Proceedings of the Pakistan Academy of Sciences 52 (4): 301–313 (2015)

ISSN: 0377 - 2969 (print), 2306 - 1448 (online)

Pakistan Academy of Sciences

Research Article

Green Agility for Global Software Development Vendors: A Systematic Literature Review Protocol

Nasir Rashid, and Siffat Ullah Khan*

Department of Computer Science & Information Technology,
University of Malakand, Lower Dir, Pakistan

Abstract: Global software development (GSD) is now-a-days pervasive in software industry aiming to develop global standard software through geographically distributed skilled teams in minimum time and cost. In order to meet the demand for green software production and frequent changes in requirements of the clients, GSD developers have revamped traditional methods and trying to incorporate green principles with agile methods for rapid and energy efficient software development. This paper presents our contribution to building a systematic literature review (SLR) protocol for green agile maturity (GAM) for GSD vendor organizations. The protocol aims to systematically review the available literature for the identification of success/risk factors that may have a direct or indirect effect on green and sustainable software development using agile methods. The desired outcome of SLR protocol will be a group of success/risk factors and their concerned practices that will be helpful for vendors to produce green and environmentally sustainable software by incorporating agile principles in global software development.

Keywords: Green software, sustainable software, green agile, global software development, SLR Protocol.

1. INTRODUCTION

Global software development (GSD) is growing rapidly due to increase in globalization of software business industry [1]. In GSD, software engineers and developers from various countries with different cultures and time zones participate in the development process. Distributed experts at diverse locations coordinate through the latest knowledge sharing and communication tools [2]. GSD offers tremendous benefits that include access to skilled pool of software developers, production of high standard software, business advantage of proximity to markets, quick access to software development updates and the possibility to use "follow-the-sun" and "round–the-clock" development. Hence software development is now considered as a globally distributed endeavour [3-4].

However, GSD unlocks new doors for software business yet it also yields a number of challenges that comprises hidden agreement costs, dearth of client involvement, splitting and allocation of work at different sites, lack of trust among the outsourcing companies and scarcity of software development outsourcing practices [5-6].

Agile software development is invigorating approach towards quick and interactive software development. It provides a conceptual structure for undertaking any software project that is co-located or globally distributed. Unlike traditional methods of software development, agile methods attempt to reduce risks and maximize software productivity by developing software in short iterations [7]. Agile approaches rely on individual developers' skills rather than formalized processes and cumbersome amount of documentation [8]. Thus, agile methods pursue to avoid suggesting overwhelming processes, having little contribution to software product [9].

Using agile methods in distributed software development offers several benefits like constant communications and scheduled delivery of software, continuous integration of software code, improved project's quality and efficiency, nominal

Received, June 2015; Accepted, November 2015
*Corresponding author: Siffat Ullah Khan; Email: siffatullah@uom.edu.pk

documentation and early expert customer feedback [10-11]. Green or sustainable software is the design and production of software, having direct or indirect negative effect on country's economy, people, society and environment that result from software pre-development, development and post-development phases are negligible and/or which have a positive impact on sustainable software production [12]. Green software engineering is an emerging paradigm and is growing rapidly that aims to develop software with green features to reduce negative impact on environment.

This research work presents a systematic literature review protocol for the development of green agile maturity model for GSD vendors as well as the preliminary results as shown in Table 5. The findings will contribute to the development of first phase of our proposed model that is aimed to assist GSD vendors to measure their green-agile maturity for the design and production of green and sustainable software [13]. The detailed structure of the proposed model is shown in Fig. 1.

