مـجلـة جـامـعـة صـحـار للأعـمـال الـمستدامـة
Sohar University Journal of Sustainable Business

Sohar University Journal of Sustainable Business
e-ISSN: 3007-5572 | Volume 1, Issue 1, 27-41
DOI: 10.69983/SUJSB/0101/03, January 2025

FACTORS IN DETERMINING INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES: A SYSTEMATIC LITERATURE REVIEW

Yahya Ateik Al-Sagheer Saeed[1], Zahidah Binti Zulkifli[1*], Sumaia Binti Shikh Nasir[1] and Wani Bilal Ahmed[1]

[1]*Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia*

ABSTRACT

Choosing the appropriate Software Development Life Cycle (SDLC) methodology is a complex task for many firms and software engineers. The challenge arises not only from a lack of understanding of the criteria needed to select a methodology that adds value to the firm but also from a limited recognition that a methodology can be applicable across multiple life cycle models. This study aims to identify the critical factors influencing the choice of Information Systems (IS) methodologies for development projects. To address this, we conducted a systematic literature review (SLR), adapting a mapping study process from existing guidelines to categorize and structure the research evidence published in the field of IS methodologies. Our synthesis of 36 relevant papers identified four key factors influencing the selection of IS development methodologies: (1) Security, (2) Quality, (3) Communication, and (4) Test Cases. These findings provide a valuable framework for selecting appropriate IS development methodologies, focusing on factors that can serve as benchmarks for making informed choices. Furthermore, the ongoing digital transformation, accelerated by the 2020 pandemic, has underscored the urgency of integrating robust IS methodologies, particularly in the context of higher education institutions, which have swiftly shifted towards online learning and digital student services.

Keywords: Information System Development, Systematic literature review

Paper Type: Research paper

INTRODUCTION

The reliance on information systems has surged over the past decades, driven by advancements in computing and communication technologies. (Liu, 2021) This sector's growing significance has prompted continuous enhancements in system development methods and techniques.

(Nikolaieva, n.d.) are employed to streamline system development in alignment with project requirements. (Alsaqqa, 2020). However, the process of choosing the most suitable methodology can be complex and requires careful consideration of multiple factors to make an informed decision. This study aims to identify the key factors influencing the selection of information systems development methodologies for project execution. (Barbara Kitchenham, Stuart M. Charters, 2007).

BACKGROUND OF THE STUDY

Information system development methodologies vary significantly between large-scale projects that require precise tracking of each development stage and small to medium-sized projects that focus primarily on critical development phases. The rise of agile software development has brought attention to lightweight methodologies that are tailored to specific types of projects and organizations involved in information system development. Despite this, numerous factors can contribute to the failure of information systems, whether these factors act in combination or individually (Gunawardhana & Perera, 2015). A major contributing factor to these failures is a lack of understanding, which can create significant problems and ongoing obstacles in information system development projects (Hart & Warne, 2007). Uwadia et al. (2006) reported that up to 25% of large systems development projects were canceled, 60% encountered cost overruns, 75% faced quality issues, and less than 1% of projects were completed on schedule.

The challenge lies in selecting an information system development methodology that aligns with the specific needs of the development firm and the nature of its projects. Firms often lack the necessary knowledge and experience to effectively evaluate and choose from the various available methodologies. Frequently, decisions are based on the recommendations of consulting firms that may promote their proprietary methodologies, leading to the selection of methodologies that are only partially suitable. This approach is a critical factor in the low adoption rates of information system development methodologies among development firms. For instance, it has been found that 60% of firms do not utilize any specific development methodology, and only 6% adhere strictly to a chosen methodology. This highlights a significant gap in the effective application of development methodologies, which can undermine project success and organizational efficiency.

