Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
A decision framework for software startups to succeed in COVID-19 environment

Sudhaman Parthasarathy
Thiagarajar College of Engineering, Madurai, India

ARTICLE INFO

Keywords:
Software project
Decision framework
Project risk
Software startup

ABSTRACT

The world is witnessing a pandemic of a severe infectious nature known as COVID-19. Many countries have used partial or complete lockdown to slow the spread of the virus, which has devastated the economies all over the world, and made hundreds of millions of people unemployed overnight. The lockdown measures and work from home practices as a response to the spread of the new coronavirus threaten the existence of many innovative startups. In fact, many software startups are either with no projects on hand and only a few of them are able to attract customers for even maintenance projects. In this paper, we provide a decision framework to help software startups choose projects which will increase their chance of success given their historically low success rate and diminished level of IT investment.

1. Introduction

Startups, a form of entrepreneurship, play an important role in providing solutions to both existing and novel problems in innovative ways and are vital for growth of individual country and global economies. They are also touted to provide novel solutions to “wicked” societal problems [1]. Growing economies in the current COVID-19 constrained environment may qualify as a “wicked” problem. Unfortunately, most startups fail. Startup Genome [2019] [2] reports that only 1 in 12 startups succeeds, making their failure rate to be higher than 90 percent. Due to COVID-19, there also has been a massive layoff of employees and a reduction in salary across industries all over the world. Level of information technology investment, particularly for startups, has also fallen dramatically [3,4].

Software startups are those companies which deliver software products or services to address both existing and new market needs and problems and/or to take advantage of novel market opportunities while operating in uncertain market conditions with limited skilled employees and restricted IT resources [5]. They are characterized by operating in highly uncertain, dynamic, unpredictable, and even chaotic market environment, evolving rapidly, acting quickly, failing fast, and learning faster to find a sustainable market niche [6]. As a consequence, software startups do not follow rigid software development practices or models such as ISO or CMM standards, but use agile and lean methodologies, as their process management has to be agile, evolutionary, and opportunistic [7]. The COVID-19 environment is likely to bring in these technologies, as their process management has to be agile, evolutionary, and opportunistic [7]. The COVID-19 environment is likely to bring in these technologies, as their process management has to be agile, evolutionary, and opportunistic [7]. The COVID-19 environment is likely to bring in these technologies, as their process management has to be agile, evolutionary, and opportunistic [7]. The COVID-19 environment is likely to bring in these technologies, as their process management has to be agile, evolutionary, and opportunistic [7]. The COVID-19 environment is likely to bring in these technologies, as their process management has to be agile, evolutionary, and opportunistic [7]. The COVID-19 environment is likely to bring in these technologies, as their process management has to be agile, evolutionary, and opportunistic [7].

Software startups are those companies which deliver software products or services to address both existing and new market needs and problems and/or to take advantage of novel market opportunities while operating in uncertain market conditions with limited skilled employees and restricted IT resources [5]. They are characterized by operating in highly uncertain, dynamic, unpredictable, and even chaotic market environment, evolving rapidly, acting quickly, failing fast, and learning faster to find a sustainable market niche [6]. As a consequence, software startups do not follow rigid software development practices or models such as ISO or CMM standards, but use agile and lean methodologies, as their process management has to be agile, evolutionary, and opportunistic [7].

There is always a silver lining in the darkest of the clouds, though. Due to COVID-19, work from home (remote working model) has been used by IT companies for many years to reduce cost in terms of physical office space and travel. This has also proven to be essential for them to keep their operations going during lockdown. Fortuitously, it is also probably the best way to implement social distancing, one of the constraints within which one has to operate in the COVID-19 environment [9].

