

Article

Causes and Mitigation Practices of Requirement Volatility in Agile Software Development

Abdulhafour Mohammad *  and Job Mathew Kollamana

School of Business, Economics and IT, University West, SE-46186 Trollhattan, Sweden

* Correspondence: abdulhafour.mohammad@hv.se

Abstract: One of the main obstacles in software development projects is requirement volatility (RV), which is defined as uncertainty or changes in software requirements during the development process. Therefore, this research tries to understand the underlying factors behind the RV and the best practices to reduce it. The methodology used for this research is based upon qualitative research using interviews with 12 participants with experience in agile software development projects. The participants hailed from Austria, Nigeria, the USA, the Philippines, Armenia, Sri Lanka, Germany, Egypt, Canada, and Turkey and held roles such as project managers, software developers, Scrum Masters, testers, business analysts, and product owners. Our findings based on our empirical data revealed six primary factors that cause RV and three main agile practices that help to mitigate it. Theoretically, this study contributes to the body of knowledge relating to RV management. Practically, this research is expected to aid software development teams in comprehending the reasons behind RV and the best practices to effectively minimize it.

Keywords: requirement volatility; requirement elicitation; requirement engineering; agile methodology; software engineering



Citation: Mohammad, A.; Kollamana, J.M. Causes and Mitigation Practices of Requirement Volatility in Agile Software Development. *Informatics* **2024**, *11*, 12. <https://doi.org/10.3390/informatics11010012>

Academic Editor: Antony Bryant

Received: 6 January 2024

Revised: 2 March 2024

Accepted: 11 March 2024

Published: 13 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Software development is a dynamic and constantly evolving field, and the ability to adapt to changing requirements is essential for organizations to remain competitive. Such changes, modifications, or uncertainties that arise in software requirements during the development process are called requirement volatility (RV) [1]. Although the volatility of requirements is seen as a unique phenomenon, there are a variety of factors that can contribute to the volatility of requirements and affect it in multiple ways [2]. Therefore, it is vital to determine what causes requirement volatility and what their potential impacts are. Before the agile period, requirement changes were discouraged, but the current state of software development necessitates embracing them as a chance to enhance software products [3]. However, the cost of requirement changes is justified in certain cases, such as meeting the changing needs of users or adapting to shifting market conditions, but it is best to try to reduce the number of unneeded changes that arise from organizational, procedural, or practice issues. This is because unnecessary changes in requirements pose major challenges to software development; for example, they can lead to project delays, cost overruns, quality degradation, increased complexity, and strained communication, which can negatively impact project outcomes and ultimately affect an organization's performance and reputation [2].

On the other hand, agile software development methodologies can accommodate changing requirements and deliver value iteratively. As a result, agile methodologies can turn the potential drawbacks of RV into opportunities for enhancing the final product's quality and relevance [2]. However, despite the many benefits of agile software development methodology in the management of RV, there is still limited knowledge regarding factors contributing to RV, its effects, and possible mitigation approaches [3] in the context of software development.

Therefore, it is important to grasp the origins and implications of RV in software projects to uncover underlying factors influencing RV in software development projects. In addition, it is significant to assess the effectiveness of various strategies and practices for mitigating RV in different settings to substantially enhance project success rates.

By comprehending these factors leading to RV and implementing strategies and best practices to minimize its impact, software development teams can more effectively manage the challenges accompanying requirement changes and deliver high-quality software that meets stakeholder expectations. Therefore, this paper aims to fill a gap: it aims to conduct an empirical investigation into RV and the usage of agile methodology in addressing it, focusing on the causes of RV in software development and identifying effective approaches for its mitigation. This will help researchers and practitioners enhance their knowledge of requirement variability. To address these objectives, a qualitative research method is used in this study to answer the following research questions.

“What are the primary causes of RV in software development projects?” and

“Which strategies have agile practitioners found to be effective in reducing RV?”

The next section introduces more details about the sources of RV and agile practices to decrease their impact.

2. Causes of Requirement Volatility

In software development, RV frequently occurs due to a variety of factors. Several studies have examined these factors, including poor communication [1–9], a lack of stakeholder engagement [5,8,10], and inadequate expertise [10–13]. Anjum et al. [4] outlined several of them that lead to changes in requirements; for instance, if developer teams need to fix bugs or if the functionality needs to be improved, requirements may need to be changed. RV may also have occurred when new requirements were added, the design was improved, the scope was changed, the wrong requirements were corrected, or conflicts were resolved. However, this study focuses exclusively on requirement change problems in automotive software development; it ignores organizational and environmental concerns such as the interplay of expertise and domain knowledge and regulatory compliance that affect the RV. Another study [5] reported the effects of late customer feedback and lengthy development cycles as challenges for requirements management. In addition, this study has revealed several crucial factors that impede software teams' ability to make the right decisions when it comes to software architecture design. Among these factors are poor communication, distorted information, and external dependencies of requirement volatility. However, the main focus of this study was on the architecture of software development rather than other factors that also play a crucial role in the success of software projects.

Madampe et al. [6] conducted a case study showing that corporate customers, who maintained long-term relationships with the company, were the primary source of changing requirements for software products. Other factors such as the fast-paced and dynamic nature of the software industry were studied also by [6]. In addition, Madampe et al. [6] highlighted that agile teams need to deliver products rapidly to the market to avoid RV caused by rapid changes in stakeholder preferences. However, this study does not discuss strategies that enable agile practitioners to be effective in reducing RV.

According to the literature, poor communication was among the most common factors found in most studies examining sources of RV mitigation [6–8]. For example, as a solution to mitigate RV, Kassab et al. [7] stressed the importance of clear communication among stakeholders to prevent misunderstandings, requirement volatility, and decreased satisfaction. Furthermore, Daud Haiderzai et al. [8] emphasize the need to build trust between end-users and developers to avoid communication breakdowns. Moreover, the developer as a mediator is a means to bridge gaps and improve communication effectiveness. Similarly, Madampe et al. [6] pointed out the significance of effective communication in preventing requirement changes and stressed the need for efficient communication

channels during the elicitation and customer negotiation phases. Despite this, additional factors that impact RV were not explored in these studies.

Another factor that leads to RV, as reported by studies [9,11], is the lack of expertise among software development teams in the customer's domain or application area, which can cause several problems, such as difficulties in understanding and interpreting requirements and challenges in determining the best way to implement them. However, there exist other details regarding the required expertise stated in the literature; for example, Canedo et al. [11] noted that a knowledge gap among software developers regarding requirement compliance and technical expertise can lead to compliance issues and security breaches, potentially putting sensitive data at risk. Therefore, some studies emphasized the need for individuals with domain knowledge to mitigate RV caused by changes within the software team and other stakeholders actively involved in the software development process [5]. When you add new members with less experience or expertise to the team, for example, the team will work slower and deliver less, at least in the short term. Experts can provide valuable information to help software architects understand and maintain requirements that are often ambiguous and poorly documented. Furthermore, expertise is essential to comply with regulations and laws [11], as the field of software development is highly volatile and subject to external factors that contribute to changes in project requirements. Therefore, compliance measures can also often exacerbate the problem of RV, leading to further complex changes in project requirements [12]. Thus, to address the challenge of RV and ensure software development practices meet the required standards, security experts, for example, must be integrated into software development teams to address the lack of security practitioners, team security awareness, and unclear security requirements [13]. For example, Moyón et al. [13] proposed a solution to achieve security compliance in large-scale agile projects by incorporating security compliance requirements into the agile development process. Even so, additional studies are required to thoroughly review how skills impact RV at various project scales.

Even though there are a handful of studies on RV in the different industry sectors, they have either focused on solutions, new frameworks, or models to mitigate RV's impacts, for example, in [6–8], or they are literature reviews that are not original research, such as in [4]. Although the literature has discussed some of the factors that impact RV, it is still fragmented. Other potential underlying causes for RV that were neither discussed nor reported in these studies need to be revealed. In the next section, key studies that have discussed agile practices used to mitigate RV in software development are introduced. However, this study tries to investigate other potential agile practices for mitigating the risks of RV that were neither discussed nor reported in previous studies.

3. Agile Practices to Reduce the Impact of RV Causes

Despite the potential risks posed by frequent requirement changes during software development, agile principles have proven to be highly effective in managing these changes [6]. By working with shorter iterations, flexible development cycles, and a continuous refinement process, agile teams can quickly respond to new requirements and deliver high-quality software within a project's constraints [6].

Several studies have examined ways to mitigate RV, including communication [2,6,7,9], stakeholder engagement [5,6,8,14], expertise [9–13,15], and documentation [4,9]. Research has indicated that selecting the right software development approach is crucial to mitigating RV [7]. Madampe et al. [14] emphasized the significance of stakeholder participation in the process of requirement changes to achieve a comprehensive understanding of their needs and priorities. In addition, this study emphasized the importance of taking into account agile teams' emotional responses when applying agile principles and practices to reduce RV. However, it is not enough to focus solely on the emotions of agile teams to reduce RV. Some other factors and aspects need to be taken into consideration.

In addition, Madampe et al. [6] highlighted that inadequate communication and rushed analysis during requirement definition can result in incorrect initial requirements.

However, agile practices such as daily stand-up meetings and close collaboration with stakeholders can significantly enhance communication [9]. Furthermore, agile teams provide regular feedback to stakeholders to ensure that requirements remain transparent and understood. Therefore, in agile environments, communication is more frequent, informal, and direct than in traditional approaches. Moreover, collaboration tools, frequent meetings, and documentation can foster effective communication in agile projects [2]. Additionally, agile methods, such as iterative development and continuous feedback, can help address ambiguity and uncertainty in requirements [2]. According to most studies, such as [2,6,9], inadequate communication is a significant factor that contributes to RV.

Additionally, to mitigate the risks associated with RV, agile artifacts, particularly user stories and documentation, serve as effective tools. By storing and prioritizing requirement-related tasks in the product backlog, agile artifacts help maintain a consistent set of requirements while continuously synchronizing user and system needs [4]. In addition, the implementation of features during iterative development phases, known as sprints, allows for efficient planning and task estimation [9]. As a result, the development team can present developed features to stakeholders, receive feedback, and re-prioritize the product backlog accordingly [15]. Furthermore, the progression during each sprint is illustrated using burn-down charts, which enable the discussion and capture of requirement changes in sprint review meetings [10]. However, these studies did not investigate further into these factors and their effectiveness in reducing RV. Furthermore, to minimize the challenge of requirement ambiguity and uncertainty, Salmani et al. [10] suggested involving domain experts or subject matter experts early in the development process. Similarly, Gupta et al. [9] explored conceptual models that are used to overcome challenges in software development's requirements engineering and management. In addition, they emphasized the importance of expertise in requirements elicitation and software development. Furthermore, they claimed that training and education can enhance this expertise, which is particularly useful for effectively eliciting and documenting requirements, especially when working with non-technical clients [16]. Therefore, agile methodology facilitates rapid learning and adaptation [10].

