerning Lean Startup and MVP.

Lenarduzzi *et al.* [3] presented a mapping study about the different definitions of MVPs in the literature. They found 22 papers, proposed a classification schema for characterizing the definition of MVP in Lean Startups, and identified a set of common key factors in the MVP definitions. While we also focus on MVPs and build on their definitions, our study has a different scope, as we aim at characterizing practices that are being used in the context of MVP development.

Paternoster *et al.* [2] conducted a systematic literature review on software development work practices in startup companies. They identified 43 primary studies, of which only 16 were entirely dedicated to software development in startups. Moreover, only 9 studies exhibited high scientific rigor and relevance. The authors did not focus their discussions on the use of MVPs in the development of software systems.

Berg *et al.* [14] presented a systematic mapping on startup research from an engineering perspective, involving 27 papers published before 2017. They aimed to identify thematic concepts involved in startup research. Although startups are known as great adopters of MVP practices, the study findings cannot be generalized to other contexts. The widespread adoption of MVPs in different contexts requires a specific approach to better understand the various practices.

III. Research Goal and Questions

Our main research goal is to **characterize software engineering practices that have been applied to develop software MVPs.** Our research questions focus primarily on the fundamental activities of specification and validation:

RQ1. Which practices have been used to ideate MVPs? We aim to provide a clear view of one of the first activities of an MVP life cycle. Comparable to typical initial requirements elicitation efforts of traditional software products, ideation aims at defining the desired features of the MVP, outlining what the MVP will provide to the customer.

RQ2. Which practices have been used to assess the technical feasibility of MVPs? It is important to assure that the desired MVP features are technically feasible, *i.e.*, can be built. This task is somewhat comparable to assessments made during typical software engineering elaboration phases. Failing to assess the technical feasibility of any software project during its early stages might imply a waste of investment and commitment to initiatives that are doomed to fail.

RQ3. Which practices have been used to estimate the MVP building effort? We aim to provide an overview of practices that are employed to estimate MVP effort. As for any software product, from a business perspective, it is important to know how much effort an MVP requires to be built.

RQ4. Which practices have been used to evaluate MVPs? One of the main goals of an MVP is to collect validated learning from the customer. To this end, the evaluation of an MVP should collect data to generate insights from the customer's perspective. In software engineering, this is akin to continuous experimentation (*e.g.*, A/B Testing) [15].

RQ5. Which type of research has been conducted regarding MVPs? To classify the types of research identified, we adopted the scheme proposed by Wieringa et al. [16]: evaluation research, solution proposal, validation research, philosophical paper, opinion paper, and experience paper.

RQ6. Which types of empirical evaluation have been performed in research regarding MVPs? We aim to identify

the types of empirical evaluation (*e.g.*, case study, controlled experiment, survey) that have been performed, especially in evaluation or validation research papers.

IV. Systematic Mapping

A systematic mapping study is a form of secondary study that provides a systematic procedure and structure of the type of research reports and results that have been published, aiming to answer a particular research question [17]. This section presents our systematic mapping protocol, detailing the search strategy, study selection, and data extraction.

A. Search Strategy

A study comparing different search strategies to perform Systematic Literature Reviews (SLRs) in software engineering [18] found that using a hybrid strategy combining a database search on Scopus with parallel backward and forward snowballing (using Google Scholar) tends to present an appropriate balance between result quality and review effort.

Having this in mind, we adopted such hybrid search strategy, involving applying a search string on the Scopus database, filtering the results using our defined exclusion criteria, and performing parallel backward and forward snowballing iterations on the remaining studies [19] [18]. By following this parallel process, papers obtained by backward snowballing are not subject to forward snowballing, and vice-versa.

To design our search string for the database search, we used the PICO criteria [20] as follows. Population (P): Software; Intervention (I): Minimal Viable Product or MVP; Comparison (C): N/A; Outcome (O): N/A. Thus, our resulting search string was "'software' AND ('Minimum Viable Product' OR 'MVP')". It was applied to titles, abstracts and keywords in Scopus in March 2021.

B. Study Selection

Our study comprised papers published by the end of 2020. The study Inclusion Criterion (IC) and Exclusion Criteria (ECs) to filter the studies are presented in Table I.

TABLE I
Inclusion and Exclusion criteria

Criteria	Description
IC1	Describe practices related to the development of software MVPs
EC1	Just mention MVPs but do not satisfactory describe the used practices
EC2	Do not mainly comprise software development-related MVPs (e.g., mainly hardware or IoT related MVPs)
EC3	Not written in English
EC4	Not published in a peer-reviewed conference, journal, or workshop
EC5	Published after 2020

Our study selection process was performed by one researcher and revised by another. Whenever they disagreed or were in doubt on whether to include a paper, a third researcher was involved to reach consensus.

