Requirement Engineering in Startups

Shatadru Shikta, Sowvik Kanti Das, Somania Nur Mahal, H. M. Mahir Shahriyar,
Kazi Bushra Al Jannat and Mahady Hasan
Software Engineering, Independent University, Dhaka, Bangladesh

Keywords: Requirement Engineering, Software Engineering, Startups, Innovation Diffusion, Entrepreneurship.

Abstract: Startup companies are usually negligent when it comes to formal requirement engineering or proper collection of requirements required for their projects which leads to their early demise before gaining enough traction in the market. This paper tries to explore the reasons as to why startups fail in context to requirement engineering and gather experiences from the industry to try and propose a framework that can help startups work with a feasible and cost-effective method towards implementation of formal requirement engineering processes to ensure a long and successful tenure in the software industry.

1 INTRODUCTION

Innovation and startups are quite often used hand-in-hand in the current software industry mainly due to the fact that startups have a knack of innovating new ideas to pin-point pain points of a current system to propose new solutions to minimize difficulties and increase efficiency in terms of cost, time or effort from the perspective of end-users.

OECD (2016) proposes the fact that tech-based startups will eventually be gamechanger in opening new entrants in the industry in the next 15 years and therefore the importance of focusing on startups to bring about a sustainable business model through innovation is unavoidable (Viki, 2016). However, around 42% of startups failure can be attributed to their product which no one wants (Patel, 2015) which in turn can be directly attributed to the fact that the requirements for the product were not feasible to start with.

Moreover, just because an idea is creative does not make it innovative since one can easily point out to the huge array of patents found that have found no commercial success whatsoever (Viki, 2016). Therefore, it is important to understand the concept sustainability of innovation, innovation diffusion of products as well as having a competitive advantage of requirement engineering (RE) in startups (Autio, Nambisan, Thomas, & Wright, 2018).

2 BACKGROUND STUDY

RE is an integral part of any software project management and has been a cornerstone of new research and development to maximize fulfilled projects. However, since all tech-based startups have any of the common constraints of time, pressure, lack of funds, lack of resources, dependency, etc., startups have an affinity to remove processes or activities they consider to be bulky in nature such as RE and quality assurance (QA) leading to ultimate product failures or face the harsh reality of adopting it later on, leading to massive investment of time and resource for reworking projects.

All startups set a goal for fulfilling user needs into account, but the challenge is introducing them to the future issues ahead of time. Innovative startups are usually started on overnight ideas; therefore, it is extremely hard for them to consider RE as the process is not broadcasted enough to be in the insight of any entrepreneur (Paternoster, Giardino, Unterkalmsteiner, Gorschek & Abrahamsson, 2014).

Tech start-ups have recently become the subject of intensive research by the software and requirements engineering communities (Alves, Cunha, & Araujo, 2020) (Unterkalmsteiner, 2016) (Nirnaya Tripathi, 2018) and although previous research sought to understand how innovative products are created by start-ups, relatively little is known about start-ups operating within ecosystems and to what degree ecosystem actors influence the

Shikta, S., Das, S., Mahal, S., Shahriyar, H., Al Jannat, K. and Hasan, M.
Requirement Engineering in Startups.
DOI: 10.5220/0010561002070214
In *Proceedings of the 16th International Conference on Software Technologies (ICSOFT 2021)*, pages 207-214
ISBN: 978-989-758-523-4

acceptance of particular practices and ultimately affect the effective production of start-ups.

According to Berg, Birkeland, Nguyen-Duc, Pappas & Jaccheri (2018), tech startups are now major drivers of economic growth, innovation, and competitiveness. As said by Blank & Dorf (2012), a start-up is a temporary organization that is looking for a viable and repeatable business model. A startup lifecycle composed of three stages, as proposed by Crowne (2002) where the first stage begins before the first sale of the product, the start-up stage begins with the idea conception, second comes the stabilization stage begins when the first buyer acquires the product and finishes when the product is delivered to new customers, and third, the growth stage involves the timeline when the product is mature and stable enough to be commissioned to an increasing customer base. It is considered a mature business if the start-up has survived these phases. This indicates that the business has introduced a successful product on the market and has a steady stream of revenue from sales. Startups have adopted agile and lean concepts to deal with volatile external factors (Bosch, Holmström Olsson, Björk, & Ljungblad, 2013) and need to look carefully at the business and product dimensions of sustainable growth to ensure success in their endeavor (Nguyen-Duc, Shah, & Ambrahamsson, 2016).