Kassab et al. [7] noted that fewer interpersonal communication paths in smaller teams reduce the risk of communication breakdown, contributing to improved estimations in all aspects of a software project. Therefore, agile teams are typically smaller, with a recommended size of five to nine team members, which seems to be more satisfactory for team members, which could be attributed to the more effective communication practices of agile teams compared to waterfall teams [7]. However, this study did not explore any other factors that might decrease RV.

Ensuring regulatory compliance is another critical challenge faced by agile teams. According to [11], educating a team about relevant regulations is essential to ensuring compliance. This understanding of regulatory compliance enables a team to work effectively and efficiently to meet project objectives while following the necessary guidelines [11]. However, it is crucial for software development teams to have compliance experience due to regulations being enforced by countries; therefore, further studies are necessary to investigate the impact of a lack of experience in these regulations on RV.

The rest of the paper is organized as follows. Section 4 describes the methodology used. Section 5 focuses on analyzing and presenting the findings obtained from the participants. The findings are followed by Section 6 which discusses the results and future research. Finally, Section 7 discusses the limitations and challenges.

4. Methods

This section is divided into five subsections that cover the research approach, sample selection, data collection methods, data analysis process, and ethical considerations. The next subsection discusses the overall approach used to guide the study.

4.1. Research Approach

The software development process is complex [17] and requires a deep understanding of the various factors that can impact a project's success [18]. Qualitative research is well suited for exploring such complex phenomena and understanding the perspectives of those involved [19]. Therefore, the research design of this study adopts a qualitative approach, focusing on understanding the causes of RV in agile software development projects and identifying the effective strategies of agile principles for managing it. The use of qualitative research helps to access the rich and detailed information provided by participants, which may not be accessible through quantitative methods. This allows us to gain insights into the real-life experiences of software development professionals, revealing the challenges they face and the strategies they employ to address RV [20]. In addition, the qualitative research design promotes flexibility, allowing our inquiry to be adapted and refined as new insights emerge during data collection. This ensures that the study remains responsive to the evolving understanding of the research problem, ultimately leading to a more robust and comprehensive analysis [20]. Furthermore, using a qualitative approach, the study can establish connections between RV and other aspects of agile software development, such as team communication, collaboration, and decision-making processes. This enables the development of a deeper understanding of the phenomenon and identifies potential areas for improvement within the software development process.

4.2. Participants

Purposive sampling has been strategically selected as the data collection method for this study on requirement volatility and agile methodology [21]. This method significantly increases the chances of gathering highly useful information by focusing on individuals with extensive experience in managing changing requirements and agile practices, thereby ensuring the collection of high-quality data [21]. The inclusion of professionals who are highly knowledgeable about agile practices ensures that their insights will greatly enhance the study. In addition, purposive sampling, in its essence, helps select participants who are in line with the study's goals, leading to a clear and comprehensive understanding of the topic [21]. It also makes efficient use of limited research resources [22], a critical factor when dealing with the fast-paced environments of requirement volatility and agile methodologies. Therefore, with its strategic benefits, purposive sampling emerges as the optimal choice for this study.

Furthermore, by employing purposive sampling in this study, specific participants with relevant experiences, characteristics, and roles were deliberately selected. This approach enriched the research findings and contributed to a deeper understanding of RV in software development. Therefore, in this study, 12 participants with experience in software development projects were included, representing a diverse range of countries, industries, and roles. The participants hailed from Austria, Nigeria, the USA, the Philippines, Armenia, Sri Lanka, Germany, Egypt, Canada, and Turkey and held roles such as project managers, software developers, Scrum Masters, testers, business analysts, and product owners. Additionally, the sample comprised participants from industries like insurance, banking, automotive, web, IoT, logistics, and both large- and small-scale projects, as well as in-house and outsourcing companies. This diverse sample ensures that various perspectives are represented, providing a more comprehensive understanding of the phenomenon under investigation [21].

The determination of the sample size in this study was informed by the findings of [23], who conducted a study that systematically assessed saturation and variability in qualitative research through coding and analyzing 60 in-depth interviews. Their research indicates that saturation typically occurs within the first 12 interviews, while basic elements for meta-themes can emerge as early as six interviews. Taking these findings into consideration, a sample size of 12 participants was deemed sufficient to ensure a thorough exploration of the research topic of RV in agile software development.

Table 1 overviews the demographic information and characteristics of the interview participants.

Table 1. Key summary details for each of the 12 interviewees.

Code	Country	Role	Experience	Company Size	Duration	Word Count
P1	Austria	Product Owner	3	600	53	8388
P2	Nigeria	Product Manager	3	300	66	9314
P3	USA	Lead Analysts	3	33k	41	6289
P4	Philippines	Tester	1	50	51	5687
P5	Armenia	Product Owner	7	70	52	6483
P6	Sri Lanka	Business Analyst	1.5	50	64	11,226
P7	Germany	Scrum Master	12	50	70	8934
P8	Egypt	Scrum Master	8	200	69	11,689
P9	Canada	Product Owner	10	2000	50	7123
P10	Turkey	Scrum Master	6	1500	91	10,583
P11	USA	Scrum Master	7	600	83	12,085
P12	Sweden	Squad Lead	5	75k	61	8055

4.3. Data Collection

A semi-structured interview guide was meticulously crafted to investigate sources of RV and agile practices flexibly and comprehensively. Kallio et al. [24] have highlighted a systematic, five-phase process for developing such a guide, including identifying the prerequisites for using semi-structured interviews, retrieving and using previous knowledge, formulating a preliminary guide, conducting pilot testing, and presenting the complete guide.

4.3.1. Identifying the Prerequisites for Using Semi-Structured Interviews

The prerequisites for using semi-structured interviews are determined by several key factors. The qualitative nature of the study necessitates a method capable of eliciting detailed and nuanced information from participants, which semi-structured interviews provide. In addition, the study's focus on identifying effective strategies for managing RV and establishing connections between RV and other aspects of agile software development, such as team communication and decision making, underscores the need for the adaptability and exploration that semi-structured interviews offer. Furthermore, the decision to utilize purposive sampling, targeting individuals with extensive experience in managing changing requirements and agile practices, reinforces the appropriateness of semi-structured interviews, given their ability to efficiently capture diverse and in-depth insights, particularly in the fast-paced environments of requirement volatility and agile methodologies.

4.3.2. Retrieving and Using Previous Knowledge

A meticulous review of peer-reviewed papers across databases like Scopus, IEEE, ACM, and Web of Science was carried out, providing insights into requirement volatility (RV) in software development, agile methodologies, and factors contributing to RV. The selected papers, chosen with a high level of agreement among researchers, detailed the adverse effects of RV, such as project risks, delays, cost overruns, and diminished software quality, alongside causal factors, including unclear requirements, inadequate communication, and regulatory changes. In addition, agile methodologies, known for their adaptability and emphasis on collaboration, were examined for their potential to mitigate these challenges. Various factors that contribute to RV in agile projects, such as inadequate communication, insufficient stakeholder engagement, and regulatory compliance, were analyzed, alongside the negative consequences of RV, namely increased development time, escalated costs, and reduced software quality. Finally, strategies and best practices to counter these impacts were reviewed, including the implementation of agile project

management and agile team structures, although challenges like uncontrolled change and the emotional toll on team members were also recognized.

4.3.3. Formulating the Preliminary Semi-Structured Interview Guide

A semi-structured interview guide was developed, amalgamating insights from the literature and aligning them with the research objectives. It acts as a navigational tool for the interviewer, marking the significant themes to be examined while maintaining scope for adaptability in the conversation.

4.3.4. Pilot Testing of the Interview Guide

The pilot testing phase for the interview guide was carried out with two participants who had prior experience in managing requirement volatility in agile software development projects. These participants closely resembled the intended interview population, enabling an evaluation of the questions' effectiveness. The interviews followed a semi-structured guide, and feedback was gathered after each interview to assess question clarity, relevance, and overall comfort during the interview. The initial interview spanned 83 min with a word count of 12,085, while the second lasted for 50 min with a word count of 7123, resulting in an average duration of 66 min. Feedback and observations from these pilot interviews, along with an assessment of the average interview length and the detail of responses, were utilized to enhance the interview guide and improve interviewing skills. This helped ensure that the guide was finely tuned to the research objectives and prepared for the main data collection stage.

4.3.5. Presenting the Complete Semi-Structured Interview Guide

The finalized guide is structured around several domains of inquiry:

- Disclaimer;
- The Definition of Volatility;
- Participants Project Experience;
- Strategies for Mitigating RV;
- Agile Practices to Address RV;
- Challenges and Lessons Learned;
- Final Thoughts and Recommendations;
- Conclusion.

This approach allowed for a thorough exploration of the topic while maintaining a focus on the individual experiences of the participants. The guide encompassed six sub-domains, featuring twenty main open-ended questions and eight additional open-ended subquestions. Subquestions were employed judiciously when a participant's response to the primary question did not sufficiently address specific topics of interest. All respondents were asked the same questions, but the interview was conducted in a manner that emphasized the importance of the participants' narratives over strict adherence to the question order. This flexible approach facilitated the capture of individual experiences effectively and enriched the data collection process [24]. Importantly, the interview guide was sent to the participants as part of the interview invitation, allowing them to familiarize themselves with the topics and prepare their thoughts in advance. This approach contributed to a more effective and engaging interview process [25].

The semi-structured interview data collection process for this study, which took place between 12 April and 1 November 2023, ensured a solid foundation for robust analysis and findings. Interviews were expertly conducted through online meetings in English, facilitating seamless communication with participants. This approach enabled the researcher to contact individuals across the globe, capturing a diverse array of data and experiences and significantly enriching the study's dataset. By tapping into the insights of a geographically widespread participant pool, the study benefits from a comprehensive understanding of the subject matter, ultimately bolstering its credibility and persuasive power. To ensure the highest level of accuracy in data capture, all interviews were recorded and then transcribed

using the innovative AI-powered transcription tool Avrio. This state-of-the-art tool generated verbatim transcriptions of the interviews, capturing the participants' exact wording while also rendering the text in a readable format. Following a standardized transcription protocol [26], any extraneous verbal fillers, inaudible segments, or overlapping speech were identified and appropriately annotated. It is worth noting that certain verbal fillers, such as "ums" and "ahs," were retained to maintain the authenticity of the participants' responses. The transcripts generated by Avrio underwent careful review to ensure consistency and accuracy. For areas of uncertainty, we reverted to the digital audio recordings, which were conveniently time-stamped for quick retrieval of the relevant audio segments. This meticulous process led to pristine transcripts that accurately conveyed the essence of the audio recordings while ensuring accessibility.

In the handling of confidential and sensitive information, a careful strategy was employed involving the use of substitution phrases or the omission of information altogether. This approach respected the privacy of the interviewees and complied with ethical guidelines for research while ensuring that the essence of the interviewees' sentiments was preserved. For the critical task of thematic analysis, the highly regarded Analysis Software for Word-based Records, specifically NVivo 14 [27], was used. As a widely used and trusted qualitative data analysis software, NVivo provides an unparalleled platform for organizing, analyzing, and visualizing textual data. Its advanced capabilities allowed it to efficiently identify patterns, themes, and insights within the interview transcripts, ultimately leading to a comprehensive and nuanced understanding of the subject matter [28]. This rigorous approach to data collection and analysis not only strengthens the study's credibility but also enhances its persuasive power.

