A Systematic Review Protocol for Identification of the Factors Affecting Sustainability Dimensions in Software Crowdsourcing

Waqas Haider[*], Muhammad Ilyas[†], Adnan Khan[‡]

Abstract

Crowdsourcing gained popularity over the past one and a half decades. The trend of software crowdsourcing is spreading at a fast rate. Various fields can apply this problem-solving model to save money and time. It is also a wonderful technique to generate novel and unconventional ideas and solutions. Nonetheless, issues remain in areas like as the maturity of software sustainability for development and its readiness as a final output for use, particularly given the growing popularity of software crowdsourcing. Clarifications are still required on sustainable software development and software sustainability. Furthermore, as software, crowdsourcing becomes popularity, norms are moving away from the sustainability of software development. Consequently, one of the weaknesses in software standards mentioned is the absence of criteria for sustainable software crowdsourcing. The aim of this research is to determine the factors that determine the aspect of sustainability in crowdsourced software development (CSD) through systematic literature review. We have created a protocol for conducting systematic literature reviews (SLR) and are already putting it into implementation. Because SLR follows a predetermined procedure, it differs from a typical literature review. Results from SLR are more comprehensive and in-depth than those from standard literature reviews. The expected results of this review concern the identification of the factors affecting the sustainability in CSD.

***Keywords*:** Systematic Literature Review, Crowdsourcing Sustainability.

Introduction

Crowdsourced software development is a new software developing method. It helps to leverage the diversified skills of people everywhere in the world (Qayyum et al., 2024). Crowdsourcing is an open invitation strategy to outsource tasks to a large and undefined community; according to the generally accepted concept provided in the articles (Chittilappilly et al., 2016; Ramachandran et al., 2025). Crowdsourced software engineering (CSE) originates out of crowdsourcing (Mao et al., 2017). It involves an open call of an online worker globally to work on various activities of software engineering, for example, coding, designing,

[*] Corresponding Author: Department of CS & IT, University of Malakand, Chakdara 18800, Pakistan, whaidermkd@gmail.com

[†] Department of CS & IT, University of Malakand, Chakdara 18800, Pakistan, milyasmkd@uom.edu.pk

[‡] Department of CS & IT, University of Malakand, Chakdara 18800, Pakistan, afridikhan541111@gmail.com

testing, and documentation (Lakhani et al., 2010; LaToza et al., 2015; Stol et al., 2014).

The primary characteristic of software crowdsourcing (SC) is the open invitation, welcoming anyone from the public to participate, irrespective of organizational boundaries (Howe, 2006). All of SC processes are conducted specially through the internet (Naik et al., 2017). Publics involved in crowdsourcing mostly work as paying out workers, while others work on small projects on a volunteer basis (Marshal Hargrave et al., 2019). This emerging distributed problem-solving paradigm (Mao et al., 2017). Parallelism's adaptive development method has been shown to accelerate product releases, lower costs, and eliminate faults (Chittilappilly et al., 2016; LaToza et al., 2015). Many popular crowdsourcing services, such as TopCoder, AppStore, uTest, Mob4Hire, and TestFlight, use CSE. The new distributed problem-solving paradigm has been utilized across various creative and design-related activities (Alonso et al., 2008; Brabham et al., 2010; Chatfield et al., 2014; Norman et al., 2011). Industrial and academic groups are becoming increasingly interested in crowdsourced software engineering.

Crowdsourcing has many benefits it allows the crowdsourcing of the software development industry to gain access to a worldwide talent pool which encourages faster innovation and diversification of problem-solving (Sarkar et al., 2023). It makes it more efficient by lowering the costs and time taken in executing the development as multiple tasks are carried out concurrently. It also assists in solving complex problems through pooling knowledge and experience as well as user feedback experience. Crowdsourcing also facilitates the involvement of the community whereby people share their knowledge and cooperate. Besides, it enables its continuous development and testing, which results in software products of better quality and user-centric (Ramachandran et al., 2025), However it faces several issues like collaboration with various staff leads to issues with management. This issue wastes time when multiple staff members create identical designs for the same assignment. Therefore, when it comes to privacy, safety, legal procedures, any software project is inappropriate for crowdsourcing (Naik et al., 2016). Crowdsourcing also consists of a few recognized issues like expert staff, communication gap, low quality submissions, loss of confidentiality and difficulty in choosing between competing solutions and copyright issues (Hatch et al., 2017). The proposed research will determine the major factors that influence the sustainability aspects in crowdsourced software development by performing a Search Literature Review (SLR).

