Framework for Modeling Risk Factors in Green Agile Software Development for GSD Vendors

Nasir Rashid[1], Muhammad Ilyas[2], Shah Khalid[3], Haseena Noureen[4], Muhammad Salam[5]
[1,2,3,4,5] Department of Computer Science and IT, University of Malakand, Pakistan
*** Correspondence:** nasir@uom.edu.pk

Citation | Rashid. N, Ilyas. M, Khalid. S, Noureen. H, Salam. M, "Framework for Modeling Risk Factors in Green Agile Software Development for GSD Vendors", IJIST, Vol. 7 Issue. 1 pp 210-234, Feb 2025
Received | Jan 02, 2025 **Revised |** Feb 01, 2025 **Accepted |** Feb 02, 2025 **Published |** Feb 04, 2025.

In the last decades, agile methodologies are commonly employed to develop and deliver valuable software, with high user satisfaction at a comparatively low cost. However in recent years, the emergence of Green Software Engineering has necessitated that software developers prioritize the development of Green And Sustainable Software (GSS). Green software development is about developing and utilizing software with restricted energy and computing resources. In recent years, as the application of Global Software Development (GSD), software engineers have applied agile methods for fast, interactive, and green software development. However, such adoption of agile methods poses certain risks. The contribution of this study is two-fold. First, it identifies 8 Risk Factors (RFs), through a Systematic Literature Review (SLR), in which 42 relevant papers are identified and reviewed. The identified RFs need to be avoided by the GSD vendors while using agile methods to deliver GSS. Second, the findings of the SLR study are empirically validated through a questionnaire survey from 106 GSD experts belonging to 25 disparate countries. The results of the SLR and survey were compared and analyzed through a two-proportion Z test using R, which shows some significant variation for some RFs. Lastly, a framework for modeling structural association among RFs was established using an interpretive structural modeling approach. Research results illustrate that the outcomes of our industrial survey are mostly coherent with the SLR findings. Future, research should focus on developing predictive models using Artificial Intelligence (AI) and Machine Learning (ML) to analyze project data in real-time, promoting proactive decision-making for GSS development.

Keywords: Agile Software Development; Green and Sustainable; Risk Factors; Empirical Study; Systematic Literature Review.

Introduction:

Agile software engineering is an exciting paradigm with iterative and lightweight approaches for quick delivery of customer-focused software development. Compared to the traditional models of software development, agile techniques try to minimize risks and maximize the development of high-quality software [1]. The agile manifesto introduced an industry-initiated vision for a self-reflective shift of the engineering paradigm in software engineering. It also presented quite innovative work in carving and disseminating the critique of standardized software processes over the previous decade, turning into a well-known practice among academicians [2].

Agile methods aim to survive in dynamic environments where customers' requirements frequently changed [3]. They prioritize the personal skills and creative efforts of developers over rigid procedures and extensive formal documentation. While maintaining detailed documentation can be time-consuming and adds to the effort required for delivering software systems, preserving essential information over time is crucial for future use [3]. Thus, agile methodologies aim to prevent predefined and formalized processes that add little effort to actual software development [1].

Agile development is based on a rapid delivery system to ensure customer satisfaction, a commitment to sustainable development, simplicity, adaptability, flexibility to accommodate changes, and a strong emphasis on communication and collaboration [2][4]. These methods usually adopt the "iterative enhancement" technique where each iteration indicates a short-scale and self-governing development life cycle [3]. These methods encompass various methodologies including "Extreme Programming, Scrum, Crystal Clear, Adaptive Software Development (ASD), Feature Development (FDD), and Dynamic System Development (DSD)" [5].

Initially, agile methods were designed for on-site/co-located team(s). However, it has been recently discovered that these methods can also be effective in distributed environments. Approaches including Scrum and XP were reconfigured as Distributed Scrum and Distributed XP [6][7]. While Agile methods provide numerous benefits in GSD environments, it has also been recognized that adapting these methods can significantly support the development of green and sustainable software, aligning with current dynamic requirements [8].

An emerging trend in software development focuses on delivering environmentally friendly systems that aim to extend battery life and reduce development costs. There is a dire need to minimize the usage of computing resources and energy to support sustainable software development to develop ecofriendly software [9].

GSS refers to the design and development of software that minimizes or eliminates direct and indirect negative impacts on a country's economy, people, society, and environment during the pre-development, development, and post-development stages while promoting sustainable production practices [10]. Information and Communication Technology (ICT) also damages the environment due to increased power and resource utilization despite their positive impact. Recently, the ICT sector has focused on exploring energy-efficient processes for software development to reduce its environmental impact [11][12]. Green ICT has emphasized the impacts of hardware on environmental sustainability. However, addressing energy utilization issues by improving software design can yield better results in green computing, as software development and execution significantly influence the operating environment [13].

Some software solutions regarding energy efficiency are well designed to better monitor and utilize the system resources and others are sustainable enough to restrict the addition of new hardware. Up to this time, there is enough research conducted in the domains of hardware but the need for further research and work is still necessary for green software [14].

Recently, some efforts [15][16] have been made to implement green software metrics to deliver sustainable software. Research conducted in [17], has focused on building software tools that can measure of development environment on surroundings in terms of energy and usage of computing resources. Some research work accentuated operating system architectures to better control the power utilization of software applications [18][19].

Green Software Engineering (GSE) aims to reduce the consumption of energy and computing resources and to reduce the negative impacts on human beings due to the development and use of software systems [10]. General solutions toward greener software processes [20][21] include virtualization, encoding efficient algorithms, designing efficient load balancing mechanisms in case of parallelism, applying green computing strategies, and designing energy-efficient routing algorithms.

Now software development requires agile principles because they enable the development of GSS [22][23]. The environmentally friendly development of software benefits from Agile principles which include requirement change acceptance alongside early software delivery with simple design approaches, and early defect detection strategies [24][25][26].

Current research is focused on the design and development of GSS by adapting the interactive agile methods in GSD. To avail the maximum benefits of the two paradigms i.e., agile and GSD, software development organizations have adapted agile in distributed software projects and coined another term, Agile GSD [27]. The software industry now recognizes the value of an integrated approach to software development, which delivers high-quality products faster, more cost-effectively, and with greater adaptability to evolving requirements [28][29][30].

While agile methods are applied in the current software industry to develop green and sustainable software, this area has attracted very little attention from the research community in the GSD environment and there is relatively scant empirical research available in this field. The benefits of agile methods do not work equally well across all development environments or all agile methods. The universal applicability of agile methods remains limited because agile methodology does not function similarly for every software development. It also poses several risks in various environments, such as limited support for distributed development, less support for safety critical software and re-usable software systems, management overhead, reliance on the implicit knowledge of developers, insufficient knowledge of the customers, and ineffective communication between the customer and developers [12].

The considerable number of publications shed light on the issues in the AGSD environment that motivated us for a systematic review to identify RFs that need to be avoided by GSD vendors while practicing agile methods for the delivery of GSS. Based on the expected findings the proposed study will be the first of its kind which will be useful in the creation of the 'green agile maturity model for GSD vendors (GAMM)', which will serve to assess the green-agile maturity of GSD vendors.

To meet the objective of systematically identifying the RFs in developing GSS using agile methods, we aim to focus on the below Research Questions (RQs).

- What are the risks involved, as identified in the literature, to be avoided by GSD vendors in the development of GSS using agile methods?
- What are the risks involved, as identified in real-world practice, to be avoided by the GSD vendors in the development of using agile methods?
- Is there any coherence between the findings, and risk factors, of the systematic literature review and industrial survey?
- What are the interrelationships among the identified risk factors?
- What are the driving and dependence power of the identified risk factors?

The objective of the Research:

This research aims to respond to the growing customer need for GSS by comprehending and managing the risks associated with agile methodologies present in GSD. Moreover, the research objectives of the study include establishing a classification of key RFs that negatively influence the development of GSS whenever agility is incorporated. All these RFs can be established using a Systematic Literature Review (SLR) that involves 42 papers found in the literature. Besides, the study intends to confirm the results of the SLR through an extended questionnaire survey with 106 GSD practitioners from 25 countries. Another goal is to establish a framework that employs interpretive structural modeling methodology to depict the structural interrelation between the various RFs. Towards that end, this framework offers a richer view of the interplays and relationships between the risks and the steps that GSD vendors can take to facilitate the use of agile methods for GSS development. This research also enriches the existing knowledge in green software engineering by providing solutions to mitigate risks and enhance sustainable practices in an agile-driven GSD setting.

Related Work:

Software productivity has dramatically improved with the adoption of the agile methodologies [3], to revamp the key processes involved in software development. Agile methods rely on designing a sequence of prototypes, using higher levels of abstraction, and modularization done by small teams. Agile methods also ensure the delivery of GSS through short increments, polymorphic design, informal communication, and optimized and energy-efficient coding [19].

However agile methods are hampered by some limitations towards sustainable software development including test automation, backlog management [20], volatility of requirements in an evolving environment, unavailability of customer representatives, and limited support for developing re-usable products [21][22].

Using agile methods in distributed teams is more common in its way to meet the current era demands. However, agile distributed development certainly poses some challenges, such as task management, delays in project delivery, and team management and communication problems due to cultural and linguistic differences [23].

Adel Hamdan et al. [24] demonstrate the strengths and weaknesses of agile development in different projects. Besides the voluminous benefits, it offers for onshore and offshore software development, agile methods still face some challenges when it comes into practice, such as inconsistency in customer interaction, difficulty in managing large teams, lack of long-term planning, lack of formal communication, and insufficient documentation.

Chin et al. [31] discovered some patterns of continuous integration that are used to support dynamic software development platforms. The authors documented that cross-platform software development faces a critical challenge in the context of practicing agile methods i.e., a lack of direct involvement of customers in the development life cycle. The absence of the customer delays the process of software builds' development and is possibly susceptible to fail, thus harming sustainable software development.

Juyun Cho [25] has discussed several benefits of agile development. However, some organizations refuse to switch to agile for several reasons, such as agile methods 1) greatly reduce the documentation and rely on the tacit knowledge of developers, 2) have limited support for safety-critical and stable projects and 3) lack the support for development of large software systems.

