Requirements Engineering is the Compass that Guides Successful Solutions: An Engineering Experience Report

Funmilayo Olaiya
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
folaiya@uwaterloo.ca

***Abstract*—This paper details a unique engineering experience during my tenure at an early-stage startup in Nigeria. To gain insights into the challenges faced by the startup, I referenced relevant academic papers to better comprehend the issues. Throughout the paper, I aimed to identify the primary concerns related to requirements engineering by narrating these experiences. Ultimately, this report provides a comprehensive overview of why requirements engineering presents significant challenges within a startup environment.**

***Index Terms*—software engineering, requirements engineering, requirements documentation, requirements validation, requirements specification, requirements management, requirements driven development, startups**

I. Introduction

Requirements engineering (RE) can be defined as an engineering discipline focused on identifying the goals and necessary specifications for the envisioned system [1]. It is a multi-faceted process comprising numerous components, each essential to ensure its successful application in any given domain.

According to a very interesting paper about the importance of requirements management [2], the authors described the requirements engineering process through an illustration as shown in Figure 1, and they further mentioned that, the term "Requirement Engineering Process" refers to the breakdown of Requirements Engineering (RE) into interconnected, non-linear activities. These activities progress from initial informal and vague individual requirement statements to a formal specification that achieves mutual understanding and agreement among all stakeholders [2].

The primary objective of this report is to gain a deep understanding of certain factors in relation to the requirements engineering process that led to the early emergence of technical debt and the continuous review, reinterpretation, and modification of requirements, particularly within the context of a startup. By examining the challenges encountered in this specific startup, the report aims to shed light on the reasons behind these issues and how they contributed to the eventual failure of the startup.

To comprehend and analyze these problems thoroughly, the report references several Requirements Engineering (RE) papers that provide valuable insights and perspectives. Through this comprehensive approach, the report seeks to offer valuable lessons and recommendations for addressing such challenges in future startup ventures.

This report is however a personal case study, focusing on the challenges encountered as a senior software engineer in an early-stage startup. As a full-stack developer, my responsibilities encompassed overseeing the entire development process of a new web application. This included managing both the front-end and back-end aspects, ensuring seamless integration, and delivering a cohesive and functional product.

By leveraging this case study, several compelling requirements engineering (RE) papers will be utilized to brainstorm and comprehend the startup's RE challenges, their causes, and potential solutions for future occurrences.

As emphasized earlier, the RE process plays a vital role, as depicted in Figure 1. However, it is essential to recognize that there are additional components in this report that are not fully represented in that figure.

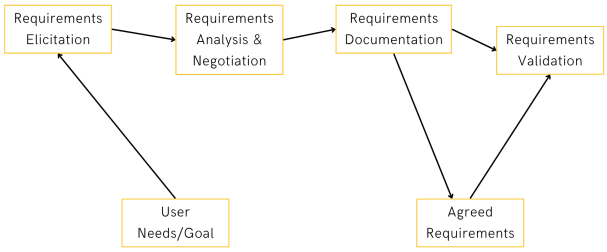

Fig. 1. Requirements Engineering Process.

II. Work Case Study

Being a recently hired senior engineer at an early-stage startup in 2021, I never anticipated a set of challenges, particularly in the process of defining requirements for specific tasks, features, and implementations.

Management's primary focus was on expediting the development of a minimum viable product (MVP) to enter the market and compete with other businesses. As previously

mentioned, the company was in its nascent stage, grappling with numerous adjustments and lacking a well-established structure, which led to certain oversights.

During that period, I lacked familiarity with the term "requirements engineering", but I was aware that our approach to determining requirements was flawed.

Furthermore, as I reflect on the past, I attempt to depict the chaotic nature of our processes related to requirements engineering through Figure 2, which practically means that we had none of the processes shown in that figure working. This visual representation sorts of illustrate the components of requirements engineering we failed to acknowledge and which brought about many challenges with the project and company.

