Developing Green and Sustainable Software using Agile Methods in Global Software Development: Risk Factors for Vendors

Nasir Rashid and Siffat Ullah Khan
Software Engineering Research Group (SERG-UOM), Department of Computer Science & Information Technology, University of Malakand, KPK, Khyber-Pakhtunkhwa, Pakistan

Keywords: Systematic Literature Review, Green and Sustainable Software, Agile Software Development, Risk Factor, Green Agile, GSD Vendors.

Abstract: Global software development (GSD) is gaining momentum due to the potential benefits it offers. GSD aims at delivering remarkable software through a widely distributed pool of experts, with reduced efforts, minimum cost and time. In recent years, GSD developers have reshaped the development processes and have adopted agile techniques and green engineering principles to cope with the frequent changes in requirements, accelerate the development in short increments and to produce energy efficient and sustainable software. However, the adoption of agile methods for developing sustainable software possesses a number of challenges. This paper presents a list of potential challenges/risks identified through systematic literature review (SLR) that need to be avoided by the GSD vendors using agile methods for the development of green and sustainable software. Our findings reveal eight risk factors that are faced by GSD vendors in the development of green and sustainable software using agile methods. GSD vendors are encouraged to address properly all the identified factors in general and the most-frequently cited critical risks in particular, such as in-sufficient system documentation, limited support for real-time systems and large systems, management overhead, lack of customer's presence, lack of formal communication and lack of long term planning.

1 INTRODUCTION

Global software development (GSD) has been grown with recent improvements in ICTs. In GSD various software engineers collaborate over temporal, geographical, cultural and linguistic distances. The major motivations for GSD are to achieve improvements in resources at low cost, high quality software, round the clock development and time to market efficiency (Alsudairi and Dwivedi, 2010). However, GSD benefits will not be achievable unless the associated risks are not managed (Noll et al., 2010). Some of the potential risks of GSD include hidden agreement costs, maintenance, lack of awareness of existing tools and lack of support for collaboration (Khan and Azeem, 2014).

In order to reduce some major risks in traditional software development, such as resources cost, frequent request for changes and timely delivery of software, the software engineering community have proposed some flexible methods called agile methods (Altameem, 2015). Some of the agile methods are Scrum, Extreme programming (XP), Crystal, Dynamic Systems Development Method (DSDM) and Lean Software Development (LSD). These methods help software developers to have more focus on requests for rapid changes in requirements, iterative development, collaborations among the developers and efficient resources (Al-Saleem and Ullah, 2015). It is obviously clear that agile methods have a positive impact on software development life cycle as it enhances the efficiency of developers and results in energy efficient software.

Agile methods promise to scale down the potential risks, detect faulty code, magnify software production and embrace frequent changes (Wrubel and Gross, 2015). Using agile methods in distributed environment also yield many benefits like smart communications, rigorous integration, sufficient document production and scheduled software (Singh et al., 2015).

This is not surprising that none of the agile

Rashid, N. and Khan, S.
Developing Green and Sustainable Software using Agile Methods in Global Software Development: Risk Factors for Vendors.
In *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering (ENASE 2016)*, pages 247-253
ISBN: 978-989-758-189-2

method is a silver bullet. It also poses a number of risks in various environments like limited support for distributed development, limited support for development of safety critical software, management overhead, reliance on the tacit knowledge of developers and in-effective communication among the customer and developers (Omar et al., 2011).

Research in the area of green agility is flourishing with increase in demand for the development of green and sustainable software. Green software engineering (GSE) is a prominent paradigm that has attracted the enthusiastic software developers, aims to develop, design and use the software with minimum economic, societal and ecological impacts (Raturi, Tomlinson et al. 2015). GSE aims at reducing the depletion of energy and natural resources and to scale down the direct and indirect negative effects on human being due to the development and use of software systems (Naumann et al., 2011).

In recent years, potential growth in green agile research has been noticed. Considerable amount of publications shed light on the issues in agile methods and in GSD as well, that motivated us for systematic review to identify risk factors in this way. The expected findings will be the first of its kind of study that will be helpful in the development of green agile maturity model in GSD that aims to measure green-agile maturity of vendor organisations. The preliminary structure of our proposed model has been published (Rashid and Khan, 2014). We have followed the concept of CMMI, IMM and SOVRM (Khan et al., 2010; Niazi, 2007; Silva et al., 2015)

This paper tries to answer the following research questions.