As there is no 100% safe COVID-19 vaccine in the market, social distancing and “working from remote locations” are being used by most IT companies to work on various projects. Even after the pandemic, the IT services companies and software startups are much interested in operating in remote mode as a cost cutting measure for some of their projects. To manage this novel situation, some of these companies have proactively created collaborative IT platforms supported by cloud enabled IT infrastructure and robust security policies to help their various teams continue working, while others are using similar platforms provided by their vendors. Worldwide lockdowns have had dramatic effect on all kinds of pollution almost everywhere on the planet. In many cities, their inhabitants have experienced this reduction in pollution first time in their lives and are rejoicing in it. It is natural that they would like to keep this lower level of pollution when lockdown will be eased out or completely removed [4]. This expectation of lower pollution level post COVID-19 provides both constraints and opportunities to all stakeholders of any economic endeavor including software startups.

A green recovery would require reduction of greenhouse emission, a constraint. In IT related pollution, data centres alone contribute to 2% of all greenhouse emission per year which is larger than entire pollution from the transport sector, and a constraint. In IT related pollution, data centres alone contribute to 2% of all greenhouse emission per year. A green recovery would require reduction of greenhouse emission, a constraint. In IT related pollution, data centres alone contribute to 2% of all greenhouse emission per year. A green recovery would require reduction of greenhouse emission, a constraint. In IT related pollution, data centres alone contribute to 2% of all greenhouse emission per year. A green recovery would require reduction of greenhouse emission, a constraint. In IT related pollution, data centres alone contribute to 2% of all greenhouse emission per year. A green recovery would require reduction of greenhouse emission, a constraint.
aviation industry, though massive efficiency gains by data centres have kept their energy consumption roughly the same over the past decade [10]. If Bitcoin gets adopted at the same rate as past broadly adopted technologies have been, it alone would increase the warming of the planet by more than 2 °C in less than three decades [11]. As pointed out earlier, dramatically reduced IT investment for startups provides another constraint for software startups either in getting new projects or managing ongoing maintenance projects [12,13].

Though there is no area of global economy which is untouched by IT, it has the most leverage in processes where there are few physical components. One can argue that for green recovery, software startups may want to target those industries where processes are dominated by physical components, rather than the ones whose processes are dominated by digital components, and where the energy efficiency gains have not been yet realized with increase in the scale of production and use. In industries where there is predominance of digital components, opportunities may lie in making further efficiency gains, as low hanging opportunities may already have been exploited.

Limited skilled employees and restricted IT resources [7] remain the main challenges faced by software startups, compounded by transformation of the working norms and culture from somewhat controlled official physical setup to remote home offices because of COVID-19, as colocation has been a vital element in new product development. It should be noted that most established IT companies have been allowing their employees for a long time to work remotely to save office space and time and expense on transportation, and is not a new constraint. The turnaround times for startups for solutions to client problems and opportunities have to be shorter; otherwise a potential revenue generating opportunity may be quickly lost because of sheer competition and a narrow window of opportunity [14].

The objective of this paper is to understand as how best software startups can succeed in the current environment and in turn can help grow economies. For this purpose, we conducted a thorough review of literature on software startups from the perspective of project management especially during this COVID-19 environment and accordingly conducted a case study at a software startup as per the guidelines provided by case study research methodologist [15]. The outcome of the case study is a decision framework for software startup. This decision framework is meant to help software startups choose projects which will increase their chance of success given their historically low success rate and diminished level of IT investment. We draw on the existing literature along with a case study in a software startup to come up with our framework. The usability of framework has also been discussed followed by its limitations. The contribution of this research work is: At first, we present a decision framework for software startup to either attract new projects or ongoing maintenance projects based on the project risk and the maturity of the startup. Second, this paper also features the usability of the framework to guide the practitioners of the software startup companies on the application of the framework to their software projects effectively.

The rest of this research paper is organized as follows: In Section 2, we provide background and related work on the software startups and their challenges during this COVID-19 scenario and outline our research objective. In Section 3, the case study research methodology and the usage of Nominal Group Technique (NGT) have been discussed. Section 4 presents the development of the proposed decision framework for software startup companies. In Section 5, the usability of framework is presented. The paper concludes in Section 6 with contributions to theory and practice.