The search in Scopus returned a total of 223 studies. First, we applied the selection criteria to the title and abstract of all candidate studies, filtering papers that were barely related to the topic. Then, we applied the selection criteria to the full text, filtering them to obtain our seed set, formed by 27 studies.

The seed set was used as input to a parallel forward and backward snowballing. In the first backward snowballing iteration, we analyzed 496 papers and included three; in the second iteration, we analyzed 110 papers but included none. In the first forward snowballing iteration, we analyzed 313 papers and included two; in the second iteration, we analyzed five papers but included none. Together, the search and snowballing procedure resulted in 32 papers. The entire procedure is detailed in our online repository [1].

The authors were aware of one additional study that was not retrieved by the search strategy [7] and analyzed the reason for missing it. Although it concerned software-based MVPs and was indexed in Scopus, the paper did not use the term "software" in its abstract. Moreover, it was published late 2020 so it was not a candidate for inclusion through backward snowballing. Finally, while the paper reported case studies with practices applied to MVP development, it had a broader scope and therefore did not refer to specific MVP research literature, also preventing its inclusion through forward snowballing. That paper was included manually.

It is noteworthy that there may be other papers that we might have missed for similar reasons. Still, our analysis indicates that this paper represents a very specific and justifiable case. Therefore, we believe that our final set of included papers comprises a representative sample to provide comprehensive and meaningful answers to our research questions.

In total, 33 papers were included in this study. An overview of these papers, organized by research type ("PP" stands for "philosophical paper") and year, is presented in Table II.

C. Data Extraction

Data extracted from each paper involved some general fields (*e.g.*, year, authors), the answers to each research question, and the advantages and disadvantages reported for some of the practices. Table III shows the data extraction form, also available as a spreadsheet in our online repository [1].

V. Results

This section presents the mapping study results based on the information extracted from the 33 selected papers.

A. Overview of the selected papers

The list of selected papers is presented in the appendix. As Figure 1 shows, the selection process returned papers published between 2013 and 2020. It is possible to observe a concentration of research on the topic in recent years.

B. RQ1. Which practices have been used to ideate MVPs?

Overall, 73% (24 out of 33) of the papers referenced MVP ideation. Our observations here focus on: approaches; end-user involvement; additional sources of ideas; and business-related, visual prototyping, and documentation practices.

Approaches. The adoption of some approaches is generally perceived by the use of some of its practices. However, some

[1]https://doi.org/10.5281/zenodo.4718759

TABLE II
PAPER DESCRIPTIONS ORGANIZED BY RESEARCH TYPE AND YEAR

Type	Year	ID	Description
Evaluation Research	2020	S5	Comparison between SCRUM and Lean Inception in the initiation phase of a software project
		S25	Evaluation of interdisciplinary scenario-building
		S28	Analysis of challenges related to the steps of an MVP development
	2019	S6	Evaluation of the effects of elements from a startup ecosystem on MVPs
		S10	Practices to perform requirements gathering in software startups
	2018	S2	Investigation on how Lean internal startup facilitates software product innovation in large companies
		S14	Adaptation of Scrum to a product innovation context
		S15	Analysis of the relationship between hypotheses and MVPs in startups
		S32	Investigation of pivots from a Lean startup perspective
	2017	S1	Investigation on how software startups approach experimentation
		S20	Investigation of factors that influence the speed of prototyping in software startups
		S29	Approach to perform effort estimation for change requests
		S31	Approach to perform effort estimation for change requests
	2016	S22	Analysis of the MVP role in software startups
	2013	S24	Approach to create MVPs in industry-academia collaborations
Personal Experience	2020	S3	Development of a conversational agent (chatbot)
		S4	Lean UX development in a fintech context
		S19	Development of a digital tool for health care
		S30	Development of an app to support dental care of deaf people
	2019	S7	Development process of an MVP to assist firefighters
		S8	Lean method contribution to a User Experience testing experiment in an academic context
		S11	Development of an MVP in a healthcare environment
		S26	Approach to streamline the requirements engineering process of mobile applications
PP	2015	S23	Method to involve users to gain meaningful feedback and learning
Proposal of Solution	2020	S33	Lean R&D approach for digital transformation
	2019	S9	Approach to develop MVPs in established companies
		S12	Method to develop MVPs in software startups
	2018	S27	Analysis of a customer touchpoint model implementation in a software development process
	2017	S16	Approach to analyze user feedback on MVPs of mobile applications
		S17	Approach to adopt agile development on MVPs of mobile applications
		S18	Approach to analyze user feedback on MVPs of mobile applications
		S21	Approach to analyze user feedback on MVPs of mobile applications
	2016	S13	Approach to analyze user feedback on MVPs of mobile applications