The aims of this study to gather information from existing literature in order to completely understand the factors that influence and contributed to the development of long-lasting, reliable, and efficient software for the crowdsourcing system. The result of this research will assist all developers, researchers, managers, stakeholders, and the software industry at large to make informed decisions. The SLR protocol is an organized guide of the study which will be implemented. Certain research questions have been formulated to lead through this process. These questions aim at describing and classifying those aspects that are related to sustainability. The findings of this work will become a basis of future research and creation of software systems which can be sustained. The study aims at answering the following research question to answer the issue of sustainability of CSD.

Related Work

Crowdsourcing refers to software development that uses people to solve many different problems, components, and tasks (Barnes et al., 2015). Some persons in this industry and on the internet are employed by software corporations to do tasks for them. They do tasks such as writing code, testing it, and designing it. These employees were responsible for a wide range of jobs, including developing, performing sophisticated micro tasks for specific functionalities, testing certain components, and assigning other duties (Estellés-Arola et al., 2012). These internet based individuals whose concerns include resource pools to solving of such jobs are commonly known as crowd. Selecting the crowd via the internet is a difficult task performed by software crowdsourcing organizations (Boughzala et al., 2014). Although the crowdsourced applications are being implemented at a high rate in the industry (Wu et al., 2013), in their working architecture, the seismic vulnerabilities that inevitably highlight the risk of software failure make survival difficult (Boughzala et al., 2014). In this regard, it may be challenging and difficult to conduct projects such as documentation. (Alelyani et al., 2016). Sustainable development, and ensuring the testing of applications before deployment (Bari et al., 2016).

It can be difficult to carry out crowdsourcing tasks while ensuring they are completed on time. Insufficient knowledge about emerging crowd sourcing software development models is cited as one of the key risk factors of project failures among others which are potential triggers of failures in software development projects (Bari et al., 2016). Organizations have the challenge of dealing with existing crowdsourcing software working paradigms including work distribution, crowd selection, and collaboration (Alelyani et al., 2016; Barnes et al., 2015). The newly created software's sustainability was challenged (Lago et al., 2015). In this case, the result is an infinite software development style when tasks are assigned to the crowd (Hegedus et al., 2013). The standardization of software and its sustainability is challenging, and the inclusion of distributed tasks among the crowd (LaToza et al., 2015). Oyedeji et al. (2024) discuss the ways of incorporating sustainability issues in crowdsource software development, identifying the fact that little develop has occurred in the industry in this regard. Using a survey and industry case study, they show applied strategies, e.g., customizing Scrum processes using sustainability-oriented tools, to promote sustainable software engineering. Sustainability can be referred to as development that meets the expectations of the current and future generations (Imperatives, 1987).

In the framework of sustainable software engineering, The sustainability concept can be understood as the ability to develop software solutions to systems with long lifetime so that they may meet the needs of the future generations too through incorporating the four dimensions of sustainability (technology, society, environment and economy) to fulfil the requirements on time (Ahmad et al., 2014). However, the achievement of software sustainability aspects is not given suitable attention. Thus, there is a clear need for software crowdsourcing to control and track aspects of sustainability. Similarly, it is contended that the standards established by the Institute of Electrical and Electronics Engineers (IEEE) and the International Organization for Standardization (ISO) lack sufficient emphasis on guidelines addressing sustainability aspects, not exclusively within the area of co-located apps evolution, although also in the process of Global Software Development (GSD), and particularly, in the most recent installment of crowdsourcing apps (Oyedeji et al., 2018).

Therefore, such a lack of sustainable development standards for crowdsourcing applications will contribute to software failures. Additionally, there is an acknowledgment that the core functional design of crowdsourcing applications harbors flaws that remain unexplored in previous research endeavors. Failure to resolve these concerns can result in severe software failures. Lastly, it is suggested that there is an imminent necessity to focus on sustainability dimensions, which are also demonstrated through United Nations environmental targets related to software development and crowdsourcing software. Ignoring some crucial phenomena can have an impact on the effective construction of a software system, leading to software failure (Malik et al., 2018).