The details of the semi-structured interview, including the full list of questions and subquestions, can be found in Appendix A of this study.

4.4. Data Analysis

In the data analysis phase of the methodology, a thematic analysis was conducted by the two authors (A.M., J.M.K.) to identify patterns and themes emerging from the interview transcripts. This analysis was guided by a codebook that was meticulously developed by the first author (A.M.) using a standard iterative process [26]. This process involved the following three stages:

1. **Definition:** In the initial stage, we identified the codes that were relevant to the study's objectives and provided a concise definition for each code. These brief definitions served as a quick reminder during the coding process [26]; see Appendix B.
2. **Full definition:** To ensure consistency and clarity, we further elaborated on each code by providing a more comprehensive definition. This full definition helped to avoid ambiguity and provided clear guidance for the coding process, ensuring that each code was applied consistently across the entire dataset [26]; see Appendix B.
3. **Example quotes:** Lastly, we compiled a selection of quotes from the interview transcripts that exemplified the use of each code. These examples served as benchmarks to ensure accurate and consistent coding throughout the analysis. To enrich the analysis and serve as a quality control procedure, the codebook was updated by researchers iteratively until the study team reached a consensus, confirming a consistent and agreed-upon approach to the analysis. With the codebook in place, the two researchers then proceeded to apply the codes to the interview transcripts and systematically organize and categorize the data [26]. This process allowed for the identification of emerging themes and patterns, which were further refined and iteratively adjusted by the two researchers (A.M., J.M.K.) to establish the credibility and validity of the results as new insights were uncovered. By adhering to this rigorous and structured approach, the study's thematic analysis ensured a comprehensive understanding of the data, ultimately leading to more robust and meaningful findings. Moreover, to further guarantee trustworthiness, reflexive conversations were conducted to discover

whether the researcher's previous attitudes about RV may have influenced his bias and consequently the outcomes of the analysis.

4.5. Ethical Considerations

The ethical considerations of this study were taken seriously to protect the well-being and rights of the participants, as well as to maintain the integrity of the research [29]. The following ethical guidelines were diligently adhered to: Firstly, informed consent: Before participating in the study, each individual was provided with a comprehensive explanation of the research's purpose, procedures, and potential implications. Participants were allowed to ask questions and clarify any concerns before voluntarily agreeing to take part in the study. This process ensured that all 12 participants were fully informed and aware of their rights, including the ability to withdraw from the study at any time without penalty. The second ethical guideline is anonymity and confidentiality: To protect the participants' privacy, all identifying information was removed from the transcripts, research outputs, and any other materials related to the study. Pseudonyms were used in place of the participants' names, and any potentially identifying details were altered or omitted. This approach ensured that participants' identities were not revealed, safeguarding their privacy and minimizing any potential harm or discomfort. The third ethical guideline is data security: All collected data, including interview recordings and transcripts, were securely stored. Upon completion of the study, the data will be retained for a predetermined period, following data protection guidelines, and then securely destroyed to prevent any unauthorized access or misuse. The fourth ethical guideline is respect for participants' well-being: The research team was mindful of participants' well-being throughout the study, ensuring that the interview process was conducted in a respectful and non-invasive manner. Participants were allowed to take breaks or pause the interview as needed and were encouraged to voice any concerns or discomfort. By adhering to these ethical guidelines, the study demonstrated a commitment to upholding high ethical standards and respecting the rights, dignity, and well-being of the participants while preserving the integrity and credibility of the research [29].

4.6. Validity and Reliability

Validity and reliability are essential for establishing confidence in research outcomes, as they collectively contribute to the overall trustworthiness of the findings. Ensuring high levels of validity and reliability allows for drawing meaningful conclusions and generating insights that can ultimately inform decision making, policy development, and future research directions [30].

4.6.1. Validity

Validity in qualitative research encompasses several important factors that contribute to the appropriateness and soundness of a study. It involves evaluating various aspects, including the alignment of the research question with the desired outcome, the suitability of the chosen methodology for addressing the research question, the validity of the research design concerning the selected methodology, the appropriateness of the sampling and data analysis techniques employed, and the extent to which the results and conclusions apply to the specific sample and contextual setting [30].

- **Internal validity:** This study ensures internal validity by employing a well-defined research design, purposive sampling, and a rigorous data analysis process. The adoption of qualitative research methods, such as semi-structured interviews and thematic analysis, allows for a comprehensive exploration of participants' experiences related to requirement volatility (RV) in agile software development. This in-depth approach facilitates a profound understanding of the phenomenon and supports the identification of effective management strategies. The emphasis on internal validity is vital for establishing causal relationships while controlling for potential confounding fac-

tors [31]. Therefore, it can be argued convincingly that this study exhibits a robust degree of internal validity, adhering to established criteria.

- **External validity:** In this study, the qualitative research design encompasses participants from diverse countries, industries, and roles, ensuring a wide range of perspectives. The inclusion of such diverse participants contributes to the external validity of the study, as it increases the potential for the generalizability of the research findings to different settings or populations. This emphasis on external validity aligns with the viewpoint emphasized by Patino et al. [31], who emphasized the importance of understanding the broader applicability of research findings. By incorporating participants from varied backgrounds, this study strengthens its external validity and provides a more comprehensive understanding of the phenomenon under investigation.

4.6.2. Reliability

Reliability is characterized by the ability to replicate processes and obtain consistent results. However, in qualitative research, reliability encounters challenges due to the presence of diverse paradigms, which make exact replication challenging. Instead, the emphasis in qualitative research is placed on consistency. It is acknowledged that some variability in the results is acceptable as long as the methodology consistently generates ontologically similar data, albeit varying in richness and context within similar dimensions [30].

- **Consistency:** To ensure consistency, this study implements a robust methodology that includes a well-defined interview guide, a standardized transcription protocol, and a meticulous codebook development process. By using the same interview guide for all participants and consistently applying the codebook during thematic analysis, the study guarantees the reliability and consistency of its findings. Kasirye et al. [32] stated that to achieve consistency, their study adheres to standardized protocols, establishes clear criteria, and maintains transparency throughout the research process.
- **Reproducibility:** This study provides a detailed overview of its research methodology, including the design, participant selection, data collection methods, data analysis process, and ethical considerations. By providing clear and specific descriptions of these aspects, the study enables other researchers to replicate the study and obtain similar results [32].
- **Accuracy:** To ensure accuracy, this study implements a rigorous data analysis process using NVivo software. This software enables the systematic identification of patterns, themes, and insights within the interview transcripts. The researcher also ensures accuracy by conducting a thorough review of the transcriptions generated by Aviro AI, confirming that the data accurately reflect the participants' experiences. Kasirye et al. [32] highlighted the importance of rigorous research methods, documenting the research process meticulously, and conducting data collection and analysis procedures with care. These practices contribute to the accuracy and reliability of the study's findings.

5. Results

The thematic analysis of qualitative interviews with seasoned agile software development practitioners yielded two major themes: (1) the primary causes of RV in software development projects and (2) strategies that agile practitioners have found to be effective in reducing RV.

The participants are all named from participant (P1) to participant (P12) to fulfill ethical considerations and maintain their anonymity. The details of each theme are presented comprehensively in the following subsections.

5.1. The Primary Causes of RV in Software Development Projects

In the causes of the RV theme, there was a similarity in the participants' answers across all sub-themes. This is due to the fact that participants share the same agile principles and practices in their projects. However, some of the participants' concerns about RV are diverse based on their level of experience in agile software development projects. Therefore, the

primary causes of RV in software development projects as reported by the participants are introduced under this theme. This theme consisted of six sub-themes: (1) market feedback and competition; (2) communication; (3) stakeholder engagement; (4) interplay of expertise and domain knowledge; (5) regulatory compliance; and (6) requirement ambiguity and uncertainty.

5.1.1. Market Feedback and Competition

Five participants believed that market competition is an important cause of RV since market competition is a critical factor that drives the success of a product or service in the rapidly evolving business landscape, which leads to RV. In addition, one participant emphasized the influence of changes in market conditions on RV, as this change leads to changes in requirements in terms of the software's functional and non-functional requirements such as security, usability, and performance. These changes can also make managing the software challenging because of the potential risks, expenses, delays, or conflicts in the software development lifecycle. A participant (P9) stated the following:

"The market conditions impact the requirements on a very granular level because when market conditions evolve, the organization changes, its shifts, its goal a little bit to accommodate that."

In addition, the participants acknowledged that the influence of competition on software development and the necessity of observing competitors and adjusting requirements to maintain competitiveness are essential. In other words, businesses must closely monitor their rivals to guarantee that they are addressing the changing needs of the market. Participant (P6) offered an example of how analyzing competitors resulted in alterations in requirements, explaining that their company examined their competitors, pinpointed areas that required improvement, and implemented the necessary changes. A participant (P6) noted the following:

"We were constantly studying our competitor, and we could see we need to do this better."

Furthermore, the participants believed that the keeping up to date with market trends is crucial for recognizing potential opportunities and risks. Customers might not be knowledgeable about the latest trends; therefore, companies need to stay informed about various developments happening beyond the technology market to discover diverse methods for addressing issues. One of the participants (P12) said the following:

"for us to consider multiple ways, we need to be aware of different things that are evolving outside in the technology market."

The previous idea was shared with another participant (P8) who believed that shifts in project strategy, driven by competition, can cause significant changes in project requirements, ultimately contributing to RV and highlighting the need for adaptability in the industry. A participant (P8) stated the following:

"If we are working in a competitive market, then this also may end up with us changing our strategy. So, for example, you plan to have a kind of feature after six months, but due to some change in the market, you start to see that the competitor is starting to build something that appeals to the customer. So, this may also end up with us changing our plan and trying to build something in a different way or in a faster way."

Another important cause of RV extracted from the participants' explanations of the causes of RV in software development is stakeholder feedback. Stakeholder feedback was considered by the participants as a cause of RV for different reasons. For instance, the participants believed that the significance of obtaining customer feedback at different stages of product development, particularly before testing a Minimum Viable Product (MVP), cannot be understated. This enables businesses to prioritize value delivery and make necessary adjustments based on feedback received from customers and other stakeholders. In this regard, a participant (P2) stated the following:

“We are done developing, like the MVP, before we do like a test and, you know, get feedback from some beta clients. . . . Taking feedback from customers is important. . . something that gets you to value faster.”

In addition, customer feedback serves as a vital factor in steering a project’s direction and shaping decisions concerning feature prioritization, technology selection, and the overall project strategy. A participant (P6) explained the following:

“Within the first two weeks, two months, I’ll be regularly talking to the customer to see how satisfied they are, what they think of it. I wouldn’t push it here to carefully get that data. Feedback is very important.”

One participant even considered effective communication and responsiveness to feedback as key elements in refining products and ensuring their success in the market. A participant (P8) noted the following:

“You ask them to have a kind of round on what you have presented today, and then you can ask them, for feedback. So this is gonna be a healthy way to build a good product.”