RQ1. What are the risks involved, as mentioned in the literature, to be avoided by agile software developers in GSD for the development of green and sustainable software using agile methods?

RQ2. Do the identified risks in green agile development, vary geographically (continent-wise)?

In the next section we describe related work, in section 3 we discuss our research methodology, and in section 4 we present the results. Section 5 discuss summary of our research, section 6 reports some limitations while section 7 presents conclusion and future work.

2 RELATED WORK

Agile methods have gained popularity in recent years and are working out to revamp the key processes involved in software development to ensure the delivery of green and sustainable software (Misra et al., 2012). However agile methods poses some limitations towards sustainability like test automation, backlog management, volatility of requirements and un-availability of customer's representative (Hushalini et al., 2014).

There is an emerging trend of introducing agile methods in GSD to achieve the maximum benefits of however this combination raises new challenges. Agile in GSD means to practice the principles of agile methods for energy efficient software delivery (Singh et al., 2015).

However it is certain that agile distributed development raises some challenges like task management, delays in project delivery, team management and communication (Jawad and Taira, 2015).

(Mohammad et al., 2013) demonstrate the strengths and weaknesses of agile development in GSD projects. Beside the voluminous benefits it offers, agile methods still face some challenges when it comes into practice, such as inconsistency in customers interaction, difficulty of managing large teams and lack of long term planning.

Current agile methods in practice produce less implementation that meet only short-term needs and thus assuredly produce software not green in nature and lack long-term suitability. (Mohan et al., 2010).

The literature described earlier explains some potential risks in the context of sustainable software development using agile methods. However, none of these factors have been identified through SLR. Our findings affirm that no research has been carried out to explore risks in sustainable software development using agile methods. The results obtained from this research work will assist the GSD agile developers to avoid the identified.

3 METHODOLOGICAL APROACH

This study adopts SLR methodology (Kitchenham and Charters 2007) as a research methodology as used by other researchers (Verner et al., 2012; Khan et al., 2015; Alzoubi et al., 2016). SLR is a systematic method for identifying, evaluating and interpreting all currently available research relevant to a particular research questions or area of interest. (Zhang et al., 2011). The following subsections describe the procedure specified in our protocol

(Rashid and Khan, 2015).

3.1 Searching Process

Developing an appropriate search strategy in SLR is quite essential. A generous was conducted to cover a broad range of relevant publications. The search process used five search engines, i.e. IEEEXplore, Science Direct, Google Scholar, Springer Link and ACM Digital Library. Manual searches for the relevant papers were conducted on the mentioned libraries using snowballing approach to increase the number of relevant research articles as used by (Wohlin, 2014).

Table 1: List of identified risk factors.

S. No	Risks Factors	Frequency N=42	Percentage
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

Search terms used were ((Global OR GSD OR Distributed) AND (Agile OR ''Agile methods'' OR "Green Agile" OR "Agile approaches") AND (Green OR Sustainable OR "Green Software Engineering" OR "Green software") AND (Limitations OR Challenges OR Issues OR Risks)).

Adjustments were made where needed to fix the syntax of different search engines used.

3.2 Publication Selection

Papers resulting from the generous search were reviewed, after considering the title if required, the abstract, rejected all other papers that were obviously irrelevant. This resulted in 42 final publications.

3.3 Data Extraction and Synthesis

Relevant data was extracted to answer the research questions. A conscientious study carried out by the author catalogued 25 groups of risk factors from 42 publications. After validation, we found 08 risks to be decisive as shown in Table 3. Among the 08 risk factors, 06 are remarkable due to high frequency and are considered as critical risk factors (CRFs) for having frequency >=40%. Criteria for criticality have been acquired from (Khan et al., 2009).

4 RESULTS

The following subsections report the results of SLR to answer the research questions.

Table 2: Continent wise paper frequency.

Continent	Frequency N=42	Percentage
Asia	12	29
N. America	09	21
Europe	18	43
Mixed	03	7

4.1 Risk Factors while using Agile Methods for Green and Sustainable Software Development (Research Question 1)

Table 1 illustrates that 'insufficient system documentation' is the most common risk factor (62%). Agile methods provide only a few implementation details of software implementation, and because of this "low-level" approach they may build the software that meet short-term individual project needs, but that do not necessarily lead to sustainable software systems (Stammel et al., 2011; Hall, 2014).