2. Background and related research work

In spite of the huge importance of startups to the economy, there is little empirical work on success factors of new startups, though there are many on the pre-startup processes [16,17]. It should be noted that there is no agreement of the definition “success” of a startup in the literature [14,18,19]. In this paper, we opt for measuring startup success by the return on investment, consistent with Casusmano [20] in the startup literature and DeLone and McLean [21] in the broader IT literature. In a review of critical success factors (CSFs) of IT startups, Santisteban and Mauricio [18] identified 21 CSFs and grouped them into three categories, organizational, individual and external.

Organizational factors define the overall characteristics of the startup: number of founding members of the startup team, years in operation since its creation, product/service innovativeness, geographical proximity to suppliers and customers, clustering of startups working in collaboration in the same domain, and partnership with external entities. Individual factors define the characteristics of the founder and/or the founding team (experience in the relevant industry, previous startup experience, academic preparedness, technological and/or business capabilities, R&D experience, management experience, leadership, gender, age, and motivation) [14]. External factors define the environment in which the startup operates such as government support during the seed phase, venture capital during growth and expansion phases, intensity of competition among startups within the same industry, dynamism of the external environment, and science and technology law [16]. A recent work by Melegati et al. [6] more or less concurs with the above findings.

Santisteban and Mauricio [18] also summarized stages of startup development into four: seed, early, growth and expansion. In the seed stage, also called as preparation and emergence, the startup team is small, ideas are sharpened, business models are refined, and funds from government seed money, friends and relatives, and small investors who believe in their future potential are used [13,22–24]. In the early or young stage, the products/services are made available in the market and continuously innovated based on feedback from real and potential customers [19,25,26]. In the growth stage, also called as early growth [22], growing [26,27], and growth and development [25], the business model has been perfected, market share is increasing, and venture investment firms start financing. In the expansion stage, also called mature [26,27], the continuous external financing becomes essential, and strategic alliances are established to increase the reach of the products/services to other markets. Some combine seed and early stage together as incubation stage [27,28], and growth and expansion together as post-incubation stage [28].

Since software is the main product of software startups, it is also important to examine what makes software projects successful in startups. Success of software projects have been traditionally measured along quality, time, scope, and cost dimensions. Software startups choose software engineering methodologies (mostly, agile or lean) opportunistically, and adapt and configure them to provide value within the constraints they have to operate [29]. Pantiuchina et al. [23] found that startups which use agile methodology, use speed related agile practices more widely than quality related practices, while they use daily standup meeting the least. They also found that startups using lean methodology do not sacrifice quality for speed more than others.

One of the major reasons of the failure of software startup is the failure of their product, i.e., software, they produce to meet potential customer demand to solve an existing or assumed problem. The products fail mostly because of inadequacies in requirements engineering [30]. Because of lack of customers initially, startups often invent requirements and validate them through frequent product releases. As product engineering consumes substantial resources in startups [5], problems in engineering practices (software requirements engineering, software design, and professional practice) lead to sub-optimal products, wasted resources and missed market opportunities, affect all aspects of company, and pose a substantial risk to the very existence of the company [30]. It is obvious that software startups must carefully navigate product engineering minefield too to succeed in the current environment, in addition to the other constraints stated earlier.

Previous research works by Zielske and Held [31] and Melegati et al. [7] on software process improvement for smaller firms involved
in software development points out that the standard-driven existing software process initiatives and implementation could not be achieved with smaller or medium firms. They suggest that a much more tailored and simplified decision support framework is required for either operating new software projects or managing ongoing projects using regular mechanism or remote mode.

We now state our research goals as follows:
1. To understand how does software startups operate and manage projects during COVID-19 environment
2. To develop a decision framework to help startups choose projects which will increase their success rate given the constraints trivial for startups during the current pandemic season and also post pandemic scenario.