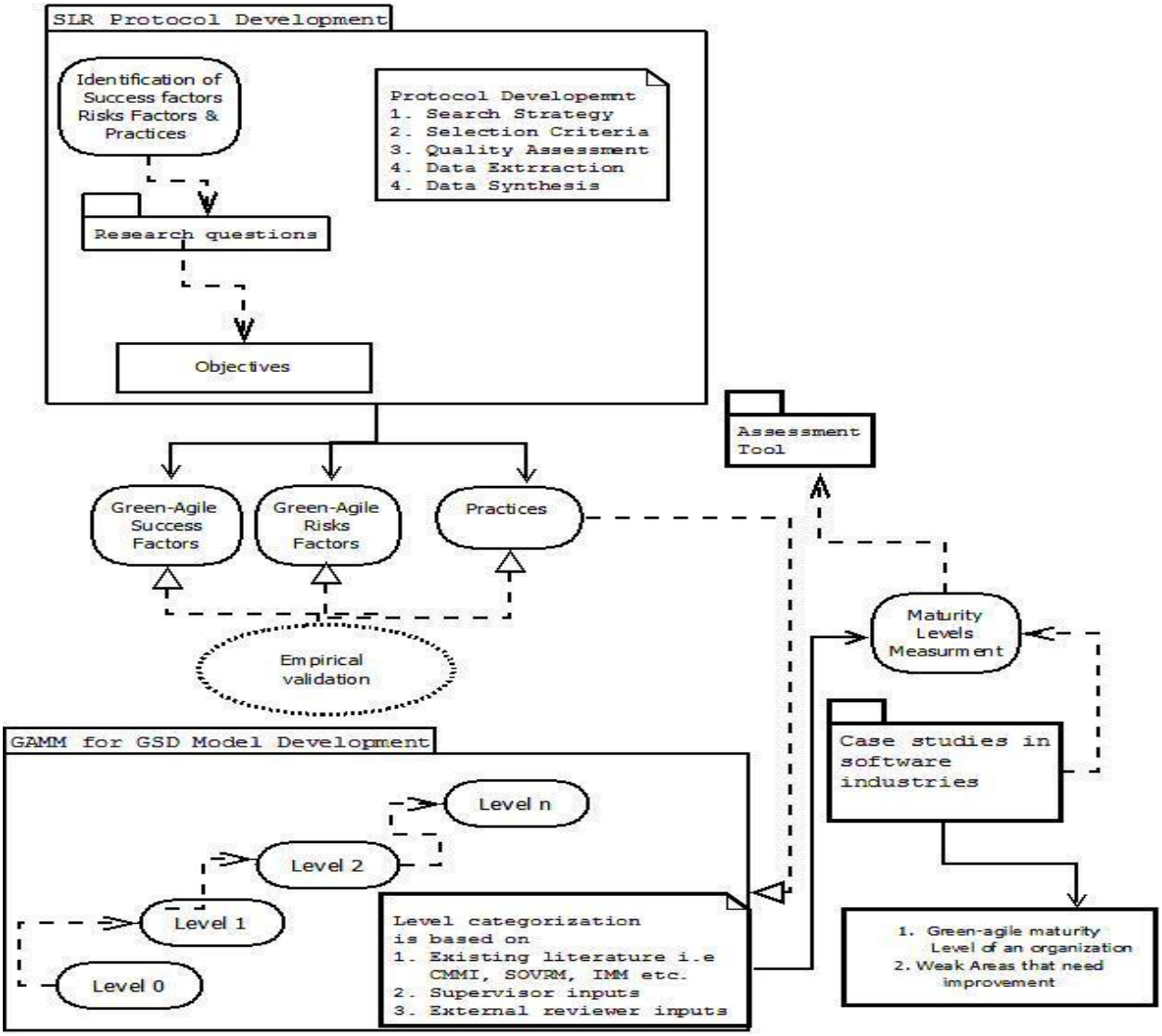

Fig. 1. Overview of proposed model (GAMM for GSD vendors).

The following research questions stimulated the research work presented in this paper:

RQ 1. What are the success factors, as mentioned in the relevant literature, for adapting agile techniques that can assist software engineers in GSD organizations for the design and production of green and sustainable software?

RQ 2. What are the risks involved, as mentioned in the relevant literature, to be avoided by software engineers in GSD organizations for the design and production of green and sustainable software using agile techniques?

RQ 3: What are the relevant practices in GSD organizations, as mentioned in the literature, to be applied by software engineers that can be valuable in the design and production of green and sustainable software using agile techniques?

Table 1. Track 1 search results.

Source	Total publications found
Google Scholar	7
Springerlink	26
ACM	4
IEEEXplore	0
Science Direct	39

Table 2. Track 2 search results.

Source	Total publications found
Google Scholar	350
Springerlink	887
ACM	42
IEEEXplore	70
Science Direct	949

2. BACKGROUND

Research in the area of green-agile is growing. A number of researchers have worked on agile methods with green aspects to promote quick and interactive development of green software. Several researchers have proposed enhanced model of agile methods specifically SCRUM and extreme programming (XP) that integrates green aspects of software engineering for co-located software development teams [14-16].

Agile methods are emerging techniques in software engineering that have reshaped software development life cycle and assure the delivery of sustainable software through iterative and quick development. Agile processes are based on a set of some major principles such as strong team work, close association between practitioners and business organizations, face-to-face meetings with customers, early and frequent delivery of workable product and accepting flexibility towards dynamic requirements from customers [17-20].

Sara et al [21] have proposed a two-level green software model that comprises sustainable software life cycle as well as software tools that claim to produce greener and environment friendly software using agile approaches. The model integrates some of the agile principles that lead to green software development. It consists of different stages that are based on hybrid processes of sequential and agile methods for production of sustainable software. Green guidelines and green processes have also been proposed for each stage of software development. Among the green agile principles flexibility in changing requirements, interactive software development that involves customer, incremental and iterative software development and early testing techniques for defects prevention have been mentioned that aid in promoting green and environmental sustainable software product.