In this section, I am going to highlight various challenges I faced during my time as an engineer at a startup as it related to requirements engineering, we will also use certain papers to gain more insights into why these challenges occurred.

A. Requirements Engineering in a Start-up in Challenging

While our main focus was to develop an MVP quickly, we faced a significant dearth of thorough research in many areas concerning the precise determination of requirements. Our requirements specification process suffered due to the absence of robust documentation procedures, resulting in frequent changes in decisions. Additionally, it's crucial to recognize that during a startup's initial stages, certain investors may exert considerable influence on the company's direction, making the operation highly sensitive. This early stage is critical as it can determine the startup's success or failure.

In this very interesting paper called; *Software development in startup companies: A systematic mapping study by Paternoster et al*, they shared something enlightening, and here's my rephrased version; *In the startup environment, it is difficult to establish an engineering process for gathering, specifying, and maintaining requirements. RE practises are frequently simplified to a few core activities. Initially, because startups frequently build software for a growing target market, customers and end users are frequently unknown, and requirements are thus market-driven rather than customer-specific* [3].

The authors of this paper put forth the argument that requirements engineering (RE) processes are invariably challenging for startups. They contend that RE is often simplified to only a handful of core activities due to other pressing priorities taking precedence within a startup environment. As a result, navigating the complexities of requirements engineering remains an enduring struggle for these early-stage companies.

B. Management Cannot Fully Control or Determine Requirements Specification

Management used to believe that once the final specifications were determined, the process should conclude. However, this perspective was proven false, especially in a company lacking a solid requirements engineering process, hastily rushing into implementation.

During this phase, I made an interesting discovery - poor software engineering practices significantly impacted requirements engineering. I advocated for the introduction of various sound software processes to ensure proper development, but I looked like the *bad guy* for some time, as the manager perceived these practices as time-consuming and detracting from feature-building efforts.

Unfortunately, we faced numerous consequences due to this approach, resulting in a considerable amount of technical debt in the back-end. Consequently, I had to invest substantial time and effort in resolving these issues all over again.

In the paper titled; *Requirements Determination is Unstoppable: An Experience Report by Berry et al*, as I rephrase, they argued that *the management seems to hold the belief that they possess full control over the extent of requirements determination. However, it is crucial to acknowledge that sufficient requirements determination is an unavoidable prerequisite for successful code creation. In case insufficient time is allocated for requirements analysts (RAs) to conduct thorough requirements engineering (RE) before generating a requirements specification (RS) for the programmers and testers, the programmers and testers will inevitably engage in additional requirements determination while performing their respective tasks.* [4].

This can also lead to potential inefficiencies and challenges in the development process, making it essential to prioritize comprehensive requirements engineering from the outset.

C. Software Requirements are Simply Uncertain

Because it was an early stage startup, software requirements was not actually that easy to really define. There was so much uncertainty with what features we could build or build first. We kept re-iterating over and over again and not because this is a bad thing but because we just did not know how to lay a ground truth for RS well.

I still firmly believe that with a well-established requirements documentation process for RS, we could have managed the situation better, and to shed more light on this, there is a paper called; *Using Real Options to Manage Technical Debt in Requirements Engineering by Abad and Ruhe*, they mentioned and I rephrase that; *because of the inherent uncertainty in software requirements, which includes uncertainties related to customer (or market) requirements, project context and environment, and the feasibility, cost, and development duration for each requirement, the requirements engineering process necessitates a more detailed focus on requirement specification* [5].

D. Requirements Specification was Hazardous

In the preceding sections, I have consistently described how disorganized our requirements gathering and specification (RS) process was. This was primarily due to the lack of a robust approach to defining requirements, which differed from what other companies employed – where they often took notes during the process.

I strongly believe that adopting the agile methodology properly by ensuring sprints are done could have significantly improved this situation. However, we did not utilize sprints; instead, our focus remained on daily stand-up meetings throughout the work-week to just focus on and keep track of building new features, and this had a detrimental impact on the flow of our requirements process.