3. Case study research methodology

We now describe our case study research methodology to develop the proposed decision framework for software startups. Our intention was to setup an early evaluation case study [32] during COVID-19 environment to understand as how startups cope up with and innovate in such an environment to succeed. Research methodologists (e.g., Wieringa and Daneva [32]) suggest that an early evaluation case study is the first step to incrementally match up and understand a new method or approach to the conditions of practice.

The case study methodology is well suited for many kinds of software engineering research, as the objects of study are contemporary phenomena, which are hard to study in isolation. Such a study seeks to demonstrate the application of the approach in a real-world context, so that both researchers and practitioners involved in trying out the new approach can learn from the experience and may come up with a list of desired properties for the proposed approach to consider while improving, enhancing or refining it. Our choice for case study methodology is justified by our intention to explore a real-life phenomenon in the context in which it happens. A case study is a particularly suitable research method in situations in which researchers studying the phenomenon seek to analyse the practice. In our case study, we expected to learn startup's subjective assessment of its maturity and accordingly it is potential to handle software projects with varying level of risks in the context of COVID-19 environment.

Our case study research is exploratory in nature. Its objective is to develop a decision framework using a practical setting which is transparent to the readers. To meet this goal, we conducted a case study in a software startup (pseudonym 'Mave Start') using a case study research method suggested by Yin [15]. Armed with our understanding of prior work on startup's success, we conducted an early evaluation case study [32] in 'MaveStart', in India, in the first half of 2020. We conducted the case study by: (1) recruiting project managers/team leads from the startup to help collect data, (2) arranging for informal interactions, followed by semi-structured interviews [33] to help them understand the Nominal Group Technique (NGT), (3) analysing the findings; (4) writing-up the results.

3.1. Nominal Group Technique (NGT)

Nominal Group Technique (NGT) [34] is a structured methodology that enables the assimilation of ideas and judgments of knowledgeable individuals towards building a group consensus for a desired outcome. With NGT, individuals are expected to generate ideas separately and anonymously, without input from their peers, by writing them on a piece of paper or typing them electronically. At the end of the session, the set of ideas from the group are extracted by merging together the pieces of paper or electronic contents. A key limitation of the NGT [34] is that study participants cannot build on one another's ideas because they never share their ideas during the session. The application of NGT does not depend on the size of the case study organization, but the size of the group. Normally, the NGT is conducted with a group size of 8–10 individuals, with the larger group consisting of 22–24 individuals.

The steps in the NGT [34] are as listed below: (i) Each one of the team members chosen for carrying out the NGT from the case study organization (in this case, it is a software startup) writes down the capability of the startup using which they assess the maturity of their startup. One of them will be designated as a team leader. (ii) Each member writes down the capability of the startup but does not discuss among themselves. The team leader then records all the points prepared by the team members. (iii) The team leader thereafter asks each team member to elucidate his points. (iv) As a next step, the team leader asks each member to write down the points that seem most significant. The team leader records the number of people who believe each criterion to be a priority. As a last step, team members rate each of the points from no importance (0) to top priority (10). The team leader then collects and adds the ratings and presents the cumulative rating for each of point that reflects the capability of the startup.

The NGT was used to find out the capabilities and characteristics of the software startup involved in our case study by means of listing out their significant features and strengths. The NGT indirectly supported the case study organization in two ways. At first, it precisely helped the case study team members to capture the strengths and the potential of the startup. Secondly, it brought a consensus among the team members to assess the maturity of the software startup as either low or high. Only if we are able to have such as an assessment on the maturity of the startup, we would be able to apply the framework to the case study organization for their utilization. The bottom-line for the use of NGT in this scenario was to outline the significant features of the startup, and the potential with which the startup had managed to overcome the challenges faced by them. This would help the members to collectively assess its maturity.

