

# A Taxonomy of Success and Failure Factors in Requirements Engineering for Sustainable Software

Sharifa Alabidi, Mohamed Hagal  
Department of Software Engineering, University of Benghazi

sharifa.alabidi@uob.edu.ly  
mohamed.hagal@benghazi.edu.ly  
DOI: <https://doi.org/10.5281/zenodo.18111885>

**Abstract.** Requirements Engineering (RE) is a crucial phase in the Software Development Life Cycle (SDLC) that significantly influences the success or failure of software projects. Despite its importance, there remains no unified taxonomy that systematically classifies the key factors influencing RE outcomes, particularly from a sustainability perspective. This study addresses this gap by conducting a systematic literature review that encompasses approximately twenty-five peer-reviewed studies published between 2010 and 2024. The identified findings were organized into six main categories: Requirements, People, Process, Communication, Tools & Technology, and Documentation, which represent the *core dimensions* of RE practice. Within each category, success and failure factors were synthesized to illustrate how they influence sustainable software outcomes. The resulting Ishikawa-based taxonomy provides a comprehensive diagnostic framework that integrates technical, human, and organizational aspects to support sustainability-oriented RE practices. The study contributes a structured foundation for assessing RE performance and guiding future sustainability modeling. Future work will focus on empirically validating the proposed taxonomy through case studies and surveys to evaluate its practical applicability in real-world software projects.

**Keywords:** Requirements Engineering. Ishikawa Model. Software Success and Failure Factors.

## 1 Introduction

Requirements Engineering (RE) is a foundational stage in the Software Development Life Cycle (SDLC), defining the extent to which software systems align with user needs, organizational goals, and sustainability objectives. It encompasses processes such as elicitation, analysis, specification, verification, and management, all aiming to translate stakeholder expectations into accurate and actionable requirements. The effectiveness of these processes directly dictates project success or failure, as numerous studies have linked poor requirements practices to project overruns, delays, and stakeholder dissatisfaction. [1].

Despite extensive research, RE continues to face persistent challenges related to vague requirements, communication breakdown, insufficient verification, and weak change management. These issues often lead to software project failures and underscore the need for structured approaches to diagnose and categorize the underlying causes [5]. Furthermore, with the growing importance of software sustainability, focusing on building maintainable, adaptable, and resource-efficient systems [4] there remains a lack of frameworks that explicitly connect RE quality to sustainable outcomes.

To address this gap, this study adopts the Ishikawa (Fishbone) model as a diagnostic and analytical framework. The Ishikawa model facilitates the systematic classification of success and failure factors by connecting them to their root causes within six key dimensions: Requirements, People, Process, Communication, Tools & Technology, and Documentation. This structure allows for a clear visualization of how interdependent elements contribute to project outcomes and provides a basis for identifying areas for improvement in RE practices.

By conducting a structured literature review, this research integrates insights from previous empirical and conceptual studies to construct a unified taxonomy that links RE performance to sustainability. The resulting classification supports both researchers and practitioners in diagnosing RE challenges, improving early-stage decision-making, and embedding sustainability principles throughout the software development process.

This paper contributes to the RE literature and practice by:

- Developing a comprehensive taxonomy of RE success and failure factors based on a structured literature review.
- Adapting and extending the Ishikawa model to classify these factors according to their root causes, enhancing the understanding of interdependencies.
- Bridging the gap between RE and software sustainability, providing a diagnostic framework that supports sustainability-aware RE practices.
- Establishing a basis for future empirical validation through surveys and case studies to test the model's applicability in real-world contexts.

This study aims to answer the following research question: *What are the critical factors influencing sustainability during the Requirements Engineering phase?*

## 2 Literature Review

This section provides a comprehensive overview of previous studies addressing success and failure factors in “Requirements Engineering” (RE), as well as recent efforts to integrate sustainability into RE processes. A total of 25 peer-reviewed studies published between 2010 and 2024 were reviewed, including 17 journal articles and 8 conference papers.

### *Inclusion and Exclusion Criteria.*

The inclusion criteria focused on peer-reviewed studies written in English that examined factors influencing success or failure in RE, or discussed sustainability-oriented

RE frameworks. Studies were excluded if they were purely conceptual without methodological grounding, duplicated earlier findings, or addressed unrelated software engineering topics such as testing or deployment.

