# Identification of Practices for the Development of Green and Sustainable Software Using Agile Methods in GSD Projects: A Systematic Literature Review Protocol

**Nasir Rashid**
Department of Computer Science and IT
University of Malakand, KPK,
Dir (L), Pakistan
nasir@uom.edu.pk.com

**Siffat Ullah Khan**
Department of Computer Science and IT
University of Malakand, KPK,
Dir (L), Pakistan.
siffatullah@uom.edu.pk

***Abstract— Context:* Over the last decade, the use of agile methodologies has been grown dramatically in software development projects. Agile methods through its principles promise to accelerate the delivery of remarkable software in increments with increased user satisfaction and reduced cost. However, in recent years, the software developers focus more on the development of green and sustainable software, in terms of both energy and resource efficiency. Green software engineering has emerged, that aims to design, develop and use the software with limited energy and computing resources to attenuate the negative effects on environment and society. The software engineering researchers are trying to integrate these two emerging trends i.e the development of green and sustainable software by practicing the agile methods to get the benefits of both.**

***Objective:* The objective of this study is to develop systematic literature review (SLR) protocol to identify agile practices for the development of green and sustainable software in GSD projects.**

***Method:* We have used systematic literature review (SLR) process as a research methodology.**

***Outcome:* The expected outcomes of the implementation of SLR protocol will be the agile practices for the development of green and sustainable software in GSD projects.**

***Keywords—* Agile methods, green software engineering, green agile, agile practices, SLR protocol.**

## I. INTRODUCTION

The purpose of this section is to give an overview of green and sustainable software engineering and its current need for the development of green and sustainable software.

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 production [1]. Green software engineering is an emerging paradigm and is growing rapidly that aims to develop software with less efforts and minimal resources to reduce negative impact on environment.

With an increasing demand of more complex software applications, Information and Communication Technology (ICT) bears a huge negative impact on the environment due to its increasing resource and power consumptions. The effect of ICT on sustainable development specifically on software is the hot topic now-a-days in software engineering. Recently ICT has been trying to find out efficient solutions for the development of software systems to reduce the negative effects on environment [2].

A variety of research work on Green ICT has mainly focused on environmental sustainability in terms of computer hardware. But revealing the issues related to energy consumption in software can be a great help in achieving green computing. Software features are responsible for CO2 emissions as are hardware components. Software has an indirect effect on the environment by operating and managing the underlying hardware [3].

Recently some efforts have been focused on building green and sustainable software, some design software processes to aid all stakeholders in building green software products. Others efforts are focused on building software tools that measure the effect of software on the environment and the effect of application development environments on software in terms of energy efficiency. There are efforts that emphasize on the operating system to help control the power consumption of applications [4, 5].

To solve the issues related to green and sustainable software development according to the current needs of the customers, there is a dire need to integrate the green engineering principles with agile methods to produce green and sustainable software [6].

Agile software development is invigorating approach towards quick and interactive software development. It provides a conceptual framework 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. Agile approaches rely on team members' skills and informal communication rather than formal and heavy documentations. Thus, agile methods pursue to avoid overwhelming processes, having little contribution, to software product [7].

Agile software engineering is an exhilarating approach for fast delivery of customer-centric software development. Unlike conventional approaches of software development, agile methods endeavor to scale down the risks and magnify software production. Agile methods embrace flexibility by trusting on developers' skills and their creativeness rather than formalized procedures and hefty documentation. Thus agile methods delve to avoid ponderous and time intense processes that have no contribution in software development. Some of the prominent agile methods are Extreme Programming, Scrum, Feature Driven Development (FDD), Crystal Clear, Dynamic System Development Method (DSD), Adaptive Software Development (ASD) etc. [8].

Originally, agile methods were developed for on-site/co-located team (s). However, later on it was realized that agile method can be supportive in distributed teams and the agile methods such as Scrum and XP were re-shaped as Distribute Scrum and Distributed XP [9].

Global software development (GSD) or distributed software development is growing rapidly due to increase in globalization of software business industry. 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. GSD offers tremendous benefits that include access to skilled pool of software developers, production of high standard software, and business advantage of proximity to market and the possibility to use “follow-the-sun” and “round–the-clock” development. Hence software development is now considered as a globally distributed endeavor [10].