studies referenced the used approaches directly, such as: Customer Development [S12, S28], Design Thinking [S2, S28], Human-centered Design [S19], Lean Inception [S5, S33], and User-centered Design [S3]. For instance, [S2] justifies complementing Lean Startup with Design Thinking, as the former does not provide an approach to ideation. In line with this need, Lean Inceptions were defined as the "combination of Design Thinking and Lean Startup to decide the Minimum Viable Product (MVP)" [21].

Paper [S25] presented its own ideation approach. It comprises "interdisciplinary scenario-building," a series of workshops involving different stakeholders, including end users, business people, and technical people, aiming to create and rank different customer scenarios. Throughout the process, user-personas are used to characterize end users. After ranking the scenarios, requirements are derived using a mapping process. The paper reported many advantages related to using these techniques, such as the elucidation of business, user, and technical needs from the outset of the project.

TABLE III
DATA EXTRACTION FORM

Information	Description
Study metadata	Name, Authors, and Year of publication
Practices used to ideate (RQ1)	Name or short description of the practices
Advantages and disadvantages	Advantages and disadvantages reported for the ideation practices
Technical feasibility assessment practices (RQ2)	Name or short description of the practices
Advantages and disadvantages	Advantages and disadvantages reported for the technical feasibility assessment practices
Effort estimation practices (RQ3)	Name or short description of the practices
Advantages and disadvantages	Advantages and disadvantages reported for the effort estimation practices
Practices used to evaluate the MVP (RQ4)	Name or short description of the practices
Advantages and disadvantages	Advantages and disadvantages reported for the evaluation practices
Type of research (RQ5)	Classification based on [16]: evaluation research, solution proposal, validation research, philosophical paper, opinion paper, or experience paper.
Type of empirical evaluation (RQ6)	Empirical evaluation type: controlled experiment, case study, survey, proof of concept, or none.

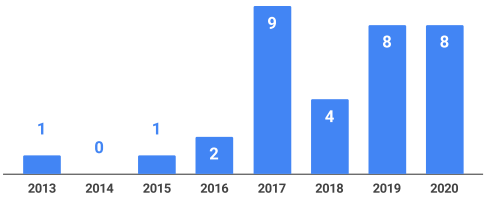

Fig. 1. Distribution of Papers per Year

End-user involvement practices. End users were reported as being actively involved in the ideation process in 13 studies [S2, S3, S6, S8, S9, S10, S12, S23, S25, S26, S27, S28, S33]. Besides directly involving end-user representatives in workshops [S2, S3, S4, S5, S12, S14, S26, S28, S33], as suggested by the previously mentioned approaches, the most cited method to actively involve end-users was through interviews [S2, S6, S8, S10, S23, S26, S27, S28], followed by surveys [S6, S23, S27], and user research [S8, S9, S23]. Moreover, brainstorming sessions [S6, S33], user observation [S2, S27], and focus groups [S26, S30] were also mentioned. For instance, in the rounds of focus groups at the beginning of each development cycle reported in [S30], end users participated alongside many other stakeholders to generate new ideas.

Additional sources of ideas. Besides the idea-gathering practices involving end users and multiple stakeholders, some papers [S6, S7, S15, S20, S26] reported the CEO/founders' vision, in a startup context, as the primary source of new product ideas and assumptions about customer problems [S15]. Paper [S26] also reported insights from industry experts. Some structured market research was used in [S1, S2, 27]. Paper [S2]

mentioned quantitative and qualitative research on existing solutions in the market and potential users, while paper [S1] described an ideation phase in which the employees' vision and customers' insights were used as inputs for the ideation.

Business-related practices. The use of visual business modeling practices was reported in two studies. Paper [S15] presented the use of the Business Model Canvas, and paper [S3] reported the use of Lean Canvas as a way of establishing a shared understanding of the problem, the target audience, and a possible solution.

Visual prototyping practices. Some visual prototyping practices were mentioned in the selected studies. For instance, sketching wireframes [S4, S15, S19, S20, S27, S33] and paper prototypes [S22, S23] were mentioned to create a common understanding about the MVP [S22, S23, S27].

Documentation practices. The main output of the ideation phase was documented mostly through features (sometimes listed as part of a Canvas, such as an MVP Canvas) and user stories [S3, S4, S11, S12, S19, S26, S27, S28, S33] – general explanation of a software feature written from the perspective of the end user. Sometimes such user stories are complemented by wireframes [S27, S33]. Paper [S27] also presented some additional documentation in the form of customer journeys and stakeholder mappings.