Tate [22] coined the importance of green and sustainable software development through iterative agile development. He suggests that culture of environment friendly software development should incorporate the principles of agile approaches in software development life cycle. According to his work presented, continuous refinement of software development by accepting changes in the requirements any time, scheduled delivery of working code, emphasis on simple design with simple contents, early defect prevention through regular testing are the core principles of agile techniques that can surely add in the eco-friendly software development.

Koontz et a. [23] have worked on re-architecting of software products and have identified the integral principles of agile methods that proved to be helpful in sustainable software development. Out of the identified principles, incremental and quick development reduced working cycle times, iterative development, mini builds with simple designs and continuous

integrations are the ones that can help to produce greener and sustainable software products.

The literature described above listed out some intrinsic principles of agile methods and their importance for the development of green and sustainable software product. However, none of the identified factors and principles has been dug out through systematic literature review. The mentioned work lacks to explore the applicability of agile principles for greener and sustainable software development in global software projects. Our proposed work, systematic literature review protocol for the identification of green agile factors for GSD vendors thus has a significant value for contribution to software engineering domain and is evident that no such effort has been spent before regarding such work.

3. RESEARCH METHODOLOGY

To achieve the ultimate objectives, the research work has been planned in three inter-dependent phases. In first portion of our research work, the success factors, risks and relevant practices, regarding the production of green software through the use of agile approaches will be investigated by means of SLR. SLR is an efficient mechanism for identifying, evaluating and interpreting all currently available research, relevant to particular research questions or area of interest [24]. SLR is now an interesting research methodology used by empiricists for conducting empirical research [25-26]. Though, it is a time consuming method and a bit hard to conduct over other review techniques but the results retrieved are unbiased as it follows a pre-defined and validated protocol [27].

Empirical study has been planned in the second phase of the research to validate the results of systematic literature review, as shown in Table 5, and to explore some new factors, if any, in GSD organizations. Same methodology has been followed by other software engineers and researchers [28-29].

In third phase of our protocol, proposed model (GAMM for GSD vendors) will be developed on the basis of the SLR outcomes and empirical study to be conducted in GSD organizations. We intend to validate the proposed model through five case studies in GSD organizations as well. Same approach has been adopted by other researchers [30].

Table 3. Data extraction form.

S. No.	Data to be extracted
1	Form No.
2	Paper review date
3	Paper tracing No.
4	Title of the paper
5	Author name(s)
6	Reference
7	Paper venue
8	Quality of publication (A/B/C/Other)
9	Country
10	Year (Publication year)
11	Strategy of study (Ordinary review, SLR, empirical study etc.)
12	Agile methodology discussed (XP, Scrum, FDD, Crystal etc)
13	Population (Sample and Target).
14	Company size (Small, Medium, Large)
15	Company type (Software industry, Academic/Research Institute etc)
16	Company scope (Local, Global)
17	Company SPI status (CMMI, ISO etc)
18	Success factors/motivators in the adoption of agile methods for the development of green software specific to GSD or in general context.
19	Challenges/risks in adoption of agile methods for the development of green software specific to GSD or in general context.
20	Practices for green agility specific to GSD or in general context.

4. SYSTEMATIC LITERATURE REVIEW PROTOCOL

Performing a systematic review involves several discrete activities that can be clustered into three major phases: planning, conducting and reporting the review. Systematic reviews deal with the problem of accumulating empirical evidences, obtained using various techniques and in broadly different contexts, used mainly in software engineering domain [31-33].

This section covers the first phase of a systematic review process (review planning).

Table 4. SLR protocol preliminary results.

S. No.	Digital library	Search string	Date constraint	Total publications found	Primary selection	Final selection
1	Science Direct	Track 2	All Years	949	28	09
2	ACM	Track 2	All Years	42	12	04
3	IEEE Xplore	Track 2	All Years	70	14	05
4	Springer Link	Track 2	All Years	887	21	16
5	Google Scholar	Track 2	All Years	350	50	15
			Total	**2298**	**125**	**49**

Publications found through snowballing: 31
Total publications found through SLR: 49
Total publications selected (final): *N*=80

Table 5. Preliminary results of the systematic literature review protocol.