In order to reap the benefits of agile methods and GSD, software organizations are blending the distributed development and agile approach called as Agile Global Software Development (AGSD). A survey conducted by Version One (https://www.versionone.com), states that organizations are constantly scaling agile methods beyond single team and single project. Several studies have reported the growing interest in applying agile methods in global software development to leverage the best features of both the approaches. Software industries have now realized the competitive advantages of the integrated approach for producing high quality software with accelerated delivery, minimal cost and flexibility to manage the requirements even late in the development process [11].

## II. BACKGROUND

As remarked in the Intel technical report [12], “The Impact of Software on Energy Consumption”, much of the computing energy consumed is based on the effectiveness of hardware energy utilization. But software also has a significant impact.

Till now greater part of work has been done within Green IT industry related to hardware that focuses on improving the energy efficiency of computer hardware. It thus seems self-evident that research needs to be broadened to include software as well. As Erdelyi [6] argues, researchers need to pay attention to the impact of software within Green IT. The trend has been changed in the last couple of years and enough work has been contributed to achieve green and sustainable software production.

Hilty [13] developed a model that combines standard Life Cycle Assessment (LCA) with “effects of ICT supply”, “effects of ICT usage” and “systematic effects of ICT” to show the potential impacts of ICTs on the life cycle of other products. He applied his model to ICT applications, which have been said to have a high carbon dioxide reduction potential and he identifies obstacles that hinder the use of these potentials.

Sara et al. [14] contributed in building a two level green software development model that covers the sustainable life cycle of a software product and the software tools promoting green and environmentally sustainable software. In the first level of the model a new green software engineering process that constitutes a hybrid mechanisms of sequential, iterative, and agile development processes to produce environment friendly software. The model proposes either guidelines or processes for green software development at each stage in the engineering process life cycle. The proposed model adds to the software life cycle the requirements stage and the testing stage. The model also includes in the first level a complete list of metrics to measure the greenness of each stage in terms of the first order effects of ICT on the environment for a green software engineering process.

Dick et al. [15] have developed a model that integrates Green IT aspects into software engineering processes with agile methods in order to design and develop green and sustainable software. The model integrates sustainability issues to software engineering processes as well as environment-friendly enhancements in to the most prominent agile method, scrum. However the energy efficiency and carbon emission of software products developed through this model cannot be measured precisely.

Tate [16] argues that culture of sustainable development should be adopted from the dogmatic methods of agile software development. Continual refinement of the product by embracing changes in the requirements, early delivery of working version of the software, emphasis on simple design and early defects prevention are the core principles of agile methods that can surely aid in the sustainable development of software.

The literature described earlier explains different approaches for the development of sustainable software. However, no research has been carried out to date in finding particular practices of agile methods through SLR that can contribute to sustainable software development. Our findings thus confirm the novelty of the work presented in this domain. The results obtained from this research work will assist the agile developers for the development of green and sustainable software in distributed software projects.

## III. Research Methodology

For the identification of agile practices for the development of green and sustainable software, we followed the systematic literature review guidelines, recommended by Kitchenham [17]. We have presented the core phases of our review protocol in the following subsections.

## IV. Systematic Literture Review Protocol

Systematic Literature Review (SLR) protocol is a set of tasks that have to be performed in order to answer the research question listed in the sub-section. Our review protocol is made of six components i.e. research questions, designing the search terms, searching strategy, publication selection criteria, publication quality assessment, data extraction and data synthesis. Each component is described in the following sections. SLR is considered to be a prominent research method in the field of software engineering and Computer Science [18-20]. We have followed all the SLR guidelines [17] by initially developing the SLR protocol as we did in our previous work [21, 22].

### *A. Research Questions*

The main objective of this research study is to find out state-of-the-art agile practices for the development of green and sustainable software development in GSD software projects. In order to achieve this goal; we outlined the following research question (RQ):

RQ: What are the agile practices, as identified in the literature, to be adopted by GSD vendor organizations which can be supportive in the development of green and sustainable software?

### *B. Constructing the Search Terms*

The following details will help in designing a search term relevant to our research questions.