Research Methodology

The research activity will be carried out to achieve the goal of our research/study project. There are two phases/stages. In the first phase, we need to develop a SLR protocol according to the SLR instructions to determine factors that affect the aspects of sustainability in CSD

(Kitchenham & Charters, 2007). Other SLR procedures have also been investigated (Ilyas et al., 2012; S. U. Khan et al., 2008; Nazir et al., 2019) for assistance in developing this SLR protocol. An SLR protocol is a literature review plan, concerning a particular research question or topic. Despite the fact that it cannot prevent publication bias and SLR is shielded from the researcher's bias by following a set process. One of the key benefits of conducting a SLR is that it brings clarity to the data collecting process, resulting in a higher level of accuracy, objectivity, and reproducibility. According to Kitchenham et al., (2007) there are three primary phases to an SLR study:

Planning the review, i.e., development of the protocol
Conducting the review, i.e., executing the protocol
Reporting the review. Dissemination of results

In the second phase, we will undertake empirical investigations using a Crowdsourced software platform to verify the SLR results and identify any new factors not found in the SLR.

This paper has been written by the first step/phase of our research project. i.e., creating a SLR protocol. We have simply discussed the initial step of our ongoing study in this report. i.e., SLR protocol. The protocol's many steps have been covered in the following sections of the publication.

Construction of Search Terms

We adhered to the instructions on formulating search keywords to our research question (Ilyas et al., 2012; R. A. Khan et al., 2014; Kitchenham & Charters, 2007). We created the following search terms.

Population: CSD
Intervention: Factors influencing sustainability aspects of CSD
Outcome of relevance: Sustainable software development
Experimental design*:* Any journals relevant to CSD, i.e. empirical study, theoretical study, case studies, expert opinion, etc.

Using the terms above, our study questions are therefore.

RQ1: [what are the factors] INTERVENTION
[That influence sustainability aspect] OUTCOME OF RELEVANCE
In
[Crowdsourced software development] POPULATION

Search Strategy

Trial search

The following search string strategy was used to perform a preliminary search in the IEEE Xplore digital library, Google Scholar, ACM, Citeseer, and Springer Link. A similar method of performing a trial

search has been used by (Ilyas et al., 2012; Nazir et al., 2019; Salam et al., 2015).
(("Factors" OR "Challenges" OR "Issues" OR "Hurdles" OR "Barriers" OR "Obstacles" OR "Impediments" OR "Hindrances") AND ("Sustainable" OR "Sustainability" OR "Green")
AND
("Crowdsourcing" OR "Crowdsource" OR "Crowdsourced")
AND
("Software"))
The papers which were discovered from this search string were used as a guideline for completion acknowledgement of main search terms.

Identification of Search Term

For the search terms identification, we use the following plan.

- We first highlighted population, intervention and outcome as the main search terms based on the research question.
- Then we will search different spelling and synonyms of the main keywords we have identified earlier.
- Next, we checked the related words in a related publication.
- When possible, make use of the 'OR' operator to string together alternate spellings and synonyms, along with using the 'AND' operator for major terms.
- In case it is required, summarize the search string.

Since there are databases where a large search string cannot be entered (just to give an example, we have Google Scholar, or CiteSeer), we might have to split the string we have constructed above into two or smaller terms.

Result for (a)

RQ1: Factor, sustainable software, crowdsourced software.

Result for (b)

Factor: ("factors" OR "challenges" OR "issues" OR "hurdles" OR "barriers" OR "obstacles" OR "impediments" OR "feature" OR "barrier" OR "motivators" OR "characteristics" OR "drivers")

Sustainable software: ("sustainable" OR "sustainability" OR "Sustainable software" OR "Sustainable software development" OR "Sustainable software engineering" OR "Green software" OR "green computing" OR "green software development" OR "sustainability" OR" software sustainability" OR "sustainability dimension" OR "green IT" OR "green ICT")

Crowdsourced software development: ("crowdsourcing" OR "Crowdsource" OR "Crowdsourced" OR "crowdsourced software" OR "crowdsourced software development" OR "crowdsourced software engineering "OR "crowd-based outsourcing" OR "collective participation" OR "crowdsourcing" OR "software crowdsourcing").