S. No	Success factor	Frequency	Percentage N=80
1	Efficient utilization of time and computing resources	21	26
2	Minimal documentation	25	31
3	Minimal rework/reengineering	18	23
4	Reduced cost	17	21
5	Improved quality	25	31
6	Improved management of product life cycle	09	11
7	E-waste minimization	09	11
8	Simple design	38	48
9	Refactoring	15	19
10	Continuous integration	14	18
11	Standard coding	27	34
12	Early defect prevention	14	18
13	Optimization of processes	05	6
14	Changes requirements anytime/late in development	01	1
15	Agile planning and agile requirements	04	5
16	Fast delivery	23	29
17	Iterative development	31	39
18	Early development	08	10
19	Early testing/continuous validation	16	20
20	Review and preview	08	10
21	Development of small size software components	02	3
22	Efficient collaborations between developers and customers	21	26
23	Energy efficient software development	01	1
24	Efficient coordination among the agile team members.	19	24

*N = Total number of finally selected publications

Identifying Search Terms

Search terms are designed according to the following rules.

a. Find major terms in research questions such as population, intervention and expected outcomes.
b. Find out synonyms for the identified major terms.
c. Verify the major terms along with synonyms in relevant literature.
d. Concatenate the major terms using 'AND' and 'OR' operators to get the final search strings.

Results for (a)

RQ 1 and RQ 2:

Green software, agile methods and global software development, success factors, risks.

Results for (b)

RQ 1 and RQ 2:

Green software: ("green software" OR "greener software" OR "sustainable software" OR "green computing" OR "green IT" OR "green software engineering")

Agile methods: (agile OR "agile methods" OR "green agile" OR "extreme programming" OR scrum)

Global software development: ("global software development" OR "distributed software development")

Results for (c)

RQ 1 and RQ 2:

green software, greener software, sustainable software, green computing, green IT, green software engineering, agile, agile methods, green agile, extreme programming, scrum, global software development, distributed software development.

Results for (d)

RQ 1 and RQ 2:

Track 1: ("green software" OR "greener software" OR "sustainable software" OR "green computing" OR "green IT" OR "green software engineering") AND (agile OR "agile methods" OR "green agile" OR "extreme programming" OR scrum) AND ("global software development" OR "distributed software development")

Track 2: ("green software" OR "greener software" OR "sustainable software" OR "green computing" OR "green IT" OR "green software engineering") AND (agile OR "agile methods" OR "green agile" OR "extreme programming" OR scrum)

Here Track1 signifies the search string being designed to explicitly retrieve available literature relevant to green agility in GSD, where Track 2 will retrieve relevant literature specific to green software development using agile methods. The results retrieved through Track 1 are almost insignificant as shown in Table 1. The tracks described earlier were discussed with experts of software engineering research group at university of Malakand and it was decided to follow Track 2 for the development of protocol, as shown in Table 2.

The identified factors as shown in Table 5 will be validated in GSD organizations through empirical studies. This will provide guidance to GSD vendor organizations to know better how these factors are applicable and adopted by software practitioners in global software development for the development of green and sustainable software using agile methods. Same method has been followed by other researchers in software engineering community [34-38].

5. PUBLICATION SELECTION

5.1. Inclusion Criteria

The following criteria are used to determine which piece of literature found by the search strings will be considered for the data extraction:

a. Studies that describe green software development using agile methods specific to GSD or in general context
b. Studies that describe applicability of agile methods in GSD that can assist towards green and sustainable software development.
c. Studies that describe green software engineering principles that are supported

by agile methods (such as XP, Scrum etc.) specific to GSD or in general context.

d. Studies that describe the agile maturity of GSD vendors that can assist software engineers for the design and development of eco-friendly software.

e. Studies that describe success or risk factors or relevant practices that can add value in the development of green software specific to GSD or in general context.

f. Studies written in English language will be considered only.

5.2 Exclusion Criteria

The following exclusion criteria describe which piece of literature found by the search strings will be excluded:

a. Studies that is not relevant to the research questions.

b. Studies that do not describe green agility.

c. Studies that are not related to green software development using agile techniques.

5.3 Selecting Primary Sources

Primary selection of literature is performed on the basis of title and abstract only. The main purpose is to exclude the irrelevant results against the research questions and problem domain. Selected literature is then carefully reviewed according to the inclusion/exclusion criteria as mentioned above. In case of any confusion regarding inclusion or exclusion principles, secondary reviewer will be contacted for guidance and expert decision.

6. PUBLICATION QUALITY ASSESSMENT

After final selection of the papers, quality and standard of the publications is measured according to the following criteria.

a. Is it clear how green agility is measured in global software development?

b. Is it clear how agile methods are adapted for green software development in GSD or in general context?

c. Is it clear how the adoption of agile methodologies is difficult to be integrated with green software development in GSD?

d. Is it clear how the factors for enhancement and improving green software development with agile methods were identified in outsourcing projects?

Each of the above list items will be marked as 'YES', 'NO', 'Partially' or N.A'.