Our findings reveal that 'Lack of customer's presence' (62%) have the same severity to the previous mentioned risk factor. Consistent presence of the customer is most essential practice of agile methods. This reduces the efforts to complete the software within defined time frame and leads to sustainable development (Mahmoud and Ahmad 2013).

In absence of customer, the software builds take a longer time to complete and are more likely to fail. Coherent, self-organizing agile teamwork and strong communication supports a green process throughout software development HSIEH and CHEN 2015).

Management of time and computing resources is an integral component that promotes green software development in all types of software development environments. However, agile methods experience

Table 3: Distribution of risk factors continent wise.

Risk Factors	Sample Size N=42								Chi-Square test (Linear- by -Linear Association) α=0.5		
	Asia N=12		N. America N=09		Europe N=18		Mixed N=03				
	Frequency	%	Frequency	%	Frequency	%	Frequency	%	X^2	df	p
Insufficient system documentation	09	75	06	67	10	56	03	100	2.105	1	0.147
Limited support for real-time systems and large systems	06	50	03	33	10	56	0	0	.208	1	.648
Management overhead	10	83	02	22	10	56	03	100	.137	1	.711
Lack of customer's presence	09	75	06	67	10	56	01	33	2.105	1	.147
Lack of formal communication	04	33	04	44	08	44	02	67	.844	1	.358
Limited support for reusability	01	8	02	22	04	22	0	0	.128	1	.669
Insufficient knowledge of the customer	06	50	04	44	04	22	01	33	2.025	1	.155
Lack of long term planning	08	67	04	44	07	39	01	33	2.25	1	1.133

the risk of 'management overhead' (60%), which may lead to over-budget, time over run and maximum use of available computing resources for software development. This can greatly influence the sustainable and green approach towards software development (Altameem, 2015).

Intelligent and long- term planning is a crucial factor to be considered for social, economic and environmental sustainability of the software (Calero et al., 2015). The 'lack of long-term planning' (48%)' is listed as the fourth highest risk concern with the use of agile methods in order to produce sustainable software. The importance of sustainability is increasingly recognized in terms ofsoftware development which needs a long term planning (Venters et al., 2015). However agile methods focus more on the immediate delivery of software according to the current needs of customers without taking into considerations its long term impact on human and society as a whole.

4.2 Assessment of the Risk Factors, Continent Wise Analysis (Research Question 2)

In order to answer the second research question, Table 3 highlights a list of 8 risk factors retrieved through SLR in distinct continents while Table 2 presents the research data from various continents. North America, Europe and Asia are considered among the continents for analysis of data whereas the rest are merged into 'Mixed category' due to small sample size. The content wise analysis diagnoses the existence of compelling variances among the risk factors. In order to observe the variances, we apply linear by linear association Chi-square (X^2) test. For the evaluation of derivations between ordinal variables, X^2 linear by linear association is used for analysis, which is considered more prevalent as compared to Pearson Chi-square test.

Table 3 interprets the frequency of publications in various. Few variations have been noticed among the identified risk factors across different continents as illustrated in Table 3. All the listed risk factors have been found in the same number in different continents except for 'Limited support for real-time systems and large systems' and 'limited support for re-usability', which do not exist in mixed continent category.

Some risk factors show high rise up in frequency in particular continents, such as 'management overhead', 'lack of customer presence', 'insufficient system documentation', 'limited support for real time systems and large systems' and 'lack of long term planning'. This shows that in certain continents the agile maturity may not have reached up to the level that can better mitigate the risks and to

contribute towards green software development.

Table 3 also highlights the high ranked risk factors among the identified ones in specific continents. In Asia 'insufficient system documentation', 'management overhead', 'lack of customer presence' and 'lack of long term planning' have got high frequency. In North America only the 'insufficient system documentation' and 'lack of customer's presence' have got significant values. While in Europe the severity of risks is the same as in Asia except for 'lack of long term planning', which shows low frequency. The three mentioned continents present slight different approach in applicability of agile methods in software development.

The findings in Table 3 shows the risks confronted when using agile methods for the development of green and sustainable software in different continents. The variation in frequency of risks across the continents may be the difference in nature and complexity of the software productions, difference in technical expertise of agile developers, selection of agile technique used or may be that some factors are more concerned regarding the milestones defined for organisations in different continents that give rise to these potential risks.

5 SUMMARY AND DISCUSSION

This study indicates a number of potential risks identified through SLR that need to be avoided by the agilists when intend to develop green and sustainable software in GSD environment. The identified risks expose some key process areas of agile methods that need a magnitude of agilists' focus in the way of greener software development.