METHODS

A systematic literature review (SLR) is a method of inquiry that meticulously evaluates all existing research evidence to provide reliable answers to specific research questions or issues (Salleh, 2011). The primary aim of an SLR is to produce a scientific synthesis of the evidence within a given area (Salleh, 2011). Unlike traditional narrative reviews, an SLR delves deeper into the literature to retrieve and analyze fact-based evidence. Since the 1990s, this method has gained popularity as a research methodology, particularly in medical research, where several well-established standards have been developed to support its application (Babar, 2009). The present study's SLR is structured around three main phases: planning, conducting, and reporting the

review. Our approach adheres to the guidelines well-established in the existing literature (Barbara et al., 2007; Vale, 2017).

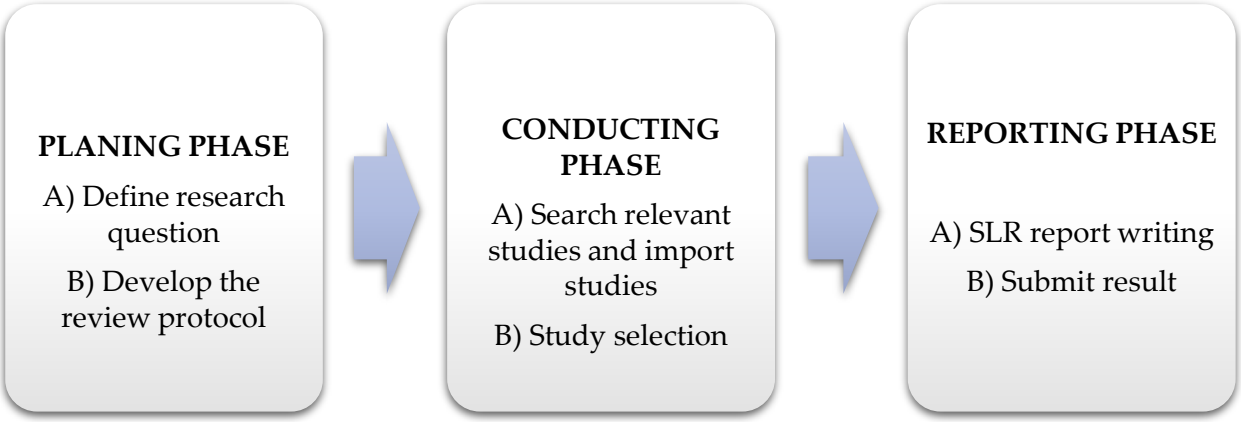

Figure 1: Mapping study process

Planning phase

The planning review is the initial phase of a systematic literature review (SLR), focusing on defining the mapping study protocol, which outlines the systematic activities for gathering and collecting information (Vale, 2017). The starting point of the SLR is to identify and formulate the research question, which is then documented in the review protocol.

Research Questions

The primary objective of formulating the research question is to guide the identification of relevant literature, determine the data to be extracted from selected studies, and establish the parameters for evaluation. This systematic literature review (SLR) specifically aims to identify the key factors for selecting appropriate Information System development methodologies. Consequently, this study focuses on systematically analyzing previous research contributions on Information Systems Methodologies published between 2017 and 2021.

RQ: *What are the factors when choosing Information Systems Methodologies for Information system development projects?*

Develop the Review protocol

After formulating the research question for the systematic literature review (SLR), the PICOC criteria were applied to further refine it. The PICOC framework—Population, Intervention, Comparison, Outcome, and Context—was used to break down the research question into distinct elements, which were then separated by commas. This approach enables the terms to be saved separately as keywords, facilitating the design of an effective search string for the literature review.