Publication's quality of selected papers is further analyzed on the basis of below mentioned criteria and only qualifying papers are selected as our final sample size. A similar method has been followed in our earlier research [39].

6.1. Criteria for Category-A Papers

In category-A, we included those papers only which satisfy the following criteria:
C=Case study
I=Interview
S=Survey

1. L=Literature review journal publications (impact factor)
2. Clear methodology
3. Must have sample size as follow;
   a. $C \geq 3$
   b. $I \geq 12$
   c. $S \geq 50$
   d. $L \geq 50$

6.2. Criteria for Category-B Papers

In this category we included only those papers which satisfy the following criteria:

1. Conference publications
2. Clear methodology
3. Must have sample size as follow:
   a. $C = 2$
   b. $5 \leq I \leq 11$
   c. $30 \leq S \leq 49$
   d. $30 \leq L \leq 49$

6.3. Criteria for Category-C Papers

In category-C we included only those papers which satisfy the following criteria:

1. Literature published in less reputed venues (Journal, Conference)
2. Clear methodology
3. Having sample size as follow;
   a. C = 1
   b. I ≤ 5
   c. 1 ≤ S ≤ 29
   d. 1 ≤ L ≤ 29

7. DATA EXTRACTION STRATEGY

In this phase the required data is extracted that answers the above mentioned research questions. Secondary reviewer is approached for necessary guidance and to resolve the ambiguities in data extraction. The secondary reviewer randomly selected research publications and compared the results produced by the primary reviewer in order to ensure the quality of publications. Extraction form as shown in Table 3 is used for data extraction process.

8. DATA SYNTHESIS

Data synthesis is the process of grouping the identified factors from finally selected publications. For research question1, the data will be synthesized in a table that will illustrate the identified factors with frequency. Complete details of each factor will be maintained separately having details (S. No, Factor group name, factor's subgroups and paper tracing number). The same process of data synthesis will be adapted for the Research Question 2.

9. RESULTS AND DISCUSSION

Table 4 depicts primary search results of the protocol. Total number of publications found through Track 2 is 2,298. Out of which 125 papers were initially selected by reading its title and abstract. After removal of the duplicate publications found, we got 118 papers. Finally a total of 80 research publications were selected, as shown in Table 6, out of which 49 have been selected by following SLR protocol and 32 by snowballing respectively. Complete list of the selected papers is shown in Table 6. Snowballing is a distinct search mechanism that uses author name, reference list of a selected paper or its citations to find more papers that may have missed by the SLR search string during its search phase. Guidelines mentioned in [40] have been used for conducting systematic reviews supported by snowballing approach. Furthermore, we have studied a number of papers [41-49] for the design and development of the presented SLR protocol.

Table 6. List of finally selected papers in the systematic literature review.

Final ID	Paper Title	Database
P1	Software evolution for industrial automation systems: Literature overview	Google scholar
P2	Collaboration in mature XP teams	Google scholar
P3	Sustainability guidelines for long-living software systems	Google scholar
P4	Appropriate information system development	Google scholar
P5	A systematic mapping study on sustainable software engineering: A research preview	Google scholar
P6	Toward an XP evaluation framework	Google scholar
P7	MI Copa: Micro credit operation automation	Google scholar
P8	Sustainable software development: An agile perspective	Google scholar
P9	Chaos issues on communication in agile global software development	Google scholar
P10	Green software engineering process : Moving towards sustainable software product design	Google scholar
P11	Enhancing software engineering processes towards sustainable software product design	Google scholar
P12	A green model for sustainable software engineering	Google scholar
P13	Bio mimicry as a super systems metaphor for software engineering?	Google scholar
P14	An IT perspective on integrated environmental modeling: The SIAT case	Google scholar
P15	Modeling to support communication and engineering of service-oriented software	Google scholar

Contd.....

Table 6 (Contd.)