C. RQ2. Which practices have been used to assess the technical feasibility of MVPs?

Only three out of 33 (9%) of our selected papers referred to some kind of technical feasibility assessment. Paper [25] reported that technical people were involved in ideation workshops to filter technically infeasible scenarios, reducing risks and possible misunderstandings between stakeholders from different backgrounds. It is noteworthy that the involvement of technical stakeholders is also suggested by the aforementioned ideation approaches. More robust assessments are discussed in [S33] and [S14]. Paper [S33] suggests exploring the architecture through a "tracerbullet" strategy [22] to assess whether it is compatible and feasible and that there is a way to solve the problem with reasonable effectiveness, as well as providing a working, demo-able skeleton with some initial implementations. Paper [S14] takes this one step further, reporting a study to explore various software architectures before the MVP development process.

D. RQ3. Which practices have been used to estimate MVP building effort?

Only two papers directly addressed MVP effort estimation [S29, S31], both focusing on change impact analysis. Paper [S31] evaluates the performance of textual similarity techniques for MVP change impact analysis based on change requests. It uses data from two industrial projects of a Canadian startup. They found that combining textual similarity with file coupling improved impact prediction, and that effort could be predicted with reasonable accuracy (72% to 84%) using textual similarity only. In paper [S29], the same authors, using different methods for textual-similarity analysis, found that the combination of machine-learning techniques with experts' manually added labels had the highest prediction accuracy. According to the authors, better prediction of change impacts allow a company to optimize its resources and provide proper timing of releases for target MVPs.

These papers provide valuable results for change impact analysis. Nevertheless, we found no approach focusing on initial effort estimation for newly outlined MVPs. While the scope of an MVP is supposed to be minimal, typically concerning quick-win implementations, we still see value in its initial effort estimation. An option would be to consider agile estimation practices for user stories, such as planning poker. However, no such practice has been mentioned in the selected papers. It is noteworthy that the Lean Inception workshop has a proxy for effort called "waves", into which sets of features are sequenced. However, there is no contextualized meaning for such "waves" in terms of effort. What we have seen in practice (*e.g.*, [7]) is that feature roadmaps are usually built based on informal estimates. Possible reasons for the lack of MVP estimation techniques are explored in the next section.

E. RQ4. Which practices have been used to evaluate MVPs?

Considering our selected papers, 70% (23 out of 33) made some reference to evaluation practices. We discuss them hereafter semantically grouping them into practices of internal validation, validation with end-user, and automated feedback.

Internal validation practices. Only one study [S24] reported that the validation was performed purely internally. In this case, the Product Owner (PO) was personally responsible for evaluating the MVP. One of the main advantages reported was that it facilitated a continuous knowledge transfer between the evaluating side (PO) and the development team. Nevertheless, we emphasize the well-known importance of validations involving customers and end-users.

Papers [S14] and [S27] also suggested some form of internal validation. While [S27] reported that the internal validation should happen during internal team meetings, [S14] reported that, during alpha testing and internal review, the team figured out that its product could not be considered an MVP, as it was missing some critical features. Both of them proposed a later external validation with end users.

Validation with end users. Validations with end users were conducted in various stages: before the public release of the MVP [S7, S8, S9, S25, S33]; testing on site [S9]; or early on, through mock-ups and early-stage prototypes [S7, S33].

Several evaluation techniques were reported. Paper [S25] reported that workshops with end-users were organized to gather feedback for each implemented feature before their release, guiding the lean development. Papers [S8], [S11], and [S23] proposed the use of usability tests as controlled experiments. This technique was justified as a way to ensure the MVP's usability, ease of use, ease of learning, and satisfaction ([S11]). Papers [S4, S8] reported the use of Think Aloud Interview Testing before the release of the MVP.

Paper [S33] emphasized the use of continuous experimentation to test MVP-related business hypotheses. Testing these

hypotheses involved measuring metrics that reflect changes in business-related results, complemented by end-user questionnaires. Paper [S1] introduced a case that heavily relied on A/B testing and customer interviews, mentioning that the former can lead to conflicts in code and the latter demands a high level of motivation to keep conducting interviews. The A/B testing was also considered hard to extract value when the sample of end users available was small, making it difficult to perform statistical analyses. Even though it recognizes the value of A/B testing as a way to reveal complex knowledge of product usage, paper [S7] justifies its non-use by listing some negative points, such as: the need for a large user base, the use of vast resources to produce variations of the same feature.