In order to pinpoint the severity of the identified risks, we take into account the frequency percentage to be >=40, as the criteria set by other researchers (Darwish and Rizk, 2015) for the identification of critical risk factors. We adopted the above mentioned criteria for criticality of the identified risk factors on RQ1, which resulted in 06 critical risks.

6 THREATS TO VALIDITY

With respect to internal validity, the threat comes from the fact that authors of selected publications may not have sufficient knowledge of the subject areas to describe the identified risks in depth. There may be an inclination towards reporting some risks in certain papers as well. Concerning external validity, first threat is the undeniable limited number of publications, as search strings are compiled and interpreted differently by various search engines. Secondly, most of the papers we found are authored by academicians who may lack the practical experience of agile methods.

Finally, regarding construct validity, a pilot questionnaire is made prior to the final implementation of the construct, which aims to ensure the improvement and assurance of the associated documentation. Probably it will result in several changes to the identified risk factors.

7 CONCLUSION AND FUTURE DIRECTIONS

This study presents a distinct approach to evaluate the use of agile methods for the development sustainable software using SLR and to find the risks along this way. This research identified 08 risk factors as listed in Table 1, from a sample of 42 papers, out of which some risks have been declared as critical due to high frequency.

Our findings suggest some key processes for future work in agile GSD industry: (1) to conduct an empirical study to validate the our findings (2) to explore some unforeseen risks apart from the identified ones, if any and (3) to conduct empirical study for the identification of relevant practices for the mitigation of the identified critical risks. Similar method has been followed by other software engineers and researchers (Garousi et al., 2015).

This study is a vital step in the long way approaching the development of our proposed green-agile maturity model that aims to measure the agile capability of vendor organization in the context of green software development. Our proposed research design consists of SLR, empirical study in GSD industry and case studies at vendor organizations. A similar research design has been used by other researchers (Niazi et al., 2015). This study also presents a continent wise analysis of the identified risks as shown in Table 3. The mentioned analysis may be useful in providing a deep insight to agile software developers about the most critical risks that need in depth focus to be avoided.

REFERENCES

Al-Saleem, S. M. and H. Ullah (2015). "A Comparative

Analysis and Evaluation of Different Agile Software Development Methodologies." IJCSNS 15(7): 39.

Alsudairi, M. and Y. K. Dwivedi (2010). "A multi-disciplinary profile of IS/IT outsourcing research." Journal of Enterprise Information Management 23(2): 215-258.

Altameem, E. (2015). "Impact of Agile Methodology on Software Development?" Computer and Information Science 8(2): p9.

Khan, S. U., & Azeem, M. I. (2014). Intercultural challenges in offshore software development outsourcing relationships: an exploratory study using a systematic literature review. Software, IET, 8(4), 161-173.

Calero, C., M. Á. Moraga, M. F. Bertoa and L. Duboc (2015). Green Software and Software Quality. Green in Software Engineering, Springer: 231-260.

Darwish, N. R. and N. M. Rizk (2015). "Multi-Dimensional Success Factors of Agile Software Development Projects." International Journal of Computer Applications 118(15).

Garousi, G., V. Garousi-Yusifoğlu, G. Ruhe, J. Zhi, M. Moussavi and B. Smith (2015). "Usage and usefulness of technical software documentation: An industrial case study." Information and Software Technology 57: 664-682.

Hall, J. J. (2014). "Quality Assurance in Business Simulation Design." Developments in Business Simulation and Experiential Learning 41.

HSIEH, C.-Y. and C.-T. CHEN (2015). "Patterns for Continuous Integration Builds in Cross-Platform Agile Software Development." JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 31: 897-924.

Hushalini, S., R. Randunu, R. Maddumahewa and C. Manawadu (2014). "Software Test Automation in Practice: Empirical Study from Sri Lanka." Compusoft 3(11): 1232.

Jawad, Z. and M. Taira (2015). "Improving Global Software Development Tools to Facilitate Distributed Agile Development." Software Engineering thesis, University of Gothemburg, Sweden.

Khan, S. U., M. Niazi and R. Ahmad (2009). Critical success factors for offshore software development outsourcing vendors: A systematic literature review. Fourth IEEE International Conference on Global Software Egineering, 2009. ICGSE 2009. Limerick, Ireland.207-2016.