Since startups offer innovative solutions under intense resource constraints, startup team's expectations about the potential of an innovative product to capture a good portion of the market, being suppliers of innovative products in either the same market or an entirely new market, is guided by actual market facts gathered through ongoing experiences (or experiments) with consumers (Levinthal, 2017). The experimentation improves validated business learning, which allows the startup team to strengthen their market assumptions and obtain better market experience and gain greater faith in them before the product is introduced to the market.

When the startups started gaining confidence in the needs of the consumer they began with delivery of the Minimal Viable Product (MVP) to further experiment with the real market by supplying early adopters with the "minimum functionality" (Gupta, Fernandez-Crehuet, Hanne, & Telesko). (Souza et al., 2019) conducted a preliminary case study on how RE processes were completed by startups in Brazil. The case study revealed that there are four key structures startups make and execute accordingly.

The first key structure, Customer-Driven Development was the core source of requirements for the software development teams. Customers would actively participate in prototype definition, prototype feature prioritization and validation. This showed collaboration within consumers to finalize construction of the project that improved development speed as well as the acceptance of the finished product.

The second key structure, Fast, Light-weight and Inexpensive Agile Practices were used during the development of the software. On one hand, this sped up the development and made management easy due to less documentation but on the other hand the lack of documentation meant transition of team members created knowledge gaps which slowed down development.

The third key structure, prototyping based Development with the help of inexpensive prototyping tools to gain user feedback. This resulted in assuring the startups of the confidence in their proposal idea from consumers. Consequently, it sped up the development and simplified their development process. The fourth key structure, Hacking Culture of the startups initiated collaborative effort among startups. The startups were hacking off their competition's processes, knowledge, tools, and methodologies to get an edge ahead without spending additional resources. Effectively, they were able to build off of failed strategies including the understanding of why they failed and starting with an alternative solution from the start.

An alternative study was conducted by Morales-Trujillo & García-Mireles (2019) on a startup over a 31-month period in New Zealand. The startup was established to complete a certain contract within a fixed period of time. During the first 2 months of the project, the RE process consisted of requirement gathering from the sponsors and users with ad-hoc implementation. Later on, the startup went from being more development specific to implementing a few agile practices such as user acceptance testing and use of wireframes. During the ad-hoc implementation, the startup used planning tools to evaluate values provided to the customer whereas the users kept increasing the feature list. Haphazard implementation and frequent feature release were followed with poor quality that resulted in a negative impact on the startup. After a 12-month period, the startup switched to automated testing tools and project management tools. This was due to inadequate quality control in their releases as well as improper planning of the deliverables. Implementation of these changes helped the startup in realizing the customer values achieved from the features in their releases.

Since Startups have an informal organizational structure with vaguely defined roles and responsibilities, team members usually play multiple roles. All these limitations impact the project timeline, so it takes longer than planned timeline and thus consequences of poor requirements engineering practice is startups leads to a bunch of reworks which include changes to the requirements and thus a major part of the system must be updated to reflect the changes in the requirements (Quispe, 2010).

Moreover, communication and coordination problems occur as the requirements elicitation process does not follow formalized methods and lastly poor visibility of the project status occurs pushing the project manager to make decisions based on uncertainty. Another issue is that requirements are mainly elicited and prioritized by startup founder's assumptions and interpretations of the market. This can create misinterpretation of the requirements (Alves, Cunha, & Araujo, 2020).

3 RESEARCH METHODOLOGY

In order to understand the current market trend, we opted to survey startups to gather data for our research. Initially, our thought was to have an interview session with startup stakeholders or project managers, but we decided to opt for online-based surveys with objective questions that would allow us to capture the essence of requirement management of the startups in the present industry.