Additionally, the participants believed that recognizing the importance of various channels for gathering customer feedback is essential. As some individuals may be hesitant to participate in surveys, businesses should explore alternative communication methods, such as video calls, chats, or collaborative platforms. Utilizing a diverse range of channels can yield more comprehensive feedback, leading to a well-rounded understanding of customer needs and preferences. A participant (P6) stated the following:

“good survey where we understand what they need and what they need to be done. And yes, we came across that.” “ . . . most leads of the video calls, we have chats with them. We collaborate because people are not always ready to answer surveys.”

5.1.2. Communication

The majority of the participants (ten participants) considered poor or miscommunication a significant cause of RV. This is because agile software projects are dynamic, so requirements are constantly revised during the project, and changes are welcomed. This creates difficulties and hinders the effective communication of all the additional requirements and changes to all members involved in a project. This theme consisted of five sub-themes: the lack of transparency and openness in communication; missing details in communication; poor communication between team members; communication between the development team and external stakeholders; and reluctance to discuss and ask questions.

- The lack of transparency and openness in communication:

According to the participants, a lack of transparency and openness in communication can significantly contribute to RV within project management. This is due to the fact that, as the participants acknowledged, when communication is unclear or insufficient, misunderstandings can arise, leading to misalignment in project objectives and expectations. This misalignment can result in frequent changes to requirements as stakeholders struggle to grasp the true scope and goals of a project. Therefore, effective communication is essential in avoiding assumptions that can negatively impact project outcomes; for instance, a participant (P2) stated the following:

“in terms of communication, I think it’s important because most people get assumption is bad. . . . if you don’t communicate, there’s no way you can know what’s going on or what the other person is thinking which is very important.”

As highlighted by participant (P8), it is crucial for team members to openly share their current status, ongoing tasks, and any blockers they may be facing. The participant (P8) further explained that when team members discover that inaccurate or incomplete information has been shared, it can significantly impact trust and relationships within the team. Additionally, transparent communication, on the other hand, facilitates effective

collaboration and alignment with the project's objectives. The participant (P8) explained the following:

"So you have this kind of communication and make sure that everyone is aware of what is the correct status of the product. . . , . . . make sure that, people are transparent, during for what they are sharing."

In addition, ten participants considered visibility and openness with clients equally important. One participant explained that when clients have a clear understanding of the development process, they are less likely to make unfounded assumptions that can lead to RV. Instead, they can actively participate in the process and collaborate with the team to achieve the desired outcomes. A participant (P11) stated the following:

"If the client is not on board with the development team or the production support team, they might think that, yeah, we are not working on it. . . , . . . We want them to have that visibility."

Furthermore, the participants believed in the benefits of clear communication in terms of tracking progress and ensuring that all team members are on the same page and mission. In addition, they felt that clear communication allows an external team, such as the product management team, to have insight into the development team's activities and assess if they are aligned with the company's mission and vision. It fosters a sense of clarity and transparency for all stakeholders involved. In this context, a participant (P12) stated the following:

"I think it's an art and it's very few people who have in the software industry, everybody must have that clarity."

- Missing details in communication:

Eight participants believed that insufficient or omitted information in communication could significantly influence the precise interpretation of a project's requirements. As a consequence, RV may emerge, adversely affecting the progress and final result of a project. In this regard, a participant (P8) stated the following:

"you get that requirement, you did the analysis, you did the design, everything is clear, but you communicated things in a bad way or not in a correct way to the development team. So this may end up that you are building a product that no one asks it for, or we have a kind of miscommunication"

In addition, based on the participants' opinions, addressing potential confusion caused by differing terminology or abbreviations is vital to maintaining clear communication within a project. By ensuring that everyone involved in the project has a shared understanding of terms and concepts, teams can work more effectively and efficiently. In this context, a participant (P12) said the following:

"fruit and the developer's best fruit or what developer would like is an orange. Are we comparing, apples to apples? No, we're not. they asked for an apple, we're doing an orange. Well, in the document it might just say, a fruit,"

Furthermore, two participants emphasized that it is important to document communication to provide a reference point and facilitate understanding. A participant (P11) noted the following:

"When we talk to somebody, it's impossible to capture everything. So what I do is, I, everything has to be documented somehow. . . , . . . So nobody, there's no room for miscommunication. If you have not, like today you have this recording. If you misunderstood something, I said, you can go back."

- Poor communication between team members:

The agile manifesto places a higher priority on people and interactions than on procedures and equipment, emphasizing in particular that in-person contact is the most efficient means of fostering cooperation and information sharing within agile teams. The participants said that teams that collaborate well can tackle difficult problems, such as RV, with

practical answers. But, in practice, this is frequently difficult due to large, dispersed teams, which divert information in unexpected ways. In addition, the participants acknowledged that inadequate communication between the development team can result in disparities in project information. These discrepancies can cause misunderstandings, misinterpretations of requirements, and unnecessary confusion within the team, making it challenging to work towards a common objective. In this regard, a participant (P5) said the following:

“the bad communication, of course, between the employees in the team, it, also decreased the productivity. . . . the miscommunication, the lack of communication between the team members. Also, I noticed there, within the same project, different people have different information”

A participant (P12) shared an experience where a miscommunication occurred regarding the deployment environment. The developer failed to discuss which environment the code should be deployed to, resulting in the deployment being carried out on two development environments instead of the designated testing environment. This miscommunication created confusion and limited the testers' access to the code, hindering the testing process. A participant (P12) said the following:

“While it was once when the developer forgot which, she forgot to discuss which environment that is, deployed. So that's one miscommunication that we had. So we, they deployed the effects on two dev, but then testers only have, access to the testing environment.”

- Communication between the development team and external stakeholders:

According to the participants, it is essential to build and maintain stakeholder engagement to reduce RV. Due to this, it is imperative to maintain constant communication and collaboration. In this regard, eight participants considered robust, unambiguous communication between the development team and external stakeholders to substantially reduce the potential for misinterpretations and abrupt changes in requirements, safeguarding against unexpected project disruptions and delays. In this context, a participant (P2) noted the following:

“managing stakeholders, keeping them updated of what's happening, especially when it comes to decision making, on products is important to control requirements.”

As reflected by one participant (P5), unclear stakeholder requirements are one of the most common challenges agile teams face. A stakeholder may understand in general what they want, but they may not be able to clearly define their requirements. Therefore, without good communication, there is a higher risk of building the wrong product and wasting valuable time and resources. The participant (P5) said the following:

“requirements have been changed, but you don't know about it because your manager didn't tell you, because of the lack of communication.”

- Reluctance to discuss and ask questions:

A few participants (three participants) believed that when team members are hesitant to discuss and ask questions, it can lead to a host of challenges, including requirement volatility. Therefore, by fostering a culture that encourages open dialogue and active engagement, teams can address potential issues early on and prevent them from escalating into larger problems later in the project. In this regard, a participant (P11) said the following:

“And if you think that you're not getting answered, ask more. There is nothing called a stupid question. So move on with your questions, and make sure that you get answers. Make sure that you question until you get the answer that you're looking for.”

Additionally, a participant (P5) shed light on a common challenge faced by managers' reluctance to engage in discussions with third parties regarding documentation requirements or potential deadline extensions. In some cases, managers may feel uncomfortable or hesitant to raise such matters, which can hinder effective communication. A participant (P5) explained the following:

“Sometimes, I also notice that many managers, don’t feel comfortable, to ask, . . . , . . . They feel shame to discuss. They feel uncomfortable discussing. during one of the meetings, when I met the third person, I asked personally, could it be possible not to ask us the same pre-preparation of the documentation from the team in such a tough, time in one day? She said, yes, of course, but why? But you, the manager didn’t discuss this issue with me. So, which is the lack of communication.”

As reflected by a participant (P5), feeling reluctant to ask questions as a team or a team member can stem from different reasons, such as a worry of judgment, a lack of self-confidence, or uncertainty about the requirements. However, it is vital to know that asking questions is an important practice for reducing conflicts and misunderstandings about requirements.

5.1.3. Stakeholder Engagement

Some of the participants (ten participants) explained that stakeholder engagement can contribute to RV, where project requirements change or evolve during the project, affecting scope, cost, and timeline. Therefore, it is important to determine and document the stakeholder’s requirements accurately and identify their needs and expectations. In addition, the participants acknowledged the importance of verifying and validating the requirements to guarantee that they are realistic, comprehensive, consistent, and aligned with the project’s scope and goals. Therefore, stakeholder engagement remains vital for aligning expectations, creating a common vision, and developing an accurate and comprehensive set of requirements. A participant (P9) shared the following:

“So we have like weekly meetings, so, not just for the requirements, but also for the product. So, every time, it is, back and forth, between requirements and the product.”

As one of the participants reflected, as a project progresses, stakeholders may develop new expectations or business objectives that were not initially considered. This can cause a ripple effect on the requirements, making it crucial for project teams to continually reassess and adapt to ensure the project remains relevant and successful. In this regard, a participant (P1) said the following:

“are always in contact with the customer. So we have like weekly meetings, so, not just for the requirements, but also for the product. So, every time, it is, back and forth, between requirements and the product”.

Furthermore, the participant emphasized the iterative nature of the requirements process. As a participant (P7) described, when stakeholders suggest changes or updates, the team is responsive and incorporates them into the project. This iterative approach ensures that the evolving needs and preferences of stakeholders are addressed, resulting in a tailored and satisfactory end product. A participant (P7) stated the following:

“And if the stakeholders think, no, maybe this could be changed. Then again, we go back to the same process. So we open the same requirement and update the requirement, and whatever has to be changed, we start working on them.”

In addition, a participant (P10) confirmed the importance of demonstrating a product’s features and capabilities to customers. This demonstration helps customers understand the capabilities of the product and ensures that it aligns with their specific requirements. In this context, a participant (P10) said the following:

“if the customers want to do something very specific to themselves, I must be sure it can be customized, that part. . . , . . . showing them a demo and demonstrate how, how this chatbot or how this product is working and what is the capabilities of it, what can be done, and what can be done in the future.”

Additionally, some of the participants (four participants) explained that there is likely to be a variety of stakeholders with different backgrounds, expertise, and interests. This diversity can lead to conflicting requirements, as stakeholders may have unique perspectives

and priorities. Effectively managing and reconciling these differences is essential to maintaining a cohesive project vision and minimizing RV. A participant (P7) noted the following:

“Every implementation will undergo some of the other kind of revision, because in the end, we have multiple people, right, pitching in with their opinions, and every person has a different opinion. So with that, some of the other amount of change is expected.”

Furthermore, some of the participants (seven participants) considered that ensuring that requirements are well defined and understood by all stakeholders is crucial to minimizing potential conflicts and reducing volatility. A lack of clarity or changing requirements can pose difficulties for the development team, leading to ambiguity and potential misdirection. In this context, a participant (P6) explained the following:

“the customers and their knowledge. It comes into play in such situations because when they don’t know what they want, they can lead us around the bush.”