### **2.1 RE Failures and Challenges**

Of the total studies, 14 papers (9 journal articles and 5 conference papers) specifically examined RE failure and success factors across different contexts. These works identified recurring issues such as ambiguous requirements, inadequate stakeholder involvement, and poor change management as primary causes of project failure [2], [3]. For instance, Charette [5] and Verner et al. [7] linked prominent project failures to insufficient requirements definitions and unrealistic objectives. Dauda et al. [9] and Dhir et al. [10] further confirmed that scope creep and frequent changes disrupt budgets and timelines. On the success side, studies by Kotowaroo and Sungkur [11] and Ibraigheeth & Fadzli [12] emphasized management clarity, user participation, and accurate planning as critical success enablers. Collectively, these findings affirm that RE remains a dominant determinant of project outcomes, and structured classification frameworks are essential to guide practitioners in identifying and mitigating risks early.

### **2.2 Integrating Sustainability into RE**

Eleven other studies focused on embedding sustainability principles within software development and RE processes. Most notably, the ENSURE framework [13] and the GREENSOFT model [14] introduced dimensions of sustainability such as environmental and economic impact. However, these frameworks did not operationalize sustainability at the RE level. Other works [15], [16] emphasized the lack of standardized sustainability metrics, while studies like Dhir et al. [10] and Nawi et al. [8] linked sustainable outcomes to effective communication, documentation, and stakeholder collaboration. Overall, efforts to integrate sustainability remain fragmented, highlighting the need for a unified approach that anchors sustainability within the diagnostic phases of RE.

### **2.3 Research Gap and Contribution**

Despite these contributions, prior studies lack a comprehensive, practice-oriented taxonomy that systematically classifies RE success and failure factors while linking them to sustainability. This research addresses that gap by using the Ishikawa (Fishbone) model as a prototypical analytical framework. The Ishikawa structure is adapted to categorize and visualize success and failure factors into six root dimensions: Requirements, People, Process, Communication, Tools & Technology, and Documentation. This framework not only systematically organizes the factors but also clarifies the underlying causes that lead to their manifestation. The visual representations in Figures 2 and 3 illustrate these relationships, helping practitioners and researchers diagnose weaknesses and identify enablers of sustainable RE practices. To summarize the key

findings from the reviewed literature, Table 1 provides a concise overview of major studies addressing RE success and failure factors.

**Table 1.** Summary of Prior Studies on Requirements Engineering Success and Failure Factors.

Author(s), Year	Context / Domain	Key Success Factors	Key Failure Factors
Charette (2005)	Large-scale software projects	–	Poor requirements definition, over-optimism
Verner et al. (2008)	Project management	–	Lack of requirements information, weak updates
Dauda et al. (2021)	Empirical analysis	Early validation, clear scope	Unclear requirements, change overload
Dhir et al. (2019)	Agile projects	Continuous elicitation, collaboration	Vague scope, poor stakeholder involvement
Kotowaroo & Sungkur (2022)	IT projects	User involvement, management clarity	Unrealistic expectations
Saputri & Lee (2020)	Sustainability integration	Quality and sustainability attributes	Lack of operationalization
Naumann et al. (2011)	Green software	Environmental focus	Limited RE linkage
Duboc et al. (2020)	Sustainability RE	Stakeholder engagement	Lack of sustainability tools
Chitchyan et al. (2016)	Sustainability practice	–	Organizational misalignment
Venters et al. (2018)	Software sustainability	Documentation, maintainability	–

From the overview in Table 1, it is evident that while numerous studies identify isolated success or failure factors, a unified taxonomy that organizes these factors into a structured, sustainability-oriented framework is missing. This gap motivates the proposed Ishikawa model, illustrated in Figures 2 and 3, which systematically categorizes and visualizes these relationships

### 3 Method

This work aims to provide a systematic and structured categorization of success and failure drivers in the RE phase to support sustainable software development practices. This was achieved by conducting a structured literature survey of relevant works and critically analyzing them to identify fundamental factors. These factors were then struc-

tured using the Ishikawa model into clear categories, providing a basis for future measurement and modeling. Figure 1 illustrates a general representation of the proposed approach.