Table 1. Summary of PICOC

Population	Information Systems Development
Intervention	Methodologies
Comparison	N/A
Outcome	Factors
Context	Information system development projects

Table 2. Keywords and Synonyms

Keyword	**Synonyms**	**Related to**
Factors	Characteristics Considerations Criteria Specifications	Outcome
Information Systems Development	Application Development Software Development Systems Development	Population
Methodologies	Method Methodology Techniques	Intervention
N/A		Outcome

To derive effective search terms for a study, one should follow a systematic process. First, it is essential to formulate the research question using the PICOC framework, which includes defining the Population, Intervention, Comparison, Outcome, and Context related to the study. This framework helps in structuring a clear and focused research question. Next, identify the key terms from this research question that will serve as the foundation for designing the search string. Following this, list synonyms and alternative terms for each keyword to ensure comprehensive coverage of the topic. This step is crucial for capturing all relevant literature, even if different terminology is used. Once the synonyms and alternative terms are identified, they should be combined using the Boolean operator "OR." This operator connects all possible variations of each keyword, such as “Information Systems” OR “IS,” expanding the search to include all relevant documents. Finally, the main keywords and their synonyms should be linked using the Boolean operator "AND." This ensures that the search string incorporates all major aspects of the research question, for example, “Information Systems” AND “Development Methodologies,” thereby focusing the search results on studies that address the complete scope of the research inquiry.

Conducting Phase

Searching relevant studies: According to Kitchenham and Charters (2007) a systematic literature review (SLR) requires an exhaustive and in-depth process to identify all relevant previous studies comprehensively. This meticulous approach ensures that the research questions (RQs) are thoroughly addressed. In this section, we describe the review strategies designed to retrieve relevant studies systematically. For the automatic search process, we used a carefully constructed search string with appropriate search terms.

("Information Systems Development" OR "Application Development" OR "Software Development" OR "Systems Development") AND ("Methodologies" OR "Method" OR "Methodology" OR "Techniques") AND ("N/A") AND ("Factors" OR "Characteristics" OR "Considerations" OR "Criteria" OR "Specifications")

This study utilized three prominent digital libraries—Scopus, IEEE Digital Library, and ProQuest—to identify relevant research in the field of criteria for selecting information systems development methodologies. Specific combinations of keywords were generated and searched across these databases. These platforms were selected for their established reputations as multidisciplinary study repositories, encompassing a wide range of peer-reviewed journals and maintaining up-to-date content. Table 3 below presents the imported studies.

Table 3: Imported studies

Database	Number of Imported studies
Scopus	128
IEEE Digital Library	58
ProQuest	77

Selection Strategy : To address the research question of this study and identify relevant primary studies on criteria for selecting information system development methodologies, a systematic selection process was employed. This process includes both inclusion and exclusion criteria tailored to the research question.

The researcher established specific selection criteria to ensure the relevance and quality of the papers included in the study. The inclusion criteria mandate that papers must be within the fields of computer science and software design. Additionally, the papers must originate from conferences, journals, dissertations, or theses to ensure academic rigor and credibility. Only open-access papers are considered to facilitate unrestricted availability. Moreover, the selected papers must be published between 2017 and 2021 to ensure the research is current and reflects recent advancements in the field.

Conversely, the exclusion criteria specify that any papers not written in English are to be omitted from the study. This language restriction ensures that the researcher can accurately interpret and analyze the content without language barriers.