Result for (C)

Crowdsourcing: crowdsource, crowdsourced, crowdsourced software, crowdsourced software development, crowdsourced software engineering, sustainability, sustainable software development, sustainable software engineering, factors, challenge, barrier, issue

Result for (d)

RQ1: ("factors" OR "challenges" OR "barriers" OR "hurdles" OR "issues" OR "obstacle" OR "hindrance" OR "feature" OR "barrier" OR "motivators" OR "characteristics" OR "drivers") AND ("crowdsourcing" OR "Crowdsource" OR "Crowdsourced" OR "crowdsourced software" OR "crowdsourced software development" OR "crowdsourced software engineering" OR "crowd-based outsourcing" OR "collective participation" OR "crowdsourcing" OR "software crowdsourcing") AND ("crowdsourcing" OR "Crowdsource" OR "Crowdsourced" OR "crowdsourced software" OR "crowdsourced software development" OR "crowdsourced software engineering" OR "crowd-based outsourcing" OR "collective participation" OR "crowdsourcing" OR "software crowdsourcing")

Result for (e)

No result

Search String Breakup

Our search word is the above-mentioned search string. Nevertheless, certain databases/libraries (e.g. Science Direct, Google Scholar) do not enable lengthy search strings. The search term above was divided into lesser-sized substrings and was separately searched in each of them. The substrings of the chosen research question RQ1are as follows. Each of the search phrases in the research question are further divided into sub-search terms.

Substrings

Search string 1: ("Factor" OR "challenge") AND ("sustainable" OR "sustainability" OR "green" AND ("crowdsourcing" OR "Crowdsource" OR "Crowdsourced") AND ("Software")

Search string 2: ("Issue" OR "hurdle") AND ("sustainable" OR "sustainability" OR "green") AND ("crowdsourcing" OR "Crowdsource" OR "Crowdsourced") AND ("Software")

Resources Search

A literature search was performed in a number of well-established digital libraries and databases, namely, IEEE Explore, Science-Direct, ACM Digital Library, Springer Link, Wiley Online Library, and Google Scholar.

Search Constraints and Validation

We conducted a comprehensive search of all relevant published articles without creating any data boundaries or limitations. A prior search was conducted using a keyword list.
("Factor" OR "Challenge" OR "Issue" OR "Hurdle" OR "Barrier" OR "Obstacle" OR "Impediment" OR "Hindrance")
AND ("Sustainable" OR "Sustainability" OR "Green")
AND ("Crowdsourcing" OR "Crowdsource" OR "Crowdsourced")
AND ("Software")
To find suitable papers, we employed various online libraries and databases: Google Scholar, IEEE Xplore, ACM Digital library, Science-Direct, Wiley Online Library, and Springer Link. These relevant papers will validate search strings before the review process begins.

Search Documentation

Table 1 shows the important digital online resources used in this investigation to locate appropriate written material.

Table 1: Searched libraries.

S.No.	Source Name	Total result	Initial selection	Final Selection
1	Science Direct	1113	15	0
2	IEEE Xplore	158	04	01
3	Springer Link	744	10	01
4	ACM Digital library	983	11	03
5	Google Scholar	319	40	12
6	Wiley online library	107	04	01
7	Snowballing			27
	Total Results	4424	84	45

The aforementioned are some of the most well-known and significant academic electronic databases. The primary motivation to include the above-mentioned online databases in the search for literature is the ability to access the information they contain. These databases additionally include access to high-impact full-text journals and conference proceedings, as well as comprehensive coverage of the software engineering area generally (Keele , 2007). From the above well-known databases a total of 4424 paper is selected. Initially we selected 84 paper on the basis of title and then apply the inclusion and inclusion criteria finally reading the full text we selected 18 articles the whole process of searching and selecting the papers as illustrated in Figure 1.

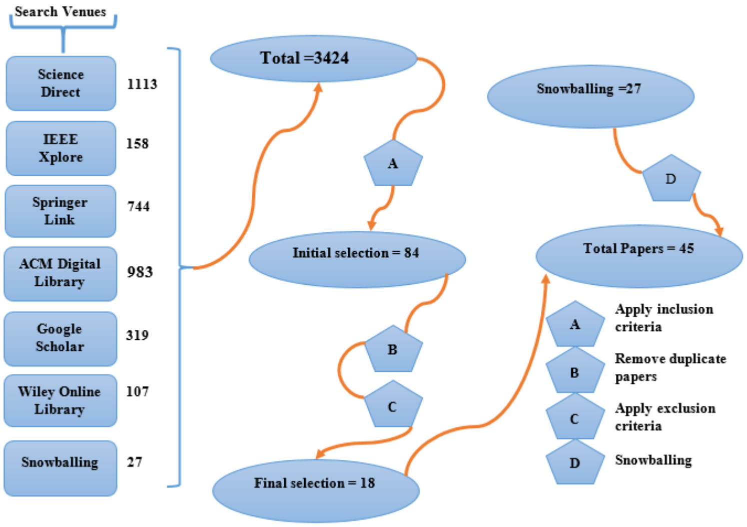

Figure: 1 Search process.