The survey was intended to collect data which we could use for further quantitative and qualitative analysis. Many phases of startups can be identified within the industry, but our target was to engage with startups who are past their ideation phase and have some form of a team, occupied in building or developing the product, or is operating in the market. For our survey, we selected 4 companies who have recently launched into the market i.e., Operating below 6 months, 7 companies who are operating in the market for at least 1 year and more and 2 companies who are operating in the market for at least more than 2 years.

3.1 Course of Action

3.1.1 Organization Classification

Our survey started with some basic classifications and we have followed the industry standard in classifying organization size with the help of employee capacity as the primary classification factor. A company will be classified as micro for up to 19 employees, small for 20 to 99 employees, medium for 100 to 499 employees.

3.1.2 RE Factor Correlation

We focused on how requirement engineering is performed at a market level will become the tipping point to show the efficacy of any given requirements for a project. Secondly, we tried locating any interconnection between the requirement engineering and the culture of an organization.

In this regard, we are looking for the combinational metric from good requirement engineering practices, satisfactory and engaging customer base, effective product with remarkable quality and the organization that hosts employees. On a statistical level, we are comparing the number of requirement engineers with the number of total employees in a project. This will provide us with the importance the organization places on requirement engineering as a whole.

3.1.3 Responsibility Association

In this stage we are ensuring the success rate of a software startup in terms of startup team knowledge, allocated resources, and implementing RE processes and their contribution to project success. In a startup ecosystem, employees are used to taking on multiple job roles at a time when a developer also works as a requirement engineer or system analyst.

In order to ensure if the requirement engineering process is prioritizing in startups, we have looked at their team structure, RE practices, allocated resources for RE process and how these factors influence a project's success rate in terms of financial profitability and number of user reach. In a standard business scenario, the return of investment is projected over a number of years which the expected return on investment for a number of years provides a strong indicator of the business's success rate. We have used the indicator for attained profitability in comparison to number of customer usage, which would reflect the current ROI of the project and also gives us the prediction for future comparisons.

3.1.4 RE Progression

In order to understand their project status and condition we needed to observe the RE challenges they are facing and how are they related to project failure. We have analysed the RE factors that are impacting on project success by taking into

consideration the amount of rework that needed to be done due to failure in collecting the right requirements.

3.2 Data Collection

The data has been collected via online surveys due to covid related complications as well as online interviews when required. Open-source online survey tools were used to obtain the data and then collected via spreadsheets to analyze the data, which has been described in the next section. Moreover, the survey was designed in such a way that included all aspects of the organization in order to have a comprehensive overview of the organization.

4 RESULTS

4.1 Company Attributes

Since the data includes PII (personally identifiable information), the data was filtered out and derivation were made on the filtered-out data. We could classify 3 as small companies while the rest 12 were classified as micro companies and the business domain variation of the startups are described in Figure 1.

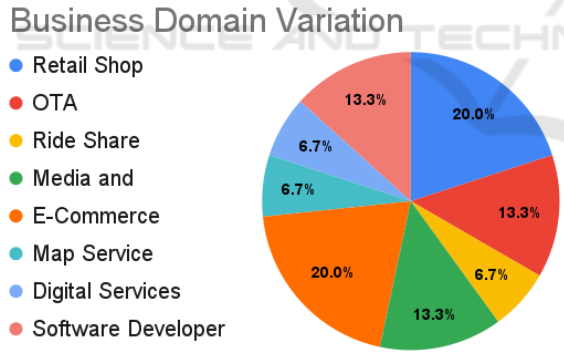

Figure 1: Business Domain Variation.

4.2 Data Analysis

Our main focus was to derive values concerning RE, product and organization valuation and determine successes and failures of the startups.

Figure 2 shows the tread line created based with Project Health percentage as the dependent value and ratio of requirement engineer to developers available on each team. The project health percentage was derived by calculating an average of the two factors, one being the ratio of number of current customers to projected customers and the other being the ratio between current revenue and projected revenue for the startups. The graph shows a steady increment of project health with the increase of ratio of requirement engineer to developers.