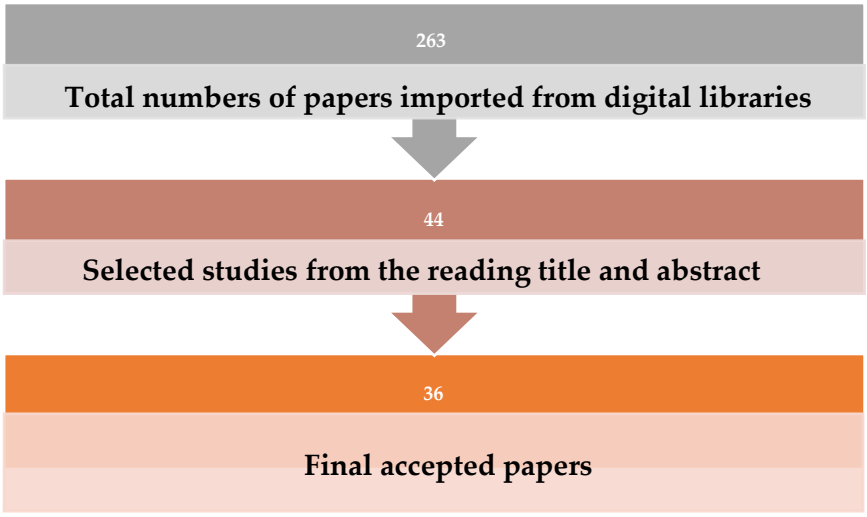

*Figure 2***:** paper selection process

Figure 2 illustrates the paper selection process employed in this study. Content-level filtration involved applying inclusion and exclusion criteria to include papers from the fields of computer science and software design, sourced from conferences, journals, dissertations, and theses. Additionally, papers with open access published between 2017 and 2021 were considered, while non-English papers were excluded. Initially, 263 papers were imported from digital libraries. After reviewing titles and abstracts, 44 papers met the criteria. Upon further scrutiny, 8 duplicate papers were identified, resulting in a final selection of 36 papers.

Reporting phase

SLR results: This section synthesizes the evidence by analyzing the extracted data from the literature search results. The systematic literature review (SLR) identified key factors that significantly influence Information System Development Methodologies. Table 5 presents a comprehensive list of these influential factors.

RQ: *What are the factors when choosing Information Systems Methodologies for Information system development projects?*

Table 4: List of Factors

No.	Factors	Studies	Total studies
1	Security	S2, I8, P2, P4, P6, P8, P12	7
2	Quality	I2, I3, I10, P9	4
3	Communication	S4, I13, I15	3
4	Test cases	S1, S6, S7	3

The study revealed consensus among authors that security has the most significant impact on the selection of system methodologies. Specifically, "Security" was identified as the most frequently studied factor influencing methodology selection (mentioned in 7 papers), followed by "Quality"

(discussed in 4 papers), "Communication" (mentioned in 3 papers), and "Test Cases" (also covered in 3 papers). Table 6 provides definitions for each of these factors.

Table 5: Factors Definitions

No.	Factors	Definition
F1	Security	To identify, mitigate, and avoid security threats to software and data assets
F2	Quality	Describes the customer satisfaction as well as development organization. While determining the customer satisfaction, a triangle that consists of the time, budget, and customer expectation requirements is considered. Thus, from an organizational viewpoint, another triangle of workflow, goodwill, and business is considered.
F3	Communication	The act of transmitting information between individuals
F4	Test cases	Test scenario measuring functionality across a set of actions or conditions to verify the expected result.

List of selected papers: As part of the study selection process, a total of 263 papers were initially imported from digital libraries. Of these, 255 papers proceeded to the data extraction phase. During this phase, the inclusion and exclusion criteria were applied to refine the selection. Specifically, papers not written in English and those unrelated to System Development Methodologies (SDM) were excluded. This rigorous filtering process resulted in 36 studies being accepted for final analysis. The complete list of all included papers is provided as an appendix.

DISCUSSION

The purpose of this systematic literature review (SLR) was to identify SDLC being practiced despite the fact that there are lot of software development models available, from the SLR we found out the most essential ones, such as Waterfall and Agile methodologies. Until date, the waterfall approach has been the most often utilized in software development. According to many in the software industry, the waterfall model has been the dominant in past and is being still practiced. However, the Agile techniques are developing and making firms switch from the old waterfall model to agile methods. Projects that need a high level of usability tend to employ conventional approaches in the early stages since the client specifies these needs. As a result, all new initiatives are based on the needs of usability. Even said, conventional software development techniques are not best suited for today's fast changing environment, which excludes the possibility of using agile approaches. However, there are certain issues with integrating Agile methodologies and usability at the same time.