Another participant (P4) considered that the incorporation of unexpected feedback into the software development process can lead to RV and pose challenges. The stakeholders may provide new information or feedback that was not initially anticipated. This can necessitate changes to the requirements and contribute to volatility. A participant (P2) stated that the following:

“you already know that, okay..., the management will come with their requirements. You might have changed from, customers you can have from, you know, outside team. So you try to touch base with all of those people and, you know, update your requirement before you start work”

5.1.4. Interplay of Expertise and Domain Knowledge

The participants believed that a team with the necessary domain knowledge and technical skills can better understand and address complex requirements, leading to successful project outcomes. This is because possessing broad experience is very beneficial because software development teams collaborate with technical teams and businesses to deliver value through software or technological solutions. Conversely, a lack of expertise can result in RV, miscommunication, and ultimately, project failure. A participant (P6) stated the following:

“What happened when feature sets were not understood when the user stories were not understood, is when things went wrong. . . , . . .technical understanding plays a huge role. It plays a big role because if you don’t understand the solution properly. . . , . . .the issue with not understanding the issue requirement. So you need to have a good technical understanding.”

In addition, the participants mentioned that insufficient domain knowledge among customers can have detrimental effects on RV. When customers lack a deep understanding of the domain, they may question the time and resources allocated to the implementation process. A participant (P2) stated the following:

“If your customer does not have the knowledge and thereby doesn’t understand the value of the technology, the first thing he’ll argue is, why do you need so many hours to implement this? Because we pay and develop hours, right? . . . , . . . the truth is, because of their lack of understanding, they don’t know the value of my time and what I’m doing.”

Furthermore, a participant (P4) further emphasized the importance of prior experience with similar products, as it enables finer details to be captured that may otherwise be overlooked. In this context, a participant (P4) said the following:

“So knowledge, when people who have prior experience with a certain product come into the picture, almost always a finer detail would be, written down. That happens very often.”

Additionally, the participants believed that a lack of domain knowledge among team members could lead to miscommunication and errors in technical specifications. A participant (P6) stated the following:

“domain knowledge is kind of a key. . . , . . . we would understand much better how the data flows and where it is restricted. And within that restriction, we can have back and forth on these requirements and things like that.”

5.1.5. Regulatory Compliance

Another important cause of RV extracted from the participants' explanations of causes of RV in software development is regulatory compliance. Regulatory compliance is the adherence of organizations to laws, regulations, guidelines, and specifications relevant to their industry, operations, or specific products and services, typically established by governmental agencies and industry-specific bodies. Ensuring compliance is crucial for maintaining a positive reputation and consumer trust and avoiding financial penalties or legal actions. As highlighted by the respondent (P2), failure to consider regulatory requirements during the planning phase can lead to volatile requirements and potential violations, resulting in significant consequences for the project and team. A participant (P6) stated the following:

“There's a very probability that when you do see the regulations, what the policy is, the government policies are concerning that program. It could change your requirements a bit. . . , . . . it should be part of your plan, your previous planning, or you know, the things you have to check before you kick off. Say, okay, these are the requirements for the product. What do the regulations say about this product? What is the policy out there?”

Another participant (P7) also confirmed this point and stated the following:

“the regulations can always change. And, if there is a change, then we will again, rework it. So, because at the end of the day, we can't sell products without their, without compliance with the regulations.”

Furthermore, the participants believed that team members must have a thorough understanding of compliance issues to prevent unintentional violations, as evidenced by the experience shared by one participant (P6) where a developer inadvertently violated General Data Protection Regulation (GDPR) laws [33]. In this context, the participant (P6) stated the following:

“The developer took a database backup from the live environment to test in his test environment back in another country. Now, when he did that, he was violating GDPR laws. If people got to know they're in big trouble”

The previous idea was shared with other participants, but they explained that compliance with the Health Insurance Portability and Accountability Act (HIPAA) in the United States [34] can cause requirement changes. In this context, a participant (P11) stated the following:

“We cannot use the production data either because the environment doesn't compliment it, or it can be because of this HIPAA violation. . . , . . .when being a product owner or being a scrum master or being this business analyst, those are the folks who should communicate that to the development team saying that we should mock up the data,”

In addition, the participants believed that geopolitical changes and evolving government policies could also contribute to RV. As described by two participants, a change in government affected their ongoing project, requiring them to adapt to new legislation and rules. Similarly, another participant spoke about the challenges of working on a project in a different country, where team members may not be aware of the local legal requirements. A participant (P1) stated the following:

“If we are doing something, for China, there are a lot of changes that take place in a very short time, it can also occur, within the project phase. So our project phase is roughly one hour, one year. And, at the start of the project phase, they might have given us a requirement, at the middle of the project phase.”

5.1.6. Requirement Ambiguity and Uncertainty

Requirement ambiguity and uncertainty are another significant cause of RV that was identified from the participants' explanations of reasons for RV in software development. Requirement ambiguity and uncertainty refer to the lack of clarity or precision in defining the needs and expectations of a software project. Participants noted that assumptions could lead to incorrect interpretations of requirements. This was attributed to the tendency of developers to make assumptions due to a lack of clarity in requirements documentation. One participant mentioned that developers may not read the user stories in detail and make assumptions based on incomplete information. A participant (P2) said the following:

"So mostly developers don't like to read. And, so if you, if you give them user stories, and of course if the user stories are much..., . . . And so basically, because they don't do that, they kind of make assumptions. . ."

Another participant (P7) also confirmed this point and stated the following:

"If it's not clear, we take it in our hands and we do what we think is correct."

In addition, the participants considered communication as a key factor causing ambiguity and uncertainty in requirements. Several participants noted that developers tended to avoid reading, while one participant stated that communication failures often led to situations where team members did not fully understand the requirements until a tangible product was available. In this regard, a participant (P6) said the following:

"in my experience, most of these things came into play because communication fails, because I don't think anyone understands what they're talking about until they have a tangible product."

Furthermore, the participants identified expertise and experience as critical components contributing to requirements' ambiguity and uncertainty. As mentioned by one participant, without the necessary experience and knowledge, software projects are more likely to encounter difficulties and potentially fail. In this regard, a participant (P6) stated the following:

"concept is great, it works excellent, the intent is great, but when it comes to the practicality on the floor, the challenges are, it is all about the people that make a difference, okay? If you don't have the right people, and if you don't have the rightly experienced people, these things all fall apart"

Moreover, three participants believed that a lack of specific details in the requirements led to ambiguity and uncertainty. For instance, one participant shared an example where the client wanted a specific date format, but the requirement was not communicated clearly. P11 highlighted a situation of confusion and the rework of a completed project where the client requested a rollback because the delivered product did not match their expectations:

"The project has been completed, but now the product is in production. But then they're asking, or the client is asking for a rollback. Why? Because it wasn't the actual product that they asked for."

Some of the participants (two participants) reported that changes during the development process can be a significant source of ambiguity and uncertainty. These changes may arise due to evolving client needs or a lack of initial clarity. As a result, development teams may need to adapt their work, causing potential disruptions, delays, and rework. A participant (P4) said the following:

"there would be, clients that aren't really, defined at the beginning. And, the client would only be able to find out, I need this, type of feature in the product while the software development life cycle is already in the middle. So, this, that's why this happens. That's why some requirements come in the middle because of that."

5.2. Strategies That Agile Practitioners Think Are Effective in Reducing RV

The agile practices that the participants have found to be effective in reducing RV are divided into three sub-themes: (1) agile ceremonies; (2) agile artifacts; and (3) agile teams.

5.2.1. Agile Ceremonies

Participants underlined the significance of various agile practices, such as scrum meetings, sprints, retrospective sprints, daily stand-up meetings, and product backlog refinement. A backlog is an organized document that contains all the requirements. Use cases or user stories are used to describe the requirements. The refinement of a product backlog is a continuous process of validating requirements. These practices and ongoing meetings enable teams to adapt to changes and prioritize tasks efficiently, thereby serving as effective strategies for reducing RV in agile software development projects. In this regard, a participant (P2) stated the following:

“sprint events, they help with communication a lot because there’s like touch points in time to time, so that helps, with communication.”

According to the participants, by incorporating daily stand-up meetings, sprint refinement sessions, retrospectives, and sprint planning meetings, agile methodologies foster frequent communication and collaboration. These gatherings ensure that team members continually share information, discuss progress, and tackle challenges together, creating a supportive environment. In this regard, a participant (P12) said the following:

“during sprint reviews, we try to go through what happened during the last sprint and try to see if we can have a demo of what happened. So we see progress in life. Otherwise, it’s more like a two-week status update of what was done. And, sprint for perspective is about three questions where we ask about the good things that happened during the sprint, what were the challenges we faced, and things that can be improved.”

In addition, the participants explained the role of daily standups as a crucial tool for effective communication and collaboration within a team. A participant (P9) stated the following:

“Communication is super important, within your team and also with, management. So managing stakeholders, keeping them updated of what’s happening. . . . important you tell them early on that, you know, keeping that for like, a week or two weeks. So of course, that’s what daily stand-ups are for, from communicating with your internal every day.”

5.2.2. Agile Artifacts

Another effective agile practice for reducing RV reported by the participants is agile artifacts, which include user stories, epics, backlogs, wireframes, and acceptance criteria, among others. These artifacts help teams break down complex projects into manageable tasks, promote effective communication, and provide a clear overview of a project’s status.

One participant explained how detailed functional requirements, system requirements, and more detailed requirements are developed and documented. A participant (P2) stated the following:

“I usually call this high-level product requirements. So basically that’s what comes out of our ideation. . . . We just have like bullet points. . . . develop that into like detail product requirement agreement. . . . that we break it down in user stories.”

In addition, the participants believed that user stories and epics play a crucial role in creating a clear vision of a product’s functionality and user experience. They serve as the foundation for agile software development, ensuring that the end-user’s needs are prioritized and addressed. In addition, user stories define features and functionality from the user’s perspective and are often broken down into developer tickets. A participant stated the following:

“feature leading user story, and then what is the functionality that this user story is expecting. And so from that, we get, how do we navigate this, certain functionality. So, and then we,

create possible scenarios to test this specific, functionality. So for example, they want a login page for, this certain, user or client. So we needed to test using the code of this client. And then we need to create, positive-negative testing. We need to test, we also need to do unit testing. So we, it's very, there's gonna be like a lot of scenarios from that."

Another participant (P9) explained epics as large user stories that cover the broader, high-level features or functionalities of a product. They are usually divided into smaller, more manageable user stories that can be completed within a specific sprint or iteration. Epics help teams identify and prioritize the most important aspects of a project, ensuring that the development process remains focused and efficient. In this regard, a participant (P9) said the following:

"we discuss if we can take it up based on the bandwidth we see for our teams, and then we, create epics and stories out of them."

Additionally, the participants believed that acceptance criteria play a pivotal role in their agile projects, ensuring that teams have a clear understanding of the expected outcomes and the desired functionality of a user story or epic. These predefined conditions serve as a benchmark for determining whether a user story or epic is complete and acceptable, which will reduce RV. Confirming this point, a participant (P10) said the following:

"our part is starting with here, acceptance criteria, and we are giving the functional requirements in here for a basic story."

Additionally, the participants explained that acceptance criteria aid in the User Acceptance Testing (UAT) phase, where the user, customer, or product owner reviews the completed work to ensure it meets the agreed-upon criteria, which will also reduce RV. In this context, a participant (P8) stated the following:

"So to make sure that we are clear about requirements and what we are building. We do have the UAT or user accepted testing, which means that we have a phase that the user or the customer or the representative or the customer, maybe in that case, the product owner, just reviewed the ticket or what we have built."