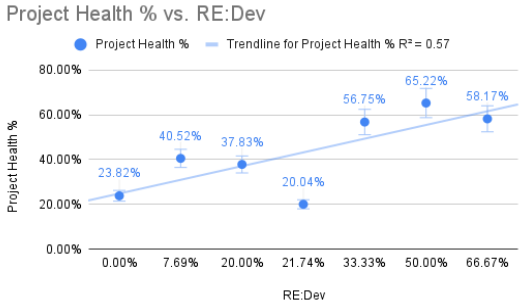

Figure 2: Project Health % vs RE:Dev.

Figure 3 shows the tread line created based with Change percentage to MVP compared to the current product as the dependent value and ratio of requirement engineer to developers available on each team. The change percentage was gathered by the percentage amount given by the personnel working on the project. The graph shows a steady decrease of change percentage to MVP compared to the current product with the increase of ratio of requirement engineer to developers.

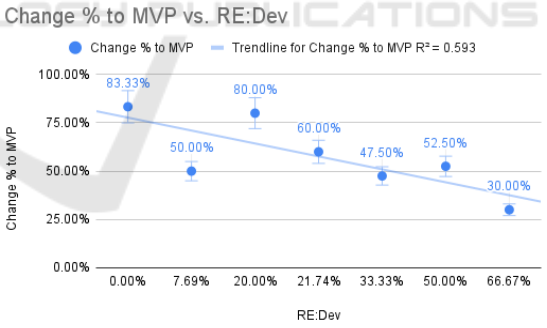

Figure 3: Change % to MVP vs RE:Dev.

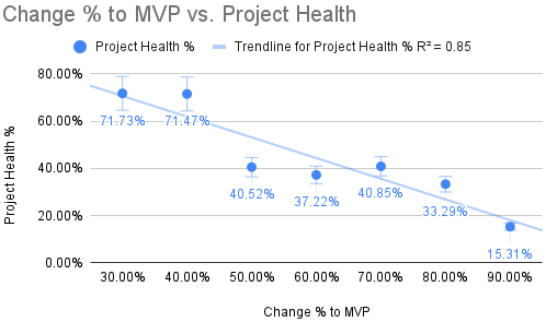

Figure 4: Project Health % vs Change % to MVP.

Figure 4 shows the tread line created with Project Health percentage as the dependent value and Change percentage to MVP compared to the current product. The graph shows a steady decrease of Project Health percentage with the increase of MVP compared to the current product.

5 PROPOSED FRAMEWORK

Based on our market research, we could clearly understand that most of the startups that have failed or are failing have the lack of requirement engineering processes involved while organization that have members of the team working as requirement engineers or is involved in some form of requirement engineering are better off in terms of project health.

Gralha, Damian, Wasserman, Goulão & Araújo (2018) state the importance of requirement engineering factors for a startup company by focusing on feedback from customers, insights from the employers which will be the founders and upper management in this case, competitive issues, demands of the market and ongoing changes in the frameworks and platforms used within the product's market ecosystem which keeps the company and the product reactive and evolutionary.

Thongsukh, Ayuthaya & Kiattisin (2017) on the other hand proposed the need for employees of a startup to collaborate and mitigate conflicts amongst themselves which extends to their views on the product. Syed, Barqawi & Mathiassen (2019) ask us the need to focus on release cycle management by ensuring identification and application of requisite changes to development and management activities to make the release cycle continuous and less chaotic.

The need for customer driven product development is considered to be a key success factor for any startup (Lorenz, Lorentzen, Stricker & Lanza, 2018) while Mattson & Sorensen (2020) state the need for a custom plan based on the requirements and available resources for any organization for success.

Since our market research is consistent with our findings from our literature review, and based on the startup study finding mentioned above, we have come up with a comprehensive yet easily implementable and feasible framework to mitigate issues related to RE for startups. For the framework, our first focus will be towards a list of objectives related to RE for the startups, given in Table 1 below.

Table 1: List of Objectives for Startups.