The result of the SLR shows that most of the researchers focused Security, Quality Assurance, developing different frameworks and Study shows most of them likely to use Agile Methodology. In fact, Agile methodologies and usability have more in common than you may think. They both

follow the cyclic development approach and user centric design, and they both place a strong emphasis on team co-ordination and communication. Despite the fact that they have a shared goal, there are numerous areas of disagreement and disagreement that might impede the progress of the project. Agile techniques thrive on achieving continuous engagement between developers and customers in order to generate excellent software, while Usability methods thrive on achieving continuous interaction between developers and users in order to develop good software. Consequently, an agile technique might overlook numerous chances for end-users if customers lack a clear grasp of the end-users. Following the introduction of agile methodologies, the process becomes straightforward and quick to implement by identifying and correcting design problems immediately following implementation. Agility is a way of working where changes are made often. There are a variety of questions and design problems from different departments that the developers must address before the process can go forwards successfully. The developers and usability testers collaborate to identify the appropriate level of test coverage for the application. Agile techniques may be successfully used for projects that are currently being done using conventional methods.

"Software Requirements Specification" is used in the classic waterfall paradigm (SRS). As soon as the SRS is implemented by the designers, they may begin working on new projects. As a result, client feedback and interactions are completely ignored, resulting in substantial losses if the project fails. Their job is to design based on user needs. Waterfall is a "over the fence" strategy. A system where the user expresses their needs and receives the product in a timely manner. Customers have a hard time defining software needs if they don't know how things have changed and progressed. The client is at the core of the agile development approach, while with agile methods the customer may add additional requirements throughout the development process. Throughout the whole development process, the clients are accessible.

CONCLUSION

In this paper, we perform a systematic literature review on the different Software Development Life Cycles, as a result of this SLR, new ideas about software development and usability might be generated. It is based on Barbara Kitchenham's systematic literature review technique. A successful project can be seen from the findings and discussion of this research, which shows that agile development methodologies are being used with usability. The product's price may even be reduced via the use of cutting-edge usability approaches like "Discount Usability."

Because of a set of standards that assist in incorporating usability into agile techniques, agile methods have been shown to favourably contribute to the usability of a product. In this paper, it is advocated for a combined strategy in which agile methodologies and usability engineering may be applied with the greatest amount of correlation possible.

REFERENCES

Abrar, M. F., Khan, M. S., Ali, S., Ali, U., Majeed, M. F., Ali, A., Amin, B., & Rasheed, N. (2019). Motivators for large-scale agile adoption from management perspective: A systematic literature review. IEEE Access.

Aizaz, F., Khan, S. U. R., Khan, J. A., Inayat-Ur-Rehman, & Akhunzada, A. (2021). An empirical investigation of factors causing scope creep in agile global software development context: A conceptual model for project managers. IEEE Access.

Akbar, M. A., Sang, J., Khan, A. A., Amin, F.-E., Nasrullah, Hussain, S., Sohail, M. K., Xiang, H., & Cai, B. (2018). Statistical analysis of the effects of heavyweight and lightweight methodologies on the six-pointed star model. IEEE Access.

Akbar, M. A., Sang, J., Khan, A. A., Fazal-E-Amin, Nasrullah, Shafiq, M., Hussain, S., Hu, H., Elahi, M., & Xiang, H. (2018). Improving the quality of software development process by introducing a new methodology–AZ-Model. IEEE Access.

Alzoubi, Y. I., & Gill, A. Q. (2020). An empirical investigation of geographically distributed agile development: The agile enterprise architecture is a communication enabler. IEEE Access.

Arrey, D. A. (2019). Exploring the integration of security into software development life cycle (SDLC) methodology (Order No. 13428588). Available from ProQuest One Academic; SciTech Premium Collection. (2191192923).

Aslam, W., & Ijaz, F. (2018). A quantitative framework for task allocation in distributed agile software development. IEEE Access.

Babar, M. Z. (2009). Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In 3rd International Symposium on Empirical Software Engineering and Measurement (pp. 346-355).

Bajaj, A., & Sangwan, O. P. (2019). A systematic literature review of test case prioritization using genetic algorithms. IEEE Access.