Furthermore, the participants emphasized that visual representation is a crucial aspect of product development, as it enables reducing RV by aiding teams to communicate and understand a product's layout and flow effectively. Wireframes, for instance, are a valuable tool for illustrating a product's structure and navigation, fostering a shared understanding among team members. In this regard, a participant (P6) stated the following:

"Typically we'll use, Wireframes, we'll use cases, to translate the requirement into something understandable by the technical solutions team."

Moreover, the majority of the participants (eight participants) believed that maintaining and managing changes in the product backlog is essential to addressing RV and ensuring a project's smooth progression. In this regard, a participant (P2) stated the following:

"Gathering these all issues created in Jira, in a backlog. I say, this is important to us because we need to use backlog, and we need to, separate the issues and order them respectively their priorities. . . , . . . deal with them at the first phase of the project, and we deal with them, a solid backlog, maybe backlog."

Some participants (four participants) considered that by documenting lessons learned from previous or ongoing projects, teams can identify potential bottlenecks, issues to avoid, and opportunities for improvement, ensuring that they do not repeat the same mistakes. This practice enables the team to maintain a record of critical insights gained from past experiences, such as changes in customer preferences or project requirements, and this helps to reduce RV. A participant (P1) said the following:

"We have documented all the lessons learned with the customer. For example, the customer has made some changes. There is some change in, the person, the contact person, the, the new contact person has, different opinion about this thing, or the different, preference

with the color of this thing, all these things are documented, and we have this as lessons learned, in a file."

Another participant referred to documentation tools that provide teams with robust platforms for capturing and managing requirements, offering a centralized location to store, access, and share project-related requirements. These tools can help streamline communication among team members, reducing the risk of misunderstandings or miscommunication that could increase the RV and conflicts in requirements. A participant (P7) said the following:

"We use git to document all the code like we have all the code, stored in the git, and if some part of the code has to be changed, we know what has to be changed. And like we have this history of, code implementations and changes. So, it has never been a problem."

5.2.3. Agile Teams

In the agile team theme, there was a lot of similarity among the participants' answers. This is due to the fact that the agile teams of the participants have common features that are shared among different agile methodologies. They stated that requirement misunderstanding, miscommunication, and poor requirement analysis are the effects of a less experienced and practiced team. The participants in this study also identified the expertise of agile teams as a highly effective technique for reducing RV. By emphasizing adaptability and collaboration, agile teams are prepared for the possibility that requirements can change at any time, allowing them to respond swiftly and deliver value to customers. This understanding highlights the importance and necessity of collaboration and communication, as a lack of these crucial elements may lead to RV. Additionally, the participants mentioned the significance of self-sufficient teams to make decisions, assume responsibility, and actively contribute to a project's success. Furthermore, agile teams are becoming more resilient and capable of tackling challenges and addressing evolving requirements. In this regard, a participant (P10) said the following:

"you need to have a different kind of person in terms of their experience in a team. If you have only seniors, it, couldn't be as well done. Or if you have only juniors, it couldn't, well. You need to have verity, different ages, different cultures, and your team must be, very, verity. . . . my team only I have juniors, so they couldn't, understand very well. . . . So all issues were some slower."

Furthermore, the participants believed that a diverse range of skills, knowledge, and technical proficiencies within agile teams is vital for reducing RV because this combination of expertise ultimately leads to a better understanding of functional and technical requirements. In this context, a participant (P10) highlighted the need for senior developers in a team:

"I need, also a senior developer in my development team for some specific project, not only juniors, because they are very new and they are not, familiar with this kind of, methodology, and they do not easily understand the requirements, and they are, that's part that causes us to decrease our velocity and maybe causing to rework in the future."

A wider range of competencies among a project team's members is necessary for an agile project to be successful in reducing RV. This is because, unlike traditional development teams, an agile team works on several phases of the development approach simultaneously rather than one at a time. For example, developers are now expected to perform a variety of tasks in their teams in addition to programming. However, this may affect their ability to respond to RV, especially in large-scale agile projects with less experienced and skilled teams. For example, SCRUM works well for smaller projects involving fewer team members; when it is used for larger projects, development efficiency is reduced.

Additionally, four of the participants considered continuous learning and knowledge sharing as critical components of agile teams to reduce RV. Staying current with the latest trends, tools, and techniques ensures teams maintain a competitive edge in the ever-evolving software development landscape. Agile practices, such as regular retrospec-

tives, foster reflection on performance, the identification of areas for improvement, and knowledge exchange among team members. This culture of continuous learning enables agile teams to respond effectively to any change in requirements. A participant (P8) stated the following:

“If we have a new person in the team who’s not used to the development process, then he’s always given, a partner, godfather kind of person. So he’s always given help who will coordinate with him and make him learn the things that we do. This helps to Clarify requirements and reduce ambiguity”

Consequently, software organizations have to train their agile teams so that they can increase their expertise by providing them with resources, training sessions, tactics, and guidelines on how to reduce RV.

Furthermore, although an agile team is distinctive in that it has limited roles, is a cross-functional, authorized, collaborative, self-organized, small team rather than independently determined, and decides how it will build a project’s artifact or service, the participants believed that the hiring or leaving of a team member may impact the ability of a team to respond efficiently to RV. Therefore, team stability leads to an increase in flexibility and responsiveness to project requirement changes, ultimately reducing RV. A participant (P9) stated the following:

“It increases the misunderstanding of requirements definitely because we lose some know-how, but, every member, who leaves the project or changes to another project has a period so that he can, educate and he can tell the new person who will continue the work, with all the know-how and, information that is required. So every time, we have somebody who’s going away from this project teaching somebody new how it is done.”

Moreover, the participants believed that by fostering a positive, supportive atmosphere, agile teams ensure that their members feel valued, motivated, and equipped to deliver their best work. A healthy emotional environment leads to higher levels of productivity and innovation, playing a crucial role in helping a team navigate RV with resilience and determination. In this regard, a participant (P4) stated the following:

“It’s important to keep your team happy. I would say motivated. . . mainly, it’s, it’s very important because if your team is happy, then. . . it’s very helpful for addressing effectively the challenges of any change in requirements. And, you have a team that wants to do work and they’re helpful and you know, they’re innovative as well.”

6. Discussion and Future Directions

The purpose of this study was to explore factors that cause RV in software development projects and agile practices to mitigate and reduce it from the perspectives of software development practitioners. The study comprehensively compares with the existing literature and research the factors influencing RV in software development projects. The findings resonate with and build upon the existing body of knowledge, providing deeper insights into the complex interplay of market feedback, competition, communication, stakeholder engagement, expertise, regulatory compliance, requirement ambiguity, and uncertainty. The thematic analysis of interview data highlights the importance of addressing these factors in managing RV and ensuring successful project outcomes. Integrating these findings with the existing literature resulted in the synthesis of key strategies and best practices for addressing these challenges. These include incorporating regulatory requirements early, fostering effective communication, providing detailed and explicit requirements, and leveraging experience and expertise.

A total of six factors and three practices were identified by the 12 participants. The findings indicate that the primary causes of RV in software development projects, as reported by the participants, are as follows: (1) market feedback and competition; (2) communication; (3) stakeholder engagement; (4) interplay of expertise and domain knowledge; (5) regulatory compliance; and (6) requirement ambiguity and uncertainty. Additionally, the agile practices that the participants have found to be effective in reducing RV are as

follows: (1) agile ceremonies; (2) agile artifacts; and (3) agile teams. The findings of the current study showed that communication received more attention from participants than the other factors. The results of this study support previous studies and reviews that also found an increased number of studies focused on communication in mitigating RV [2,6,7,9]. However, this indicates a gap regarding other factors such as regulatory compliance, which requires significant changes in project requirements due to various regulations and policies, such as GDPR [30].

Like the findings of the current study, persuasive evidence from various studies [1,3] has consistently emphasized that robust communication is essential for reducing RV and underscored the significance of prioritizing communication for the success of software development projects. Furthermore, transparent communication was reported by the participants as a practice to enable project teams to avoid misunderstandings, assumptions, and ambiguous expectations among team members. However, it is vital to acknowledge that excessive communication can present challenges in managing RV. Therefore, an optimal balance between open communication and efficient requirements management is crucial for successfully navigating RV complexities.

Additionally, there is no consensus among the study's participants regarding some of the factors that are identified in Section 5. This may be explained based on the participant's experiences in certain contexts and project settings; see Table 1. The findings of the current study showed that several factors that cause RV are raised according to the project's complexity, the methodology used, and the organization's size and structure. In this regard, several factors can be overcome by applying appropriate settings and using proper practices and techniques. For example, requirement ambiguity and uncertainty are not a problem when the scope is pretty well defined, while the latter factor (uncertainty) increases with the ambiguity of the scope. Therefore, the alignment between organizational setting complexities and agile appropriateness should be evaluated before adopting an agile approach. This is due to the importance of organizational context in requirements engineering since agile methods stress responsiveness and informal communication that are rigid to guarantee across multidisciplinary teams in a large organization [34]. Therefore, this challenge demands different solutions to improve agile practice to manage requirements, taking into consideration the high capacity for change in requirements and the abilities of agile teams to work in diverse organizational settings and contexts, for example, the need to include the classic role of requirements engineering in large agile projects and complex organization settings to handle the difficulties of RV. However, several previous studies have described specific contexts and settings in which these factors were identified, such as in [35], while others have identified factors in more general contexts and settings, such as in this study.

Furthermore, this study emphasized the interconnectedness of these factors and the need for a holistic approach to managing RV in software development projects [36]. By understanding the intricate relationships between these factors, software teams can develop more effective strategies for mitigating RV and improving overall project success. Similar to the findings reported by [37–43], the participants in the current study considered regulatory compliance to be one of the most challenging issues. This difficulty in reducing RV caused by this factor, as reported by the participants, might be due to complex interactions between regulatory compliance issues and other issues such as the interplay of expertise and domain knowledge. For instance, to mitigate RV, it is necessary to ensure that a project adheres to strict data protection laws such as HIPA [35], which requires adequate expertise and legal issues to determine what kind of data should be considered private or public and the type of access control that should be applied. In this regard, the most challenging issue reported by the participants in reducing RV was regulatory compliance. However, few studies have handled the impacts of this factor and its effects on a project's success [40,42]. Therefore, further research is recommended on this factor.

Furthermore, stakeholder engagement was reported as an important factor that causes RV due to the fact that inadequate stakeholder involvement often leads to incomplete or

missing requirements, resulting in suboptimal communication [40]. Therefore, the study underscores the importance of effective communication with all types of stakeholders, such as internal team members, suppliers, logistics personnel, and other departments involved in a project. Confirming this point, the findings of the current study showed that software projects still suffer from a lack of a clear plan to engage stakeholders, especially resistant and neutral stakeholders. Similar to our findings, Bano et al. [40] also reported that the importance of the stakeholder engagement aspect was stressed by the participants in group interviews, where the majority of the participants considered that unanticipated feedback from stakeholders can further increase RV by necessitating changes to a project's requirements. Consequently, further research is recommended on employing effective practices for stakeholder engagement to mitigate RV.