Final ID	Paper Title	Database
P16	Smart green infrastructure for innovation and transformation hosting environments	IEEE
P17	Impact of web 2.0 and cloud computing platform on software engineering	IEEE
P18	GETA for information technology: Go green, eat green, think green, and act green information technology	Google scholar
P19	The agile manifesto	Google scholar
P20	Towards better understanding of agile values in global software development	Google scholar
P21	Review of agile methodologies in software development	Google scholar
P22	Analysis and design of a novel agile methodology	Google scholar
P23	Sustainability in software engineering	Google scholar
P24	Cloud software Finland	IEEE
P25	Processes and practices for quality scientific software projects	
P26	Adopting key lessons from agile manufacturing to agile software product development: A comparative study	Google scholar
P27	Success factors of agile software development	Science direct
P28	Review on traditional and agile cost estimation success factor in software development project	Google scholar
P29	Limitations of agile software processes	IEEE
P30	A survey study of critical success factors in agile software projects	Science direct
P31	Performance evaluation of software development models	Google scholar
P32	The agile software development series	IEEE
P33	Usage and perceptions of agile software development in an industrial context: An exploratory study	IEEE
P34	Agile software development practices: Evolution, principles and criticisms	Google scholar
P35	Agile software development: novel approaches for software engineering	IEEE
P36	What is agile software development?	Springer link
P37	Learning from agile software development	IEEE
P38	Agile methodologies and process discipline	Google scholar
P39	Odyssey and other code science success stories	Google scholar
P40	Agile process for integrated service delivery	IEEE
P41	Role of agile methodology in software development	Google scholar
P42	Human resource planning in agile projects	Google scholar
P43	Extreme programming – agile method used in project management	Google scholar
P44	Using factor analysis to generate clusters of agile practices	IEEE
P45	Scaled agile framework: A blight	Google scholar
P46	Software development methodologies for reducing project risks	Google scholar
P47	Green software engineering with agile methods	IEEE
P48	Common agile practices in software processes	IEEE
P49	Factors influencing the agile methods in practice literature survey & review	IEEE
P50	Introducing agile development practices from the middle	IEEE
P51	Green as the new lean: How to use lean practices as a catalyst to greening your supply chain	IEEE
P52	Supporting distributed extreme programming	Springer link
P53	Outsourcing and offshoring with agility: A case study	Springer link
P54	Agile processes in software engineering and extreme programming	Springer link
P55	Sustainable software: A study of software product sustainable development	Springer link
P56	Agile software construction	Springer link

Contd.....

Table 6 (Contd.)

Final ID	Paper Title	Database
P57	REM in agile projects	Springer link
P58	Assessing software agility: An exploratory case study	Springer link
P59	Comparative analysis of agile maturity model	Springer link
P60	A Failure to Learn in a Software Development Team: The unsuccessful introduction of an agile method	Springer link
P61	Derivation of green metrics for software	Springer link
P62	Aggregated survey of sustainable business models for agile mobile service delivery platforms	Springer link
P63	Agility meets system engineering: A catalog of success factors from industry practice	Springer link
P64	Factors affecting effectiveness of agile usage: Insights from the BBC worldwide case study	Springer link
P65	Obstacles to agile software development	Springer link
P66	Creating environmental awareness in service oriented software engineering	Springer link
P67	Introduction to mechanism design for sustainability	Springer link
P68	Empirical studies of agile software development: A systematic review	Science direct
P69	Progressive outcomes: A framework for maturing in agile software development	Science direct
P70	Risks in distributed agile development: A review	Science direct
P71	Agile software architecture: Aligning agile processes and software architectures	Science direct
P72	Drivers of agile software development use: Dialectic interplay between benefits and hindrances	Science direct
P73	Essential communication practices for extreme programming in a global software development team	Science direct
P74	Review of life cycle assessment towards sustainable product	Science direct
P75	Model of efficient and sustainable improvements in a lean production system through processes of environmental innovation	Science direct
P76	Processes versus people: How should agile software development maturity is defined?	Science direct
P77	A literature review of agile practices and their effect in scientific software development	ACM
P78	Necessary and neglected? An empirical study of internal documentation in agile software development teams	ACM
P79	A systematic literature review of agile software processes and user centered design integration	ACM
P80	Effort estimation in agile software development: A survey on the state of practice	ACM

A list of identified factors through SLR has been shown in Table 5. There are a total of 24 factors, extracted from finally selected papers (N=80). All these factors are considered as green factors of agile methods that can help software development organizations to produce green and sustainable software in GSD. Among the identified factors from selected publications, simple design, iterative development, minimal documentation, efficient utilization of time and computing resources, standard coding and fast delivery of software have shown high frequency. Our results reveal that the identified factors, as shown in Table 5, should be taken into account by vendor organizations for the development of green and sustainable software using agile techniques.

10. CONCLUSIONS

This paper presents a distinct approach towards the development of software by describing a systematic literature review protocol. We have a

particular focus on agile methods in, that support the development of sustainable software. Agile methods promise quick and scheduled delivery of software in short increments. It has reduced the complexity of software development through customer involvement and continuous interaction. To avail green agility, there is a need to explore green principles of agile methods that can contribute in software development life cycle that yields green and sustainable software. Keeping in view the importance and potential benefits of global software development, we intend to develop green agility maturity model that will help to measure the agile maturity of software organization and will also help to identify the weak areas that need to be addressed.