Objective	Priority
Problem and product Identification	High
Ensuring on-time Releases	High
Improve release Processes	Medium
Build a basic RE infrastructure	Medium
Build team confidence in the product	Low
Ensure customer satisfaction	Low
Implement continuous deployment	Low

The priorities listed are based on a phase-by-phase manner and each consequent objective must be completed before the next objective can be started. In this way, the team can organize their focus based on the resource and time at hand and can easily distribute the workload amongst the team. Based on these 7 objectives, 7 phases are designed in order to implement the objectives and each phase will have multiple sub-objectives in order to complete the final objective. Before going into the phases, a few things must be considered. Firstly, the startup team would consist of a founder and a co-founder. The founder is considered to be the brain of the project, based on whose vision the whole product will be built. Secondly, the team must consist of a technical consultant who will be either the founder or the co-founder who can analyze the technical feasibility of the vision of the founder at hand. Based on this two-man team, the project will start.

5.1 Phase 1

The first phase is based on the objective of problem and product identification. Based on the outline of the main problem the founder has focused on, the problem has to be broken down to assess each component of the problem and solutions, both operational and technical, must be sought out. Next the technical consultant must do a feasibility analysis of the technical solutions at hand and propose a basic estimation of the resource and time required to build technical solutions to each sub-problem.

Based on the estimation provided, the founder and co-founder must both elicit and prioritize the list of sub-problems and consider the sub-problems to be available on the first release of the software product. Finally, the founder and co-founder will set the requirements for the MVP.

5.2 Phase 2

Next the required technical resources must be hired based on the requirements of the MVP, and a

detailed technical scope with functional and non-functional requirements must be prepared by the technical consultant of the project parallelly. During this stage, the technical consultant will be considered to be the head of RE and will be responsible for all RE related processes. Once the developers are hired, the technical consultant must handover the MVP scope document to the developer with an estimated deadline. The technical consultant (who is also the RE head) will oversee the development of the MVP and make sure that the MVP is released on time.

5.3 Phase 3

Once the MVP is released, stage 3 starts, whereby the RE head must take analyze the current status of the product based on market feedback from the customers. The RE head must take into account the bugs found in the initial release by performing testing as well as the imminent requirements based on customer feedback. Based on the bugs and the requirements, the RE head has to again to elicit and prioritize the requirements and bug fixes and create and lock the scope for at least 3-5 releases of the product and push it into development.

5.4 Phase 4

Once the development is ongoing, the technical consultant must hire a dedicated tester for the team and gather all the bugs in the system. As the RE head, the technical consultant then must elicit and prioritize the bugs found in the system by the dedicated tester and push these bug fixes into the next releases.

Additionally, the RE head must appoint a dedicated Requirement Analyst who will be responsible in During phase 4, no new functionality development must be pushed into the locked scope from phase 3 but high priority bug fixes must be prioritized and put into the scope of work. Phase 4 will end once all the releases planned in phase 3 is completed.

5.5 Phase 5

Once phase 4 has ended, phase 5 will start by analyzing the market value of the product by the RE head. This can be done through the valuation considered under the Software value pyramid and the SIG maintainability Model. (de Groot, Nugroho, Back, & Visser, 2012) This can help the team, especially the RE head as well as the founder understand the viability of the requirements that are in pipeline as well as the manintaibility of the project. This in turn will help the organization move in the right direction and also give visibility of the current and ensuing probable progress and of the software, helping build team confidence on the product itself.

Moveover, a dedicated requirement engineer should be hired under the RE head, who will facilitate the analysis of the RE and project metrics as well as take handover of all the exisiting RE processes performed by the RE head. However, the final elicitation and prioritization of the requirements must be approved by the existing RE head.

5.6 Phase 6

Once the metrics are put into place and the team has confidence on the product, customer confidence needs to be achieved. The role of a customer excellence officer has to be assigned to the dedicated requirement engineer who would know about the existing bugs of the system, the ins and outs of the product as well as look at the product from the perspective of the customer/end-user. The customer excellence officer can list down the requirements as well as the bugs from the customer perspective, then elicit and prioritize the requirements and bug fixes required and push it to the RE lead for final prioritization and the RE lead will push the prioritized list to the development team for release. This will in turn help build the confidence on the product from a customer's point of view and help grow a steady and loyal customer base.