Papers [S8], [S11], and [S30] conducted statistical usability tests. Six papers [S1, S2, S12, S16, S21, S32] reported that MVP evaluation was performed by analyzing data generated from the interaction with end users after the release. While paper [S12] stated that market response from the release was analyzed, the others reported more detailed processes, like face-to-face customer interviews [S2] and usage tracking and explicitly collected feedback [S16, S21]. Paper [S32] analyzed in a case study how pivots can be made in a mobile app based the number of downloads, ratings and reviews.

Automated Feedback. Four papers [S13, S16, S18, S21] describe a framework to automate mining of usage tracking and explicit feedback from end users, and reported that the framework allowed to easily capture usage trends.

F. RQ5. Which type of research has been conducted?

Figure 2 presents the distribution of research types reported in the selected papers. As depicted, *evaluation research* leads with 14 papers, followed by *proposal of solution* and *personal experience*, both with nine papers. *Philosophical paper* category was represented by only one study. This makes sense, as papers reporting industry cases or experiences, where MVPs are widely explored to deliver valuable products, typically involve evaluation research.

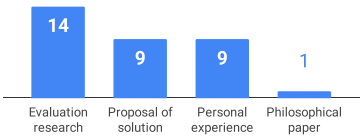

Fig. 2. Distribution of Research Type

G. RQ6. Which type of empirical evaluations have been performed?

Figure 3 presents the distribution of empirical evaluations reported. *Case study* evaluations were reported in 13 papers, while three *survey* evaluations and one *proof of concept* were reported. 16 papers did not report any kind of empirical evaluation. It is also worth noticing that no study presented a controlled experiment as an empirical evaluation.

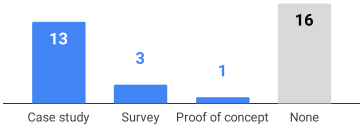

Fig. 3. Distribution of Empirical Evaluations

VI. Discussion

In this section we summarize our findings and discuss identified gaps and threats to validity.

A. Synthesis of Results

Our systematic mapping allowed identifying 33 studies concerning the development of software MVPs published by the end of 2020. We analyzed these studies to characterize the practices that have been used to ideate, assess the feasibility, estimate effort, and evaluate such MVPs. Furthermore, we analyzed the types of research and empirical studies conducted.

Figure 4 shows the mapping of the analyzed MVP activities against their research type and empirical evaluation type. It is easy to observe that the identified studies focus on MVP ideation (*cf.* Section V-B) and evaluation activities (*cf.* Section V-E). We also noticed that there are much more evaluation research papers than solution proposals. Indeed, many proposals in this area did not emerge from academia (*e.g.*, approaches such as Lean Startup, Lean Inception, and Design Thinking; and practices such as continuous experimentation, A/B testing) and academia has mainly assessed the use of such proposals and their related practices through case studies.

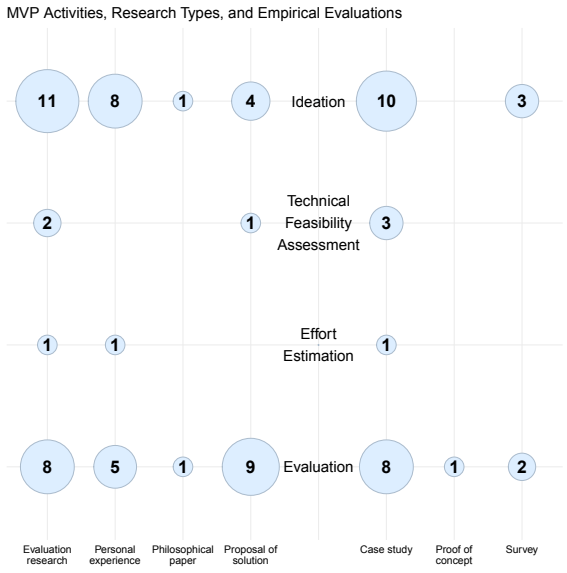

Fig. 4. MVP Activities, Research Types, and Empirical Evaluations

The rather scarce research concerning technical feasibility assessment and effort estimation is also worth mentioning.

Regarding technical feasibility assessment, this scarcity may be related to the technical simplicity of many MVP projects or to the fact that this activity is probably not receiving enough attention when developing software MVPs. However, not all MVPs are technically simple. In particular, in the digital transformation context, MVPs commonly involve applying new technologies in domains in which they have not been applied before. Failing to properly assess the feasibility may imply in wasted investments. Hence, there seems to be a need for solution proposals concerning lightweight software MVP technical feasibility assessment.