This research work presents the development of SLR protocol and its subsequent results as shown in Table 4 and Table 5. While implementing the protocol using the designed search string as mentioned in Track 2, we got publication sample of (N=2,298) , out of them (N=125) have been primary selected by reading the title and abstract of the papers. After a thorough review of the full text of the primary selected publications, we got a final sample of N=80.

We have almost finalized the data extraction of final selected publications by elicitation of the required data using an extraction form, as shown in Table 3. We have initially synthesized the extracted data in the form of factors as our preliminary results. These factors will be beneficial to software vendors for the production of green and sustainable software products using agile methods in global projects. Furthermore, these findings will ultimately help us to develop Green Agile Maturity Model for GSD Vendors (GAMM for GSD Vendors), in order to measure their green agile maturity for GSD projects. The detailed view is shown in Fig. 1.

11. REFERENCES

1. Holmström, B. Fitzgerald & J. Agerfalk. Agile practices reduce distance in global software development. *Information Systems Management* 23(3): 7-18 (2006).
2. Sahay, S. Global software alliances: The challenge of' standardization. *Scandinavian Journal of Information Systems* 15(1): 11-17 (2003).
3. Smite, D., C. Wohlin & T. Gorschek. Empirical evidence in global software engineering: A systematic review. *Journal of Empirical Software Engineering* 15(1): 91-118 (2010).
4. Khan, S.U., M. Niazi & R. Ahmad. Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. *Journal of Information and Software Technology* 53(1): 693–706 (2011).
5. Khan, H.H., N. Mahrin & B. Chuprat, Factors generating risks during requirement engineering process in global software development environment. *International Journal of Digital Information and Wireless Communications (IJDIWC) 4*(1): 63-78 (2014).
6. Zafar, A., S. Ali & R. K. Shahzad. Investigating integration challenges and solutions in global software development. In: *IEEE Conference, Frontiers of Information Technology (FIT),* Islamabad, Pakistan, p. 291- 297 (2011).
7. Vithana, V.N., Fernando & M. Kapurubandara. Success factors for agile software development: A case study from Sri lanka. *International Journal of Computer Applications* 113(17): 119-125 (2015).
8. Alistair., A. *Agile Software Development: The Cooperative Game*. Pearson Education (2006).
9. Highsmith, J. & A. Cockburn. Agile software development. *The Business of Innovation Computer* 34(9): 120-127 (2001).
10. Shrivastava, S. V. & H. Date. Distributed agile software development: A review. *Journal of Computer Science* 7(3): 10–16 (2010).
11. Singh, A., K. Singh & N. Sharma. Agile in global software engineering: an exploratory experience. *International Journal of Agile Systems and Management* 8(1): 23-38 (2015).
12. Kern, E., M. Dick & M. Naumann. Impacts of software and its engineering on the carbon footprint of ICT. *Environmental Impact Assessment Review* 52: 53-61 (2015).
13. Rashid., N. & S.U Khan. Green agile maturity model for global software development vendors. *Science International Journal* (Lahore, Pakistan) 5(26): 2041-2043 (2014).
14. Mohan, K., B. Ramesh & L. Cao. Managing disruptive and sustaining innovations in Green IT. *IT Professional* 6: 22-29 (2012).
15. Layman, L., Williams & D. Damian. Essential communication practices for extreme programming in a global software development team. *Journal of Information and Software Technology* 48(9): 781-794 (2006).
16. Ray, S. Green software engineering process: moving towards sustainable software product design. *Journal of Global Research in Computer Science* 4(1): 25-29 (2013).