Population: GSD Vendor Organization

Intervention: practices/solutions.

Outcomes of relevance: Green Agile maturity model for GSD vendor [23].

Experimental Design: Empirical studies, systematic literature review, theoretical studies, case studies, expert‘s opinions.

### *C. Searching Strategy and Search Terms*

The following search strategy is used for the construction of search terms.

a) Use the Research Questions for the derivation of major terms, by identifying population, intervention and outcome.

b) For these major terms, find the alternative spellings and synonyms;

c) Verify the key words in any relevant paper.

d) Use of Boolean Operators for conjunction if the database allows, in such a way, to use ‘OR’ operator for the concatenation of alternative spellings and synonyms whereas ‘AND’ for the concatenation of major terms.

We have designed the following two search strings for searching our selected online digital libraries.

S1: (“agile methods” OR “agile software development”) AND (“Green software” OR “sustainable software”) AND (‘practices OR ‘techniques’) AND (“Global software development” OR ‘GSD’)

S2: (“agile methods” OR “agile software development”) AND (“Green software” OR “sustainable software”) AND (‘practices OR ‘techniques’).

Where S1 denotes search string to retrieve publications regarding the agile practices for the development of green and sustainable software in GSD projects, while S2 denotes search string to retrieve publications regarding the agile practices for the development of green and sustainable software in general context. The results of search string (S1) were very poor and almost negligible as shown in Table 1. Consequently we decided to implement search string S2 after detailed discussions with experts of software engineering research group (SERG-UOM) at the university, to implement search string S2. The search results of S2 are showed in Table 2. The practices, identified through the SLR (using search string S2), will be validated through empirical studies in GSD industry in order to know whether these findings are applicable specifically, or can be adopted, in GSD environment.

Moreover limited numbers of empirical research has been conducted in the context of green and sustainable software in general and GSD in particular [24].

TABLE 1. S1 RESULTS

| Digital Libraries | Total publication Found |
|---|---|
| Science Direct | 00 |
| ACM | 00 |
| IEEE Xplore | 06 |
| Springer Link | 02 |
| Google Scholar | 20 |
| **Total** | **28** |

TABLE 2. S2 RESULTS

| Digital Libraries | Total publication Found |
|---|---|
| Google Scholar | 194 |
| ACM | 156 |
| IEEE Xplore | 338 |
| Springer Link | 221 |
| Science Direct | 79 |
| **Total** | **988** |

### D. *Resources Searched*

The following digital libraries and databases have been searched to search S2 as a search string.

- https://scholar.google.com.pk/
- http://dl.acm.org/
- http://ieeexplore.ieee.org/
- http://www.springer.com/
- http://www.sciencedirect.com/

### E. *Publication Selection*

The main purpose of the publication selection procedure is to choose only those search results which are relevant to our research questions.

The following inclusion and exclusion criteria have been used for the relevant literature selection.

a. Inclusion Criteria

- Research studies and articles that are related to our research questions.
- Research work that describe green and sustainable software development using agile methods in GSD projects/general context.
- The research papers/ articles /books/review paper is in English Language.
- The article/paper is available in full text.
- Research papers that describes success factors, risk factors and their relevant agile practices for the development of green and sustainable software in GSD projects/general context.

b. Exclusion Criteria

Article/papers/books etc. that do not fulfill inclusion criteria as mentioned above will be excluded.

### F. *Publication Selection*

The main purpose of quality assessment is to check and assess the quality of finally selected papers. The quality checklist contains the following questions:

a. Is the objective of the research clearly defined?
b. Is the conclusion of the research related to the objective of the research?
c. Whether the green and sustainable software development using agile methods in GSD projects/general software development context is discussed clearly?
d. Is it clear how the factors/agile practices were identified?

Each of the above questions will be marked as ‘YES’, ‘NO’, or ‘N.A’.

Apart from the above quality criteria we will also use the following quality criterion which is based on our own experience. We have used the same criteria in our previous research [21].