Mahmoud, S. S. and I. Ahmad (2013). "A green model for sustainable software engineering." International Journal of Software Engineering and Its Applications 7(4): 55-74.

Misra, S., V. Kumar, U. Kumar, K. Fantazy and M. Akhter (2012). "Agile software development practices: evolution, principles, and criticisms." International Journal of Quality & Reliability Management 29(9): 972-980.

Mohammad, A. H., T. Alwada'n and J. M. A. Ababneh (2013). "Agile Software Methodologies: Strength and Weakness." International Journal of Engineering Science and Technology (IJEST) 5(03): 455-459.

Mohan, K., B. Ramesh and V. Sugumaran (2010). "Integrating software product line engineering and agile development." Software, IEEE 27(3): 48-55.

Naumann, S., M. Dick, E. Kern and T. Johann (2011). "The greensoft model: A reference model for green and sustainable software and its engineering." Sustainable Computing: Informatics and Systems 1(4): 294-304.

Niazi, M., S. Mahmood, M. Alshayeb and A. Hroub (2015). "Empirical investigation of the challenges of the existing tools used in global software development projects." Software, IET 9(5): 135-143.

Noll, J., S. Beecham and I. Richardson (2010). "Global software development and collaboration: barriers and solutions." ACM Inroads 1(3): 66-78.

Omar, M., S.-L. Syed-Abdullah and A. Yasin (2011). "The impact of agile approach on software engineering teams." American Journal of Economics and Business Administration 3(1): 12.

Rashid, N., & Khan, S. U. (2014)."Green-Agile Maturity Model (GAMM) for Global Software Development (GSD) Vendors" (2014) The 1st Agile Conference Pakistan (Islamabad), Sci. Int.(Lahore),vol. 26(5),pp. 2041 -2043.

Raturi, A., B. Tomlinson and D. Richardson (2015). Green Software Engineering Environments. Green in Software Engineering, Springer: 31-59.

Singh, A., K. Singh and N. Sharma (2015). "Agile in global software engineering: an exploratory experience." International Journal of Agile Systems and Management 8(1): 23-38.

Stammel, J., Z. Durdik, K. Krogmann, R. Weiss and H. Koziolek (2011). Software evolution for industrial automation systems: literature overview, KIT, Fakultät für Informatik.

Venters, C., C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, G. Rodriguez-Navas and N. Seyff (2015). "Mind the Gap: Bridging the Sustainable Software Systems Research Divide."

Verner, J. M., O. P. Brereton, B. Kitchenham, M. Turner and M. Niazi (2012). "Systematic literature reviews in global software development: A tertiary study."

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ACM. p.38.

Wrubel, E. and J. Gross (2015). "Contracting for Agile Software Development in the Department of Defense: An Introduction.", Technical report (CMU/SEI-2015-TN-006), Carneige Mellon University.

Zhang, H., M. A. Babar and P. Tell (2011). "Identifying relevant studies in software engineering." Information and Software Technology 53(6): 625-637.

Khan, S. U., & Niazi, M. (2010, June). A preliminary structure of software outsourcing vendors' readiness model. In Proceedings of the 11th International Conference on Product Focused Software (pp. 76-79). ACM.

Niazi, M., A Framework for Assisting the Design of Effective Software Process Improvement Implementation Strategies. Journal of Systems and Software.2007,78(2):204-222.

Rashid, N., & Khan, S. U. (2015). Green Agility for Global Software Development Vendors: A Systematic Literature Review Protocol. Proceedings of the Pakistan Academy of Sciences 52(4): pp.301-313.

Khan, R. U., Khan, S. U., Khan, R. A., & Ali, S. (2015). Motivators in Green IT-outsourcing from Vendor's Perspective: A Systematic Literature Review. Proceedings of the Pakistan Academy of Sciences 52(4): pp.345-360.

Alzoubi, Y. I., Gill, A. Q., & Al-Ani, A. (2016). Empirical studies of geographically distributed agile development communication challenges: A systematic review. Information & Management, 53(1):pp. 22-37.

Kitchenham, B. A., & Charters, S. (2007). Procedures for performing systematic literature reviews in software engineering. Keele University & Durham University, UK.

Silva, F. S., Soares, F. S. F., Peres, A. L., de Azevedo, I. M., Vasconcelos, A. P. L., Kamei, F. K., & de Lemos Meira, S. R. (2015). Using CMMI together with agile software development: A systematic review. Information and Software Technology, 58, 20-43.