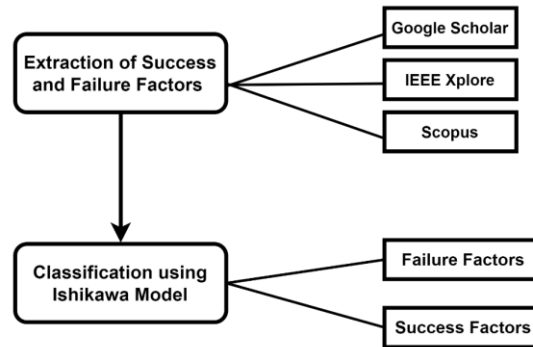


Fig. 1. Conceptual overview of the research approach

### 3.1 Extraction of Success and Failure Factors in RE

As mentioned earlier, Table 1 in the previous section served as the starting point for identifying and classifying success and failure factors in Requirements Engineering (RE). The summarized findings in that table provided the foundation for recognizing recurrent themes and patterns reported across previous studies, thereby guiding the systematic extraction process.

A systematic literature review was then conducted using three reputable academic databases—Google Scholar, IEEE Xplore, and Scopus—to ensure comprehensive coverage. The search employed the following keywords: “*Requirements Engineering Failure*,” “*Success Factors*,” “*Project Failure Causes*,” and “*Sustainable Requirements Engineering*.”

The review focused on studies addressing:

- Software project failures related to the RE phase.
- Key drivers of successful RE implementation.
- RE practice classification models or frameworks.

The retrieved studies were screened and analyzed to extract core factors influencing RE outcomes. These factors were then grouped into thematic categories that represent both technical and human-oriented aspects of RE. Finally, the categories were organized within the Ishikawa (Fishbone) framework, forming the analytical structure for subsequent sections 2 and 3.

#### Systematic Review Protocol.

To enhance methodological transparency, the review covered studies published between 2010 and 2024, focusing exclusively on peer-reviewed journal and conference papers written in English.

Inclusion criteria:

- Studies examining RE success or failure factors.
- Research addressing sustainability or quality aspects within the RE phase.

**Exclusion criteria:**

- Non-peer-reviewed works (e.g., theses, reports).
- Studies unrelated to RE (e.g., testing or deployment).
- Duplicates or incomplete data.

A considerable number of studies were initially retrieved from the selected databases. After applying the inclusion and exclusion criteria, a final set of 25 peer-reviewed publications (including both journal and conference papers) was retained for in-depth analysis. These selected works formed the empirical foundation for identifying the success and failure factors detailed in the following subsections.

**3.2 Failure Factors**

Key root causes that frequently lead to project failure were identified in the literature:

- **Inadequate Needs Identification:** Insufficient gathering of needs results in software systems that poorly meet user requirements [2].
- **Ambiguity and Lack of Clarity in Requirements:** Unclear requirements allow for multiple interpretations among development teams, leading to inconsistencies in implementation [20].
- **Poor Requirements Change Management:** Constant, unmanaged changes disrupt project progress [3].
- **Late Handling of Non-Functional Requirements (NFRs):** Late consideration of NFRs can cause significant performance and security issues [3].
- **Weak Requirements-Related Risk Management:** Ignoring proactive risk analysis can lead to unforeseen obstacles that threaten continuity [5].
- **Inadequate Time and Resource Planning:** Results in schedule overruns and excessive costs [21].
- **Poor Stakeholder Communication:** Insufficient interaction with clients or end-users leads to systems that do not align with actual needs [22].
- **Lack of Documentation:** Poorly documented requirements hinder traceability and future maintenance [10].
- **Reliance on Unproven Technologies:** Use of immature tools increases technical risk and reduces stability [6].
- **Lack of Experience Among Analysts:** Inexperienced analysts may misunderstand or overlook critical needs [20].
- **Failure to Clearly Prioritize Requirements:** Leads to focus on less important needs [11].
- **Conflicting Stakeholder Perspectives:** Result in contradictory requirements and added complexity [9].

- Lack of Requirements Traceability: Makes it difficult to assess future change impacts [3].
- Inability of Users to Articulate True Needs: Leads to inaccurate requirements gathering [20].
- Cultural and Language Barriers: Impede communication and understanding in cross-cultural teams [20], [19].