With respect to effort estimation, while there is some research on change impact analysis, there is an absence of proposals and evaluations concerning initial MVP effort estimation. This absence may be related to the intended minimal MVP scope. We hypothesize that the limited scope makes stakeholders feel more confident in informal estimates and in informally sketched feature development roadmaps. However, even for MVPs there may be strict time-to-market or cost constraints. Hence, effort estimation should not be neglected and effective techniques should be proposed (or adapted) and evaluated for the MVP context.

Finally, the absence of validation research and controlled experiments within the selected papers is an indication that academia might still be catching up on investigations concerning the MVP topic. Despite of the widespread use and importance of MVPs in the software industry - in the startup context and beyond - this topic still lacks research maturity.

B. Threats to Validity

In this section, we critically review our study regarding its threats to validity. We also address the efforts employed to mitigate each of its identified threats.

1) Internal Validity: To perform this mapping study, we followed the guidelines proposed by [17] and a hybrid search strategy that has been effective in identifying relevant software engineering research [18].

2) External Validity: Concerning the generalizability of the results, we invested effort in defining our protocol to enable reproducing our results. Moreover, the intermediate spreadsheets used to control the study selection and the extracted data from all the studies are available and auditable [1].

3) Reliability: The mapping protocol was discussed with other researchers. The study selection and extraction process was revised by an independent researcher to reduce the risk of bias during paper selection and the possibility of errors during data extraction. When we had some divergence, a third researcher was included in the discussion, so that it was possible to reach a consensus.

VII. Conclusion

In this paper, we presented the results of a systematic mapping study on the use of software engineering practices to develop MVPs. We identified and analyzed 33 papers, published between 2013 and 2020. From these papers, we extracted information related to practices concerning the MVP ideation, technical feasibility assessment, effort estimation, and evaluation. We also analyzed the type of research and the type of empirical evaluation.

Our results show that most papers presented practices regarding MVP ideation and evaluation. For ideation we found the use of several different approaches (e.g., Customer Development, Design Thinking, Lean Inception), with an emphasis on end-user involvement practices (e.g., workshops, interviews, surveys) and lightweight documentation (e.g., features, user stories, wireframes). Regarding evaluation there is an emphasis on end-user validations based on different practices (e.g., usability tests, A/B testing, usage data analysis). Another relevant finding is that there is limited research regarding MVP technical feasibility assessment and effort estimation. As future work, we plan to explore practices related to these two activities and analyze their use in this context through case studies. We hope that our results provide some guidance to practitioners when performing technical feasibility assessments and effort estimation of MVPs.

Finally, we observed an absence of validation studies and controlled experiments, which sheds light on a limited academia involvement in research on the topic. Indeed, the available evidence does not yet allow to draw precise conclusions on the benefits and limitations of the identified practices, which represents a road still to be paved by future research.

References

[1] E. Ries, *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Currency, 2011.

[2] N. Paternoster, C. Giardino, M. Unterkalmsteiner *et al.*, "Software development in startup companies: A systematic mapping study," *Information and Software Technology*, vol. 56, no. 10, pp. 1200–1218, 2014.

[3] V. Lenarduzzi and D. Taibi, "Mvp explained: A systematic mapping study on the definitions of minimal viable product," in *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2016, pp. 112–119.

[4] A. N. Duc and P. Abrahamsson, "Minimum viable product or multiple facet product? the role of mvp in software startups," in *International Conference on Agile Software Development (XP)*, 2016, pp. 118–130.

[5] M. M. Mardhia and E. D. Anggraini, "Implement a lean ux model: Integrating students' academic monitoring through a mobile app," in *International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, 2019, pp. 1–5.

[6] J. Münch, F. Fagerholm, P. Johnson *et al.*, "Creating minimum viable products in industry-academia collaborations," in *Int. Conf. on Lean Enterprise Software and Systems (LESS)*, 2013, pp. 137–151.

[7] M. Kalinowski, H. Lopes, A. F. Teixeira *et al.*, "Lean R&D: An agile research and development approach for digital transformation," in *International Conference on Product-Focused Software Process Improvement (PROFES)*, 2020, pp. 106–124.

[8] D. Dennehy, L. Kasraian, P. O'Raghallaigh *et al.*, "A lean start-up approach for developing minimum viable products in an established company," *Journal of Decision Systems*, vol. 28, no. 3, pp. 224–232, 2019.

[9] H. Edison, N. M. Smørsgård, X. Wang *et al.*, "Lean internal startups for software product innovation in large companies: enablers and inhibitors," *Journal of Systems and Software*, vol. 135, pp. 69–87, 2018.