Agile methods provide limited implementation details of the development processes. This characteristic may result in software systems that address only short-term needs, potentially leading to software that is not environmentally sustainable and lacks long-term viability. The lack of insufficient documentation also gives rise to enhanced complexity, lack of system familiarity, arduous maintenance, and difficulty accessing change in all sorts of software development [26].

The literature explains some potential risks of agile methods when adapted by the development team to deliver green and sustainable software. However, no empirical research work, using SLR, has been carried out to date in finding the RFs of agile methods, when adapted for delivery of GSS using agile methods. The authors further validated the RFs, initially identified through SLR, via industrial survey. The findings of the current paper contribute to the first phase of our proposed GAM model [32]. The GAM model intends to assist the GSD vendors in assessing their agile maturity for GSS development.

Research Method:

In this study, we have used a Systematic Literature Review (SLR) and questionnaire survey for extracting and validating the RFs in GSS development (GSSD) using agile methods, as used by other researchers [33][34][35][36][37].

Using guidelines for conducting SLRs [38], we performed a comprehensive search of RFs related to using agile methods for GSSD. SLR is a protocol-based approach that provides higher accuracy and objectivity than traditional literature reviews [39][40][41][42][43][44][45][46][47][48]. The different phases of our SLR process are described in the following section.

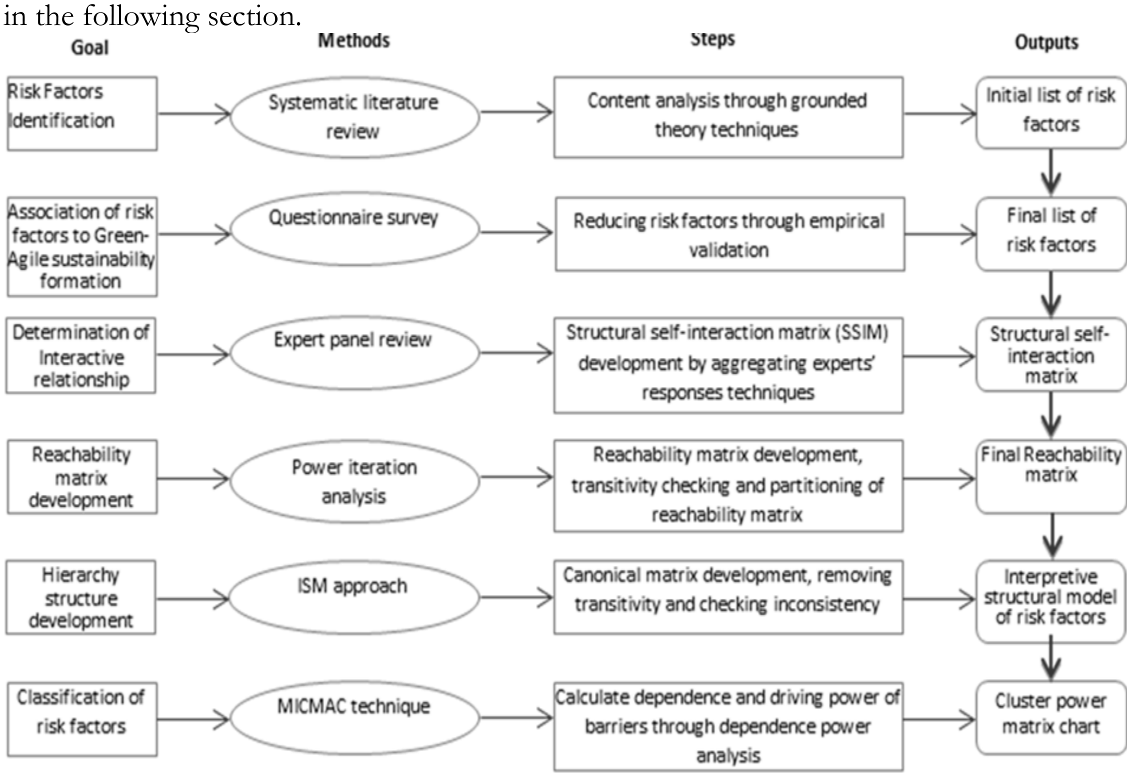

Figure 1. Research Methodology

SLR guidelines were followed while developing the search strategy [49] and it was implemented in the identified digital libraries, as shown in Table 1.

The SLR follows specific steps which begin with establishing research goals and questions while creating a thorough review plan and extend to database search efforts and quality assessment of included studies followed by organized data extraction and synthesis until the final presentation of results. The guidelines establish standards for a review process that maintains rigorous methodologies transparent procedures and reproducible research practices.

Title and abstract were first used to select papers and later screened according to the inclusion and exclusion criteria using full text. A selection process for titles used predefined keywords that matched the research objectives to maintain study alignment with its specified scope. Quality was evaluated alongside data extraction based on factors such as the provision of concrete examples of how agile methods may be adapted for green development and the establishment of RFs.

The authors selected 42 papers using predefined criteria that examined publication in peer-reviewed journals or conferences while meeting the research objectives and scope of GSS development using agile methods. From the 42 papers, data were collected based on specific parameters such as study approach, type of organization/industry, agile methodologies, and risks under consideration. A descriptive summary at baseline analysis revealed 16 risk factor groups that were later reduced to 8 groups: 6 of these were critical RFs where the frequency was ≥40%, the criteria for critical factors being used by [39][50][51].

Table 1. Search results of various digital venues

Digital library	No. of publications	Primary selection	Final selection
Science direct	162	52	9
ACM	58	20	6
IEEE Xplore	92	16	8
Springer link	150	23	7
Google Scholar	266	36	12
Total	**728**	**147**	**42**

The questionnaire was pre-tested through a pilot study that involved 10 GSD practitioners who were randomly selected from the list of participants who agreed to participate in the main study. They included practitioners from various geographical areas and had considerable experience in agile methods for GSS development.

During pre-testing, the focus was on assessing the validity, reliability, and sensitivity of the items included in the questionnaire. The participants were also asked to comment on the wording, format, and general functionality of the questionnaire. Several changes were made as follows: The language in some items was made plain and clear; Some ambiguous questions were rephrased to be consistent; Some questions were realigned for logical sequence. Additionally, the Likert scale options were reviewed to ensure they aligned with the study's aims and objectives, and minor adjustments were made to improve the clarity of the responses. This pre-testing process helped to develop a good and clear version of the questionnaire to be used in the survey conducted among 106 GSD practitioners.

To validate the findings of the SLR, an online questionnaire survey was administered to 106 Agile-GSD practitioners across 25 countries with a >50% response rate. Respondents were recruited from the LinkedIn professional groups. The survey was conducted with the help of a 7-point Likert scale; however, the special focus was made on 'extremely agree and agree' responses. This survey also aimed to find out other RFs that practitioners in the IT industry have recognized as important.

We reached out to practitioners through LinkedIn groups and professional networks; 71 from Asia, 53 from Europe, 39 from North America, and 27 from other regions.

Systematic Expert Panel Review:

The next step was to determine the structural relationship within the RFs through a review from an expert panel. ISM was employed to map experts' perceptions of the relative closeness of the identified RFs to deduce the direct and indirect connections between the RFs. To this end, we selected a pool of ten specialists from respondents of the survey due to their experience and specialization in the ASD field. For the ISM technique, only RFs that most of the experts selected in the survey were used for further analysis. To establish pair-wise relations among the RFs, the opinion of the experts was sought in terms of the following four categories: achieved by, leads to, bidirectional, and no relation concerning the table showing barriers. The result was then translated based on Table 10.

Interpretive Structural Modeling (ISM):

ISM employs the application of MCDM that explains the complex pattern of associations by incorporating simple notations of graph theory [52][53][54].

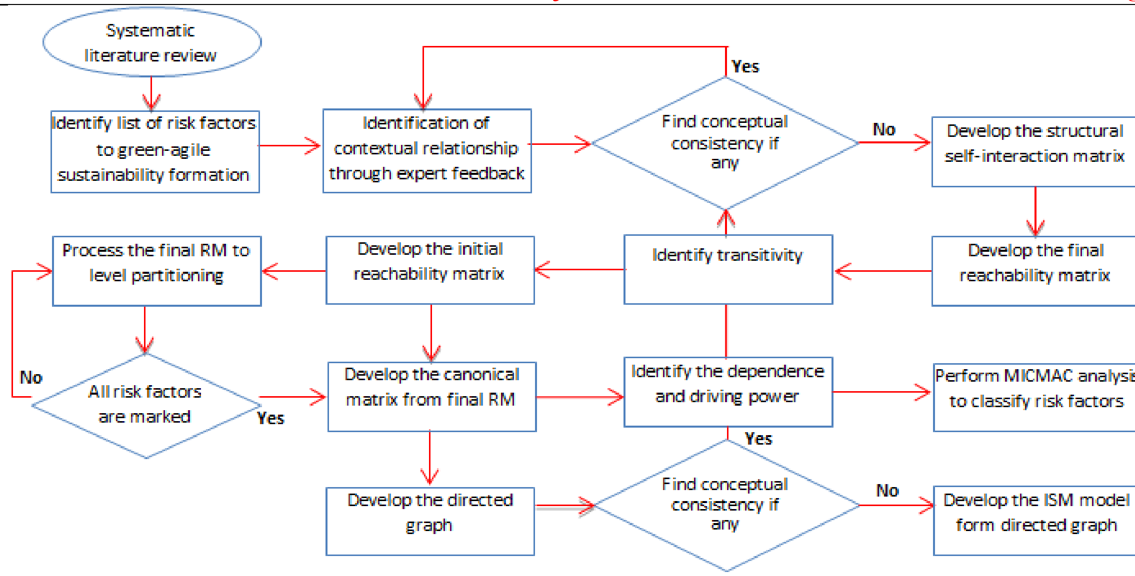

Figure 2. Analytical framework for analyzing and modeling structural associations among risk factors for green-agile sustainability

It is developed based on an interpretive method (derived based on the opinions of industrial and academic specialists) for developing the relative relationship between the identified RFs which may be associated or independent and those that have a positive or negative impact on a problem or an issue of interest to the research community [55][56][57]. It must be noted that unlike, TOPSIS, AHP, and ANP for developing the association amongst the enumerated factors, the ISM approach does not require This helps reduce the expert's bias and objectivity at the same time as enhancing the relationship between several predictors and eventually enhancing the accuracy of the model. It gives a clear understanding of the model variables and helps the expert to identify structure within the identified variables [55][56][57].