To shed more light on this, there is a paper called; *Core - A Method for Controlled Requirements Specification by Mullery* where the author argued and I rephrase that, *attempts to adequately specify requirements usually fail, sometimes fatally, and one cause is the lack of a process that is formally defined and directly addresses the needs of requirement specification* [6].

E. What about Requirements Validation?

Indeed, without a well-established requirements validation (RV) process, it becomes challenging to ensure proper validation when there isn't even a requirements analyst (RA) or requirements engineering (RE) process in place, especially in the context of an early-stage startup. While we lacked a formal RV process, there is a somewhat equivalent practice in software engineering known as Test Driven Development (TDD). Unfortunately, we executed this in a haphazard manner, and there was a lack of accountability in the process.

In order to effectively validate requirements, it is essential to have a comprehensive requirements document that encompasses well-defined requirements specifications.

In this paper called, *Empirical Studies of Requirements Validation Techniques by Raja*, it was mentioned and I rephrase that, *many businesses resort to ad-hoc requirements validation due to several factors, including a shortage of trained personnel or insufficient training and exposure to requirements validation practices. As a result, more emphasis is often placed on the software testing phase, which is typically conducted towards the end of the software project after all modules have been integrated, and despite the challenges, there are various techniques available for requirements validation, such as requirements reviews, prototyping, use case-based modeling, and other methods* [7].

These RV techniques can help ensure that the specified requirements are accurate, complete, and aligned with the stakeholders' needs before the software development process progresses further.

Without a clear and structured documentation of requirements, the validation process becomes significantly more difficult.

F. Requirements Management

It is evident that the lack of proper requirements specification and documentation significantly hindered the development of a comprehensive requirements management plan. Without well-defined requirements and a documented account of these specifications, the foundation for a requirements management plan is absent. Additionally, the absence of sensible requirements determination further compounds the problem.

In this paper called; *Importance of Requirement Management: A Requirement Engineering Concern by Pandey and Pandey*, they argued and I rephrase that, *inevitably, due to system errors and a deeper grasp of consumers' genuine demands over time, requirement modifications become inevitable. Requirements management tasks encompass keeping the project plan current with requirements, managing versions, monitoring their status, and tracing them* [2].

A requirements management plan is dependent on having a clear understanding of the project's requirements, their priorities, and how changes to requirements will be managed throughout the project lifecycle. In the absence of this critical information, it becomes challenging to establish a structured and effective requirements management process.

Amidst rapid changes and fast-paced progress, our lack of a robust requirements management (RM) system left us without adequate support during moments of chaos and urgency. Consequently, this situation led to knowledge gaps among specific team members.

G. Did we ever stop "Requirements Engineering"?

The absence of clear guidelines on when and how to halt the influx of requirements resulted in a continuous stream of new features and APIs being added, driven by excitement rather than a focus on essential priorities. As a consequence, undocumented processes and events became deeply entrenched, leading to certain engineers being left uninformed and disconnected from various processes.

In the paper called *Requirements Determination is Unstoppable: An Experience Report by Berry et al*, they argued and I rephrase that, *in order to inform the Requirements Analysts (RAs) in the Requirements Engineering (RE) team about the suitability of the Requirements Specification (RS) meeting the condition, it is necessary for the RE team to have participation from both the eventual coding team and the eventual testing team, with at least one member from each team* [4].

On numerous occasions, during core meetings, certain individuals from a particular group were absent, and as a result, feedback was not conveyed back to them. Indeed, having a robust requirements documentation (RD) process in place could have greatly benefited everyone involved, enabling better collaboration and success as a cohesive team.

Considering the sequence of these specific events that transpired at the startup and as described in this section, the eventual outcome was that we ran out of funding.

III. Discussion

In this section, I will outline specific key points and discoveries that, if implemented, could have provided some relief to the startup I worked at and can also serve as valuable insights to enhance our overall understanding.