Moreover, the participants in this study emphasized that market feedback and competition can contribute to RV by driving companies to adapt their product requirements to meet the changing demands of the market. In addition, this study showed that by staying responsive to market changes and incorporating customer feedback, companies can improve their products or services, helping mitigate the impact of RV from market factors. Similar to our findings, Anjum et al. [4] emphasized that late customer feedback and lengthy development cycles significantly contribute to RV. Furthermore, the rapidly changing software industry, characterized by swiftly evolving technology and competitors, intensifies these issues. In this context, refs. [6,14] highlighted that rapidly changing stakeholder preferences place additional pressure on delivering products quickly to the market to avoid any additional changes in requirements. Therefore, to address these challenges, organizations must prioritize early customer feedback and adapt to change. In addition, embracing early customer feedback can lead to a better understanding of requirements and fewer changes later in the development cycle [4]. Similarly, Madampe et al. [6] stated that continuously adapting development strategies allows organizations to accommodate evolving needs and maintain competitiveness in the fast-paced software industry.

According to the apparent findings in this study, the participants reported a preference for written communication, like email and group chats, over verbal communication, as it provides a record of instructions and minimizes miscommunication. This aligns with Madampe et al. [6], who recommended using tools that enhance communication and collaboration, such as video conferencing, chat applications, and documentation tools. In addition, the role of user stories as documentation in agile software development was debated among the participants of this study. Some participants contend that user stories are the predominant form of documentation, while others observe that teams often provide limited detail in user stories. Concerns have been raised that user stories may not encompass all aspects of software development, including vulnerabilities, bugs, unexpected termination, and undefined behavior [41]. Nonetheless, a study has demonstrated that the majority of respondents acknowledge the significance of documentation in agile development. However, the absence of documentation becomes particularly problematic when new team members join a project, as highlighted by [9]. In this regard, this study also posits that documentation as a means of communication and knowledge transfer can serve as solutions to address challenges arising from employee turnover.

There are many reasons for organizations to adopt agile principles, including increasing speed to market, meeting customer demands, or improving team productivity. Besides these reasons, participants reported that agile methodologies offer flexibility in accommodating fluctuating project requirements. However, an interesting observation is that some software companies opt for an agile transformation without ensuring their teams have a thorough understanding of agile principles. This typically results in a superficial adoption of agile practices, where pre-existing teams are simply rebranded as 'agile' teams, and members are designated roles without a thorough understanding of agile principles. This practice has been aptly referred to by a respondent as akin to serving an 'old drink in a new bottle'.

Even though the Agile principles are designed to accommodate change, uncontrolled changes can still present challenges [38,39]. It is crucial to understand that agile team members may experience a range of emotions when requirements change, which can impact their strategies and efficiency [37]. In this context, the results from this study revealed the importance of keeping the team motivated and satisfied, as this fosters increased productivity and innovation. Open communication and feedback are also essential in motivating agile teams, helping them stay aligned and make necessary adjustments. However, changes in a project's scope and requirements can extend the project's timeline, causing frustration and decreased morale. Despite this, this study showed that some team members may complain, but they still manage to get the job done. This highlights the importance of resilience and adaptability in agile teams, even when faced with challenges and changing circumstances.

Even though this study included participants with different roles in the agile team, such as scrum master, product owner, project manager, and tester, there is a lot of similarity in the results. In addition, the differences between these roles did not appear clearly in the results. Therefore, this study did not show the treatment of the participants' roles as it did not have effects on the results. This similarity in findings is due to the cross-functional agile team structure, where each member has a good set of different skills. In addition, collaboration between team members about requirements offers a shared understanding among team members about the different challenges encountered regarding RV. Therefore, further research is recommended to reveal more details regarding the relationship between RV and agile team members' roles and experiences.

The study's scope, which includes diverse industries and various project types, highlighted the variability in RV's causes based on these factors. Consequently, further research focusing on specific industries, project types, and agile methodologies is crucial for a more comprehensive understanding of RV in the context of agile software development. Future research could investigate industry-specific factors influencing RV, the susceptibility of certain project types or agile methodologies to RV, and potential differences between various agile methodologies regarding their ability to manage RV. This approach would contribute to developing tailored strategies for addressing RV's unique challenges within specific agile frameworks and lead to improved project outcomes and more efficient software development processes. Additionally, employing longitudinal research designs to track RV's evolution over time could enable researchers to identify patterns useful for predicting and mitigating RV in future projects, better understand RV dynamics in agile projects, and develop evidence-based recommendations for practitioners.

7. Challenges and Limitations

There are specific limitations that the current study faces. Despite Guest et al. [23] claiming that a sample size of 12 participants was deemed sufficient, it is worth noting that the relatively small sample size may constrain the generalizability of the findings. Additionally, the characteristic of a qualitative study is that it does not seek to be generalizable [44]. Furthermore, we only used one research method for data collection. This shows that no other complementary method was used to validate the results. Having a hybrid method to triangulate the qualitative findings with quantitative ones may have offered us a high degree of internal validity. Utilizing various approaches in data collection could aid the researchers in verifying the reliability and validity of the collected data [23]. Moreover, this research provided the participants' observations and experiences about software development. In these types of studies, reviewing the impartiality and objectivity of the participants' answers is challenging, and the presented descriptions may not be enough. Therefore, further surveys are suggested concerning the attitudes of team members toward requirement volatility in software projects while designing and implementing software applications.

8. Conclusions

This research study offers compelling evidence that adopting agile methodologies can significantly accommodate RV. The study, which included 12 participants with diverse

backgrounds from various countries, industries, and roles, employed purposive sampling to ensure a comprehensive understanding of RV in software development. The findings indicated six primary factors that cause RV in software development projects, as follows: (1) market feedback and competition; (2) communication; (3) stakeholder engagement; (4) interplay of expertise and domain knowledge; (5) regulatory compliance; and (6) requirement ambiguity and uncertainty. In addition, the findings indicated that by addressing identified challenges and fostering a culture of effective communication, stakeholder engagement, and adaptability, teams can minimize RV's negative impact on projects and improve their overall outcomes. Agile methodologies, including ceremonies, artifacts, and teams, have proven effective in reducing RV and promoting collaboration, transparency, and rapid response to changes.

This study showed that some factors and practices to mitigate RV received more attention from the participants than others that have a negative impact on RV in software development projects. Therefore, further research is required on these factors.

Author Contributions: Conceptualization, A.M. and J.M.K.; methodology, A.M. and J.M.K.; validation, A.M. and J.M.K.; formal analysis, A.M. and J.M.K.; investigation, A.M.; resources, A.M.; data curation, A.M. and J.M.K.; writing—original draft preparation, A.M.; writing—review and editing, A.M.; visualization, A.M.; supervision, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and Swedish law [42]. Ethical approval was not required since data were anonymized and not personally sensitive according to the Swedish Ethical Review Act (SFS 2003:460 3 §§) [42].

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data that support the findings of this study are available upon request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Semi-Structured Interview Guide

Disclaimer

- This interview is being conducted for research purposes and may be recorded and transcribed for further analysis.
- The information provided will be kept confidential and anonymous.
- Participation in this interview is voluntary, and you may withdraw at any time.
- Open-ended questions will be asked, but you are free to discuss any additional information you feel is relevant.
- The results of this interview may be published, but no personal identifying information will be included.
- By agreeing to participate in this interview, you acknowledge that you have understood the disclaimer.

What is Requirement Volatility?

Requirement volatility is the degree of instability, change, modification, or deletion of software requirements during the development process, caused by incomplete requirements, evolving technology, changing stakeholder needs, scope creep, miscommunication, and regulatory or legal changes.

General Questions

1. Name
2. Country of Residence or Work Location
3. How many years of experience do you have in agile software development projects?

4. What is your role in the Agile software development process?
5. Size of the organization (number of employees)

Requirement Volatility

1. In your experience, what is the main source of changing requirements for software products?
2. Major impacts of requirement volatility on Agile software development projects?
3. How has RV affected project timelines, budgets, and overall quality?
4. How do you manage requirement volatility in software development projects?
5. How do you document changes in requirements?
6. Communication
 - a. Have you experienced any communication barriers during software development projects in the past?
 - b. Have you ever faced any misunderstandings or requirement changes due to communication breakdowns in software development projects?
7. Stakeholder
 - a. How important do you believe stakeholder engagement is for the success of software development projects?
 - b. How frequently do you use user stories to convey requirements?
8. Expertise
 - a. How often does your team face difficulties in understanding and interpreting requirements?
 - b. To what extent do you think a lack of domain knowledge among software developers or customers impacts requirements, compliance, and security?
9. Regulatory Compliance
 - a. To what extent do you believe regulatory compliance affects requirement volatility in software development projects?
 - b. How do you ensure that the software development process meets the required regulatory standards?

Strategies for Mitigating Requirement Volatility:

1. What strategies have you found to be effective in mitigating requirement volatility in Agile projects?
2. How do you incorporate these strategies into your Agile development process?
3. Can you share any specific examples or case studies where these strategies were particularly successful?

Agile Practices to Address Requirement Volatility

1. In your opinion, what improvements could be made to Agile practices to address requirement volatility more effectively?
2. Are there any tools or techniques that could be particularly beneficial in managing requirement volatility in Agile projects?
3. How can organizations support Agile teams in adapting to and managing requirement volatility more effectively?
4. How can Agile ceremonies, such as daily stand-up meetings and sprint reviews, improve communication among team members?

Challenges and Lessons Learned

1. What challenges have you encountered while managing requirement volatility in Agile projects?
2. Can you share any lessons learned or best practices that have emerged from your experiences in managing requirement volatility?

Final Thoughts and Recommendations

1. Based on your experiences, what recommendations would you give to Agile practitioners seeking to better manage requirement volatility in their projects?
2. Is there anything else you'd like to share or any additional insights regarding requirement volatility in Agile software development?

Conclusion

Thank you for the time and valuable insights.

Appendix B

Codebook: definitions

Source of RV

1. Brief definition: Market feedback and competition Full definition: Influence of customer feedback, competitor strategies, and market trends on the project requirements. Example: Introduction of a new product by a competitor, customer feedback suggesting a change in product features.
2. Brief definition: Lack of communication Full definition: Insufficient or ineffective communication among project stakeholders. Example: Misunderstandings between team members, unclear project objectives or requirements.
3. Brief definition: Lack of stakeholder engagement Full definition: Inadequate involvement of project stakeholders in decision-making and collaboration. Example: Stakeholders not participating in project meetings, limited input from stakeholders during project planning.
4. Brief definition: Lack of expertise Full definition: Limited knowledge or skills among team members, leading to difficulties in project execution. Example: Team members not familiar with specific technologies, lack of experience in a particular domain.
5. A brief definition: Regulatory compliance definition Full definition: Adherence to legal, industry, or organizational regulations and standards. Example: Compliance with data protection regulations, meeting industry-specific safety requirements.
6. Brief definition: Requirement ambiguity and uncertainty Full definition: Unclear or ill-defined project requirements leading to confusion and misinterpretation. Example: Vague project goals, conflicting requirements from different stakeholders.