5.7 Phase 7

With both the team and the customers having confidence on the product, continuous deployment must be achieved. In order to do so, the RE processes must be automated using tools like Modern Requirements or Jama Software which can be integrated with project management tools like JIRA for elicitation, prioritization and pushing requirements to the developers seamless to ensure continuous development of the product. The dedicated requirement engineer will, under this phase, become the RE lead and implement the tools required for the automation of the RE management.
A summary of the entire framework is given below in Table 2.

Table 2: Framework Summary.

P#	Important Actions	Achievements
1	Breakdown of the problem into subproblems, brainstorm to find the solution to the subproblems, resource and timeline estimation for development of solution for each subproblem and based on the estimation, elicit and prioritize requirements for the MVP.	Technical Feasibility study of the MVP with high level scope.
2	Hire resources based on the estimations, create low level technical scope and start developing the project.	Technical consultant will become RE head, low level scope for MVP and delivery of MVP.
3	Take market feedback of MVP and prepare and lock scope for the next 3-5 releases.	Release processes are improved and scope ready and locked for next 3-5 releases.
4	Hire dedicated tester to analyze bug fixes required and elicit and prioritize these bug fixes under the planned releases.	Better understanding of technical issues in the system and product stability increases.
5	Implement metrics on product to understand maintainability and ROI while hiring a dedicated requirement engineer.	Team has visibility on the market viability on the product, which in turn gives the team confidence and direction.
6	Role of customer excellence officer is given to the dedicated requirement engineer who will continuously collect market feedback from customers and based on feedback, new requirements and bug fixes are put into scope.	Ensure higher customer satisfaction.
7	Automate requirement management through tools and push requirements continually for development.	Continuous development and deployment are achieved.

The state of requirement engineering practices in startups has still been largely unexplored. In the proposed framework the whole process is divided into 7 phases and these combinations and achievements are designed in such way that it could be easily adapted by any startups despite having resource shortages. The framework considered the common limitations faced by startups in the requirement engineering process.

6 CONCLUSION

As much as we would like to propose a framework for startups at every stage, our proposed model will only help the startups which are still in their early stages. The primary limitation of the proposed model is the startup's maturity level. Mature startups beyond the development phase cannot really use it. They may also need to adapt the framework model based on their targeted industry.

The study concludes the inclusion of RE practices in the startup ecosystem promotes customer loyalty, overall revenue whereas reducing rework which translates to project cost reduction. Later we will evolve the framework in such a way that it can be adapted by mature startups.

Furthermore, future prospects of requirement management can be looked into through research and development and machine learning algorithms can be implemented for automated requirement elicitation (Darwish, N. R., Mohamed, A. A., & Abdelghany, A. S., 2016).

Since proper implementation of requirements engineering processes help in ensuring innovation diffusion easily for the products (Autio, Nambisan, Thomas, & Wright, 2018) and can help encourage the team into exploring further innovations regarding their products (Rajapathirana & Hui, 2018). So, in our future work we will work on more unexplored requirements engineering practices and case studies in startups around the world.

REFERENCES

OECD. (2016). Start-ups and innovative entrepreneurship.

Viki, T. (2016). *A Lean Startup Definition of Innovation*. From Medium: https://medium.com/the-corporate-startup/a-lean-startup-definition-of-innovation-af5bb72c836d

Patel, N. (2015). *90% Of Startups Fail: Here's What You Need To Know About The 10%.* From Forbes: https://www.forbes.com/sites/neilpatel/2015/01/16/90-of-startups-will-fail-heres-what-you-need-to-know-about-the-10/?sh=33151ffc6679

Autio, E., Nambisan, S., Thomas, L., & Wright, M. (2018). Digital affordances, spatial affordances, and the genesis of entrepreneurial ecosystems. *Strategic Entrepreneurship Journal, 12*(1), 72-95.