### 3.3 Success Factors

The critical factors for successful RE implementation include:

- Active User Participation: Ensures realistic and accurate requirements reflecting user needs [9].
- Comprehensive Requirements Elicitation: Reduces costly rework later [2].
- Clear and Unambiguous Requirements: Prevents misinterpretation and improves implementation accuracy [3].
- Early Integration of NFRs: Ensures system performance and security from early stages [3].
- Effective Change Management: Maintains project stability and control [3].
- Requirements Risk Management: Minimizes unexpected project disruptions [7].
- Accurate Scheduling and Estimation: Helps in achieving time and cost objectives [23].
- Qualified Requirements Analysts: Improve understanding and analysis quality [20].
- Early Verification: Catches issues early, reducing rework [20].
- Use of Specialized RE Tools: Enhances traceability and impact analysis [20].
- Requirements Prioritization: Focuses development on critical features [11].
- Early Conflict Resolution: Establishes consensus early and avoids future disagreements [9].
- Overcoming Cultural and Language Barriers: Promotes communication within diverse teams [20].

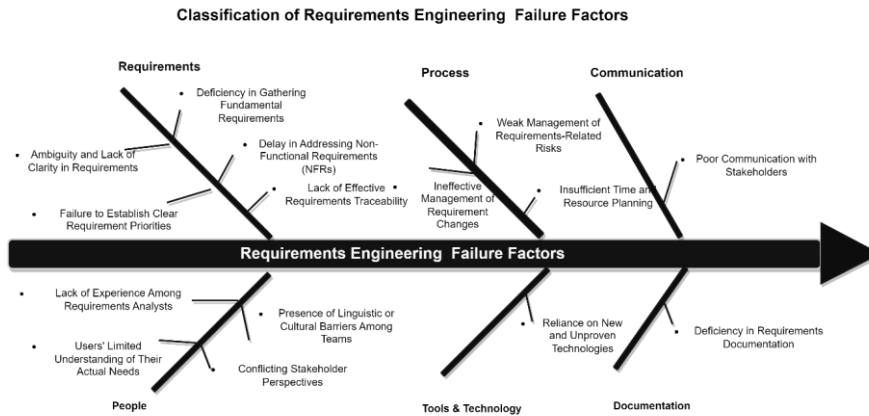
### 3.4 Classification of Factors using the Ishikawa Model

The Ishikawa (Fishbone) Diagram, developed by Kaoru Ishikawa in the 1940s, is a problem-solving analysis tool used to systematically identify root causes. Based on the extracted findings, and given the observed non-overlap between the identified factors, they were classified into six fundamental categories, each representing a root dimension contributing to the outcomes of RE activities. These categories are:

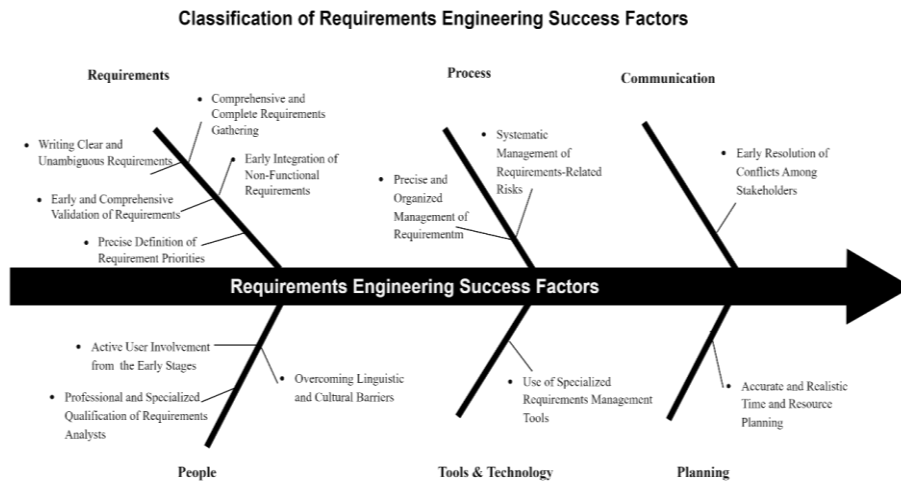
- |                       |                  |
|-----------------------|------------------|
| 1. Requirements       | 2. People        |
| 3. Process            | 4. Communication |
| 5. Tools & Technology | 6. Documentation |

Figures 3.2 and 3.3 illustrate the classification of failure and success factors, respectively, showing how each factor is placed under its corresponding root cause category. These visual diagrams offer a clear and comprehensive view of how human, technical,

and procedural elements within the RE process interact to impact sustainability and project outcomes.