Snowballing Technique

Finding literature that an automatic search might have overlooked is accomplished by using the snowballing technique. By examining the references in the chosen articles and authors, it locates applicable research. According to the guidelines of SLR (Budgen et al., 2007), the snowballing forward & backward) strategy is applied to all chosen publications in order to include as many important or related sources as possible. The forward snowballing method is searching throughout the reference lists of relevant publications to locate more relevant articles to include in the study, whereas the backward snowballing method involves locating articles that reference the pertinent articles that are part of the research. The snowballing approach was also employed since the size of the sample was very small, the snowballing mechanism allowed us to increase our 18 publication sample size to 45 publication sample size. Table 1 summarizes the search procedure and the list of the final paper as shown in the figure 2 below as per database.

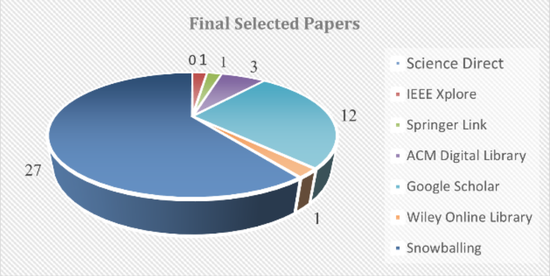

Figure 2: Final selection.

Publication Selection Criteria

Figure 3 below shows the criteria for selecting articles. It provides a visual explanation of the selection process.

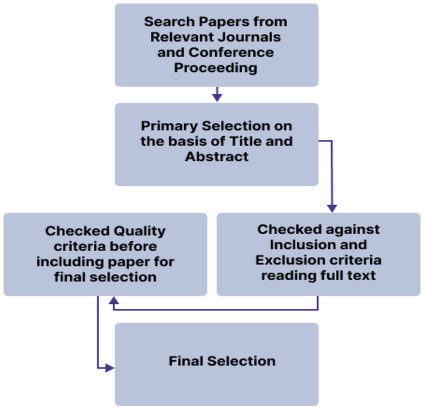
Search Papers from Relevant Journals and Conference Proceeding
Primary Selection on the basis of Title and Abstract
Checked against Inclusion and Exclusion criteria reading full text
Checked Quality criteria before including paper for final selection
Final Selection

Figure 3: Publication selection criteria.

Only publications related to the research questions will be chosen using the publication selection criteria. Only articles that are related to crowdsourcing software development will be considered.

Search Result Management

Each search's results will be saved as HTML pages in a directory. Furthermore, the top selection list will be formatted as follows. The tracing number is the number discovered while searching for a specific document. If a documented is cited in more than one database, it will only appear once in the final selection list to prevent duplication. The "S.No." field in the final.

Inclusion Criteria

The following criteria for inclusion were employed to decide which literature items (such as articles, technical reports) that were found using the search terms would be used to extract data. Only English-language publications about crowdsourced software development and engineering will be taken into consideration. Below is a summary of the requirements.

1. Research which finds out factors which contribute to long-term sustainability at any stage.
2. Publication on the issues and challenges of creating sustainable software in a CSD.
3. Research papers which outline the sustainability feature of CSD.
4. Literature that outlines sustainability criteria for CSD.
5. Research on crowdsourced software describes the aspect of software sustainability.

Exclusion Criteria

This section details the exclusion criteria that will be relied upon in the decision as to which of the pieces of literature that are obtained by the search keyword will subsequently be disregarded or excluded. The following are the requirements. The requirements are as follows.

1. Unrelated publication irrelevant to the objectives of the study
2. Research that are not related to CSD.
3. Studies that do not describe sustainability aspect factors or challenges/problems in CSD.
4. Non-crowdsourcing studies in the environment

Primary Selection

The title, keywords, and abstract will initially select primary sources. The results which cannot be related to the problem/research questions are ignored or excluded. The inclusion and exclusion criteria indicated above will be reviewed by carefully reading the entire text of the primary sources chosen during the first selection process.

Final Selection

The final selection will be on a full reading of the paper.

Publication Quality Assessment

The final selection of publications will be followed by a quality review. Along with data extraction, the quality of publications will also be assessed. The following questions will be used to assess the quality. Is it clear how the factors influencing the sustainability aspect of CSD were identified? Each of the above elements will be labeled as "YES," "NO," "PARTIAL," or "N.A." A small fraction will be graded randomly as the second reviewer by my supervisor to validate the results.