Alves, C., Cunha, J., & Araujo, J. (2020). On the Pragmatics of Requirements Engineering Practices in a Startup Ecosystem. *2020 IEEE 28th International Requirements Engineering Conference (RE)*.

Unterkalmsteiner, M. a. (2016). Software Startups - A Research Agenda. *E-Informatica Software Engineering Journal*.

Nirnaya Tripathi, E. K. (2018). An anatomy of requirements engineering in software startups using multi-vocal literature and case survey. *Journal of Systems and Software, 146*, 130-151.

Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I., & Jaccheri, L. (2018). Software startup engineering: A systematic mapping study. *Journal Of Systems And Software, 144, 255-274.* doi: 10.1016/j.jss.2018.06.043

Blank, S., & Dorf, B. (2012). *Startup Owner's Manual.* [S.l.]: Wiley.

Crowne, M. (2002). Why software product startups fail and what to do about it. Evolution of software product development in startup companies. *IEEE International Engineering Management Conference*.

Bosch, J., Holmström Olsson, H., Björk, J., & Ljungblad, J. (2013). The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups. *Lean Enterprise Software and Systems*.

Nguyen-Duc, A., Shah, S., & Ambrahamsson, P. (2016). Towards an Early Stage Software Startups Evolution Model. *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*.

Gupta, V., Fernandez-Crehuet, J., Hanne, T., & Telesko. (n.d.). Requirements Engineering in Software Startups: A Systematic Mapping Study. *Applied Sciences, 10*(17), 6125.

Souza, R., Malta, K., Silva, R., Masiero, P., Almeida, E., & Machado, I. (2019). A Case Study about Startups' Software Development Practices. Proceedings Of The XVIII Brazilian Symposium On Software Quality. doi: 10.1145/3364641.3364663

Morales-Trujillo, M., & García-Mireles, G. (2019). Evolving with patterns: a 31-month startup experience report. *Proceedings Of The 2019 27Th ACM Joint Meeting On European Software Engineering Conference And Symposium On The Foundations Of Software Engineering.* doi: 10.1145/3338906.3340447

Quispe, A. &. (2010). Requirements Engineering Practices in Very Small Software Enterprises: A Diagnostic Study. *International Conference of the Chilean Computer Science Society, SCCC*, 81-87.

de Groot, J., Nugroho, A., Back, T., & Visser, J. (2012). What is the value of your software? *2012 Third International Workshop on Managing Technical Debt (MTD)*.

Darwish, N. R., Mohamed, A. A., & Abdelghany, A. S. (2016). A hybrid machine learning model for selecting suitable requirements elicitation techniques. *International Journal of Computer Science and Information Security*, 1-12.

Rajapathirana, R., & Hui, Y. (2018). Relationship between innovation capability, innovation type, and firm performance. *Journal of Innovation & Knowledge, 3*(1), 44-55.

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

Levinthal, D. (2017). Resource Allocation and Firm Boundaries. *Journal Of Management, 43(8), 2580-2587.* doi: 10.1177/0149206316667458

Lorenz, R., Lorentzen, K., Stricker, N., & Lanza, G. (2018). Applying User Stories for a customer-driven Industry 4.0 Transformation. *IFAC-Papersonline, 51(11), 1335-1340.* doi: 10.1016/j.ifacol.2018.08.345

Thongsukh, S., Ayuthaya, S., & kiattisin, S. (2017). Startup Framework based On Scrum Framework. *2017 International Conference On Digital Arts, Media And Technology (ICDAMT).* doi: 10.1109/icdamt.2017.7905012

Gralha, C., Damian, D., Wasserman, A., Goulão, M., & Araújo, J. (2018). *The evolution of requirements practices in software startups.* Proceedings Of The 40Th International Conference On Software Engineering. doi: 10.1145/3180155.3180158

Syed, K., Barqawi, N., & Mathiassen, L. (2019). Release cycle management: an action research study into a software company. *International Journal Of Business Information Systems*, *30*(2), 152. doi: 10.1504/ijbis.2019.097533

Mattson, C., & Sorensen, C. (2020). *Product development* (1st ed.). Springer International Publishing.