Kitchenham, B., & Charters, S. M. (2007). Guidelines for performing systematic literature reviews in software engineering. Keele University and Durham University Joint Report.

Campbell, J. (2019). Exploring The Strategies Software Developers Need to Maintain Security Requirements During Agile Development Processes. Doctoral dissertation. Colorado Technical University.

Cvetkovic, N., Morača, S., Jovanović, M., Medojević, M., & Lalić, B. (2017). Enhancing the agility and performances of a project with lean manufacturing practices. In Annals of DAAAM and Proceedings of the International DAAAM Symposium (pp. 661-670). Danube Adria Association for Automation and Manufacturing, DAAAM.

Dar, H., Lali, M. I., Ashraf, H., Ramzan, M., Amjad, T., & Shahzad, B. (2018). A systematic study on software requirements elicitation techniques and its challenges in mobile application development. IEEE Access.

De Silva, I. J. (2019). A Framework to Evaluate Candidate Agile Software Development Processes. Doctoral dissertation. University of Minnesota.

Gunawardhana, D. N. T., & Perera, C. (2015). Classification of failure factors in information systems. International Journal for Innovation Education and Research, 3(11).

Fasano, J. W. (2020). Agile Software Development Practices for Customer Service and Support Doctoral dissertation. Capella University.

Fernández-Diego, M., Méndez, E. R., González-Ladrón-De-Guevara, F., Abrahão, S., & Insfran, E. (2020). An update on effort estimation in agile software development: A systematic literature review. IEEE Access.

Horton, S. (2020). Are software security issues a result of flaws in software development methodologies?(Order No. 27962083). Available from ProQuest One Academic. (2405158801).

Johnston, R. A. (2018). A multivariate Bayesian approach to modeling vulnerability discovery in the software security lifecycle. The George Washington University.

Jose, C. R. (2020). Exploring security process improvements for integrating security tools within a software application development methodology. [Unpublished dissertation or thesis].

Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2021). Systematic mapping study on security approaches in secure software engineering. IEEE Access.

Khatibsyarbini, M., Isa, M. A., Jawawi, D. N. A., & Hamed, H. N. A., & Mohamed Suffian, M. D. (2019). Test case prioritization using firefly algorithm for software testing. IEEE Access.

Kipreos, M. (2019). The impact of leadership style on the adoption of agile software development: A correlational study. [Unpublished dissertation or thesis].

Kochhar, P. S., Xia, X., & Lo, D. (2019). Practitioners' views on good software testing practices. In Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019 (pp. 61-70). Institute of Electrical and Electronics Engineers Inc.

Liu, S. (2021, October 22). Information technology (IT) spending on enterprise software worldwide. Retrieved from Statista: https://www.statista.com/statistics/203428/total-enterprise-software-revenue-forecast/

Lunesu, M. I., Tonelli, R., Marchesi, L., & Marchesi, M. (2021). Assessing the risk of software development in agile methodologies using simulation. IEEE Access.

Pawaskar, S. (2014). Maximizing the Performance of Agile Teams for IoT Development. Retrieved from Bepress [works.bepress.com]

Nikolaieva, A. (n.d.). 8 best software development methodologies. Retrieved from Uptech: https://www.uptech.team/blog/software-development-methodologies

Patel, S. (2021). Dynamic modeling of the effectiveness of software development methods on DoD programs (Order No. 28716962). Available from ProQuest Central; ProQuest One Academic; Publicly Available Content Database. (2572595494).

Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., & Leppänen, V. (2021). Security in agile software development: A practitioner survey. Information and Software Technology.

Salleh, N. M. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. IEEE Transactions on Software Engineering, 37(4), 509–525.

Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies (iJIM), 14(25).

Sami, M. (2012, March 21). Choosing the right software development life cycle model. Retrieved from Melsatar: https://melsatar.blog/2012/03/21/choosing-the-right-software-development-life-cycle-model/

Santolucito, M. (2020). A Modular Synthesis Framework for Software Deployment, Design, and Implementation. Doctoral dissertation. Yale University.