A. Determining Requirements

When it comes to determining requirements, the absence of proper requirements documentation (RD) poses a significant obstacle. Without a well-defined RD, successful requirement

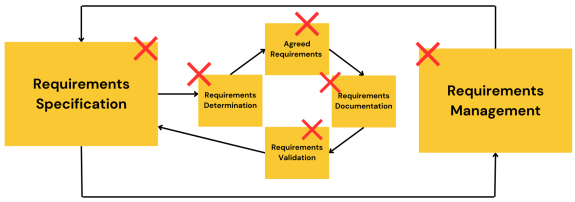

Fig. 2. Chaos Overview

determination becomes challenging, leading to unnecessary changes or constant verification of requirements. Additionally, a comprehensive RD acts as the authoritative source for decisions made and guides the way forward.

In the domain of requirements engineering, Oberg et al. authored a paper titled *Applying Requirements Management with Use Cases*. Within this paper [8], the authors presented an illustrative image, as shown in Figure 3, which visually depicts their recommendations or concept related to when it comes to determining requirements, as they believe that involving a representative from each team in the activity is the optimal approach. This ensures that everyone stays informed about certain requirements throughout the process.

However, in my view, I contend that everything becomes meaningless without a reliable repository for storing requirements specification (RS). Without a well documented RS accessible to all team members, any further efforts made to determine requirements on a regular basis may become futile.

It is worth noting that when bad software engineering (SE) practices exist, effective requirements engineering (RE) practices cannot thrive. Conversely, if strong RE practices are already in place, they naturally encourage good SE practices. They both complement each other and act as the foundation for a highly successful project.

Additionally, RE plays a crucial role in enforcing product quality assurance (QA) and facilitating the reduction of technical debt. An example of this is shown in Figure 4 where requirements and software documentation have to work hand in hand to serve as a successful point of reference.

B. Requirements Management Tools

If there are automated requirements management tools on the market, who employs them? and how are they used? I'm sure we all have been curious about them. In my opinion, they may be a huge assistance in terms of validating requirements as they evolve in order to assess the quality of the requirements at the time. It can be rather pricey, and others might not be aware of it, according to some ideas spreading online.

But to share a thought worth noting that was presented in this paper titled *How to Select a Requirements Management Tool: Initial Steps by Gotel et al*. They pointed out as I rephrase that, *requirements management tools constitute a fundamental component of many requirements management solutions* [9].

C. Requirements Driven Development

Anyone who has worked in the software industry for a while will be familiar with Kent Beck. He was a software engineer who formalized Test Driven Development (TDD) for software projects in an effort to raise the standard of how software was being developed.

The belief is that just as Test-Driven Development (TDD) significantly improved the quality of the software engineering process, applying a similar approach to requirements could also enhance software engineering practices effectively.

There is a blog post on this topic called *Requirement Driven Development Or: Building the Right Thing by Isaac Hildebrandt*, in which he attempts to illustrate that this approach can certainly be applied to requirements and software engineering practises. Here are some worthy things mentioned in the blog post as I rephrase [10]:

- Requirement Driven Development (RDD) centers around the product attributes that the development team must prioritize and be mindful of during their work.
- At the end of the day, two essential goals are pursued in terms of development: verification and validation. Verification assesses whether the thing was built correctly, while validation evaluates whether the right thing was built.

Reflecting on the comprehensive range of topics discussed in this section, it becomes evident that if the startup I worked at had adopted even half of these strategies, we could have likely successfully launched our product and attained a significant level of achievement and recognition in the market. The incorporation of these valuable approaches could have had a profound impact on our overall progress and potential for success.