[10] E. Lindgren and J. Münch, "Raising the odds of success: the current state of experimentation in product development," *Information and Software Technology*, vol. 77, pp. 80–91, 2016.

[11] I. Sommerville, *Software engineering, tenth edition*. Pearson, 2016.

[12] J. Pantiuchina, M. Mondini, D. Khanna *et al.*, "Are software startups applying agile practices? the state of the practice from a large survey," in *International Conference on Agile Software Development (XP)*, 2017, pp. 167–183.

[13] A. Nguyen-Duc, X. Wang, and P. Abrahamsson, "What influences the speed of prototyping? an empirical investigation of twenty software startups," in *International Conference on Agile Software Development (XP)*, 2017, pp. 20–36.
[14] V. Berg, J. Birkeland, A. Nguyen-Duc *et al.*, "Software startup engineering: A systematic mapping study," *Journal of Systems and Software*, vol. 144, pp. 255–274, 2018.
[15] B. Fitzgerald and K.-J. Stol, "Continuous software engineering: A roadmap and agenda," *Journal of Systems and Software*, vol. 123, pp. 176–189, 2017.
[16] R. Wieringa, N. Maiden, N. Mead *et al.*, "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," *Req. Engineering*, vol. 11, no. 1, pp. 102–107, 2006.
[17] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.
[18] E. Mourão, J. F. Pimentel, L. Murta, M. Kalinowski, E. Mendes, and C. Wohlin, "On the performance of hybrid search strategies for systematic literature reviews in software engineering," *Information and Software Technology*, vol. 123, p. 106294, 2020.
[19] E. Mourão, M. Kalinowski, L. Murta, E. Mendes, and C. Wohlin, "Investigating the use of a hybrid search strategy for systematic reviews," in *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2017, pp. 193–198.
[20] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.
[21] P. Caroli, "Lean inception: how to align people and build the right product," *Editora Caroli*, 2018.
[22] D. Thomas and A. Hunt, *The Pragmatic Programmer: your journey to mastery*. Addison-Wesley Professional, 2019.

APPENDIX

[S1] M. Gutbrod, J. Münch, and M. Tichy, *"How Do Software Startups Approach Experimentation? Empirical Results from a Qualitative Interview Study,"* in Product-Focused Software Process Improvement (PROFES), 2017.

[S2] H. Edison, N. M. Smørsgård, X. Wang *et al.*, *"Lean Internal Startups for Software Product Innovation in Large Companies: Enablers and Inhibitors,"* Journal of Systems and Software, vol. 135, 2018.

[S3] E. Ruane, R. Smith, D. Bean *et al.*, *"Developing a conversational agent with a globally distributed team: an experience report,"* in Int. Conf. on Global Software Engineering (ICGSE), 2020.

[S4] A. A. Hendriadi and A. Primajaya, *"Optimization of financial technology (fintech) with lean UX development methods in helping technical vocational education and training financial management,"* Conference Series Materials Science and Engineering, vol. 830, 2020.

[S5] I. Braga, M. Nogueira, N. Santos *et al.*, *"Does the Lean Inception Methodology Contribute to the Software Project Initiation Phase?,"* in Computational Science and Its Applications (ICCSA), 2020.

[S6] N. Tripathi, M. Oivo, K. Liukkunen, *et al.*, *"Startup ecosystem effect on minimum viable product development in software startups,"* Information and Software Technology, vol. 114, 2019.

[S7] N. Shaghaghi, S. Patel, B. Pabari *et al.*, *"Implementing Communications and Information Dissemination Technologies for First Responders,"* in Global Humanitarian Technology Conference (GHTC), 2019.

[S8] M. M. Mardhia and E. D. Anggraini, *"Implement a Lean UX Model: Integrating Students' Academic Monitoring through a mobile app,"* in International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA), 2019.

[S9] D. Dennehy, L. Kasraian, P. O'Raghallaigh *et al.*, *"A Lean Start-up approach for developing minimum viable products in an established company,"* Journal of Decision Systems, vol. 28, no. 3, 2019.

[S10] R. Chanin, L. Pompermaier, A. Sales, *et al.*, *"Collaborative Practices for Software Requirements Gathering in Software Startups,"* in International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 2019.

[S11] A. Reis, F. Coutinho, J. Ferreira *et al.*, *"Monitoring System for Emergency Service in a Hospital Environment,"* in Portuguese Meeting on Bioengineering (ENBENG), 2019.

[S12] L. Pompermaier, R. Chanin, A. Sales *et al.*, *"MVP Development Process for Software Startups,"* in International Conference on Software Business (ICSOB), 2019.