**Fig. 2.** Failure factors classification.



**Fig. 3.** Success factors classification.

To further enhance clarity, examples are provided below to illustrate how individual factors are mapped to their respective categories:

- **Requirements:** The factor "Ambiguous Requirements" is assigned to this category because it directly reflects issues in defining, analyzing, and verifying user needs.
- **People:** "Lack of Analyst Experience" is categorized here because it originates from individual competency gaps that affect the accuracy and quality of RE.
- **Process:** "Poor Change Management" belongs to this category as it reflects procedural weaknesses in handling evolving requirements.
- **Communication:** "Cultural and Language Barriers" is placed in this category because it hinders effective collaboration and shared understanding among stakeholders.

- **Tools & Technology:** "Reliance on Unproven Technologies" fits this category as it relates to immature or unsuitable technological choices impacting RE stability.
- **Documentation:** "Lack of Documentation" is classified here due to its impact on traceability, maintainability, and long-term sustainability of project requirements.

This classification framework provides a structured and diagnostic perspective on RE performance, allowing practitioners and researchers to identify root causes of success or failure within each dimension. Furthermore, the Ishikawa model offers a visual mechanism for diagnosing weaknesses and designing targeted improvements, particularly in sustainability-oriented RE contexts.

## 4 Discussion

The objective of this research was to synthesize a systematic taxonomy of the success and failure factors that occur in the RE phase to lay the foundation for supporting sustainable software development. This section interprets the implications across the six Ishikawa categories and discusses how this classification contributes to both theoretical understanding and practical application in Requirements Engineering.

### 4.1 Category Implications and Impact on Sustainability

Each category within the Ishikawa model highlights a pivotal aspect that shapes software sustainability during RE activities:

- **Requirements:** The accuracy, completeness, and prioritization of requirements directly impact system resilience, adaptability, and maintainability. Vague or volatile requirements can create technical debt and diminish long-term sustainability.
- **People:** Analyst expertise and stakeholder involvement are essential for accurate elicitation and analysis. A lack of alignment can lead to unmet needs and conflicting expectations, ultimately resulting in fragile systems.
- **Process:** Robust change management and risk management processes allow for adaptation while maintaining stability. Weak processes can lead to scope creep, cost overruns, and delays, threatening project viability.
- **Communication:** Open and timely communication helps avoid misunderstandings. Cultural or language barriers can produce incomplete or misinterpreted requirements.
- **Tools & Technology:** The use of mature, appropriate RE tools enhances traceability and consistency, while immature tools increase integration risk.
- **Documentation:** Adequate documentation supports traceability, knowledge sharing, and change management, while poor documentation leads to maintainability issues and knowledge loss.

## 4.2 Sustainability Enablers and Comparison to Previous Models

The main research question; what are the key factors influencing software sustainability during the Requirements Engineering phase? is answered by the systematic classification of both success and failure factors. These are structured across the Ishikawa diagram to identify root causes impacting sustainability. Key failure factors include ambiguous requirements [20], poor change management [3], late handling of NFRs [3], and insufficient documentation [10]. Conversely, early user participation [9], clear prioritization [11], early verification [20], and comprehensive documentation are crucial success drivers. These findings align with previous studies that emphasize stakeholder engagement and documentation as cornerstones of sustainable software [17], [25]. Compared to earlier frameworks like GREENSOFT [14] and ENSURE [13], which primarily focused on high-level environmental aspects of sustainability, this research integrates sustainability thinking directly into the RE processes. By integrating both technical and human-centric factors, the framework serves not only as a diagnostic tool but also provides a roadmap for sustainable requirements practices in real-world contexts.

The taxonomy can serve as a checklist for software project managers to assess RE sustainability readiness and identify improvement areas early in the SDLC.

## 5 Conclusion and Future Work

### 5.1 Conclusion

This paper presents a systematic taxonomy of success and failure factors in the Requirements Engineering phase to promote sustainable software development. The study identified six core categories: Requirements, People, Process, Communication, Tools & Technology, and Documentation, which form the basis of the Ishikawa model framework. These categories capture both enablers and inhibitors of RE success, unifying fragmented research into an actionable model for evaluation and improvement. The findings emphasize that software sustainability must be established starting from the requirements phase. The classification framework offers analytical and practical value for future sustainability modeling, process evaluation, and enhancing software quality.

This study is limited to literature-based analysis without empirical validation; future work will test the taxonomy in real-world RE projects.