*Criteria for A-Quality papers:* In this category we list those papers which fulfill the following criteria's:

- Paper published in impact factor Journal
- Having clear methodology
- Having sample size of the following condition
  a. Case Study ≥ 3
  b. Interviews ≥ 12
  c. Survey ≥ 50
  d. Literature Review ≥ 50

*Criteria for B-Quality papers:* In this category we list those papers which fulfill the following criteria's:

- Paper published in well reputed conference
- Having clear methodology
- Having sample size of the following condition
  a. Case Study = 2
  b. Interviews: ≥ 5and ≤ 11
  c. Survey: ≥ 30 and ≤ 49
  d. Literature Review: ≥ 30 and ≤ 49

*Criteria for C-Quality papers:* In this category we list those papers which fulfill the following criteria's:

- Experienced reports/ articles, published in less reputed venues (Journal, Conference)
- Having clear methodology
- Having sample size of the following condition
  a. Case Study: 1
  b. Interviews ≤ 5
  c. Survey: ≥ 1 and ≤ 29
  d. Literature Review: ≥ 1 and ≤ 29.

### G. *Data Extraction*

The process of extracting data from finally selected relevant literature is called data extraction. The main aim of our protocol is to extract the data which satisfies our research question. The analysis will be commenced by a single researcher, who will be responsible for the data extraction. A secondary reviewer will be approached for guidance in case of an issue regarding the data extraction. The secondary reviewer will select few publications randomly from the list of publication already chosen by the primary reviewer. The secondary reviewer will independently extract data from the randomly selected publications. The results will then be compared with the results produced by the primary reviewer. We have used similar approach in previous research [25, 26].

### H. *Data Synthesis*

The analysis of extracted data from finally selected papers is called data synthesis. For the RQ, the data will be synthesized by creating one summary table having the columns (S.No, Factors/practices, Frequency, Percentages) showing the list of all the factors along with their frequencies and percentages.

## V. Results of Systematic Literture Review Protocol

The protocol preliminary results are shown in Table 3. The protocol has retrieved total N=988 research papers. From which N=290 were selected initially by reading title and abstract of the papers. Finally we selected 53 research papers. We are now in the process of data extraction. The expected outcomes of the SLR will be the identification of agile practices that can be supportive in the development of green and sustainable software in GSD projects.

Table 3. Preliminary results of the SLR protocol

| **Digital Libraries** | **Total publication Found** | **Initial selection** | **Final selection** |
|---|---|---|---|
| Google Scholar | 194 | 54 | 21 |
| ACM | 156 | 36 | 5 |
| IEEE Xplore | 338 | 61 | 13 |
| Springer Link | 221 | 84 | 9 |
| Science Direct | 79 | 55 | 7 |
| Total | 988 | 290 | 53 |

## VI. Conclusion and Future Work

Agile software development is an emerging trend to cope with the failures of software projects. As there is a dire need to develop green and sustainable software with low energy and computing resources in order to have positive impact on the society and human beings. Consequently, researchers have worked out to integrate these two emerging research areas to produce eco-friendly and sustainable software through agile methods to get the maximum benefits of both trends. Research in this area will assist GSD vendors to deliver green software and sustainable software in general and to measure the green agile maturity level of GSD vendors in particular. In this paper we have presented our SLR protocol to identify agile practices for the development of green and sustainable software in GSD. During implementation of the protocol, we have identified *N=988* research papers out of them (*N=290*) have been selected initially by reading the title and abstract of the papers. After applying the inclusion/exclusion criteria, we left with 53 publications, as shown in Table 3. We are now in the process of data extraction. The expected outcomes of the SLR will be the identification of agile practices that can be supportive in the development of green and sustainable software in GSD projects. The ultimate goal of this research study is the design and development of green agile maturity model for GSD vendors [27].

## References

[1] S. Naumann, M. Dick, E. Kern, and T. Johann, "The greensoft model: A reference model for green and sustainable software and its engineering," Sustainable Computing: Informatics and Systems, vol. 1, pp. 294-304, 2011.

[2] S. Agarwal, A. Nath, and D. Chowdhury, "Sustainable approaches and good practices in green software engineering," International Journal of Research and Reviews in Computer Science (IJRRCS), vol. 3, 2012.

[3] S. Ray, "Green software engineering process: moving towards sustainable software product design," Journal of Global Research in Computer Science, vol. 4, pp. 25-29, 2013.