Solutions to address RV:

1. A brief definition: Agile Ceremonies Full definition: Regular meetings and events in agile project management that facilitate communication, collaboration, and decision-making. Example: Sprint planning, daily stand-ups, sprint reviews, and sprint retrospectives.
2. A brief definition: Agile Artifacts Full definition: Tangible outputs in Agile project management used to track progress and facilitate collaboration. Examples: Product backlog, sprint backlog, user stories, and burndown charts.
3. A brief definition: Agile Team Full definition: A cross-functional, self-organizing group of individuals working together to achieve common project goals in an Agile environment. Example: Developers, testers, designers, and product owners collaborating to deliver increments of the product.

References

1. Nerur, S.; Mahapatra, R.; Mangalaraj, G. Challenges of migrating to agile methodologies. *Commun. ACM* **2005**, *48*, 72–78. [[CrossRef](#)]
2. Alsaqaf, W.; Daneva, M.; Wieringa, R. Understanding Challenging Situations in Agile Quality Requirements Engineering and Their Solution Strategies: Insights from a Case Study. In Proceedings of the 2018 IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, 20–24 August 2018.
3. Curcio, K.; Navarro, T.; Malucelli, A.; Reinehr, S. Requirements engineering: A systematic mapping study in agile software development. *J. Syst. Softw.* **2018**, *139*, 32–50. [[CrossRef](#)]
4. Anjum, S.K.; Wolff, C.; Toledo, N. Adapting Agile Principles for Requirements Engineering in Automotive Software Development. In Proceedings of the 2022 IEEE European Technology and Engineering Management Summit (E-TEMS), Bilbao, Spain, 22–24 May 2022.

5. Dasanayake, S.; Aaramaa, S.; Markkula, J.; Oivo, M. Impact of requirements volatility on software architecture: How do software teams keep up with ever-changing requirements? *J. Softw. Evol. Process* **2019**, *31*, e2160. [CrossRef]
6. Madampe, K.; Hoda, R.; Grundy, J. A Faceted Taxonomy of Requirements Changes in Agile Contexts. *IEEE Trans. Softw. Eng.* **2022**, *48*, 3737–3752. [CrossRef]
7. Kassab, M.; DeFranco, J.; Graciano Neto, V. An Empirical Investigation on the Satisfaction Levels with the Requirements Engineering Practices: Agile vs. Waterfall. In Proceedings of the 2018 IEEE International Professional Communication Conference (ProComm), Toronto, ON, Canada, 22–25 July 2018.
8. Daud Haiderzai, M.; Vranić, V. Identifying and Involving the Real End User in Software Development: Towards a Pattern Language. In Proceedings of the 27th European Conference on Pattern Languages of Programs, Irsee, Germany, 6–10 July 2022; pp. 1–11.
9. Gupta, A.; Poels, G.; Bera, P. Using Conceptual Models in Agile Software Development: A Possible Solution to Requirements Engineering Challenges in Agile Projects. *IEEE Access* **2022**, *10*, 119745–119766. [CrossRef]
10. Salmani, A.; Imani, A.; Bahrehvar, M.; Duffett-Leger, L.; Moshirpour, M. A Data-Centric Approach to Evaluate Requirements Engineering in Multidisciplinary Projects. In Proceedings of the 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Prague, Czech Republic, 9–12 October 2022.
11. Canedo, E.D.; Toffano Seidel Calazans, A.; Cerqueira, A.J.; Teixeira Costa, P.H.; Seidel Masson, E.T. Agile Teams' Perception in Privacy Requirements Elicitation: LGPD's compliance in Brazil. In Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, 20–24 September 2021.
12. Moyon, F.; Beckers, K.; Klepper, S.; Lachberger, P.; Bruegge, B. Towards Continuous Security Compliance in Agile Software Development at Scale. In Proceedings of the ICSE '18: 40th International Conference on Software Engineering, Gothenburg, Sweden, 29 May 2018; pp. 31–34.
13. Moyón, F.; Méndez, D.; Beckers, K.; Klepper, S. How to Integrate Security Compliance Requirements with Agile Software Engineering at Scale? Product-Focused Software Process Improvement. In Proceedings of the 21st International Conference, PROFES 2020, Turin, Italy, 25–27 November 2020.
14. Madampe, K.; Hoda, R.; Singh, P. Towards understanding emotional response to requirements changes in agile teams. In Proceedings of the ICSE '20: 42nd International Conference on Software Engineering, Seoul, Republic of Korea, 27 June–19 July 2020; pp. 37–40.
15. Herdika, H.R.; Budiardjo, E.K. Variability and Commonality Requirement Specification on Agile Software Development: Scrum, XP, Lean, and Kanban. In Proceedings of the 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), Yogyakarta, Indonesia, 15–16 September 2020.
16. Lorca, A.L.; Burrows, R.; Sterling, L. Teaching motivational models in agile requirements engineering. In Proceedings of the 2018 IEEE 8th International Workshop on Requirements Engineering Education and Training (REET), Banff, AB, Canada, 21 August 2018; IEEE: Piscataway, NJ, USA; pp. 30–39.
17. Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 6th ed.; Project Management Institute: Newtown Square, PA, USA, 2017.
18. Moran, A. *Managing Agile: Strategy, Implementation, Organization and People*; Springer International Publishing: Zurich, Switzerland, 2015.
19. Tenny, S.; Brannan, G.D.; Brannan, J.M.; Sharts-Hopko, N.C. *Qualitative Study*; StatPearls Publishing: Treasure Island, FL, USA, 2017.
20. Dybå, T.; Prikładnicki, R.; Rönkkö, K.; Seaman, C.; Sillito, J. Qualitative research in software engineering. *Empir. Softw. Eng.* **2011**, *16*, 425–429. [CrossRef]
21. Campbell, S.; Greenwood, M.; Prior, S.; Shearer, T.; Walkem, K.; Young, S.; Bywaters, D.; Walker, K. Purposive sampling: Complex or simple? Research case examples. *J. Res. Nurs.* **2020**, *25*, 652–661. [CrossRef] [PubMed]
22. Palinkas, L.A.; Horwitz, S.M.; Green, C.A.; Wisdom, J.P.; Duan, N.; Hoagwood, K. Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. *Adm. Policy Ment. Health Ment. Health Serv. Res.* **2015**, *42*, 533–544. [CrossRef] [PubMed]
23. Guest, G.; Bunce, A.; Johnson, L. How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. *Field Methods* **2006**, *18*, 59–82. [CrossRef]
24. Kallio, H.; Pietilä, A.-M.; Johnson, M.; Kangasniemi, M. Systematic methodological review: Developing a framework for a qualitative semi-structured interview guide. *J. Adv. Nurs.* **2016**, *72*, 2954–2965. [CrossRef] [PubMed]
25. Bryman, A. *Social Research Methods*, 4th ed.; Oxford University Press: Oxford, UK; New York, NY, USA, 2012.
26. MacQueen, K.M.; McLellan, E.; Kay, K.; Milstein, B. Codebook Development for Team-Based Qualitative Analysis. *CAM J.* **1998**, *10*, 31–36. [CrossRef]
27. Lumivero. Company—About Us—Lumivero. Denver. 2023. Available online: <https://lumivero.com/company> (accessed on 19 May 2023).
28. Leech, N.L.; Onwuegbuzie, A.J. Beyond constant comparison qualitative data analysis: Using NVivo. *Sch. Psychol. Q.* **2011**, *26*, 70–84. [CrossRef]
29. Rana, J.; Dilshad, S.; Ahsan, M.A. Ethical Issues in Research. In *Global Encyclopedia of Public Administration, Public Policy, and Governance*; Farazmand, A., Ed.; Springer International Publishing: Cham, Switzerland, 2021.
30. Leung, L. Validity, reliability, and generalizability in qualitative research. *J. Fam. Med. Prim. Care* **2015**, *4*, 324–327. [CrossRef] [PubMed]
31. Patino, C.M.; Ferreira, J.C. Internal and external validity: Can you apply research study results to your patients? *J. Bras. Pneumol.* **2018**, *44*, 183. [CrossRef] [PubMed]

32. Kasirye, F. *A Conceptual Paper on Ensuring Quality in Qualitative Research*; Department of Communication, International Islamic University Malaysia: Kuala Lumpur, Malaysia, 2021.
33. Ke, T.T.; Sudhir, K. Privacy rights and data security: GDPR and personal data markets. *Manag. Sci.* **2023**, *69*, 4389–4412. [[CrossRef](#)]
34. Neto, F.G.D.O.; Horkoff, J.; Knauss, E.; Kasauli, R.; Liebel, G. Challenges of Aligning Requirements Engineering and System Testing in Large-Scale Agile: A Multiple Case Study. In Proceedings of the IEEE 25th International Requirements Engineering Conference Workshops, Lisbon, Portugal, 4–8 September 2017; pp. 315–322.
35. Oakley, A. HIPAA, HIPPA, or HIPPO: What Really Is the Health Insurance Portability and Accountability Act? *Biotechnol. Law Rep.* **2023**, *42*, 306–318. [[CrossRef](#)]
36. Mäkiö, E.; Mäkiö, J.; Kowal, J. Implementation of Task-Centric Holistic Agile Approach on Teaching Cyber Physical Systems Engineering. In Proceedings of the Americas Conference on Information Systems: A Tradition of Innovation, AMCIS 2017, Boston, MA, USA, 10–12 August 2017.
37. Marnada, P.; Raharjo, T.; Hardian, B.; Prasetyo, A. Agile project management challenge in handling scope and change: A systematic literature review. *Procedia Comput. Sci.* **2022**, *197*, 290–300. [[CrossRef](#)]
38. Beck, K.; Beedle, M.; Van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; et al. Manifesto for Agile Software Development. 2001. Available online: <https://agilemanifesto.org/> (accessed on 20 April 2023).
39. Cohen, D.; Lindvall, M.; Costa, P. An Introduction to Agile Methods. *Adv. Comput.* **2004**, *62*, 1–66. [[CrossRef](#)]
40. Bano, M.; Zowghi, D.; da Rimini, F. Power and Politics of User Involvement in Software Development. In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, Christchurch, New Zealand, 28–29 June 2018; pp. 157–162.
41. Dalpiaz, F.; Brinkkemper, S. Agile Requirements Engineering with User Stories. In Proceedings of the 2018 IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, 20–24 August 2018.
42. Nilsson Tengstrand, S.; Tomaszewski, P.; Borg, M.; Jabangwe, R. Challenges of Adopting SAFe in the Banking Industry—A Study Two Years after Its Introduction. In Proceedings of the 22nd International Conference on Agile Software Development, Virtual Event, 14–18 June 2021.
43. Villamizar, H.; Kalinowski, M.; Viana, M.; Fernandez, D.M. A Systematic Mapping Study on Security in Agile Requirements Engineering. In Proceedings of the 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Prague, Czech Republic, 29–31 August 2018; pp. 454–461.
44. Smith, J.A.; Larkin, M.H.; Flowers, P. *Interpretative Phenomenological Analysis: Theory, Method and Research*; Sage: London, UK, 2009.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.