[S13] K. Holl, F. Elberzhager, and C. Tamanini, *"Optimization of mobile applications through a feedback-based quality assurance approach,"* in International Conference on Mobile and Ubiquitous Multimedia (MUM), 2016.

[S14] A. Abdul, J. M. Bass, H. Ghavimi *et al.*, *"Product innovation with scrum: A longitudinal case study,"* in International Conference on Information Society (i-Society), 2017.

[S15] D. Khanna, A. Nguyen-Duc, and X. Wang, *"From MVPs to Pivots: A Hypothesis-Driven Journey of Two Software Startups,"* in International Conference on Software Business (ICSOB), 2018.

[S16] S. A. Scherr, F. Elberzhager, and K. Holl, *"An Automated Feedback-Based Approach to Support Mobile App Development,"* in Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2017.

[S17] M. Bauer, K. Sergieieva, and G. Meixner, *"Enabling Focused Software Quality Assurance in Agile Software Development Processes for Mobile Applications using Text and Usage Mining Methods:,"* in International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), 2017.

[S18] F. Elberzhager, K. Holl, B. Karn *et al.*, *"Rapid Lean UX Development Through User Feedback Revelation,"* in Product-Focused Software Process Improvement (PROFES), 2017.

[S19] S. M. Haddad, R.T. Souza, J.G. Cecatti *et al.*, *"Building a Digital Tool for the Adoption of the World Health Organization's Antenatal Care Recommendations: Methodological Intersection of Evidence, Clinical Logic, and Digital Technology,"* Journal of Med. Int. Res., vol. 22 (10), 2020.

[S20] A. Nguyen-Duc, X. Wang, and P. Abrahamsson, *"What Influences the Speed of Prototyping? An Empirical Investigation of Twenty Software Startups,"* in XP Conference Workshop on Agile Processes in Software Engineering and Extreme Programming, 2017.

[S21] F. Elberzhager and K. Holl, *"Towards Automated Capturing and Processing of User Feedback for Optimizing Mobile Apps,"* Procedia Computer Science, vol. 110, 2017.

[S22] A. N. Duc and P. Abrahamsson, *"Minimum Viable Product or Multiple Facet Product? The Role of MVP in Software Startups,"* in XP Conference Workshop on Agile Processes in Software Engineering and Extreme Programming, 2016.

[S23] L. Hokkanen and M. Leppänen, *"Three patterns for user involvement in startups,"* in European Conference on Pattern Languages of Programs (EuroPLOP), 2015.

[S24] J. Münch, F. Fagerholm, P. Johnson *et al.*, *"Creating Minimum Viable Products in Industry-Academia Collaborations,"* in International Conference on Lean Enterprise Software and Systems (LESS), 2013.

[S25] I. Jennes, M. Friedrich, J. Van der Bank *et al.*, *"The benefits of interdisciplinary scenario-building for hybrid radio applications,"* Telematics and Informatics, vol. 54, 2020.

[S26] P. Collins, N. Shaghaghi, and S. Lanthier, *"Developing High-Value Technologies for First Responders,"* in 2019 IEEE Global Humanitarian Technology Conference (GHTC), 2019.

[S27] T. Sauvola, M. Kelanti and J. Hyysalo, *"Continuous Improvement and Validation with Customer Touchpoint Model in Software Development."* in Int. Conf. on Software Engineering Advances (ICSEA), 2018.

[S28] L. Pompermaier and R. Prikladnicki, *"Brazilian Startups and the Current Software Engineering Challenges: The Case of Tecnopuc,"* in Fundamentals of Software Startups, 2020.

[S29] M. Nayebi, S. J. Kabeer, G. Ruhe, C. Carlson, and F. Chew, *"Hybrid Labels Are the New Measure!,"* IEEE Software, vol. 35, no. 1, 2018.

[S30] V. Campos, R. Cartes-Velásquez, and C. Bancalari, *"Development of an app for the dental care of Deaf people: Odontoseñas,"* Universal Access in the Information Society, vol.19, 2018.

[S31] S. J. Kabeer, M. Nayebi, G. Ruhe *et al.*, *"Predicting the vector impact of change: an industrial case study at brightsquid,"* in Int. Symposium on Empirical Software Engineering and Measurement (ESEM), 2017.

[S32] J. Ochoa-Zambrano and J. Garbajosa, *"An analysis of the bluetooth terminal development pivots from lean startup perspective: experience and lessons learnt,"* in XP Conference Scientific Workshops, 2017.

[S33] M. Kalinowski, H. Lopes, A. F. Teixeira *et al.*, *"Lean r&d: An agile research and development approach for digital transformation,"* in International Conference on Product-Focused Software Process Improvement (PROFES), 2020.