Table 2. Linguistic terms for RFs

Linguistic Terms	Meaning	Corresponding symbols
Achieved by	Risk factor ***a*** will help to achieve risk factor ***b***	V
Leads to	Risk factor ***a*** will lead to risk factor ***b***	A
Bidirectional	Risk factors ***a*** and ***b*** will achieve each other	X
No relation	Risk factors ***a*** and ***b*** have no relation to achieving each other	O

A complete flow diagram of ISM and MICMAC-based analysis methodology is illustrated in Figure 1 and Fig 2. Various steps involved in ISM methodology are given as follows:

Identification of RFs Through SLR and Survey:

To identify variables related to the issues or problems under consideration, in the first step, various RFs to Green agile development formation were identified via SLR and empirical survey.

Development of Structural Self-Interaction Matrix (SSIM):

To become aware of the contextual associations among the RFs, in the second step, an initial pair-wise association among the identified RFs was developed in the form of a Structural Self-Interaction Matrix (SSIM). Experts' evaluations were used to determine the value of the SSIM scale presented in Table 2 for translation.

Development of Initial Reachability Matrix (RM) From SSIM:

The SSIM reveals the first-order correlation matrix between the RFs whereas the RM reveals both the first and second-order correlation matrix. According to the SSIM two actions were performed to build the RM. To establish the direct connections between the RFs an initial reachability matrix (RM) was derived from the SSIM of Step 2.

Obtain Final RM from Initial RM by Including Transitivity:

The first RM based only on SSIM demonstrates only direct dependencies between variables without indirect ones; it is crucial to find the indirect associations between the RFs by using the check of transitivity. Consequently, in the fourth stage from the initial RM, the final RM was determined through power iteration analysis for transitivity. Shen et al postulate [58], that, if Rf is the final RM then Ri can be given as the initial RM whereby, Rf can be derived as (1).

$$R_f = R_i^k = R_i^{k+1} \quad (1)$$

Partition of RM into Different Levels:

To derive the hierarchy structure and to identify various levels in the hierarchy, the following from the final RM of Step 4 will be determined: – Reachability set R – Antecedent A – Intersection set R ∩ A for each risk factor. Based on set R, A, and R ∩ A, level partitions will be performed. To determine the level of each risk factor, set R was compared to the intersection set R ∩ A [58]. Challenges that R = R ∩ A will ensure attaining level 1 are as follows; the one that was labeled will be removed from the rest and the same will be done over and over until all RFs are labeled.

Formation of Canonical Matrix and Development of Digraph:

Finally, from the last RM of associations among the barriers, a canonical matrix is derived in this step. A digraph (directed graph) is drawn with the help of a canonical matrix. A digraph is created with the help of the 1's present in the canonical matrix.

MICMAC Analysis Techniques:

To counteract the dispersion of the RFs, MICMAC also known as cross-impact matrix multiplication applied to the classification analysis technique was used in the last step to categorize the RFs into four categories depending on the dependence power and driving power of each risk factor [59].

Results:

This section describes the outcomes of the SLR process and the industrial survey. Here we discuss the results and analyses of the identified RFs to answer RQ1, RQ2, and RQ3. The details are given in the following subsections.

Risk Factors in the Development of Green and Sustainable Software Using Agile Methods (RQ1):

To answer RQ1, we identified a list of 8 RFs, through SLR, as presented in Table 3. Amongst the identified RFs, some are pinpointed as high-ranked and considered critical RFs (CRFs). We set the criteria for a critical factor based on its frequency percentage (%) in the publications' sample (N=42). If the frequency % of a particular risk factor is >=40%, it was included in the list of critical factors. A similar approach is adapted in [51][60][61]. However, GSD experts are intended to determine their criteria for measuring the significance of a factor to be critical. The details of these CRFs are as follows:

Table 3 Risk factors identified through SLR

S. No	Risk factors	Frequency N=42	%
1	Insufficient System Documentation	26	62
2	Limited Support for Real-Time Systems and Large Systems	19	45

3	Management Overhead	25	60
4	Lack of Customer's Presence	26	62
5	Lack of Formal Communication	18	43
6	Limited Support for Reusability	07	17
7	Insufficient Knowledge of the Customer	15	36
8	Lack of Long-Term Planning	20	48

CRF1: Insufficient System Documentation:

Table 3 portrays that 'insufficient system documentation' is the most critical risk factor (62%) among the identified factors. This is evident that the development of GSS needs proper and updated documentation regarding the requirements, architecture design, and testing [15][62].

Agile methodologies cater to limited implementation details regarding the product development process, and because of this "low-level" approach they may build software that meets short-term individual project needs, but that does not necessarily lead to GSS [63]. A lack of or insufficient system documentation may be a big hindrance to achieving sustainability in software systems and lead to increased system complexity, degraded maintainability, and lack of system familiarity [37].

CRF2: Lack of Customers' Presence:

Our findings "revealed that 'Lack of customer's presence' (62%) have the same severity as the previously mentioned risk factor. The consistent presence of the customer and his/her active involvement in software development is the most essential component of agile methods. This reduces the efforts to complete the software within a defined time frame and leads to sustainable software development [38]. In the absence of a customer, the software builds take a longer time to complete and are more likely to fail, thus harming sustainable software development "[15].

Coherent, self-organizing agile teamwork and strong communication with customers is an agile principle that supports a green process throughout software development [31]. Agile methods state that the customer should be an active participant in each iteration of software development from design to implementation and then maintenance for long periods, which is not fully applicable in the GSD environment [64].

CRF3: Management Overhead:

Management of time and computing resources is an integral principle that promotes GSS development [43]. However, agile methods undergo the challenge of 'management overhead' (60%), which results in over budget, time overrun, and full utilization of available and accessible computing resources for developing software. This can go a long way to affect the green and sustainable way to software development [64].

CRF4: Lack of Long-Term Planning:

Intelligent and long-term planning is a crucial factor to be considered for the social, economic, and environmental sustainability of the software that deals with the usage, business, and technical aspects in terms of energy efficiency [65]. The 'lack of long-term planning' (48%)' is listed as the fourth highest risk factor in using agile methods to design and deliver green and sustainable software. "The importance of sustainability is increasingly recognized in terms of software development which needs long-term planning for the design, use, and maintenance of software [66]. However agile methods focus more on the immediate delivery of software according to the current needs of customers without taking into consideration its long-term impact on humans and society as a whole [67].

CRF5: Limited Support for Real-Time Systems and Large Systems: We found the 'limited support for real-time systems and large systems' with 45% frequency, as a critical risk factor of agile methods, adapted to deliver green and sustainable software, as shown in Table 3. The design and development of real-time and large systems in terms of energy efficiency is a complex task. An approach to developing such systems needs a variety of objectives which include strategic decision-making, long-term planning, designing complex architecture, and coping with various interrelated internal and external factors [68]. Agile methods have proved successful with a focus on software development in small, short-lived nature, and simple in structure with support for dynamic customers' requirements [69]. Fundamental assumptions in agile software development hamper the development and maintenance of real-time and large projects with sustainability aspects [70][71].

CRF6: Lack of Formal Communication:

We also found the 'lack of formal communication' with 43% frequency, as a critical risk factor of agile methods, adapted by the development team to deliver green and sustainable software, as shown in Table 3. One of the major principles of agile methods is to emphasis more on informal communication for fast delivery of users' requirements among the team members and customers. However, this feature of agile methods fails to document the critical attributes of software regarding its architecture design, code design, and reengineering [72][73].

Both formal communication and structured documents regarding software processes are essential for effective sustainable software development [74]. The lack of formal communication fails to describe the major decisions, user requirements, and progress of the software project and thus leads to software development that does not meet the sustainability criteria.

Risk Factors Validated through Questionnaire Survey (RQ2)

Table 4 presents a list of RFs that were obtained through the analysis of the questionnaire survey to answer RQ2. A questionnaire survey explained in section (Research Method) was conducted to find out the perception of both local and global experts in the software industries using the Google online survey forms. Out of one hundred and six GSD experts, in our research domain, from 25 different countries, Asia-71, Europe-53, North America-39, and other-27 completed the survey.

The data from questionnaire survey feedback, as shown in Table 4, demonstrates that all the identified RFs have got >=50% positive responses in our sample. Such a positive response indicates that all the identified RFs are of significant value for GSD vendors while adapting agile methods in delivering green and sustainable software.

Table 4. Summary of GSD experts' feedback regarding the risk factors

Risk Factors	Total responses from industry practitioners=106									
	Positive (+ve)				Negative (-ve)				None	
	ES	MS	SS	Positive%	SD	MD	ED	Negative%	None	%
Insufficient system documentation	31	36	23	85	5	2	5	12	4	4
Limited support for real-time systems and large systems	25	34	19	74	10	2	5	17	11	11
Management overhead	26	32	17	71	14	6	6	26	5	5
Lack of customer's presence	24	34	19	73	10	9	5	24	5	5
Lack of formal communication	26	35	23	79	9	3	5	17	5	5
Limited support for reusability	18	27	24	65	13	7	2	22	15	15
Insufficient knowledge of the customer	24	32	19	71	14	7	6	27	4	4

Lack of long-term planning	23	40	15	74	8	7	6	21	7	7

Analysis of the RFs Across SLR and Questionnaire Survey (RQ3):

To answer RQ3, we have compared the SLR results with the questionnaire response (Positive only) by applying a two-proportion Z test using R programming. This analysis purports to analyze the significant differences, if any, among the risk factors in two data sets, as shown in Table 5.