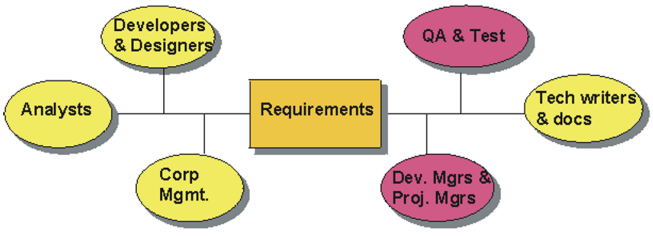

Fig. 3. Cross-Functional Requirements Team.

IV. Conclusion

In this paper, I presented a case study of my work experience as a software engineer at an early-stage startup in Nigeria, where I highlighted several engineering problems we encountered and how we addressed them through the lens of requirements engineering. I will now reiterate the key points

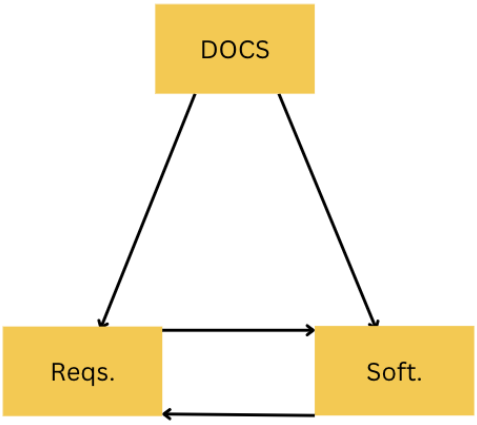

Fig. 4. SE and RE Documentation Work Collaboratively.

discussed already. By emphasizing the significance of the requirements engineering life cycle, I stress that neglecting it can lead to adverse consequences. An example is when many software projects experience extended implementation times due to inadequate, poor and negligent RE procedures.

Requirements engineering necessitates the application of sound software engineering practices, and it is reasonable to assume that effective requirements engineering practices can quickly identify shortcomings in every software engineers' work. Regardless of the software engineers' expertise or management style, failure to follow due process in requirements engineering can result in undesirable outcomes, even for the most skilled professionals.

As we are all aware, startups often rush to create a minimum viable product (MVP) or build rapidly to stay afloat, leading many to disregard the requirements engineering process. Unfortunately, more than 90% of startups still fail. Interestingly, there is no reported instance of a startup that dared to take the path of good requirements engineering and emerged successful.

Exploring or witnessing such a case in the present or future would be highly beneficial for all of us.

REFERENCES

[1] A. van Lamsweerde, "Requirements engineering in the year 00: a research perspective," in *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*, 2000, pp. 5–19.

[2] D. Pandey and V. Pandey, "Importance of requirement management: A requirement engineering concern," *International Journal of Research and Development-A Management Review (IJRDMR)*, vol. 1, no. 1, pp. 66–70, 2012.

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

[4] D. M. Berry, K. Czarnecki, M. Antkiewicz, and M. AbdElRazik, "Requirements determination is unstoppable: An experience report," in *2010 18th IEEE International Requirements Engineering Conference*, 2010, pp. 311–316.

[5] Z. S. H. Abad and G. Ruhe, "Using real options to manage technical debt in requirements engineering," in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, 2015, pp. 230–235.

[6] G. P. Mullery, "Core - a method for controlled requirement specification," in *Proceedings of the 4th International Conference on Software Engineering*, ser. ICSE '79. IEEE Press, 1979, p. 126–135.

[7] U. A. Raja, "Empirical studies of requirements validation techniques," in *2009 2nd International Conference on Computer, Control and Communication*, 2009, pp. 1–9.

[8] R. Oberg, L. Probasco, and M. Ericsson, "Applying requirements management with use cases," *Rational Software Corporation*, vol. 21, 1998.

[9] O. Gotel and P. Mader, "How to select a requirements management tool: Initial steps," in *2009 17th IEEE International Requirements Engineering Conference*, 2009, pp. 365–367.

[10] I. Hildebrandt, "Requirement driven development or: Building the right thing," 2022. [Online]. Available: https://phase2online.com/2022/01/20/requirement-driven-development/