'Mave Start' was founded in 2014 with 4 individuals in India, by reaching out to potential customers for developing new innovative software products to meet their current requirements. Over the years, it has been able to complete a majority of its projects successfully, and had grown to 32 employees at the beginning of 2020. Fifty percent of its employees have an experience of 2–6 years while others have 7–13 years. It started out by developing innovative IT products with cutting edge technologies for food processing and aerospace industries. Since last year, it has been offering cloud, mobile and on-premise solutions using AI and Data analytics for various manufacturing companies. Since the start of COVID-19 spread, it has had hard time to even receive an acknowledgement for its proposals sent to various existing and potential customers. Since the last week of February 2020, it has been working only on two of its ongoing smaller projects received from its new customers. During the period of this case study, it was on constant look out for more projects to keep itself financially viable. In conducting this case study our goal was to find out factors which are allowing this startup to survive in the current COVID-19 constrained environment and would potentially, we hope, other startups to survive and even flourish. We initially interacted and conducted ‘semi-structured’ interviews with 8 of their senior level employees (3 project managers, and 5 team leads). Then we asked each of them to write-down the characteristics of the startup which allowed them carry out various software projects. We subsequently used nominal group technique (NGT) to build consensus among these participants.

We briefly reproduce below some of the statements made by the participants:

(i) "For the past 6 years, ‘Mave Start’ is successfully delivering new products across domains such as sea food processing and aerospace and recently logistics and supply chain too. It was able to respond to all their customers’ orders on-time".

(ii) "Customers of our previously completed projects were satisfied with our products. This is evident from the fact that these customers are still using our implemented products without any serious maintenance issue as well as continuing relationship with us for future up-gradation of the delivered products".
(iii) “We have sufficient employees well trained to work on cutting-edge technologies to keep us up with the demand from the customers to offer cloud, mobile or on-premise software products”.

(iv) “We are growing year by year, which is indicated by our increasing number of successfully delivered products across various domains and increased employee strength”.

(v) “Two of our earlier projects witnessed a sudden marginal cut in the budget, followed by a request for earlier delivery of the finished products thereby forcing us to compress project schedule and some project tasks. However, we managed it well to complete those projects on-time and delivered the product successfully to the customers”.

The outcome of the NGT was the consensus among the 8 case study participants that can be captured by two factors, startup maturity and project risk. It was the group’s consensus that the ability of the startup by using its maturity to select projects of appropriate risk allowed it to thrive in past and so far, whether the COVID-19 storm is present or not. They discounted the use of green software projects as a general success factor, and suggested that software startups may choose to limit themselves to only green projects, though they warned that doing so may increase their risk of failure in the current environment of reduced IT investment for startups.

4. Proposed software startup decision framework

The term ‘startup maturity’ can be described as the overall capability of the startups to produce necessary software to meet customer and/or market needs in a given context, and can be considered to include more or less all the success factors identified in the “Prior Work” section above [6]. In a nutshell, it is their ability to adapt new engineering practices and reactivity to keep up with emerging technologies and markets [5]. The risks may arise from change in roles and responsibilities of team members, organizational structures, business processes, technical factors, and/or financial factors during the execution of the project [9].

In the context of projects, the case study helped us identify four types: ‘new projects’, ‘maintenance projects’, ‘partially cancelled projects’ and ‘outrightly cancelled projects’. The term ‘new projects’ refers to the newly commenced software projects which are in the initial phase of the project lifecycle. The ‘maintenance projects’ refers to those software projects in which changes are being made in the previously delivered software products upon specific request from the customer.

The ‘partially cancelled projects’ refers to those projects where the customers may have initially required multiple software modules for multiple core functionalities of their organization, however, as a cost cutting measure or for some other technical/business constraints, they now require delivery of one or just a few of their highest priority standalone modules. The ‘outrightly cancelled projects’ refer to those projects which were not completed. It is also possible that they would have been able to remain financially viable in adverse conditions. These kinds of startups would be able to succeed on all four types of projects, but their payoff would be probably high on ‘Maintenance Projects’ as they would have probably produced and implemented them in the first place, and would have know-how to complete them with much less resource commitments.