Data Extraction Strategy

Primary Study Data

The primary objective of the research is to gather information from published sources that address the questions of the study. From each final publication, we will extract the following information.

- Information about the publication, such as the title, authors, journal/conference title, and so on.
- Data that will help you answer your research queries.

Following data will be collected from the study in order to answer the questions it has:

- The study's background information.
- Factors that will positively or negatively influence crowdsourced software development's sustainability.

The extraction procedure involves a data extraction form, which includes fields, S.No., paper Id, date of review, title, Authors, Reference, database, methodology, sample population, target population, publication year, organization type, company size, project size, product type, country/ location of analysis and publication quality description.

Data Extraction Process

Analysis and extraction of the data will be conducted by one researcher. This will enable to maintain a transparent process. The primary reviewer provides data extraction after the retrieval of the papers. The primary reviewer can follow the suggestions of the secondary reviewer in respect to addressing any issue that happened during data extraction. The inter-rater reliability test will also be conducted by the second reviewer after the data extraction by the first reviewer is done. The secondary reviewer will choose a few articles at random from the list that the first reviewer has already selected. The secondary reviewer will select a random number of publications and search for them. The findings will be compared to those of the primary reviewer.

Primary Reviewer: Waqas Haider

Secondary Reviewer: Muhammad Ilyas

Data Storage

The descriptive information of each publication would be captured and stored in an electronic record in a Microsoft Word document.

Data Synthesis

The information will be compiled into a single summary table with columns (S.No., factors, Frequency, Percentages) listing all of the parameters that determine the sustainability of crowdsourced software development. In a separate table, each of the following columns will be displayed for every factor specified in the Summary table (factors group name, S.No. of factors).

Validation of the Review Protocol

Shah Khalid & Rafiq Ahmad Khan, two of my research colleagues, were the first to receive the protocol. Dr. Muhammad Ilyas, my supervisor, then reviewed it. Lastly, this was forwarded to the Software Engineering Research Group at the University of Malakand (SERG UOM).

Discussion

A Systematic Literature Review (SLR) was used to carry out the research as described in Section 3 research methodology. Despite the increasing popularization of crowdsourced software development (CSD), the sustainability of such development and its preparation as a final product still poses serious challenges. Our work fills this gap by determining factors that influence sustainability aspects in CSD through a protocol-based SLR, which provides a more rigorous and thorough examination than other reviews. The previous study has provided the useful information about both crowdsourcing and software sustainability. As an example, Hasteer et al. (2016) pointed out the advantages and issues of crowdsourcing in software development. Whereas, Asiegbu et al. (2017) highlighted its transformative significance and potential for adoption from sustainability perspective. Imran et al. (2019) introduced an SLR on practices and challenges. Penzenstadler et al. (2012) identified sustainability as a primary quality issue encountered in software engineering. In a similar manner, Leicht et al. (2016) also explored the special advantages of crowdsourcing in software testing. In spite of these contributions, not many studies have explored the intersection of sustainability and CSD in detail. The gap we expect to fill in our findings is the establishment of factors that influence sustainability aspects in crowdsourced software development. Nevertheless, some constraints should be recognized. The SLR was restricted to 6 digital libraries and English-language studies that may have omitted any relevant works. Even though the snowballing was used to reduce this, the newer studies published later after the review date may be missed. Although it has these shortcomings, the systematic approach will bolster the strength and reliability of our analysis.

The projected results of this research offer practical recommendations to developers, practitioners and the wider software community. The study can guide project development, involvement of contributors and resource mobilization by determining sustainability aspects of CSD. In the future, we intend to further this research by finding solutions and best practices and then an empirical study to prove them in actual projects. The overall result will be a sustainability model designed to support the CSD, providing a clear and practical direction to establish credible, efficient and long-lasting software systems through crowdsourcing.

Conclusion

The protocol paper will present a step-by-step approach to the investigation of the factors that determine sustainability in crowdsourced software development within the greater context of Global Software Engineering (GSE). The study underlines the opportunity of software crowdsourcing as an effective and low-cost technique of developing high-quality and long-lasting software products by utilizing a large number of software participants. The paper employed a systematic literature review (SLR) to come up with the most important factors that influence sustainability aspects in crowdsourced CSD projects. The study identifies the key factors of long-term project success, which include economic, social, environmental, and technological challenges. The combination of these findings will bridge the gap between theoretical knowledge and real applications, thereby benefiting both platform stakeholders, software industry, developer and researchers. Crowdsourcing platforms will be used to validate results and come up with answers on factors identified after the implementation phase of the SLR is being continuously implemented. The paper also contributes to the emergent concept of sustainable software engineering and eventually contribute to a significantly better-performing software ecosystem with increased lifecycle in laying out the groundwork to future research and innovation into approaches into crowdsourced development methodology.