17. Misra, S., V. Kumar & K. Fantazy. Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management* 29(9): 972-980 (2012).
18. Brhel, M., H. Meth & A. Maedcher, Exploring principles of user-centered agile software development: A. *Information and Software Technology* 61: 163-181 (2015).
19. Dybå, T. & T. Dingsøyr. Empirical studies of agile software development: a systematic review. *Information and Software Technology* 50(9): 833-859 (2008).
20. Dingsyr, T., S. Nerur & V. Balijepally. A decade of agile methodologies: towards explaining Agile software development. *Journal of Systems and Software* 85(6): 1213-1221 (2012).
21. Mahmoud, S.S. & I. Ahmad. A green model for sustainable software engineering. International Journal of Software Engineering and Its Applications 7(4): 55-74 (2013).
22. Tate, K. *Sustainable Software Development: An Agile Perspective*. Addison-Wesley Professional (2005).
23. Koontz, R.J. & R.I. Nord. Architecting for sustainable software delivery. Cross Talk (May/June 2012), In: *Rapid and Agile Stability*, p. 14–19 (2012).
24. Khan, S.U., M.K. Niazi & R. Ahmad. Barriers in the selection of offshore software development outsourcing vendors: an exploratory study using a systematic literature review. *Information and Software Technology* 53(7): 693-706 (2011).
25. Zhang, H., M.A. Babar & P. Tell. Identifying relevant studies in software engineering. *Information and Software Technology* 53(6): 625-637 (2011).
26. Kitchenham, B., S.L. Pfleeger & L.M. Pickard. Preliminary guidelines for empirical research in software engineering. *Software Engineering, IEEE Transactions* 28(8): 721-734 (2002).
27. Khan, S. U., M. Niazi & R. Ahmad. Factors influencing clients in the selection of offshore software outsourcing vendors: an exploratory study using a systematic literature review. *Journal of Systems and Software* 84(4): 686-699 (2011).
28. Khan, A.W. & S.U. Khan. Critical success factors for offshore software outsourcing contract management from vendors' perspective: an exploratory study using a systematic literature review. *Journal of IET Software* 7(6): 327-338 (2013).
29. Khan, S.U. & M.I. Azeem. Intercultural challenges in offshore software development outsourcing relationships: an exploratory study using a systematic literature review. *Journal of IET Software* 8(4): 161-173 (2014).
30. Firdaus, A., I. Ghani & N.M.I. Yasin. Developing secure websites using feature driven development (FDD): a case study. *Journal of Clean Energy Technologies* 1(4): 322-326 (2013).
31. Brereton P., B.A. Kitchenham & D. Budgen. Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software 80(4): 571-583 (2007).
32. Kitchenham, B., O.P. Brereton & D. Budgen. Systematic literature reviews in software engineering–a systematic literature review. *Information and Software Technology* 51(1): 7-15 (2009).
33. Niazi, M., N. Ikram & M. Bano. Establishing trust in offshore software outsourcing relationships: an exploratory study using a systematic literature review. *Journal of IET Software* 7(5): 283-293 (2013).
34. Korkala, M. & F. Maurer, Waste identification as the means for improving communication in globally distributed agile software development. Journal of Systems and Software 95(1): 122-140 (2014).
35. Shrivastava, S.V. & U. Rathod. Categorization of risk factors for distributed agile projects. *Information and Software Technology* 58*:* 373-387 (2015).
36. Ramesh, B., L. Cao & R. Baskerville. Agile requirements engineering practices and challenges: An empirical study. *Information Systems Journal* 20(5): 449-480 (2010).
37. Niazi, M., D. Wilson & D. Zowghi. A maturity model for the implementation of software process improvement: An empirical study. *Journal of Systems and Software 74*(2): 155-172 (2005).
38. McIntosh, S., Y. Kamei & B. Adams. An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering* 4(1)*:* 1-44 (2015).
39. Ali, I. & S.U. Khan. Usability of agile methods in software development outsourcing: A systematic literature review protocol. *Science International Journal* (Lahore, Pakistan) 5(26): 2023-202 (2014).
40. Petersen, K., S. Vakkalanka & L. Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: an update. *Information and Software Technology* 64: 1-18 (2015).
41. Niazi, M., S. Mahmood & M. Alshayeb. Empirical investigation of the challenges of the existing tools used in global software development projects. *Journal of IET Software,* 9(5): 135-143 (2015).
42. Verner, J.M., O.P. Brereton, B.A. Kitchenham, M.K. Niazi. Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology* 56: 54-78 (2014).

43. Verner, J.M., O.P. Brereton, B. Kitchenham, M. Turner & M.K. Niazi. Systematic literature reviews in global software development: a tertiary study. In: *16th International Conference on Evaluation & Assessment in Software Engineering (EASE),* Spain, p. 2-11 (2012).
44. Niazi, M. Software process improvement: a road to success. *Product-Focused Software Process Improvement:* Springer, p. 395-401 (2006).
45. Niazi, M.K. An instrument for measuring the maturity of requirements engineering process. In: *6th International Conference on Product Focused Software Process Improvement*. *LNCS*, Oulu, Finland , p. 13-16 (2005).
46. Khan, S.U., M. Niazi & R. Ahmad. Empirical investigation of success factors for offshore software development outsourcing vendors. *Journal of IET Software* 6: 1-15 (2012).
47. Niazi, M., D. Wilson & D. Zowghi. A model for the implementation of software process improvement: an empirical study. In: *Product Focused Software Process Improvement.* Springer, p. 1-16 (2004).
48. Niazi, M., D. Wilson & D. Zowghi. A model for the implementation of software process improvement: a pilot study. In: *Third IEEE International Conference on Quality Software,* (*QSIC*03). IEEE Computer Society, Dallas, USA, p. 196-203 (2003).
49. Niazi, M. Software process improvement implementation: avoiding critical barriers. *The Journal of Defense Software Engineering* 22(1): 24-27 (2009).