Saraiva, R., Medeiros, A., Perkusich, M., Valadares, D., Gorgônio, K. C., Perkusich, A., & Almeida, H. (2020). A Bayesian networks-based method to analyze the validity of the data of software measurement programs. IEEE Access.

Sarwar, A., Hafeez, Y., Hussain, S., & Yang, S. (2020). Towards taxonomical-based situational model to improve the quality of agile distributed teams. IEEE Access.

Taaibosch, S. (2017). Factors influencing the transition to Agile software development M.Com Thesis. University of Johannesburg.

Tran, H. K. V., Unterkalmsteiner, M., Börstler, J., & Ali, N. B. (2021). Assessing test artifact quality—A tertiary study. Information and Software Technology.

Tunio, M. Z., Luo, H., Cong, W., Fang, Z., Gilal, A. R., Abro, A., & Shao, W. (2017). Impact of personality on task selection in crowdsourcing software development: A sorting approach. IEEE Access.

Uwadia, C. O., Ifinedo, P. E., Nwamarah, G. M., Eseyin, E. G., & Sawyerr, A. (2006). HEof management information systems for Nigerian universities. Information Technology for Development, 12(2), 91-111.

Vale, T. D. (2017). Software product lines traceability: A systematic mapping study. Information and Software Technology, 84, 1–18.

Van Wyk, L. (2018). A framework to measure the value of agile software projects. M.Com Thesis. University of Johannesburg.

Venkatesh, V., Thong, J. Y. L., Chan, F. K. Y., Hoehle, H., & Spohrer, K. (2020). How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills. Information Systems Journal.

Wang, J., Wang, Z., & Li, J. (2018). A recommendation method for social collaboration tasks based on personal social preferences. IEEE Access.

Younas, M., Jawawi, D. N. A., Mahmood, A. K., Ahmad, M. N., Sarwar, M. U., & Idris, M. Y. (2020). Agile software development using cloud computing: A case study. IEEE Access.

Yu, D., Zhou, Z., & Wang, Y. (2020). Crowdsourcing software task assignment method for collaborative development. IEEE Access.

*CORRESPONDING AUTHOR

Yahya Ateik Al-Sagheer Saeed can be contacted at: *yahya.ateik@live.iium.edu.my*

CITATION

Saeed, Y. A. A., Zulkifli, Z. B., Nasir, S. B. S., Ahmed, W. B. (2025). Factors in determining information systems development methodologies; A systematic literature review. *Sohar University Journal of Sustainable Business*, 1(1). 1-16.

Appendix: List of Papers

Code	Papers	Factors
S1	Assessing test artifact quality—A tertiary study	Central artifacts in software testing (Tran et al., 2021)
S2	Security in agile software development: A practitioner survey	Security Engineering Security design (Rindell et al., 2021)
S3	How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills	Unambiguous role perceptions and thereby (Venkatesh et al., 2020)
S4	An Empirical Investigation of Geographically Distributed Agile Development: The Agile Enterprise Architecture is a Communication Enabler	Active communication efficiency communication effectiveness (Alzoubi and Gill, 2020)
S5	Agile Software Development Using Cloud Computing: A Case Study	Agile Development and Cloud Computing (ADCC) framework (Younas et al., 2020)
S6	Practitioners' Views on Good Software Testing Practices	Test cases (Kochhar et al., 2019)
S7	Test Case Prioritization Using Firefly Algorithm for Software Testing	Automated test case classification automated prioritization of system test cases (Khatibsyarbini et al., 2019)
S8	Enhancing the agility and performances of a project with lean manufacturing practices	Lean Kanban manufacturing principles (Cvetkovic et al., 2017)
I1	Statistical Analysis of the Effects of Heavyweight and Lightweight Methodologies on the Six-Pointed Star Model	Lightweight methodologies for small-scale projects, heavyweight methodologies for medium and large-scale projects (Akbar et al., 2018)
I2	Improving the Quality of Software Development Process by Introducing a New Methodology–AZ-Model	Quality (Akbar et al., 2018)
I3	A Systematic Study on Software Requirements Elicitation Techniques and its Challenges in Mobile Application Development	Quality assessment criteria, time and cost factors, resource effectiveness, domain understanding, applicability (Dar et al., 2018)
I4	Motivators for Large-Scale Agile Adoption From Management Perspective: A Systematic Literature Review	Quick delivery of software products with minimal cost and user satisfaction (Akbar et al., 2019)