Acknowledgments

We acknowledge the support and guidance from the authors and reviewers; we also have some thanks to the software engineering research group (SERG UOM).

References

Ahmad, R., Baharom, F., & Hussain, A. (2014). *A systematic literature review on sustainability studies in software engineering.* Paper presented at the Knowledge Management International Conference (KMICe), Langkawi, Malaysia.

Alelyani, T., & Yang, Y. (2016). *Software crowdsourcing reliability: An empirical study on developers behavior.* Paper presented at the Proceedings of the 2nd International Workshop on Software Analytics.

Alonso, O., Rose, D. E., & Stewart, B. (2008). *Crowdsourcing for relevance evaluation.* Paper presented at the ACM SigIR Forum.

Asiegbu Baldwin, C., Oluigbo Ikenna, V., Ajakwe Simeon, O., & Onyike Gerald, O. (2017). Crowdsourcing software development: concept, benefits, and adoption. *International Journal of Scientific Research in Computer Science and Engineering, 5*(3), 7-16.

Bari, E., Johnston, M., Wu, W., & Tsai, W.-T. (2016). *Software crowdsourcing practices and research directions.* Paper presented at the 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE).

Barnes, S. A., Green, A., & de Hoyos, M. (2015). Crowdsourcing and work: individual factors and circumstances influencing employability. *New Technology, Work and Employment, 30*(1), 16-31.

Boughzala, I., De Vreede, T., Nguyen, C., & De Vreede, G.-J. (2014). *Towards a maturity model for the assessment of ideation in crowdsourcing projects.* Paper presented at the 2014 47th Hawaii International Conference on System Sciences.

Brabham, D. C., Sanchez, T. W., & Bartholomew, K. (2010). *Crowdsourcing public participation in transit planning: preliminary results from the next stop design case.* Paper presented at the TRB 89th Annual Meeting Compendium.

Budgen, D., Kitchenham, B., Charters, S., Turner, M., Brereton, P., & Linkman, S. (2007). *Preliminary results of a study of the completeness and clarity of structured abstracts.* Paper presented at the 11th international conference on Evaluation and Assessment in Software Engineering (EASE).

Chatfield, A. T., & Brajawidagda, U. (2014). *Crowdsourcing hazardous weather reports from citizens via twittersphere under the short warning lead times of EF5 intensity tornado conditions.* Paper presented at the 2014 47th Hawaii International conference on system sciences.

Chittilappilly, A. I., Chen, L., & Amer-Yahia, S. (2016). A survey of general-purpose crowdsourcing techniques. *IEEE Transactions on Knowledge and Data Engineering, 28*(9), 2246-2266.

Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information science, 38*(2), 189-200.

Hasteer, N., Nazir, N., Bansal, A., & Murthy, B. (2016). Crowdsourcing software development: Many benefits many concerns. *Procedia Computer Science, 78*, 48-54.

Hatch, M. R. (2017). *The maker revolution: Building a future on creativity and innovation in an exponential world*: John Wiley & Sons.

Hegedus, P. (2013). *Revealing the effect of coding practices on software maintainability.* Paper presented at the 2013 IEEE International Conference on Software Maintenance.

Howe, J. (2006). The rise of crowdsourcing, Wired. http://www. *wired. com/wired/archive/14.06/crowds. html*.

Ilyas, M., & Khan, S. U. (2012a). Software integration challenges in global software development environment: A systematic literature review protocol. *IOSR Journal of Computer Engineering (IOSRJCE), 1*, 29-38.

Ilyas, M., & Khan, S. U. (2012b). *Software integration model for global software development.* Paper presented at the 2012 15th International Multitopic Conference (INMIC).

Imperatives, S. (1987). Report of the World Commission on Environment and Development: Our common future. *Accessed Feb, 10*.

Imran, A., & Kosar, T. (2019). Software sustainability: a systematic literature review and comprehensive analysis. *arXiv preprint arXiv:1910.06109*.

Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering: Technical report, ver. 2.3 ebse technical report. ebse.