The comparison of the findings from both data sets is also illustrated in Figure 3. The results of the two-proportion Z-test for each risk factor are summarized below:

Table 5. Two-Proportion Z test analysis of SLR and Survey results

Risk Factor	SLR %	Survey Positive %	Z-Score	P-Value
Insufficient System Documentation	62	85	-3.43	0.0006
Limited Support for Real-Time Systems	45	74	-4.01	0.00006
Management Overhead	60	71	-1.55	0.12
Lack of Customer's Presence	62	73	-1.57	0.12
Lack of Formal Communication	43	79	-5.00	0.0001
Limited Support for Reusability	17	65	-6.77	0.0001
Insufficient Knowledge of the Customer	36	71	-4.81	0.0001
Lack of Long-Term Planning	48	74	-3.61	0.0003

Table 5 presents the results of comparing the SLR with survey findings. By applying the two-proportion Z-test, similarities, and differences in the identified RFs for GSS development using agile methodologies can be highlighted. The p-values of most RFs indicate that there is a significant difference in the frequency of these RFs in the two data sets such as "Insufficient System Documentation", "Limited Support for Real-Time Systems", Lack of Formal Communication", "Limited Support for Reusability", "Insufficient Knowledge of the Customer", and "Lack of Long-Term Planning". These outcomes imply that even though these factors are described in the literature as critical risks, they may not be viewed as critical or as frequently occurring within actual agile settings by the practitioners. Such differences may mean that there are differences between the contexts under which academic research is developed and the contexts in which it is implemented, between the scope of research work and problems in practice, or between the focus of research work and the focus of practical work. However, "Management Overhead" and "Lack of Customer's Presence" RFs are not significantly different between the two datasets. This alignment shows that these risks are well identified in the literature and acknowledged by agile practitioners as important for GSS development projects.

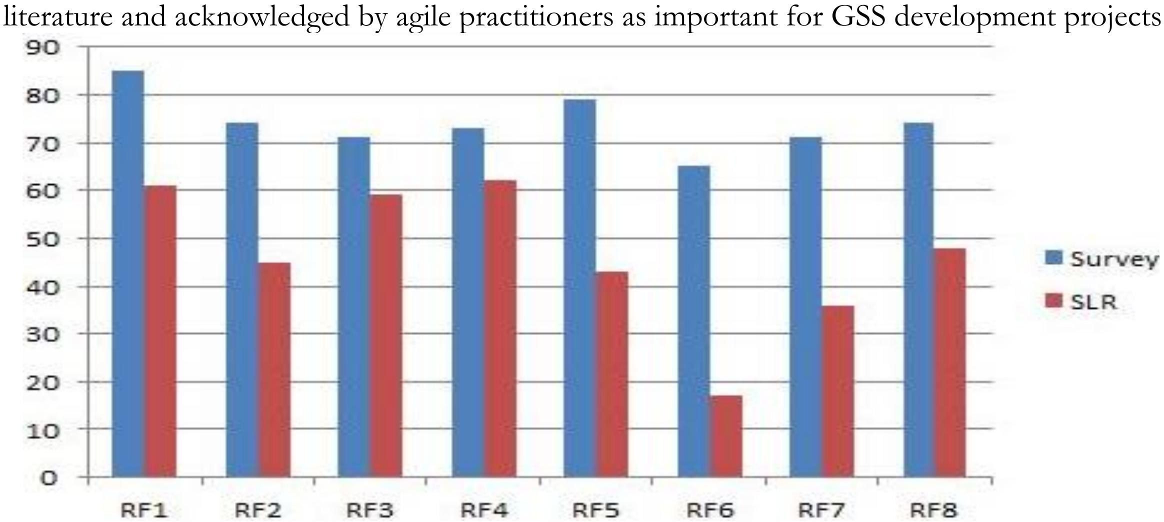

Figure 3. Comparison of SLR and questionnaire survey results.

Further, the identified RFs are compared in both data sets in terms of ranking, RFs are ranked in descending order, as shown in Table 6.

Table 6. Comparison of the Ranks of risk factors of SLR and questionnaire survey data

Risk factors	Occurrence in SLR (N=42)		Marked agree in questionnaire survey (N=106)		Average Rank
	%	Rank	%	Rank	
Insufficient system documentation	62	1	85	1	1
Limited support for real-time systems and large systems	60	2	74	3	3
Management overhead	43	3	71	5	4
Lack of customer's presence	36	4	73	4	4
Lack of formal communication	30	5	80	2	4
Limited support for reusability	27	6	65	6	6
Insufficient knowledge of the customer	27	6	71	5	6
Lack of long-term planning	24	7	74	3	5

As shown in Table 6 the comparison of rankings from the SLR and survey identifies the degree of consistency and inconsistency in the perception of RFs. When comparing the two datasets, three factors reveal consistency across both: "Insufficient System Documentation" ranks highest in both. The same is true for such factors as "Limited Support for Reusability" and "Insufficient Knowledge of the Customer", which are identified both by academic sources and by agile practitioners. Other factors include "Limited Support for Real-Time Systems and Large Systems" and "Management Overhead" which, although their ranks have changed slightly, are still considered highly important in both settings. However, "Lack of Long-Term Planning" constitutes a larger gap according to the survey results, ranked 7th in the SLR and 5th in the survey because practitioners are more concerned with strategic planning when using Agile due to difficulties in achieving sustainable long-term objectives in an iterative environment. In general, the concordance of most rankings indicates mutual concerns.

Further, to assess the RFs significance of SLR and survey data, we applied Spearman's rank-order correlation, as presented in Table 7. It is worth mentioning that the ranks of the RFs across both data sets are not similar. Spearman's correlation coefficient is 0.430, which indicates disagreement between the ranks for both data sets. In addition, the scatter plot, shown in Figure 4, illustrates more differences than similarities in the ranks of the RFs across both data sets. This is because the points in the scatterplot do not follow any linear pattern, showing that the relationship between the two variables (SLR and survey) is non-linear.

Table 7. Spearman's rank-order correlation.

			SLR	Survey
Spearman's rho	SLR	Correlation coefficient Sig. (2-tailed) N	1.000 8	0.430 0.287 8
	Survey	Correlation coefficient Sig. (2-tailed) N	0.430 0.287 8	1.000 8

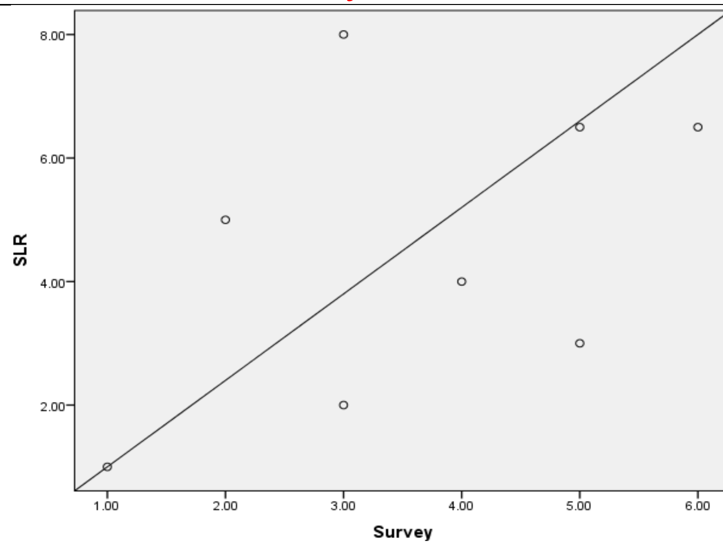

Figure 4. SLR and Survey Ranks Correlation scattered Graph of the RFs.

To obtain more accurate correlation results, the independent t-test was also used to compare the means of the SLR, and the survey as depicted in Table 8-9. Since Levene's Test is not significant (that is, .525>0.05), then we go for "equal variances not assumed". As for this assumption, the t-test value showed that there was no significant difference in the SLR and the questionnaire survey (t = .653, p = .525 > 0.05).

Table 8. Group Statistics

	Type	N	Mean	Std. Deviation	Std. Error Mean
Factor	SLR	8	4.25	2.121	.750
	Survey	8	3.63	1.685	.596

Table 9. Independent samples T-test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	Lower
Factor	Equal variances assumed	.670	.427	.653	14	.525	.625	.958	-1.429	2.679
	Equal variances not assumed			.653	13.31	.525	.625	.958	-1.439	2.689

Interpretive Structural Modeling (ISM) Analysis (RQ4):

In this section, the findings concerning the third research question are reported. In the first step of our methodology, 08 RFs were extracted from 42 papers which were included in the SLR. Before answering RQ3, we checked the identified RFs' validity through SLR, using a questionnaire survey. Finally, after considering the majority's opinion, only 08 RFs were taken for further analysis using the ISM technique. For the ISM study, out of the participants of the major survey, we have chosen a panel of 10 experts based on their experiences. To minimize single-point bias, the experts selected were from both academia and practitioners. To establish a pair-wise relationship among the identified 08 RFs as presented in table 6. The experts were requested to express their opinions regarding four kinds of relationships (achieved by, leads to, bidirectional, no relation) in the row and column.

Development of Structural Self-Interaction Matrix (SSIM):

In the second step, to build a pair wise relation among the identified 08 RFs as mentioned in Table 11 the following SSIM is created based on the relation found between 08 RFs across row and column of Table 11. A pathway connecting the listed RFs with the symbols O, X, V, and A has been described. Entry (a, b) will be marked as: a

- when no relation exists among the RFs
- ***X*** when bi-directional association exists among the RFs
- ***V*** if barrier ***a*** will help to achieve risk factor ***b***
- ***A*** if risk factor ***b*** will help to achieve factor ***a***

SSIM matrix is established based on the interrelationship among the listed RFs. The SSIM matrix is discussed with experts involved in agile software development. Because of experts, risk factor "R1" (i.e. 'insufficient system documentation') will help to address risk factor 'R7' (i.e., 'insufficient knowledge of the customer"), so symbol 'A' is assigned. Risk factor "R2" (i.e. 'insufficient system documentation') leads to risk factor "R8" (i.e. 'Lack of long-term planning'), therefore, it is represented by the character 'V'. Further, the relation between "R1" (i.e. 'insufficient system documentation') and risk factor "R6" ('limited support for reusability') is bi-directional therefore it is denoted by the symbol 'X'. Both can help to achieve each other. Additionally, risk factor 'R1' (i.e. 'insufficient system documentation') and risk factor 'R2' (i.e. 'limited support for real-time systems and large systems') are unrelated and are therefore denoted by the symbol 'O' and so on (see Table 12).

Development of Reachability Matrix (RM) from SSIM:

To develop initial reachability among the RFs, initial RM is obtained from the SSIM as illustrated in Table 10.