### 5.2 Future Work

This paper lays the foundation for a follow-up study aimed at further enhancing sustainability in Requirements Engineering (RE). Based on the identified success and failure factors, an operational framework will be applied to a real-world case study. The factors will be operationalized into a detailed checklist to evaluate the effectiveness of RE practices in promoting sustainable software development. The observations and insights derived from the application and assessment of this checklist will guide iterative refinements to the framework, ensuring its practical relevance and adaptability across

various project contexts. Future work will also explore broader empirical validation through case studies and surveys, refining the framework to support structured, sustainability-aware RE practices and provide actionable guidance for practitioners.

## References

1. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
2. Charette, R. N. (2005). Why software fails. *IEEE Spectrum*, 42(9), 42–49. <https://doi.org/10.1109/MSPEC.2005.1502528>
3. Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., & Venters, C. C. (2015). *The Karlskrona Manifesto for Sustainability Design*. <https://doi.org/10.48550/arXiv.1505.06742>
4. Hall, T., Beecham, S., & Rainer, A. (2002). Requirements problems in twelve software companies: An empirical analysis. *IEE Proceedings - Software*, 149(5), 153–160. <https://doi.org/10.1049/ip-sen:20020733>
5. Firesmith, D. (2007). Common requirements problems, their negative consequences, and the industry best practices to help solve them. *Journal of Object Technology*, 6(1), 17–33. <https://doi.org/10.5381/jot.2007.6.1.a17>
6. Charette, R. N. (2005). Why software fails. *IEEE Spectrum*, 42(9), 42–49. <https://doi.org/10.1109/MSPEC.2005.1502528>
7. Verner, J. M., Overmyer, S. P., & McCain, K. W. (2008). In the 25 years since *The Mythical Man-Month*, what have we learned about software project management? *Information and Software Technology*, 50(3), 135–142. <https://doi.org/10.1016/j.infsof.2007.06.002>
8. Dauda, A., Ismail, A., Misra, S., Olajide, M., & Fernandez-Sanz, L. (2021). A systematic literature review on factors causing software project failures. *International Journal of Advanced Computer Science and Applications*, 12(5), 524–536. <https://doi.org/10.14569/IJACSA.2021.0120562>
9. Dhir, S., Singh, M., & Suri, P. K. (2019). Failure factors and success factors in agile software development methodology. *International Journal of Innovative Technology and Exploring Engineering*, 8(11), 2189–2195. <https://doi.org/10.35940/ijitee.K2092.0981119>
10. Kotowaroo, M. Y., & Sungkur, R. K. (2022). Success and Failure Factors Affecting Software Development Projects from IT Professionals' Perspective. *University of Mauritius Research Journal*. <https://www.researchgate.net/publication/364038253>
11. Ibraigheeth, M., & Fadzli, S. A. (2019). Core factors for software projects success. *JOIV: International Journal on Informatics Visualization*, 3(1), 69–74. <https://doi.org/10.30630/joiv.3.1.194>
12. Saputri, T. R. D., & Lee, S.-W. (2020). ENSURE: An enhanced sustainability requirements engineering framework. *Journal of Software: Evolution and Process*, 32(12), e2281. <https://doi.org/10.1002/smr.2281>
13. Naumann, S., Dick, M., Kern, E., & Johann, T. (2011). The GREENSOFT model: A conceptual reference model for green and sustainable software. In *Proceedings of the 25th International Conference on Informatics for Environmental Protection* (pp. 91–98). [https://doi.org/10.1007/978-3-642-19536-5\\_9](https://doi.org/10.1007/978-3-642-19536-5_9)
14. Duboc, L., Penzenstadler, B., Betz, S., Chitchyan, R., Porras, J., Seyff, N., & Venters, C. C. (2020). Requirements engineering for sustainability. *Empirical Software Engineering*, 25(3), 2453–2498. <https://doi.org/10.1007/s10664-020-09807-6>