[4] M. Thiry, L. Frez, and A. Zoucas, "GreenRM: Reference Model for Sustainable Software Development," in SEKE, 2014, pp. 39-42.

[5] F. Albertao, "Sustainable software development," Harnessing Green IT, pp. 63-83, 2012.

[6] K. Erdelyi, "Special factors of development of green software supporting eco sustainability," in Intelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium on, 2013, pp. 337-340.

[7] A. Cockburn, Agile software development: the cooperative game: Pearson Education, 2006.

[8] A. M. M. Hamed and H. Abushama, "Popular agile approaches in software development: Review and analysis," in Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on, 2013, pp. 160-166.

[9] R. Akbar, M. Haris, and M. Naeem, "Agile framework for globally distributed development environment (the DAD model)," in Proceedings of the

8th Conference on Applied Informatics and Communications, 2008, pp. 423-428.

[10] S. Sahay, "Global software alliances: the challenge of'standardization'," Scandinavian Journal of Information Systems, vol. 15, p. 11, 2003.

[11] L. Ahuja and R. Priyadarshi, "Agile approaches towards Global Software Engineering," in Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on, 2015, pp. 1-6.

[12] B. Steigerwald, C. Lucero, C. Akella, and A. Agrawal, "Impact of software on energy consumption," White paper, Intel Corp, 2011.

[13] L. M. Hilty, "CO2 reduction with ict: Prospects and barriers," Environmental Informatics and Systems Research, EnviroInfo Warsaw, pp. 12-14, 2007.

[14] S. S. Mahmoud and I. Ahmad, "A green model for sustainable software engineering," International Journal of Software Engineering and Its Applications, vol. 7, pp. 55-74, 2013.

[15] M. Dick, J. Drangmeister, E. Kern, and S. Naumann, "Green software engineering with agile methods," in Green and Sustainable Software (GREENS), 2013 2nd International Workshop on, 2013, pp. 78-85.

[16] K. Tate, Sustainable software development: an agile perspective: Addison-Wesley Professional, 2005.

[17] B. Kitchenham and C. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering, Keele University and Durham University Joint Report," EBSE 2007-001, 2007.

[18] J. I. M. de Souza, V. r. M. Silva, D. N. Prata, and P. Letouze, "A Further Analysis of the Importance of Systematic Reviews to Computer Science," International Journal of Social Science and Humanity, vol. 5, p. 682, 2015.

[19] M. Staples and M. Niazi, "Experiences Using Systematic Review Guidelines," Journal of Systems and Software., vol. 80, pp. 1425-1437, 2007.

[20] Y. I. Alzoubi, A. Q. Gill, and A. Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review," Information & Management, vol. 53, pp. 22-37, 2016.

[21] N. Rashid and S. U. Khan, "Green Agility for Global Software Development Vendors: A Systematic Literature Review Protocol," proceedings of the Pakistan academy of sciences, vol. 52, pp. 301-313, 2015.

[22] M. Yaseen, S. U. Khan, and A. U. Alam, "Software Multi-Sourcing Risks Management From Vendor’s Perspective: A Systematic Literature Review Protocol," Gomal University Journal of Research, vol. 29, 2013.

[23] N. Rashid and S. U. Khan, "GREEN AGILE MATURITY MODEL FOR GLOBAL SOFTWARE DEVELOPMENT VENDORS," Science International, vol. 26, 2014.

[24] R. Sriram and S. K. Mathew, "Global software development using agile methodologies: A review of literature," in Management of Innovation and Technology (ICMIT), IEEE International Conference, 2012, pp. 389-393.

[25] S. U. Khan, M. Niazi, and N. Ikram, "Software Development Outsourcing Relationships Trust: A Systematic Literature Review Protocol," in Evaluation and Assessment in Software Engineering, EASE 2010, Keele University, UK, 2010.

[26] S. U. Khan and M. I. Azeem, "Intercultural challenges in offshore software development outsourcing relationships: an exploratory study using a systematic literature review," Software, IET, vol. 8, pp. 161-173, 2014.

[27] N. Rashid and S. U. Khan, "Green-agile maturity model (GAMM) for global software development (GSD) vendors," Science International, vol. 26, pp. 2041-2043, 2014.