Table 10. Reachability matrix (RM)

Code	1	2	3	4	5	6	7	8	Driving power	Ranked
R1	1	1*	1	1*	1	1	0	1	7	3
R2	1*	1	0	0	0	1	0	1	4	7
R3	1	1*	1	1	1	1*	0	1	7	2
R4	0	1*	0	1	0	1*	0	1	4	6
R5	0	1*	0	1	1	1*	0	1	5	4
R6	1	0	1*	0	1*	1	0	1*	5	5
R7	1	1*	1	1	1*	1*	1	1	8	1
R8	1*	1	0	0	0	1	0	1	4	8
Dependence Power	6	7	4	5	5	8	1	8	44	
Ranks	4	3	7	6	5	2	8	1		

Table 11. Final reachability matrix

Code	Reachability set	Antecedent set	Intersection	Level
R1	{1, 2, 3, 4, 5, 6, 8}	{1, 2, 3, 6, 7, 8}	{1, 2, 3, 6, 8}	V
R2	{1, 2, 6, 8}	{1, 2, 3, 4, 5, 6, 7, 8}	{1, 2, 8}	II
R3	{1, 2, 3, 4, 5, 6, 8}	{1, 3, 6, 7}	{1, 3, 6}	V
R4	{2, 4, 6, 8}	{1, 3, 4, 5, 7}	{4}	III
R5	{2, 4, 5, 6, 8}	{1, 3, 5, 6, 7}	{5, 6}	IV
R6	{1, 3, 6}	{1, 2, 3, 4, 5, 6, 7, 8}	{1, 3, 6}	I
R7	{1, 2, 3, 4, 5, 6, 7, 8}	{7, 8}	{7, 8}	VI
R8	{1, 2, 6, 8}	{{1, 2, 3, 4, 5, 6, 7, 8}	{1, 2, 6, 8}	I

For obtaining an initial RM, the interaction created in SSIM is converted into a binary matrix (0, 1), substituting symbols O, X, V, A by 0 and 1 by the following rules: If the intersection of (a, b) in the SSIM is as follow: O then (a, b) = (b, a) = 0 in the reachability matrix.

- X then (a, b) = (b, a) = 1 in the reachability matrix
- V then (a, b) =1 and (b, a) =0 in the reachability matrix
- A then (a, b) =0 and (b, a) =1 in the reachability matrix

According to the four rules mentioned above, the first estimate of the relative risk is given in Table 10. In the initial RM, the final RM is obtained by using the transitivity in the computation of power analysis through the transitivity rule. Transitivity is assumed in the ISM approach that if a risk factor 'X' is related to 'Y', and 'Y' is related to 'Z', then 'X' is related to 'Z'. The last value of RM is derived from the initial RM by making the addition of transitivity by hand.

Table 12. Conical matrix (CM)

Code	8	6	4	2	5	1	3	7	Driving power
R8	1	0	0	0	0	0	0	0	1
R6	0	1	0	0	0	0	0	0	1
R4	1	0	1	0	0	0	0	0	2
R2	1	1	0	1	0	0	0	0	3
R5	1	0	1	0	1	0	0	0	3
R1	1	1	0	0	1	1	0	0	4
R3	1	0	1	0	1	1	1	0	5
R7	1	0	1	0	0	1	1	1	5
Dependence Power	7	3	4	1	3	3	2	1	**44**

The last RM is shown in Table 11 and 1v means forward transitivity and 1a means backward transitivity. Forward transitivity is attained by symbol V while backward transitivity is attained by symbol A Table 14 also shows the driving (as a row) and dependence power (as a column) of each risk factor with their respective ranks. The driving and dependence power calculation of RFs is done based on the final reachability matrix and is explained as follows:

Driving Power of the RFs: To obtain the driving power of RFs, we count the number of 1's across the rows in the final RM.

Dependence Power of the RFs: To obtain the dependence power of RFs, we counted the number of 1's across the columns in the final RM.

Ranks of the RFs: We ranked the RFs based on the driving and dependence power such that an influential with the highest driving and dependence power was assigned a high rank.

The final reachability matrix is further used for level partitioning of the RFs for building ISM hierarchical structure, while both driving and dependence power of the risk factor are used to help in conducting the MICMAC analysis.

Partition of Final RM into Different Levels: From the final RM, as illustrated in Table 11, the reachability set R and antecedent A for each risk factor is obtained. The final RM is partitioned into various levels according to the following procedure:

Reachability set: Set R of risk factor a will contain itself and all other RFs that help to achieve them in the row.

Antecedent set: Set A of risk factor A will contain itself and all other barriers that help to achieve them in the column.

R ∩ A is obtained for all RFs which give a set of RFs common in both R and A.

A top-level node in the hierarchy: A risk factor with reachability set R and intersection set R ∩ A will be assigned level 1 where R = R ∩ A. A top-level risk factor is a risk factor that has been attained with assistance from other RFs but in return does not contribute towards attaining any other risk factor. The top-level risk factor can be over one putting pressure on the other at the same level of the risk factor. In the second step, after the top-level RFs were designated, they were distinguished from the remaining RFs.

The same process was continued until all possible levels were highlighted. In the present study, the process was taken up to eleven iterations. The results per level are given in Table 11. These levels were further used to arrive at a conical matrix and digraph and to build the last hierarchy of ISM. Thus, from figure 6, it is clear that 'insufficient system documentation' (R1) and 'limited support for reusability' (R6) are marked for level-I during the first iteration. The responses that are placed at the second level of the ISM model include 'Insufficient system documentation' (R1), 'limited support for real-time system and large system' (R2), and 'lack of long-term planning' (R8). These were marked during the second iteration. After finishing the 3rd iteration 'lack of customers' presence' (R4) is the only risk factor placed at the III level. During the 4th iteration, 'lack of formal communication' (R5) and limited support for reusability (R6) are placed at level IV. Similarly, 'Insufficient system documentation' (R1), 'management overhead' (R3), and 'limited support for reusability' (R6) were marked during iteration seventh while 'insufficient knowledge of the customer' (R7) and 'lack of long-term planning' (R8) were marked during the last iteration, as shown in Table 11.

Formation of a Conical Matrix (CM) and Development of Digraph:

A conical matrix is obtained by arranging RFs by the same level in the columns and rows of the final RM. The CM assists in developing the structural model and this is presented in table 12 below. CM will be used in connection with the generation of the digraph. As can be seen in this step transitive links have been omitted. The initial digraph is constructed with transitivity from which, a final digraph can be derived by eliminating the transitive edges. The last digraph is presented in figure 5. This digraph will be finally converted into the ISM-based green-agile sustainability framework model.

Formation of the ISM-Based Structural Model:

ISM for RFs was arrived at from the developed digraph in Figure 5 by replacing the nodes of the graph with a verbal statement in the respective risk factor. The structural model presents the interconnection between the RFs. If a relationship between RFs b and a exists then it can be found by an arrow from a to b, respectively.

The developed ISM model was verified and scrutinized for any kind of conceptual conflict. This was seen in the result where the various RFs are grouped into eleven levels in the hierarchy model. A hierarchical system aims at 'Limited support for reusability' (R6) and 'lack of long-term planning' (B3) and is positioned at the top level that depends on 'limited support for real-time system and large systems' (B6). This risk factor at level II is realized by the lower level risk factor 'lack of'. Customer's presence (R). The only risk factor at the fourth level is 'lack of formal communication' (B23). It is directly influenced by 'insufficient system documentation' (B34) and 'management overhead' (B11). Both RFs can be achieved through 'insufficient knowledge of the customer' (R4). Figure 6 and Figure 7 demonstrate a distribution of the RFs into various hierarchical levels.

MICMAC analysis techniques (RQ4):

The RFs were classified into four groups – MICMAC technique [59]. The categories are explained as follows:

Autonomous: Where both the driving and dependence power of RFs are weak, they would be regarded as autonomous RFs. These are comparatively less associated with the rest. These RFs are presented in Quadrant-I.

Dependent: RFs with high dependence power but low driving power were regarded as dependent RFs. These RFs are described in the Quadrant-II.

Linkage: Those RFs that had both driving and dependence power to a great extent were considered linkage RFs. Such RFs are on the one hand modified by the lower-level RFs of the model and on the other hand, they impact the large number of other RFs in the model. These RFs are best described in Quadrant-III.

Independent: The RFs having high driving power but low dependence power were regarded as independent RFs. These RFs are described in Quadrant-IV. In figure 10, the cluster power matrix of the RFs according to the dependence and driving power is demonstrated. The final iteration level of each risk factor is presented in Table 12 below.

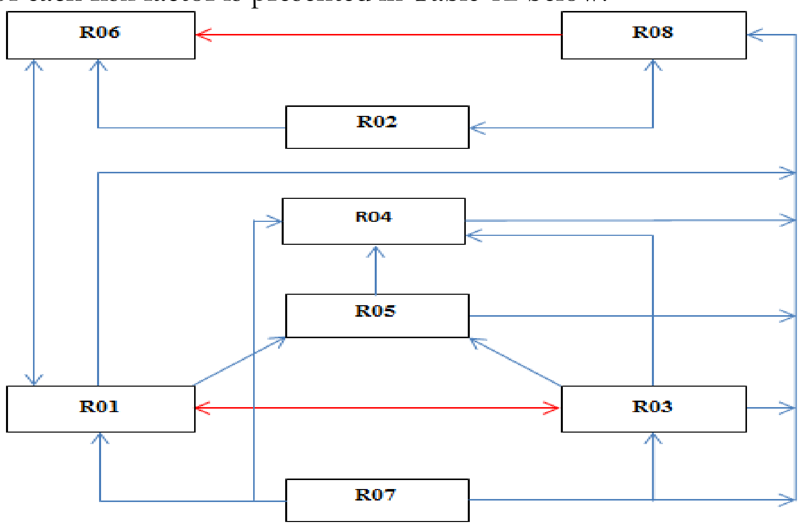

Figure 5. Digraph model for the risk factors

In the above figure, the red color arrow represents the association at the same level while the sky color arrow represents the association with other levels. In response to RQ3, the contextual relationship among the RFs is modeled by developing the ISM model, as shown in Figure 10. The ISM model mixes the green-agile RFs in a hierarchy of six distinct levels with horizontal and vertical arrows showing the interrelationship among the RFs. The top RFs are those that have high dependence power but low driving power in the canonical matrix (Table 12). It means these RFs are affected by most of the RFs, but it does not affect any risk factor except those at the same level. On the other hand, the bottom RFs have high driving power and low dependence power in the canonical matrix (Table 12). It means these RFs help in achieving theoretically all risk factors, but it does not depend on any risk factor except those at the same level.