For ‘New Projects’, they would be able to utilize their experience on past projects in streamlining various tasks and activities and probably reusing some of the modules developed for them. Such projects would still provide high payoffs, but probably not as high as ‘Maintenance Projects’. Though, they would still succeed on previously “Partially Completed Projects’ and ‘Outrightly Cancelled Projects’, the risk would be higher as they would have to forecast the work of prior developers in fulfilling the project requirements, which may not be streamlined with their own expertise and then rework all those requirements which were not completed. It is also possible that they would have to rework previously completed features/modules as it would be probably more difficult to integrate their work on uncompleted features/modules with the completed features/modules by previous developers. To increase their chance of success, software startups with low maturity should select ‘New Projects’. Such startups would be just starting out and may not have had enough experience to weather adverse conditions. If they are a few years old, they may not have learned lessons adequately from past failures. They would be better off not working on the remaining three types of projects until they have accumulated both experience and financial chest.

5. Usability of software startup decision framework

Startups with high level of maturity would have been in existence for many years. They would have worked on projects which could be categorized as both successful and unsuccessful [17]. They would have learned lessons from their both successful and unsuccessful efforts in terms of what made them succeed on certain projects, while fail on others. They would have been able to remain financially viable in adverse conditions. These kinds of startups would be able to succeed on all four types of projects, but their payoff would be probably high on ‘Maintenance Projects’ as they would have probably produced and implemented them in the first place, and would have know-how to complete them with much less resource commitments.

For ‘New Projects’, they would be able to utilize their experience on past projects in streamlining various tasks and activities and probably reusing some of the modules developed for them. Such projects would still provide high payoffs, but probably not as high as ‘Maintenance Projects’. Though, they would still succeed on previously “Partially Completed Projects’ and ‘Outrightly Cancelled Projects’, the risk would be higher as they would have to forecast the work of prior developers in fulfilling the project requirements, which may not be streamlined with their own expertise and then rework all those requirements which were not completed. It is also possible that they would have to rework previously completed features/modules as it would be probably more difficult to integrate their work on uncompleted features/modules with the completed features/modules by previous developers. To increase their chance of success, software startups with low maturity should select ‘New Projects’. Such startups would be just starting out and may not have had enough experience to weather adverse conditions. If they are a few years old, they may not have learned lessons adequately from past failures. They would be better off not working on the remaining three types of projects until they have accumulated both experience and financial chest.

6. Conclusions

This research work is an effort to explore the choices of software projects a software startup could consider for succeeding in this COVID-19 environment and even during the post COVID-19 period. We consider this work as the first of its kind and there are no similar or related decision frameworks in the software engineering research or information systems research in the context of startups for managing software projects during post COVID-19 period. Though a few recent research works [12,13,24] outlines the challenges faced by software startups in attracting new software projects or managing their existing projects during this pandemic situation, they have not suggested any models or frameworks to overcome the problem.
It is hoped that our software startup decision framework (SSDF) would enable software startups to increase their chance of success by helping them choose projects whose risk would be compatible with their own maturity level. The startups after assessing the projects for their risk and preferring a quadrant from the SSDF rather than commensurate with their maturity level may consider evaluating the trade-offs among the benefits, costs, and risk. To explore the generalizability of our proposed framework, we recognize the need for future studies in the software startup companies of different sizes and maturity. Future research may also want to operationalize startup maturity.

We agree that one could not generalize results from a few case studies [15,32]. However, the observations made in this research work would definitely help the software startup community to expand their understanding of the software development and project management activities of software startups. To generalize the results, a few more case studies with software startups are required. Such case studies should be conducted by bringing in a larger team of members from startups involved in software project development, deployment and maintenance. We would like to carry forward the current research work in this line in the future. In future, we would like to expand the current research work by using the design thinking so as to provide a creative problem-solving approach for startups to manage their software development activities.