I5	An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review	Team and project factors (Fernández-Diego et al., 2020)
I6	Assessing the Risk of Software Development in Agile Methodologies Using Simulation	Risk factors (Lunesu et al., 2021)
I7	An Empirical Investigation of Factors Causing Scope Creep in Agile Global Software Development Context: A Conceptual Model for Project Managers	Ability to manage and control the change elements (Aizaz et al., 2021)
I8	Systematic Mapping Study on Security Approaches in Secure Software Engineering	Software security measures (Khan et al., 2021)
I9	A Bayesian Networks-Based Method to Analyze the Validity of the Data of Software Measurement Programs	Trustworthiness (Saraiva et al., 2020)
I10	A Systematic Literature Review of Test Case Prioritization Using Genetic Algorithms	Quality and reliability (Bajaj et al., 2019)
I11	A Quantitative Framework for Task Allocation in Distributed Agile Software Development	Task allocation to team members (Aslam et al., 2018)
I12	Impact of Personality on Task Selection in Crowdsourcing Software Development: A Sorting Approach	Personality on task selection (Tunio et al., 2017)
I13	Crowdsourcing Software Task Assignment Method for Collaborative Development	Developer collaboration (Yu et al., 2020)
I14	Towards Taxonomical-Based Situational Model to Improve the Quality of Agile Distributed Teams	Situational factors (Sarwar et al., 2020)
I15	A Recommendation Method for Social Collaboration Tasks Based on Personal Social Preferences	Social-collaboration tasks (Wang et al., 2018)
P1	Agile Software Development Practices for Customer Service and Support	Agile software development practices (Fasano & John, 2020)
P2	Are Software Security Issues a Result of Flaws in Software Development Methodologies?	Software Security (Horton & Sandra, 2020)

P3	A Framework to Measure the Value of Agile Software Projects	Agile software development practices (Van Wyk & Louis, 2018)
P4	Exploring The Strategies Software Developers Need to Maintain Security Requirements During Agile Development Processes	Software Security (Campbell & John, 2019)
P5	A Modular Synthesis Framework for Software Deployment, Design, and Implementation	Framework Development (Santolucito & Mark, 2020)
P6	A Multivariate Bayesian Approach to Modeling Vulnerability Discovery in the Software Security Lifecycle	Software Security (Johnston et al., 2018)
P7	Dynamic Modeling of the Effectiveness of Software Development Methods on DoD Programs	Evaluating Different Software methodologies for DoD programs (Patel & Shirali, 2021)
P8	Exploring Security Process Improvements for Integrating Security Tools within a Software Application Development Methodology	Software Security (Jose & Crispin, 2020)
P9	Factors Influencing the Transition to Agile Software Development	Software Quality (Taaibosch & Shari, 2017)
P10	The Impact of Leadership Style on the Adoption of Agile Software Development: A Correlational Study	Evaluating Different Software methodologies (Kipreos & Mike, 2019)
P11	A Framework to Evaluate Candidate Agile Software Development Processes	Project Cost and Risk (De Silva & Ian, 2019)
P12	Exploring the Integration of Security into Software Development Life Cycle (SDLC) Methodology	Software Security (Arrey et al., 2019)
P13	Maximizing the Performance of Agile Teams for IoT Development	Agile and IOT development (Moedt & Wouter, 2019)