15. Chitchyan, R., Becker, C., Betz, S., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., & Venters, C. C. (2016). Sustainability design in requirements engineering: State of practice. In *Proceedings of the 38th International Conference on Software Engineering Companion* (pp. 533–542). <https://doi.org/10.1145/2889160.2889217>
16. Nawi, C., Rahman, A., & Ibrahim, R. (2011). Government ICT project failure factors: A revisit. *International Journal of Computer Science Issues*, 8(5), 466–474.
17. Mandal, S., & Pal, S. (2015). A framework for software project success. *International Journal of Computer Applications*, 116(11), 1–5. <https://doi.org/10.5120/20415-2574>
18. Azbaki, O., Hagal, M. A., & Ramadhan, A. R. (2021). Ishikawa model for improving the use of brainstorming technique to elicit user requirements. In *\*2021 International Conference on Microelectronics (MI-STA)\** (pp. 254–258). <https://doi.org/10.1109/MI-STA52233.2021.9464427>
19. Venters, C. C., Jay, C., Lau, L., Griffiths, M. K., Holmes, V., Ward, R., ... & Dibsedale, C. E. (2018). Software sustainability: The modern tower of Babel. *IEEE Software*, 35(2), 10–14. <https://doi.org/10.1109/MS.2017.4541034>
20. Robertson, S., & Williams, J. (2006). Understanding project failure. *British Computer Society*. <https://www.bcs.org/articles-opinion-and-research/understanding-project-failure/>
21. Sharma, T. B., Singh, S., & Goyal, D. P. (2020). Requirements engineering challenges in agile software development. *International Journal of Information Management*, 52, 102045. <https://doi.org/10.1016/j.ijinfomgt.2020.102045>
22. Damian, D. E., & Zowghi, D. (2003). An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences* (p. 10). <https://doi.org/10.1109/HICSS.2003.1173659>
23. Ashraf, M. A., Alam, H., & Awan, S. M. (2021). Origins and consequences of misunderstood/slipped requirements on large-scale software projects. *VFAST Transactions on Software Engineering*, 9(4), 62–67. <https://doi.org/10.21015/vtse.v9i4.744>
24. Paleri, M., Khan, K., & Chelliah, P. (2016). A study on the factors affecting IT project failures. *International Journal of Computer Applications*, 975(8887), 12–17. <https://doi.org/10.5120/ijca2016911932>
25. Horkoff, J. (2016). Requirements for sustainability: State of the art. In *IFIP Advances in Information and Communication Technology* (Vol. 474, pp. 533–542). [https://doi.org/10.1007/978-3-319-44805-3\\_47](https://doi.org/10.1007/978-3-319-44805-3_47)

# تصنيف لعوامل النجاح والفشل في هندسة المتطلبات من أجل برمجيات مستدامة

شريفة العبيدي، محمد حجل

قسم هندسة البرمجيات، جامعة بنغازي  
sharifa.alabidi@uob.edu.ly  
mohamed.hagal@benghazi.edu.ly  
DOI: <https://doi.org/10.5281/zenodo.18111885>

**الملخص.** تُعد هندسة المتطلبات مرحلة أساسية في دورة حياة تطوير البرمجيات، إذ تؤثر بشكل مباشر في نجاح أو فشل المشاريع البرمجية. وعلى الرغم من أهميتها، لا يزال هناك غياب لتصنيف موحد يُحدد بصورة منهجية العوامل الجوهرية التي تؤثر في نتائج هندسة المتطلبات، ولا سيما من منظور الاستدامة. تهدف هذه الدراسة إلى سد هذه الفجوة من خلال إجراء مراجعة منهجية للأدبيات شملت نحو خمس وعشرين دراسة محكمة نُشرت بين عامي 2010 و2024. وقد تم تصنيف النتائج ضمن ست فئات رئيسية تمثل الأبعاد المحورية لممارسات هندسة المتطلبات، وهي: المتطلبات، الأفراد، العمليات، الاتصال والتواصل، الأدوات والتقنيات، والتوثيق. تم في إطار كل فئة تحليل عوامل النجاح والفشل لتوضيح كيفية تأثيرها على استدامة البرمجيات. ويقدم التصنيف المقترح، المستند إلى نموذج إيشكاوا، إطارًا تشخيصيًا متكاملًا يجمع بين الجوانب التقنية والبشرية والتنظيمية لدعم ممارسات هندسة المتطلبات الموجهة نحو الاستدامة. تُسهم هذه الدراسة في إرساء أساس علمي منظم لتقييم أداء هندسة المتطلبات وتوجيه جهود نمذجة الاستدامة المستقبلية. كما ستركز الأبحاث المستقبلية على التحقق التجريبي من فعالية التصنيف المقترح عبر دراسات حالة واستبيانات لقياس مدى قابليته للتطبيق في المشاريع البرمجية الواقعية.

**الكلمات المفتاحية:** هندسة المتطلبات، نموذج إيشكاوا، المتطلبات عوامل نجاح وفشل البرمجيات.