From a theoretical perspective, our research has contributed a decision framework to the body of software engineering and information system literature, specifically in the context of startup during this COVID-19 ecosystem. We deem that this research work is the first of its kind in the context of startup for initiating a new software project or managing ongoing projects with their available resources. Our proposed framework, at first, considers the maturity of startup and, in that; we leverage existing empirical knowledge on software project risks [35,36]. This has to lead to the development of a decision support framework for startup to succeed in COVID-19 environment.

The proposed framework is technology independent and hence it could be utilized by any startups for their software project irrespective of the domain such as banking, healthcare, insurance and manufacturing. This is of greatest advantage to the startup as they build the product across various domains. Finally, to the practitioners involved in the projects of startup, our research suggests a simplified and easier way to structure and channelize their process and workflow with regard to their project management activities. This would pay off to the startup in the long run in terms of avoiding challenging software projects which at a later stage of software development which would obviously demand additional costs, time and resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The author would like to thank Prof Srinarayan Sharma, Founding team member of DioneDawn, a startup and formerly Professor of IT & Systems, Indian Institute of Management (IIM) Kashipur, India for his guidance to carry out this research work.

References

[1] S. Hill, N. Bosma, A. Ionescu-Somers, J. Levie, A. Tarnawa, Global Entrepreneurship Monitor, 2019/2020 Global Report, Global Entrepreneurship Research Association, London Business School, Regents Park, London, UK, 2020.

[2] Startup Genome LLC, Global startup ecosystem report 2019, 2020, published May, 2019, Available at https://startupgenome.com/reports/modal-get-report (Accessed 22 May 2020).

[3] G. Gruman, How the COVID-19 pandemic is starving tech startups worldwide, in: Computerworld, 2020, published Apr 21, 2020, Available at https://www.computerworld.com/article/3538915/how-the-covid-19-pandemic-is-starving-tech-startups-worldwide.html (Accessed 20 May 2020).

[4] P. Ralph, et al., Pandemic programming how covid-19 affects software developers and how their organizations can help, Empir. Softw. Eng. 05 (2020).

[5] C. Giardino, M. UnterKalmeister, N. Paternoster, T. Gorscheck, P. Abrahamsson, What do we know about software development in startups? IEEE Softw. 31 (5) (2014) 28–32.

[6] J. Melegrati, A. Goldman, F. Kon, X. Wang, A model of requirements engineering in software startups, Inf. Softw. Technol. 109 (1) (2019) 92–107.

[7] J. Melegrati, E. Guerra, X. Wang, Understanding hypotheses engineering in software startups through a gray literature review, Inf. Softw. Technol. 133 (2021) 106465.

[8] A. Winston, Is the COVID-19 outbreak a black swan or the new normal, MIT Sloan Manag. Rev. 16 (2020) 154–173.

[9] J. Fu, M. Mishra, Fintech in the time of COVID-19: Technological adoption during crises, J. Financial Intermediation 50 (2020) 100945.

[10] Northwestern University, Data centers use less energy than you think: New, comprehensive analysis presents a more nuanced presents of global energy use, Sciencedaily, 2020, Retrieved May 2, 2020 from https://www.sciencedaily.com/releases/2020/02/200227144131.htm.

[11] C. Mora, R.L. Rollins, K. Talabady, M.B. Kantar, M.K. Chock, M. Shimada, E.C. Franklind, Bitcoin emissions alone could push global warming above 2 °c, Nature Clim. Change (2018).

[12] A. Kuckertz, et al., Startups in times of crisis – A rapid response to the COVID-19 pandemic, J. Bus. Ventur. Insights 13 (2020) e00169.

[13] D.A. Shepherd, F.P. Saade, J. Wincen, How to circumvent adversity? Refugee-entrepreneurs' resilience in the face of substantial and persistent adversity, J. Bus. Ventur. 35 (4) (2020) 105940.

[14] J. Melegrati, E. Guerra, X. Wang, HyMap: eliciting hypotheses in early-stage software startups using cognitive mapping, Inf. Softw. Technol. (2022) 106807.

[15] R.K. Yin, Case Study Research, fifth Ed, Sage, Thousand Oaks, 2013.