Khan, R. A., & Khan, S. U. (2014). Communication and Coordination Challenges in Offshore Software Development Outsourcing Relationship from Vendors' Perspective: Preliminary Results. *Science International, 26*(4).

Khan, S. U., Niazi, M., & Ahmad, R. (2008). Systematic Literature Review Protocol for Software Outsourcing Vendors Readiness

Model (SOVRM). *School of Computing and Maths, Keele University, UK TR/08-01*.

Kitchenham, B., & Charters, S. (2007). Technical report title: Guidelines for performing Systematic Literature Reviews in Software Engineering, EBSE 2007-001. *Keele University and Durham University Joint Report*.

Lago, P. (2015). *Challenges and Opportunities for Sustainable Software*.

Lakhani, K. R., Garvin, D. A., & Lonstein, E. (2010). Topcoder (a): Developing software through crowdsourcing. *Harvard Business School General Management Unit Case*(610-032).

LaToza, T. D., Chen, M., Jiang, L., Zhao, M., & Van Der Hoek, A. (2015). *Borrowing from the crowd: A study of recombination in software design competitions.* Paper presented at the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.

LaToza, T. D., & Van Der Hoek, A. (2015). Crowdsourcing in software engineering: Models, motivations, and challenges. *IEEE software, 33*(1), 74-80.

Leicht, N., Knop, N., Müller-Bloch, C., & Leimeister, J. M. (2016). When is crowdsourcing advantageous? the case of crowdsourced software testing.

Malik, M. N., & Khan, H. H. (2018). Investigating Software Standards: A Lens of Sustainability for Software Crowdsourcing. *IEEE Access, 6*, 5139-5150.

Mao, K., Capra, L., Harman, M., & Jia, Y. (2017). A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software, 126*, 57-84.

Marshal Hargrave. (jul 8, 2019). "Crowdsourcing". from https://www.investopedia.com/terms/c/crowdsourcing.asp

Naik, N. (2016). *Crowdsourcing, open-sourcing, outsourcing and insourcing software development: A comparative analysis.* Paper presented at the 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE).

Naik, N. (2017). *Software CROWD-sourcing.* Paper presented at the 2017 11th International Conference on Research Challenges in Information Science (RCIS).

Nazir, S., Fatima, N., & Chuprat, S. (2019). *Individual Sustainability Barriers and Mitigation Strategies: Systematic Literature Review Protocol.* Paper presented at the 2019 IEEE Conference on Open Systems (ICOS).

Norman, T. C., Bountra, C., Edwards, A. M., Yamamoto, K. R., & Friend, S. H. (2011). Leveraging crowdsourcing to facilitate the discovery of new medicines: American Association for the Advancement of Science.

Oyedeji, S., Chitchyan, R., Adisa, M. O., & Shamshiri, H. (2024). Integrating sustainability concerns into agile software development process. *arXiv preprint arXiv:2407.17426*.

Oyedeji, S., Seffah, A., & Penzenstadler, B. (2018). A catalogue supporting software sustainability design. *Sustainability, 10*(7), 2296.

Penzenstadler, B., Bauer, V., Calero, C., & Franch, X. (2012). *Sustainability in software engineering: A systematic literature review.* Paper presented at the 16th international conference on evaluation & assessment in software engineering (EASE 2012).

Qayyum, S., Imtiaz, S., Hayat Khan, H., Almadhor, A., & Karovic, V. (2024). Working with agile and crowd: human factors identified from the industry. *Frontiers in Computer Science, 6*, 1400750.

Ramachandran, R., & Pillai, S. (2025). Open Innovation and Crowd Sourcing: Harnessing Collective Intelligence *Evolving Landscapes of Research and Development: Trends, Challenges, and Opportunities* (pp. 235-260): IGI Global Scientific Publishing.

Salam, M., & Khan, S. U. (2015). Systematic literature review protocol for green software multi-sourcing with preliminary results. *Proc. Pak. Acad. Sci, 52*, 285-300.

Sarkar, D. (2023). Social media and entrepreneurship. *International Journal For Multidisciplinary Research, 5*(3), 3401.

Stol, K.-J., & Fitzgerald, B. (2014). *Two's company, three's a crowd: a case study of crowdsourcing software development.* Paper presented at the Proceedings of the 36th International Conference on Software Engineering.

Wu, W., Tsai, W.-T., & Li, W. (2013). An evaluation framework for software crowdsourcing. *Frontiers of Computer Science, 7*(5), 694-709.