As shown in Figure 7, at level VI (from the bottom), the risk factor R07, "Insufficient knowledge of the customer," has a driving power of 5 in the conical matrix (Table 12). This indicates that the risk factor contributes to the achievement of other theoretical RFs but is not influenced by any other risk factor. If left unaddressed, it becomes challenging to address all the factors that rely on it. Level V has two RFs R01 ("Insufficient system's documentation") and R03 ("Management overhead"). These RFs have dependence powers of 4 and 5, respectively. For example, R01 and R03 are both dependent on R07, indicating that most issues arise from a lack of formal communication.

Additionally, R01 is influenced by R06 and R03, while R03 is affected by R04 and R01. This suggests that "insufficient system documentation" and "management overhead" are primarily caused by "insufficient knowledge of the customer." Level IV has only one risk factor i.e. R05 "Lack of formal communication", which depends on R01 and R03. Further level III consists of only one risk factor i.e. R04 "Lack of customer's presence", which is directly dependent on R05 and R07.

Level II consists of one risk factor i.e. R02 "Limited support for real-time systems and large systems". This risk factor is dependent on R08 i.e. "Lack of long-term planning", which means that if we have to develop real-time and large systems the organization must have proper and long-term planning.

In last, level I consists of two RFs i.e. R06 "Limited support for reusability" and R08 "lack of long-term planning", having driving power of 1 in a conical matrix, as shown in Table 12. R06 depends on R01, R08 and R02. Similarly, R08 depends on R02, R01, R04, R05, R03 and R07 as well. The risk factor is quite integral and weak to survive as it depends on several RFs to be addressed while developing GSS through agile methods in GSD.

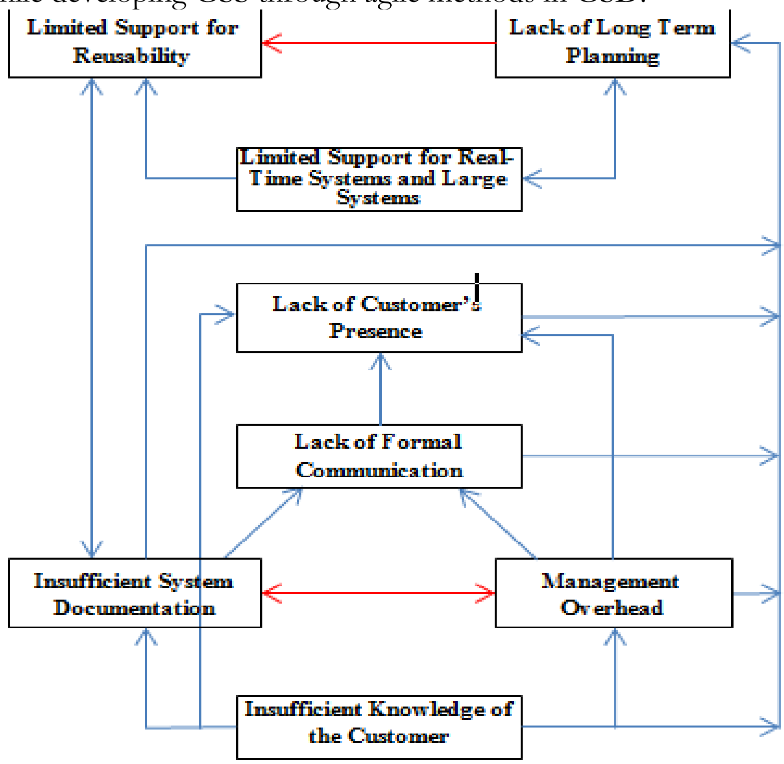

Figure 6. ISM-based Risk factors prioritization model

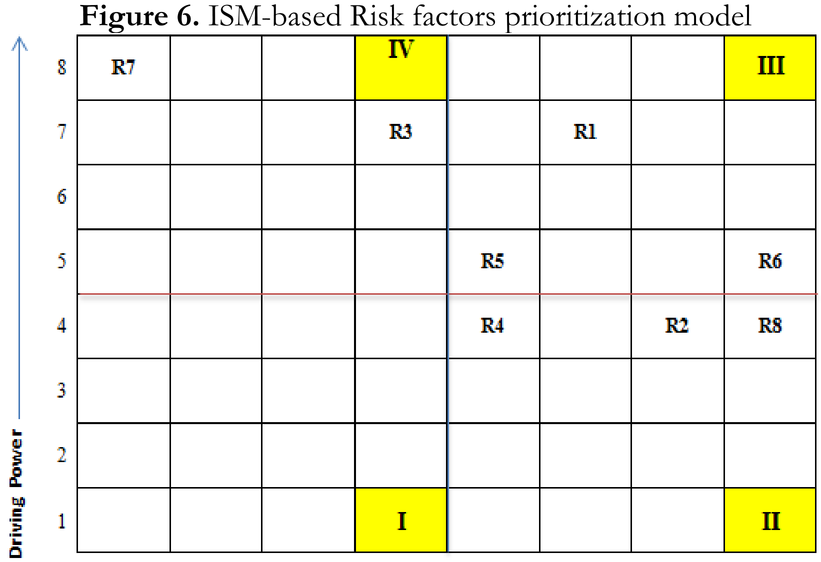

Figure 7. Representation of cluster power diagram for MICMAC analysis

For RQ5, the dispersion of the RFs is addressed by performing MICMAC analysis. For MICMAC analysis, the RFs are distributed into four Quadrants as illustrated in the dependence – driving power matrix (Figure 7).

Quadrant-I: The RFs under Quadrant-I have weaker driving and dependence power and, therefore are known as autonomous RFs. It is clear from Figure 7 that only no risk factor falls on the extreme ends of the autonomous quadrant.

Quadrant-II: The RFs under Quadrant-II have very weak driving power and are known as dependent RFs. In this study, three RFs such as R2 ('limited support for real-time systems and large systems'), R4 ('lack of customer's presence'), and R8 ('lack of long-term planning') are marked as dependent. These RFs are the ones that are influenced by independent RFs.

Quadrant-III: The barriers under Quadrant-II have a strong dependence as driving power and are called linkage barriers in the MICMAC exploration. In this study, barriers such as R1 ('insufficient system's documentation'), R5 ('lack of formal communication'), and R6 ('limited support for reusability') are put in the linkage category.

In the above figure, levels show priority such that level 1 represents high priority while level 10 shows lower priority. The risk factors with high dependence and low driving power are put at the top. Driving risk can achieve dependent risk. Top lever risk(s) are achieved by all others, but it cannot achieve any risk except at the same level if any.

Quadrant-IV: The risk factor under Quadrant-IV has strong driving but weak dependence power and is known as an independent risk factor. In this study, only one risk factor is found independent i.e. R3 ('management overhead').

Limitations:

The following are the limitations of the research methods that have been used in this study and should be noted. The internal validity threat is that some of the reported RFs that have been described in some of the articles could not have explained the causes accurately. In this case, we are not able to contain such menace. The authors of such research articles did not mention the intrinsic reasons.

This could also mean that some studies are predisposed to find out only a specific set of RFs. Most of the chosen articles were case studies, experience reports, and some empirical studies that could be publication-biased.

We only restricted our search to five research venues. In addition, because of a higher number of publications in the field in recent years, there is a possibility that some of the relevant articles were not included when compiling the results of the SLR. However, it is not a systematic omission as described in the literature [39][75]. Nevertheless, we believe that our presented findings (RFs) are based on the extraction from the most relevant published literature.

In the second stage, we carried out an online questionnaire survey in the GSD industry to validate the SLR results, and to find out new factor (s) in addition to the identified list. In this regard, we got 106 questionnaires as the final sample from 25 different countries. It was better that we could have invited more GSD practitioners, but it seemed quite difficult due to limited resources and time constraints. Due to limited international informants in a questionnaire survey, one should be vigilant when generalizing the results.

We selected a questionnaire as a tool to get practitioners' perspectives on the RFs. On major drawback of this method is that respondents are provided with a list of factors, which tends to confiscate the respondents. We tried to cope with this issue by supporting the survey informants to provide additional factors, other than the presented list. However, like other researchers [73][76][77][78], we are also fully confident in our findings because we acquired the data from GSD experts with disparate roles and active involvement in agile GSD projects.

Conclusion:

This study demonstrates that green and sustainable practices must be integrated with agile software development as GSD becomes more prevalent in modern development practices. An SLR analysis helped us identify 8 major RFs that GSD vendors need to address properly to achieve success with agile methods for delivering green and sustainable software. A survey of 106 GSD experts from 25 countries validated the importance of these RFs and revealed some variations through two-proportion Z-test statistical analysis. The interpretive structural modeling framework developed offered a structured mapping of the RFs interdependencies which presented important insights to help practitioners.

Research findings demonstrate that software development requires an equilibrium between agility and sustainability to maintain both efficient user-oriented development and environmental responsibility. The survey results match the findings from the SLR review thus demonstrating the significance of recognized risks in practical software development scenarios for future research development and practical applications.

Future research should focus on the integration of Artificial Intelligence (AI) and Machine Learning (ML) approaches to bolster risk identification and risk mitigation for agile methodologies in GSD environments. Further, the use of AI and ML can produce predictive models that analyze real-time project data for RFs detection while enabling proactive sustainable software development decision-making.

References:

[1] M. Kuhrmann *et al.*, "What Makes Agile Software Development Agile?," *IEEE Trans. Softw. Eng.*, vol. 48, no. 9, pp. 3523–3539, Sep. 2022, doi: 10.1109/TSE.2021.3099532.

[2] A. M. & Y. I. Alzoubi, "Structured software development versus agile software development: a comparative analysis," *Int. J. Syst. Assur. Eng. Manag.*, vol. 14, pp. 1504–1522, 2023, doi: https://doi.org/10.1007/s13198-023-01958-5.