[16] B. Kim, H. Kim, Y. Jeon, Critical success factors of a design startup business, Sustainability 10 (9) (2018) 2981.

[17] A. Morita, T. Nauliat, E. Lutz, Accelerators as drivers of cooperation among early-stage startups, Technovation (2021) 10278.

[18] J. Saijntieb, D. Mauricio, Systematic literature review of critical success factors of information technology startups, Acad. Entrepreneurship J. 23 (2) (2017) 1–23.

[19] A. Marcon, J. Ribeiro, How do startups manage external resources in innovation ecosystems? A resource perspective of startups' lifecycle, Technol. Forecast. Soc. Change 171 (2021) 120965.

[20] M.A. Cusumano, Evaluating a startup venture, Commun. ACM 56 (10) (2013) 26.

[21] W.H. Delone, E.R. McLean, Information systems success: The quest for the dependent variable, Inf. Syst. Res. 3 (1) (1992) 60–95.

[22] L. Pirolo, M. Presutti, The impact of social capital on the start-ups’ performance growth, J. Small Bus. Manag. 48 (2) (2010) 197–227.

[23] J. Pantuchina, M. Mondini, D. Khanza, X. Wang, P. Abrahamsson, Are software startups applying Agile practices? The state of the practice from a large survey, in: H. Baumeister, H. Lichter, M. Riebisch (Eds.), Agile Processes in Software Engineering and Extreme Programming, XP 2017, Cologne, Germany, May 22–26, in: 18th International Conference Proceedings, 2017, pp. 167–183.

[24] R. Camara, M. Marinho, S. Sampoiol, S. Cadete, How do Agile software startups deal with uncertainties by Covid-19 pandemic? Int. J. Softw. Eng. Appl. 11 (4) (2020) 1–20.

[25] A. Ng, D. Macbeth, G. Southern, Entrepreneurial performance of early-stage ventures: Dynamic resource management for development and growth, Int. Entrepreneurship Manag. J. (2014) 1–19.

[26] N. Bocken, Sustainable venture capital catalyst for sustainable start-up success? J. Cleaner Prod. 108 (2015) 647–658.

[27] L. Yoon-Jun, Technology strategy by growth stage of technology-based venture companies, Inf. Rev. Bus. Res. Pap. 6 (6) (2010) 216–234.

[28] S. Almakzeni, A. Bramantoro, W. Rashideh, A survivability model for Saudi ICT start-ups, Int. J. Comput. Sci. Inf. Technol. 7 (2) (2015) 145–157.

[29] N. Paternoster, C. Giardino, M. UnterKalmeister, T. Gorscheck, P. Abrahamsson, Software development in startup companies: A systematic mapping study, Inf. Softw. Technol. 56 (2014) 1200–1218.

[30] E. Kotinis, M. UnterKalmeister, T. Gorscheck, Software engineering in start-up companies: An analysis of 88 experience reports, Empir. Softw. Eng. 24 (2019) 68–102.

[31] M. Ziejske, T. Held, Application of agile methods in traditional logistics companies and logistics startups: Results from a German Delphi Study, J. Syst. Softw. 177 (2021) 110950.
[32] R.J. Wieringa, M. Daneva, Six strategies for generalizing software engineering theories, Sci. Comput. Program. 101 (1) (2015) 136–152.

[33] N. King, C. Horrock, Interviews in Qualitative Research, Sage, 2010.

[34] S.A. Olsen, Group Planning and Problem Solving Methods in Engineering Management, Wiley, New York, 1982.

[35] C. Giardino, M. Paternoster, Unterkalmsteiner T., P. Gorschek, N. Abrahamsson, Software development in startup companies: The greenfield startup model, IEEE Trans. Softw. Eng. 42 (6) (2016) 585–604.

[36] V. Gupta, et al., Fostering product innovations in software startups through freelancer supported requirement engineering, Results Eng. 8 (2020) 100175.