[3] M. P. Adam Alami, Oliver Krancher, "The journey to technical excellence in agile software development," *Inf. Softw. Technol.*, vol. 150, p. 106959, 2022, doi: https://doi.org/10.1016/j.infsof.2022.106959.

[4] J. Sheffield and J. Lemétayer, "Factors associated with the software development agility of successful projects," *Int. J. Proj. Manag.*, vol. 31, no. 3, pp. 459–472, Apr. 2013, doi: 10.1016/J.IJPROMAN.2012.09.011.

[5] I. L. I. Alina-Mădălina GHEORGHE, Ileana Daniela GHEORGHE, "Agile Software Development," *Inform. Econ.*, vol. 24, no. 2, 2020, doi: 10.24818/issn14531305/24.2.2020.08.

[6] K.-J. S. Ronnie de Souza Santos, Paul Ralph, Arham Arshad, "Distributed Scrum: A Case Meta-Analysis," *ACM Comput. Surv.*, vol. 56, no. 4, 2023, doi: 10.1145/3626519.

[7] K. Duehr, P. Efremov, J. Heimicke, E. M. Teitz, F. Ort, and A. Weissenberger-Eibl, Marion, Albers, "THE POSITIVE IMPACT OF AGILE RETROSPECTIVES ON THE COLLABORATION OF DISTRIBUTED DEVELOPMENT TEAMS – A PRACTICAL APPROACH ON THE EXAMPLE OF BOSCH ENGINEERING GMBH," *Proc. Des. Soc.*, vol. 1, pp. 3071–3080, 2021, doi: 10.1017/pds.2021.568.

[8] M. Dick, J. Drangmeister, E. Kern, and S. Naumann, "Green software engineering with agile methods," *2013 2nd Int. Work. Green Sustain. Software, GREENS 2013 - Proc.*, pp. 78–85, 2013, doi: 10.1109/GREENS.2013.6606425.

[9] M. D. Eva Kern, Stefan Naumann, Achim Guldner, "Green software and green software engineering - definitions, measurements, and quality aspects," *Conf. First Int. Conf. Inf. Commun. Sustain.*, 2013, [Online]. Available: https://www.researchgate.net/publication/272086338_Green_software_and_green_software_engineering_-_definitions_measurements_and_quality_aspects

[10] L. O. F. Ugochukwu Okwudili Matthew, Olasubomi Asuni, "Green Software Engineering Development Paradigm: An Approach to a Sustainable Renewable Energy Future," *Adv. Softw. Eng. Through AI, Fed. Learn. Large Lang. Model.*, vol. 14, pp. 281–29, 2024, doi: 10.4018/979-8-3693-3502-4.ch018.

[11] T. N. Nchofoung and S. A. Asongu, "ICT for sustainable development: Global comparative evidence of globalisation thresholds," *Telecomm. Policy*, vol. 46, no. 5, p. 102296, Jun. 2022, doi: 10.1016/J.TELPOL.2021.102296.

[12] C. C. Venters *et al.*, "Sustainable software engineering: Reflections on advances in research and practice," *Inf. Softw. Technol.*, vol. 164, p. 107316, Dec. 2023, doi: 10.1016/J.INFSOF.2023.107316.

[13] "(PDF) GREEN SOFTWARE ENGINEERING PROCESS : MOVING TOWARDS SUSTAINABLE SOFTWARE PRODUCT DESIGN." Accessed: Feb. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/276146350_GREEN_SOFTWARE_ENGINEERING_PROCESS_MOVING_TOWARDS_SUSTAINABLE_SOFTWARE_PRODUCT_DESIGN

[14] L. Ardito, G. Procaccianti, M. Torchiano, and A. Vetrò, "Understanding green software development: A conceptual framework," *IT Prof.*, vol. 17, no. 1, pp. 44–50, Jan. 2015, doi: 10.1109/MITP.2015.16.

[15] "A green model for sustainable software engineering." Accessed: Feb. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/285762532_A_green_model_for_sustainable_software_engineering

[16] S. Naumann, E. Kern, M. Dick, and T. Johann, "Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects," *Adv. Intell. Syst. Comput.*, vol. 310, pp. 191–205, 2015, doi: 10.1007/978-3-319-09228-7_11.

[17] S. Wang, H. Chen, and W. Shi, "SPAN: A software power analyzer for multicore computer systems," *Sustain. Comput. Informatics Syst.*, vol. 1, no. 1, pp. 23–34, Mar. 2011, doi: 10.1016/J.SUSCOM.2010.10.002.

[18] "GreenRM: Reference model for sustainable software development | Request PDF." Accessed: Feb. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/290559298_GreenRM_Reference_model_for_sustainable_software_development

[19] F. Albertao, "Sustainable software development," *Harnessing Green IT*, pp. 63–83, 2012.

[20] "Optimization of Operating Systems towards Green Computing | Request PDF." Accessed: Feb. 02, 2025. [Online]. Available: https://www.researchgate.net/publication/220651916_Optimization_of_Operating_Systems_towards_Green_Computing

[21] C. T. D. Lo and K. Qian, "Green computing methodology for next generation computing scientists," *Proc. - Int. Comput. Softw. Appl. Conf.*, pp. 250–251, 2010, doi: 10.1109/COMPSAC.2010.31.

[22] K. Erdélyi, "Special factors of development of green software supporting eco sustainability," *SISY 2013 - IEEE 11th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 337–340, 2013, doi: 10.1109/SISY.2013.6662597.

[23] B. Nikolic, J. Bachrach, E. Alon, K. Asanovic, and D. Patterson, "Specialization for energy efficiency using agile development," *2015 4th Berkeley Symp. Energy Effic. Electron. Syst. E3S 2015 - Proc.*, Nov. 2015, doi: 10.1109/E3S.2015.7336777.

[24] S. Aguado, R. Alvarez, and R. Domingo, "Model of efficient and sustainable improvements in a lean production system through processes of environmental innovation," *J. Clean. Prod.*, vol. 47, pp. 141–148, May 2013, doi: 10.1016/J.JCLEPRO.2012.11.048.

[25] D. Spinellis, "Sustainable Software Development: An Agile Perspective," *Addison-Wesley Prof.*, vol. 4, no. 10, p. 49, 2005, doi: https://doi.org/10.1145/1189276.1189292.

[26] S. A. Mcxin Tee, Rusli Abdullah, "A Systematic Literature Review of Green Software Development in Collaborative Knowledge Management Environment," *Study Pract.*, 2015, doi: 10.18535/ijact.

[27] S. Jalali and C. Wohlin, "Agile practices in global software engineering - A systematic

map," *Proc. - 5th Int. Conf. Glob. Softw. Eng. ICGSE 2010*, pp. 45–54, 2010, doi: 10.1109/ICGSE.2010.14.
[28] A. Danait, "Agile offshore techniques - A case study," *Proc. - Agil. Confernce 2005*, vol. 2005, pp. 214–217, 2005, doi: 10.1109/ADC.2005.9.
[29] L. Ahuja and R. Priyadarshi, "Agile approaches towards Global Software Engineering," *2015 4th Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2015*, Dec. 2015, doi: 10.1109/ICRITO.2015.7359249.
[30] M. N. REHAN AKBAR, MUHAMMAD HARIS, "Agile Framework for Globally Distributed Development Environment (The DAD Model)," *8th WSEAS Int. Conf. Appl. INFORMATICS Commun.*, pp. 20–22, 2008, [Online]. Available: https://www.researchgate.net/profile/Rehan-Akbar-2/publication/228745285_Agile_framework_for_globally_distributed_development_environment_the_DAD_model/links/5ae15cdf458515c60f65fe2c/Agile-framework-for-globally-distributed-development-environment-the-DAD
[31] C.-T. C. Hsieh C.-Y., "Patterns for Continuous Integration Builds in Cross-Platform Agile Software Development," *J. Inf. Sci. Eng.*, vol. 31, no. 3, pp. 897–924, 2015, [Online]. Available: https://www.researchgate.net/publication/283521886_Patterns_for_Continuous_Integration_Builds_in_Cross-Platform_Agile_Software_Development
[32] N. and S. U. K. Rashid, "Green-agile maturity model (GAMM) for global software development (GSD) vendors," *Sci. Int.*, vol. 26, no. 5, pp. 2041–2043, 2014.
[33] M. Ilyas and S. U. Khan, "Software integration in global software development: Challenges for GSD vendors," *J. Softw. Evol. Process*, vol. 29, no. 8, p. e1875, Aug. 2017, doi: 10.1002/SMR.1875.
[34] A. A. Khan, J. Keung, M. Niazi, S. Hussain, and A. Ahmad, "Systematic literature review and empirical investigation of barriers to process improvement in global software development: Client–vendor perspective," *Inf. Softw. Technol.*, vol. 87, pp. 180–205, Jul. 2017, doi: 10.1016/J.INFSOF.2017.03.006.
[35] M. Niazi *et al.*, "Challenges of project management in global software development: A client-vendor analysis," *Inf. Softw. Technol.*, vol. 80, pp. 1–19, Dec. 2016, doi: 10.1016/J.INFSOF.2016.08.002.
[36] P. Mikalef, I. O. Pappas, J. Krogstie, and M. Giannakos, "Big data analytics capabilities: a systematic literature review and research agenda," *Inf. Syst. E-bus. Manag.*, vol. 16, no. 3, pp. 547–578, Aug. 2018, doi: 10.1007/S10257-017-0362-Y/METRICS.
[37] R. Jabangwe, H. Edison, and A. N. Duc, "Software engineering process models for mobile app development: A systematic literature review," *J. Syst. Softw.*, vol. 145, pp. 98–111, Nov. 2018, doi: 10.1016/J.JSS.2018.08.028.
[38] G. Carrera-Rivera, A., Ochoa, W., Larrinaga, F., & Lasa, "How-to conduct a systematic literature review: A quick guide for computer science research," *MethodsX*, vol. 9, no. 1, p. 101895, 2022, doi: 10.1016/j.mex.2022.101895.
[39] S. U. Khan, M. Niazi, and R. Ahmad, "Factors influencing clients in the selection of offshore software outsourcing vendors: An exploratory study using a systematic literature review," *J. Syst. Softw.*, vol. 84, no. 4, pp. 686–699, Apr. 2011, doi: 10.1016/J.JSS.2010.12.010.
[40] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Human Behav.*, vol. 51, pp. 915–929, Oct. 2015, doi: 10.1016/J.CHB.2014.10.046.
[41] S. Mahmood, S. Anwer, M. Niazi, M. Alshayeb, and I. Richardson, "Identifying the factors that influence task allocation in global software development: Preliminary results," *ACM Int. Conf. Proceeding Ser.*, vol. 27-29-April-2015, Apr. 2015, doi:

10.1145/2745802.2745831.
[42] N. M. Mohammed, M. Niazi, M. Alshayeb, and S. Mahmood, "Exploring software security approaches in software development lifecycle: A systematic mapping study," *Comput. Stand. Interfaces*, vol. 50, pp. 107–115, Feb. 2017, doi: 10.1016/J.CSI.2016.10.001.
[43] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Inf. Softw. Technol.*, vol. 58, pp. 148–169, Feb. 2015, doi: 10.1016/J.INFSOF.2014.06.011.
[44] M. Niazi, "A comparative study of software process improvement implementation success factors," *J. Softw. Evol. Process*, vol. 27, no. 9, pp. 700–722, Sep. 2015, doi: 10.1002/SMR.1704.
[45] M. Niazi, "Do Systematic Literature Reviews Outperform Informal Literature Reviews in the Software Engineering Domain? An Initial Case Study," *Arab. J. Sci. Eng.*, vol. 40, no. 3, pp. 845–855, Mar. 2015, doi: 10.1007/S13369-015-1586-0/METRICS.
[46] M. Niazi, S. Mahmood, M. Alshayeb, A. M. Qureshi, K. Faisal, and N. Cerpa, "Toward successful project management in global software development," *Int. J. Proj. Manag.*, vol. 34, no. 8, pp. 1553–1567, Nov. 2016, doi: 10.1016/J.IJPROMAN.2016.08.008.
[47] S. and S. U. K. Ali, "Systematic literature review protocol for software outsourcing partnership (SOP).," *IOSR J. Comput. Eng.*, vol. 2, 2012, doi: 10.9790/0661-0210818.
[48] and A. U. A. Yaseen, M., S.U. Khan, "SOFTWARE MULTI-SOURCING RISKS MANAGEMENT FROM VENDOR'S PERSPECTIVE: A SYSTEMATIC LITERATURE REVIEW PROTOCOL," *Gomal Univ. J. Res.*, vol. 29, no. 2, 2013, [Online]. Available: http://www.gujr.com.pk/index.php/GUJR/article/view/238
[49] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009, doi: 10.1016/j.infsof.2008.09.009.
[50] S. U. Khan, M. Niazi, and R. Ahmad, "Empirical investigation of success factors for offshore software development outsourcing vendors," *IET Softw.*, vol. 6, no. 1, pp. 1–15, Feb. 2012, doi: 10.1049/IET-SEN.2010.0038.
[51] M. M. and S. L. Ahmed, "A Systematic Literature Review on Challenges in Service Oriented Software Engineering," *Int. J. Softw. Eng. Its Appl.*, vol. 9, no. 6, pp. 173–186, 2015.
[52] V. Raci and R. Shankar, "Analysis of interactions among the barriers of reverse logistics," *Technol. Forecast. Soc. Change*, vol. 72, no. 8, pp. 1011–1029, Oct. 2005, doi: 10.1016/J.TECHFORE.2004.07.002.
[53] D. W. Malone, "An Introduction to the Application of Interpretive Structural Modeling," *Proc. IEEE*, vol. 63, no. 3, pp. 397–404, 1975, doi: 10.1109/PROC.1975.9765.
[54] M. D. Singh, R. Shankar, R. Narain, and A. Agarwal, "An interpretive structural modeling of knowledge management in engineering industries," *J. Adv. Manag. Res.*, vol. 1, no. 1, pp. 28–40, Oct. 2003, doi: 10.1108/97279810380000356/FULL/XML.
[55] R. W. Hawthorne and A. P. Sage, "On applications of interpretive structural modeling to higher education program planning," *Socioecon. Plann. Sci.*, vol. 9, no. 1, pp. 31–43, Feb. 1975, doi: 10.1016/0038-0121(75)90039-7.
[56] G. Kannan and A. N. Haq, "Analysis of interactions of criteria and sub-criteria for the selection of supplier in the built-in-order supply chain environment," *Int. J. Prod. Res.*, vol. 45, no. 17, pp. 3831–3852, Sep. 2007, doi: 10.1080/00207540600676676.
[57] V. R. Pramod and D. K. Banwet, "ISM for understanding the enablers of telecom service supply chain," *Int. J. Bus. Excell.*, vol. 8, no. 5, pp. 537–565, 2015, doi: 10.1504/IJBEX.2015.071277.

[58] L. Shen, X. Song, Y. Wu, S. Liao, and X. Zhang, "Interpretive Structural Modeling based factor analysis on the implementation of Emission Trading System in the Chinese building sector," *J. Clean. Prod.*, vol. 127, pp. 214–227, Jul. 2016, doi: 10.1016/J.JCLEPRO.2016.03.151.

[59] A. Kumar and G. Dixit, "An analysis of barriers affecting the implementation of e-waste management practices in India: A novel ISM-DEMATEL approach," *Sustain. Prod. Consum.*, vol. 14, pp. 36–52, Apr. 2018, doi: 10.1016/J.SPC.2018.01.002.

[60] S. Ali and S. U. Khan, "Critical success factors for software outsourcing partnership (SOP): A systematic literature review," *Proc. - 2014 IEEE 9th Int. Conf. Glob. Softw. Eng. ICGSE 2014*, pp. 153–162, Oct. 2014, doi: 10.1109/ICGSE.2014.12.

[61] S. U. Khan and M. Niazi, "Critical Challenges in Offshore Software Development Outsourcing: An Empirical Study," *Softw. Eng. / 781 Control Appl.*, pp. 29–36, Jul. 2012, doi: 10.2316/P.2012.780-019.

[62] M. A. K. M. Mohankumar, "Green based Software Development Life Cycle Model for Software Engineering," *Indian J. Sci. Technol.*, vol. 9, no. 32, pp. 1–8, 2016, doi: 10.17485/ijst/2016/v9i32/88499.

[63] D. Chang, C. K. M. Lee, and C. H. Chen, "Review of life cycle assessment towards sustainable product development," *J. Clean. Prod.*, vol. 83, pp. 48–60, Nov. 2014, doi: 10.1016/J.JCLEPRO.2014.07.050.

[64] A. Eman, "Impact of Agile Methodology on Software Development," *Comput. Inf. Sci.*, vol. 8, no. 2, pp. 9–9, 2024, doi: 10.5539/cis.v8n2p9.

[65] C. Calero, M. Á. Moraga, M. F. Bertoa, and L. Duboc, "Green Software and Software Quality," *Green Softw. Eng.*, pp. 231–260, Jan. 2015, doi: 10.1007/978-3-319-08581-4_10.

[66] N. S. Venters, C., C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, G. Rodriguez-Navas, "Mind the Gap: Bridging the Sustainable Software Systems," *Res. Divid.*, 2015.

[67] D. Kemp, K. Akroyd-Wallis, I. Mactaggart, R. Evans, K. Harrison, and H. Woodcock, "'Fifty Shades of Agile': AN ANALYSIS OF DIFFERENT PERSPECTIVES OF AGILE SYSTEMS ENGINEERING," *INCOSE Int. Symp.*, vol. 26, no. 1, pp. 691–712, Jul. 2016, doi: 10.1002/J.2334-5837.2016.00187.X.

[68] A. Q. G. Jovita Vasauskaite, "Rethinking enterprise architecture for sustainable energy system development," *J. Electron. Sci. Technol.*, 2015, doi: 10.11989/JEST.1674-862X.505151.

[69] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Stahl, "The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at Ericsson," *Int. Symp. Empir. Softw. Eng. Meas.*, pp. 348–356, 2013, doi: 10.1109/ESEM.2013.53.

[70] N. B. M. Dingsøyr, T., "Towards principles of large-scale agile development," *Int. Conf. Agil. Softw. Dev.*, 2014.

[71] B. Fitzgerald and K. J. Stol, "Continuous software engineering and beyond: Trends and challenges," *1st Int. Work. Rapid Contin. Softw. Eng. RCoSE 2014 - Proc.*, pp. 1–9, Jun. 2014, doi: 10.1145/2593812.2593813.

[72] M. Korkala and F. Maurer, "Waste identification as the means for improving communication in globally distributed agile software development," *J. Syst. Softw.*, vol. 95, pp. 122–140, Sep. 2014, doi: 10.1016/J.JSS.2014.03.080.

[73] Y. I. Alzoubi, A. Q. Gill, and A. Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review," *Inf. Manag.*, vol. 53, no. 1, pp. 22–37, Jan. 2016, doi: 10.1016/J.IM.2015.08.003.

[74] K. D. Batra, D., W. Xia, D. VanderMeer, "Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects:

A Case Study from the Cruise Line Industry," *Commun. Assoc. Inf. Syst.*, vol. 27, 2010, [Online]. Available: https://aisel.aisnet.org/cais/vol27/iss1/21/

[75] E. Hossain, M. Ali Babar, and H. Y. Paik, "Using scrum in global software development: A systematic literature review," *Proc. - 2009 4th IEEE Int. Conf. Glob. Softw. Eng. ICGSE 2009*, pp. 175–184, 2009, doi: 10.1109/ICGSE.2009.25.

[76] N. Baddoo and T. Hall, "Motivators of Software Process Improvement: an analysis of practitioners' views," *J. Syst. Softw.*, vol. 62, no. 2, pp. 85–96, May 2002, doi: 10.1016/S0164-1212(01)00125-X.

[77] A. R. Beecham, S., T. Hall, "Software process improvement problems in twelve software companies: An empirical analysis," *Empir. Softw. Eng.*, vol. 8, no. 1, pp. 7–42, 2003.

[78] R. Giuffrida and Y. Dittrich, "Empirical studies on the use of social software in global software development – A systematic mapping study," *Inf. Softw. Technol.*, vol. 55, no. 7, pp. 1143–1164, Jul. 2013, doi: 10.1016/J.INFSOF.2013.01.004.